Universidad del País Vasco Euskal Herriko Unibertsitatea
BILBOKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE BILBAO

MÁSTER UNIVERSITARIO EN

INGENIERÍA MECÁNICA

TRABAJO FIN DE MÁSTER

DESARROLLO DE UNA APLICACIÓN EN MATLAB PARA LA PARTICULARIZACIÓN DE LOS MÉTODOS DE FATIGA MULTIAXIAL A LOS CASOS UNIAXIAL Y TORSIÓN PURA

Estudiante *Apodaka Manzarbeitia, Pedro*
Director *Abasolo Bilbao, Mikel*
Departamento *Ingeniería mecánica*
Curso académico *2021/2022*

Bilbao, 03 de junio de 2022

Resumen ejecutivo

- **Alumno:** Pedro Apodaka Manzarbeitia.
- **Titulación:** Máster en Ingeniería Mecánica.
- **Director:** Mikel Abasolo Bilbao.
- **Título del proyecto:** Desarrollo de una aplicación en Matlab para la particularización de los métodos de fatiga multiaxial a los casos uniaxial y torsión pura.
- **Centro:** Escuela de Ingeniería de Bilbao.
- **Departamento:** Departamento de Ingeniería Mecánica.
- **Fecha de comienzo del proyecto:** Septiembre de 2021.
- **Fecha de finalización del proyecto:** Junio de 2022.
- **Resumen:**

En este trabajo se muestra una aplicación informática, desarrollada mediante el software Matlab, para la particularización de los métodos de análisis de fatiga con tensiones multiaxiales a los casos de fatiga uniaxial y de torsión pura. Por lo tanto, dentro de las diferentes ramas en las que se puede clasificar la Ingeniería Mecánica, pertenece al ámbito de Diseño y dentro de éste a la fatiga o, más concretamente, a la fatiga con tensiones multiaxiales.

Pese a que existen numerosos métodos de fatiga multiaxial, no hay ninguno que corresponda a la solución definitiva por lo que, en cada situación (material, estado tensional…) conviene emplear uno u otro en función del que mejor se ajuste. Para ello, en principio se deberían realizar ensayos simulando las características a las que va a estar sometida la pieza, pero esto resulta excesivamente costoso y, por tanto, únicamente se suele realizar en casos particulares. Así pues, mediante este software se pretende conseguir una herramienta de ayuda en la identificación del método que mejor se ajuste para cada material, sin tener que realizar los ensayos de fatiga multiaxial.

De los múltiples métodos que existen, se van a particularizar seis de los más empleados como son: Sines, Crossland, Matake, Findley, Papuga PCr y Dang Van. Como se puede apreciar, todos ellos corresponden a métodos avanzados de fatiga multiaxial que, a partir de los datos que introducirá el usuario, se particularizarán tanto para el caso uniaxial como de torsión pura y se representarán en sendas gráficas. Junto con estos, se tendrá la posibilidad de representar los resultados experimentales de diversos materiales que se

encuentren grabados en la propia librería de la aplicación e incluso, de materiales importados por el propio usuario, para poder visualizar el método que mejor se ajusta.

Por último, dado que este trabajo se encuentra estrechamente ligado a la asignatura de *Métodos de Análisis y Diseño para Fractura y Fatiga* que se imparte en el primer curso del Máster Universitario de Ingeniería Mecánica de la UPV/EHU, se pretende conseguir una herramienta de apoyo en la docencia de la misma. En este sentido, además de servir de acompañamiento en su enseñanza, se aspira a mostrar a los alumnos que se inician en ella, la posibilidad de llevar a cabo este tipo de aplicaciones que, como se puede apreciar, pueden resultar de gran interés.

- **Abstract:**

In this work, it is shown a computer application, that has been developed in the graphic environment of Matlab, for the particularization of the multiaxial fatigue methods to the uniaxial and pure torsional cases. Therefore, whithin the different mechanical engineering trades, it belongs to the Design field; and more specifically, to the area of fatigue with multiaxial stresses.

There are a lot of multiaxial fatigue methods. In spite of this, there is not a definitive method since in each case (material, type of stresses…) it should be used the one that best fits. The correct way would be to carry out experimental tests simulating the real characteristics of the part, but this is excessively costly so, as a result, it is usually carried out only in particular cases. Hence, the development of this software is aimed to provide a tool that could help on the identification of the method that best fits to the material without the need of carry out multiaxial fatigue tests.

Despite the many methods that exist, those that are going to be particularized are Sines, Crossland, Matake, Findley, Dang Van and Papuga PCr since they are six of the most commonly used. All of them are advanced multiaxial fatigue methods which, using the material data entered by the user, will be particularized for both the uniaxial and pure torsional cases and then, they could be plotted. In addition, in order to be able to identify the method that best fits, it will be posible to plot the failure points saved in the application's own library together with those that are added by the user.

Finally, as a result of the relationship between this work and the *Fracture and Fatigue Analysis and Design Methods* subject that is taught in the first year of the UPV/EHU Master´s Degree in Mechanical Engineering, it is intended to provide a complementary tool for its teaching. In this way, as well as be a teaching aid, the app is to show the students who are introducing to the multiaxial fatigue issues, the possibility of carrying out this type of works as they can be interesting and great potential for this field.

- **Laburpena:**

Lan honetan, Matlab softwarea baliatuz fatiga multiaxialeko metodoak kasu uniaxialera eta bihurdura purura zehazteko aplikazio informatiko bat aurkezten da. Hortaz, Ingeniaritza Mekanikoa dituen arlo desberdinetatik, Diseinu eremuaren barnean materialen fatigarekin zerikusia dauka; tentsio multiaxialen eraginpenean dauden materialen fatigarekin hain zuzen ere.

Fatiga multiaxialerako hainbat método badaude ere, ez dago bakar bat behin betikoa denik zeren eta egoera bakoitzean (materiala, tentsio motak…) bata edo bestea erabiltzea komeni da hoberen egokitzen denaren arabera. Horretarako, errealitatean lan egingo den materialarekin saiakuntza experimentalak egitea aproposa izango litzateke, aldiz, oso garestia denez, kasu partikularretan baino ez da egiten. Hori dela medio, software honen xedetariko bat, fatiga multiaxialeko saiakuntzak egin barik, material bakoitzeko metodo aproposena identifikatzen laguntzen duen tresna bat eskuratzea da.

Gaur egun makina bat metodo daude baina lan honetan erabiliko direnak Sines, Crossland, Matake, Findley, Dang Van eta Papuga PCr dira; batez ere gehienbat erabiltzen diren metodoak direlako. Aipatutako guztiak, fatiga multiaxialeko metodo aurreratutzat har daitezke eta, erabiltzaileak aplikazioan sartutako balioak erabiliz, kasu uniaxialera eta bihurdura purura zehaztuko dira, geroago marrazteko aukera izateko. Honez gain, grafiko berberetan hainbat materialen hutsegite puntuak sartzeko aukera egongo da, bai aplikazioaren liburutegian gordeta dauden balioak bai erabiltzaileak gehitutakoak.

Azkenik, lan honek Euskal Herriko Unibertsitateak eskaintzen duen Ingeniaritza Mekanikoko Masterrean *Fatiga eta Hausturaren analisi eta diseinaketa metodoak* deitutako irakasgaiarekin lotura duela medio, bere irakaskuntzarako tresna baliagarria izan daitekela uste da. Horrez gain, tentsio multiaxialen eraginpenean dauden materialen fatiga ikasten hasi direnentzat, hemen aurkezten den antzeko proiektuetan sakontzeko aukera erakutsi nahi da; izan ere, garatutako aplikazio gisako tresnak interesgarriak direla ondoriozta daiteke.

Tabla de acrónimos

Acrónimo	Término
UPV/EHU	Universidad del País Vasco/Euskal Herriko Unibertsitatea
EIB/BIE	Escuela de Ingeniería de Bilbao/Bilboko Ingeniaritza Eskola
MAyDFF	Métodos de Análisis y Diseño de Fractura y Fatiga
ASTM	American Society for Testing and Materials
ASM	American Society for Metals
σ	Tensión normal
τ	Tensión cortante
$[\sigma(t)]$	Tensor de tensiones (variable en el tiempo)
σ_m	Tensión normal media
σ_a	Tensión normal alterna
τ_m	Tensión cortante media
τ_a	Tensión cortante alterna
N	Duración a fatiga (número de ciclos)
LCF	Fatiga de cíclos bajos (Low Cycle Fatigue)
HCF	Fatiga de ciclos altos (High Cycle Fatigue)
σ_{yp}	Tensión de fluencia
σ_{ur}	Tensión última de rotura
σ_{ut}	Tensión última de tracción
σ_{-1}	Límite de fatiga a tensión alterna
σ_0	Límite de fatiga a tensión pulsante
τ_{-1}	Límite de fatiga a torsión alterna
τ_0	Límite de fatiga a torsión pulsante
τ_{ut}	Resistencia a torsión
k	Fracción entre σ_{-1} y τ_{-1}
R	Relación de carga
d	Daño a fatiga
D	Daño que produciría el fallo a fatiga
σ_{eq}	Tensión normal equivalente
σ_{fallo}	Tensión normal que causaría el fallo a fatiga
τ_{oct}	Tensión cortante octaédrica
σ_h	Tensión hidrostática
α	Parámetro alfa del método
β	Parámetro beta del método

MSSR	Maximum Shear Stress Range
MD	Maximum Damage
$\{n\}$	Vector normal al plano
GAM	Gonçalves, Araújo, Mamiya
Papuga PCr	Papuga Critical Plane
SPM	Spagnoli modificado
QCP	Quadratic Critical Plane
Papuga PI	Papuga Integral
LZ	Liu y Zenner

Índice

Índice de Figuras

Índice de Tablas

I. MEMORIA

1 Introducción

1.1 Puesta en escena de la fatiga

A lo largo del tiempo se ha apreciado como los materiales que se encuentran sometidos a cargas variables fallan antes que cuando las cargas son constantes. A este fenómeno se le denomina fatiga y se debe al daño progresivo que se introduce sobre el material y que al final puede llegar a producir la rotura de un elemento mecánico, incluso cuando la carga tiene un valor menor al necesario para que se produjese la rotura estática bajo una carga constante.

Otra manera más formal de definir la fatiga es la que se emplea en la norma ASTM E-1823 y ASM International. Según estos, la fatiga es un proceso de cambio estructural localizado, progresivo y permanente que se produce en un punto del material que se encuentra sometido a condiciones que producen tensiones y deformaciones variables y que pueden originar la aparición de grietas o la rotura completa al cabo de un número suficiente de ciclos o fluctuaciones.

La mayoría de los fallos en servicio de los componentes mecánicos están relacionados con la fatiga. En este sentido, se estima que en torno al 90% de los fallos de los elementos estructurales se deben a fatiga; suponiendo un coste de alrededor del 3-5% del producto interior bruto de los países desarrollados [1].

Junto con esto, cabe remarcar el notable impacto que causan los fallos a fatiga en diferentes ámbitos ya que, se trata de fallos muy complicados de prever y que suelen resultar catastróficos. En primer lugar, se destaca el ámbito social; cuando el fallo del componente da lugar a daños humanos, y seguido de este se encuentra el aspecto económico debido a la gran cantidad de consecuencias que produce aguas abajo del fallo; como por ejemplo la parada de la máquina.

Históricamente ha habido numerosos fallos debidos a fatiga que se han producido a medida que han ido apareciendo nuevos sistemas mecánicos sometidos a diferentes cargas variables. Sin embargo, los primeros análisis de fatiga se comenzaron a realizar en torno a la segunda mitad del siglo XIX con la aparición de ferrocarril ya que éste dispone de una gran cantidad de componentes que se encuentran sometidos a cargas variables y numerosos ciclos. Un ejemplo de esto se puede encontrar en las ruedas y los ejes en donde cada vuelta del elemento rodante corresponde a un ciclo de carga.

A partir de entonces, vista la importancia que albergan los análisis a fatiga y la carencia de estudios previos, cada vez más industrias los han promovido. Entre otras, se pueden

destacar los ejemplos de la industria automovilística, aeronáutica, máquina herramienta o, la propia industria del ferrocarril [2].

Así pues, hoy en día se tienen innumerables grupos de investigación tanto privados como respaldados por administraciones públicas trabajando con proyectos relacionados con la fatiga. En ellos, se invierten importantes cantidades económicas para evitar que ocurran catástrofes como la de los aviones civiles Comet de Havilland (ver Figura 1.1) en donde el avión se desintegró como consecuencia de un diseño inapropiado desde el punto de vista de fatiga en las ventanas de la aeronave.

Figura 1.1. Avión Comet de Havilland [3].

1.2 Análisis y diseño a fatiga

Tal y como se ha podido intuir, el análisis a fatiga suele corresponder a uno de los parámetros fundamentales de diseño; especialmente si el componente pertenece a sistemas de gran responsabilidad como las aeronaves. Sin embargo, a pesar de que la mayoría de las cargas que actúan sobre los elementos de máquinas son variables, no siempre es necesario realizar un análisis a fatiga. Un caso muy común de esto se puede encontrar en el cálculo de una estructura que, a pesar de estar sometida a cargas fluctuantes como el viento, se calcula a estática. Así pues, en el siguiente subapartado se van a tratar los diferentes posibles fallos del material en función de las relaciones existentes entre una serie de parámetros.

1.2.1 Análisis estructural y resistente

Tal y como se ha mencionado previamente la inmensa mayoría de las piezas se encuentran sometidas a cargas variables, pero, sin embargo, siempre y cuando las condiciones lo permitan es más ventajoso y sencillo realizar el análisis resistente estático. Sin embargo, y antes de comenzar a exponer los diferentes casos en los que realizar un tipo de análisis u otro, se van a presentar los conceptos de análisis estructural y resistente.

Para estudiar si una pieza es válida para el desempeño de una determinada tarea y, por consiguiente, que no se produzca el fallo, se deben realizar el análisis estructural y

resistente. Mediante el análisis estructural, se analiza el comportamiento del sistema sin tener en cuenta que se puede dar el fallo del material y, se obtienen las tensiones que deberá soportar el material. Actualmente, esto se realiza mediante EF y se pueden distinguir tres tipos: estático, cuasiestático y dinámico; siendo este último el menos preciso y el más costoso por lo que, se reservará para aquellos casos en los que resulte estrictamente necesario.

A partir de los resultados del análisis estructural se realiza el análisis resistente, en donde se estudia si las tensiones que actúan sobre el sistema van a producir el fallo del material. En este caso, tal y como se observará más adelante, se van a distinguir dos tipos de comportamiento del material o posible fallo: estático o fatiga.

Tras esto, para poder recoger en una tabla los diferentes casos que se pueden dar siguiendo una nomenclatura identificable, en la ecuación (1.1) se presenta la expresión genérica de la tensión en el punto crítico del material en función del tiempo.

$$\sigma(t) = \sigma_m + \sigma_a = \frac{F_m}{A} + \frac{F_a}{A} \cdot D(\bar{\omega}) \cdot \sin(\bar{\omega} t) \tag{1.1}$$

Donde:

- σ_m: Tensión media
- σ_a: Tensión alterna
- F_m: Fuerza media
- F_a: Fuerza alterna
- A: Área o sección
- $\bar{\omega}$: Frecuencia de la fuerza
- $D(\bar{\omega})$: Amplificación dinámica

Cabe mencionar que, aunque se trate de un caso poco habitual, en el caso en que las fuerzas no sean armónicas se realiza la correspondiente descomposición de las fuerzas con la transformada de Fourier y a continuación, se trabaja con la frecuencia más alta representativa.

Por lo tanto, teniendo en cuenta la expresión general (1.1), se pueden distinguir los cinco casos que se recogen en la Tabla 1.1. Como se puede apreciar, estos casos se muestran en función de cómo sea la relación entre la frecuencia de la fuerza ($\bar{\omega}$) y la frecuencia del primer modo de vibración (ω_1), la fracción entre las fuerzas ($^{F_a}/_{F_m}$) y el tipo de amortiguamiento (ξ).

Tabla 1.1. Tipos de análisis a realizar en función de las características del sistema

Caso	$\bar{\omega}/\omega_1$	F_a/F_m	ξ	Análisis	
				Estructural	**Resistente**
1	Bajo	Bajo	Indiferente	Estático	Estático
2	Bajo	Alto	Indiferente	Cuasiestático	Fatiga
3	Alto	Bajo	Bajo	Más probable estático que dinámico	Probable estático, muy improbable fatiga
4	Alto	Bajo	Alto	Muy probable estático	Muy probable estático
5	Alto	Alto	Indiferente	Dinámico	Probable fatiga

A la vista de la Tabla 1.1, el tipo de análisis tanto estructural como resistente que se debe realizar en cada caso está fuertemente influenciado por el resultado de los dos primeros cocientes ($\bar{\omega}/\omega_1$) y (F_a/F_m). Sin embargo, también se debe prestar atención al valor del amortiguamiento en los casos en los que el cociente de las frecuencias no es bajo ya que, cuanto menor sea este, mayor será $D(\bar{\omega})$ y por consiguiente, facilitaría la excitación del sistema.

Junto con esto, destacar como uno de los objetivos más importantes que se tiene a la hora de realizar el diseño de un componente de máquinas es que el factor de amplificación dinámica sea en torno a la unidad para que el sistema no entre en resonancia [4]. Para ello, las frecuencias de las cargas deben ser inferiores a las frecuencias naturales, pero como la carga viene dada, el diseñador únicamente puede actuar sobre la pieza y más concretamente, sobre su rigidez y masa ya que ω_1 depende de éstas.

Por último, como conclusión de todo esto se puede decir que los estudios a fatiga se realizan cuando por las características del sistema no se pueden realizar los cálculos a estática. Además, en estos casos los cálculos a realizar tienen que ser a fatiga ya que para evitar que un componente falle a fatiga, debe de estar diseñado a fatiga; no pudiendo extrapolar resultados de análisis estáticos ni realizar acciones semejantes.

1.2.2 Complejidad de los estudios a fatiga

Tal y como se ha comentado previamente, los análisis a fatiga se realizan cuando no se cumplen las condiciones para poder realizar los cálculos a estática; es decir, cuando es estrictamente necesario. Esto se debe a varios motivos, pero principalmente, está relacionado con lo problemático que resulta realizar los estudios a fatiga en piezas complejas con cierta precisión debido a diferentes motivos que se van a exponer a continuación.

En primer lugar, no existe una teoría sólida para general para estudiar el comportamiento a fatiga ya que todas las existentes tienen su campo de aplicación y a su vez sus propias limitaciones. Por otro lado, afectan multitud de factores como por ejemplo la forma de trabajo o el acabado superficial que, en estática no se tiene en cuenta. Asimismo, en ocasiones la fatiga se combina con otros procesos de deterioro como el creep, la corrosión o las altas temperaturas lo cual complica aún más el análisis y, además, a todo esto, hay que añadirle la elevada dispersión que tienen los resultados y, que es inherente al propio fenómeno.

Como consecuencia de todo lo anterior, se constata que los cálculos a fatiga se deben considerar como orientativos o de anteproyecto ya que su precisión es limitada. Esto significa que faltaría ensayar prototipos y verificar los resultados para poder tomarlos como válidos; lo cual debe figurar en la memoria de todos aquellos proyectos que incluyan cálculos de esta materia.

En este sentido, cabe decir que la decisión sobre realizar los ensayos la debe tomar la empresa en cuestión y que, normalmente, se suele tomar en función de la responsabilidad que vaya a tener la pieza que se está diseñando. Así, aproximadamente la totalidad de las piezas en el sector aeronáutico se someten a dichos ensayos (ver Figura 1.2) mientras que, en el sector automovilístico, dicha cantidad se reduce de manera considerable.

Figura 1.2. Ensayo a fatiga del ala del avión Boeing 777X [5].

Por último, remarcar que los métodos que se deben emplear para llevar a cabo los cálculos a fatiga deben ser siempre aquellos que estén aceptados y hayan sido reconocidos ya que, de lo contrario resultarían inservibles. Sin embargo, existe una excepción, cuando el método utilizado ha sido validado mediante ensayos y, por tanto, se asume como un método contrastado.

1.2.3 Técnicas para el análisis a fatiga

Una vez destacada la importancia de utilizar métodos de cálculo contrastados, en este subapartado se va a tratar sobre las diferentes técnicas desde las que se puede enfocar el estudio a fatiga. Principalmente, se pueden distinguir tres grandes escalas: atómica, cristalina y macroscópica; cuya relación con la ingeniería mecánica, objetivo de este trabajo, se va a explicar con mayor detalle a continuación.

Actualmente, el estudio de una pieza a fatiga a escala atómica es el nivel menos factible de los tres que se han presentado ya que no existen medios para ello, por lo que no tiene gran interés desde el punto de vista de la ingeniería mecánica. Se emplea a nivel científico y las unidades en las que se trabaja son el angstrom y el nanómetro.

Por su parte, el diseño a fatiga a escala cristalina, a pesar de que hoy en día no sea todavía viable, en un futuro con ordenadores de mayor potencia y una teoría sólida podrían ser adecuados. Se emplea en ingeniería de materiales y las unidades con las que se trabaja son las micras.

Por último, el análisis de la pieza a escala macroscópica es el que se emplea en ingeniería mecánica ya que su principal objetivo no está en el mecanismo de fallo sino en obtener piezas con buena respuesta a fatiga. Las unidades con las que se trabaja son décimas de milímetro y dado que el trabajo se enmarca dentro de este grupo, en adelante cuando se trate sobre el análisis a fatiga, se estará haciendo referencia al que se lleva a cabo en este sentido.

A la hora de realizar el análisis de un componente a fatiga, existen diferentes procedimientos en función del tipo de pieza que sea. Así, se pueden clasificar en tres grupos: normas, métodos de propósito general y protocolos internos cuyo campo de aplicación es el que sigue.

Las normas se emplean en elementos comerciales como por ejemplo los engranajes o rodamientos y se trata de un procedimiento directo ya que su realización está totalmente documentada. Sin embargo, cuando la pieza que se desea estudiar no se encuentra recogida en normas, se emplean los métodos de propósito general o métodos internos de la empresa, dependiendo el caso.

Si la empresa dispone de un protocolo documentado que ha sido contrastado mediante ensayos experimentales, se empleará este, pero cuando no es así, se utilizan los métodos de propósito general. Estos últimos son genéricos y se clasifican en métodos de vida total y métodos de propagación de grieta que, aunque ambos sean complementarios, este trabajo va a estar enfocado a los primeros.

1.2.4 Enfoques de diseño a fatiga

Después de tratar los procedimientos de cálculo a fatiga, se van a exponer los diferentes enfoques de diseño a fatiga que existen ya que cada uno de ellos tiene su cobertura y en función de las necesidades es más conveniente uno u otro.

En primer lugar, se tiene el diseño a vida infinita cuya principal característica es la obtención de resultados conservadores ya que tiene como objetivo que las tensiones se encuentren en el rango elástico. Se utilizan cuando el componente en cuestión va a estar sometido a millones de ciclos como, por ejemplo, las ruedas de una locomotora. Junto con esto, cabe decir que, la temática del presente trabajo estará relacionada con la fatiga de ciclos altos por lo que, en adelante se profundizará en ella.

Por otro lado, mediante el enfoque safe-life, se busca un diseño a vida finita de forma que, se reemplaza el componente antes de alcanzar la vida calculada bajo un determinado margen de seguridad. La determinación de este margen es una decisión compleja ya que se debe buscar un parámetro óptimo porque de lo contrario, se pueden dar las siguientes situaciones. Si el margen es pequeño, pueden ocurrir el fallo antes de lo esperado y puede resultar catastrófico mientras que, si el margen es grande, hay que reemplazar los componentes cada menos tiempo del necesario y esto da lugar a pérdidas económicas.

Otro tipo de enfoque a vida finita es el fail-safe, cuyo objetivo es evitar que el sistema completo se vea afectado cuando se produzca el fallo de un determinado componente. De esta forma, la máquina puede continuar operando hasta que se encuentre un momento adecuado para solucionarlo, por ejemplo, una parada de mantenimiento o un cambio de turno. Un caso típico de diseño fail-safe se puede encontrar en uniones atornilladas redundantes donde, puede ocurrir el fallo de uno de los tornillos y el resto siguen en funcionamiento.

Por último, se tiene el diseño damage-tolerant (tolerancia al daño) que se puede considerar como un refinamiento del enfoque safe-life. Se asume que la pieza dispone de grietas utilizando la mecánica de la fractura junto con sucesivas inspecciones, se trata de predecir cuánto tiempo durará el componente. Este enfoque se emplea en aplicaciones punteras donde se pueden llevar a cabo las citadas inspecciones periódicas.

1.3 Fallo por fatiga

Tras haber realizado una revisión general sobre el análisis y diseño a fatiga, en este subapartado, se va a tratar el fallo por fatiga. Para ello, se va a comenzar por su origen y a continuación, se van a explicar las diferentes fases que tienen lugar.

1.3.1 Origen del fallo por fatiga

Tal y como se ha mencionado previamente, en el comportamiento a fatiga de una pieza influyen multitud de factores y en este mismo sentido, el fallo puede deberse a otras tantas causas que se van a recoger en cuatro tipos de fallo: de diseño, fabricación, mal uso y montaje.

El fallo de diseño corresponde a un fallo de la etapa inicial del proyecto en donde se ha actuado de manera errónea. Ejemplos típicos de ello son el diseño a estática o semejantes en lugar de a fatiga y la aproximación errónea de las cargas y condiciones de contorno de la pieza; de modo que los cálculos se realizan considerando unas circunstancias que no representan la realidad a la que se someterá el componente.

Puede ocurrir también que, aunque el diseño de la pieza sea correcto, se produzca el fallo en su fabricación o montaje; es decir, antes de que comience a operar. Un caso común de fallo de fabricación se puede encontrar en piezas de forja ya que ésta deja defectos internos y por su parte, ejemplo de fallo de montaje sería el rayado de una pieza para introducirla en un ensamblaje lo cual, como se ha comentado, es perjudicial.

Por último, el fallo por mal uso se da como consecuencia de someter a la pieza a cargas no comprendidas en la fase de diseño. Ejemplos habituales de este tipo de fallos se puede encontrar en el sector automovilístico donde, las compañías comercializan los modelos en cada país en función de sus características (carreteras, meteorología, consideración del peso máximo autorizado…). Es por ello que, por ejemplo, al exportar un vehículo de un país desarrollado a uno subdesarrollado pueden ocurrir fallos como el de la Figura 1.3 donde el cigüeñal ha fallado por estar sometido a elevados esfuerzos de torsión.

Figura 1.3. Fallo a fatiga del cigüeñal por estar sometido a elevados esfuerzos de torsión [6].

1.3.2 Fases del fallo por fatiga

Independientemente de la causa que origine el fallo por fatiga, en rasgos generales se pueden distinguir tres fases: iniciación de la grieta, propagación y rotura [1]. Para poder

visualizar y tener un orden de magnitud aproximado de las zonas donde ocurren dichas fases, en la Figura 1.4 se presenta un ejemplo de una chapa plana sometida a esfuerzos axiales; donde las zonas A, B y C corresponden a las tres fases mencionadas.

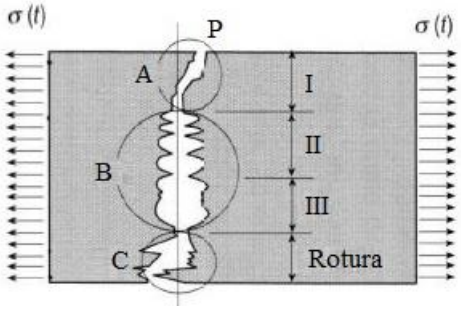

Figura 1.4. Ejemplo de las fases de fallo a fatiga sobre una pieza.

Tras identificar las zonas donde tienen lugar cada una de las fases del fallo por fatiga, a continuación, en la Figura 1.5 se muestra de forma esquematizada la secuencia de operaciones que se producen antes de que finalmente ocurra la rotura estática.

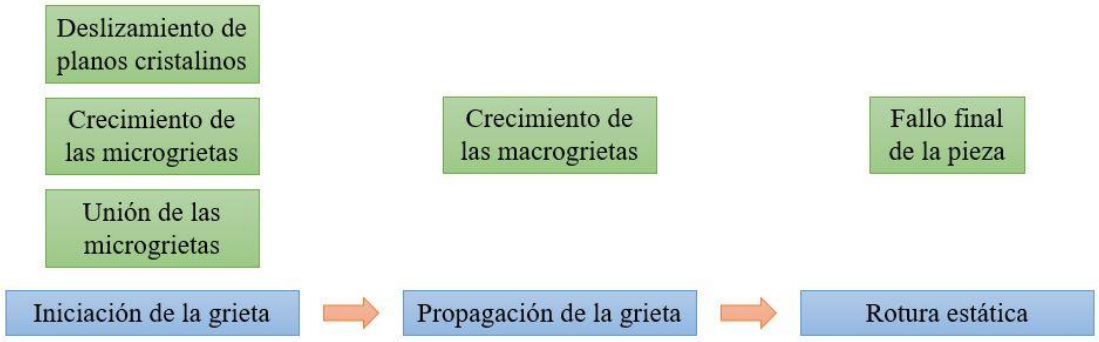

Figura 1.5. Esquema de las fases del fallo a fatiga.

La primera fase del fallo por fatiga es la iniciación de la grieta que, habitualmente, suele estar en la superficie de la pieza por los siguientes dos motivos [7]. Por un lado, se necesita una superficie libre para que se inicien las grietas por lo que, solamente comenzarán en el interior de una pieza si esta no es perfectamente maciza, es decir, si tiene poros o inclusiones. Además, las tensiones máximas suelen estar en las superficies de las piezas ya que es ahí donde se encuentran las concentraciones de tensión, soldaduras y demás aspectos que influyen negativamente en el comportamiento a fatiga.

El fundamento principal de esta primera fase se encuentra en el desplazamiento irreversible por cortadura que se produce entre dos planos cristalinos sucesivos. Junto con esto, cabe mencionar que el trayecto de la línea de deslizamiento inicial viene dado como consecuencia del tipo de cristal y de las discontinuidades que se comportan como si

fuesen concentradores de tensión. Pese a ello, la dirección aproximada de dichos deslizamientos, dado que se da por cortadura, se encuentra a 45º con respecto a la dirección de la tensión normal como se aprecia en la Figura 1.6.

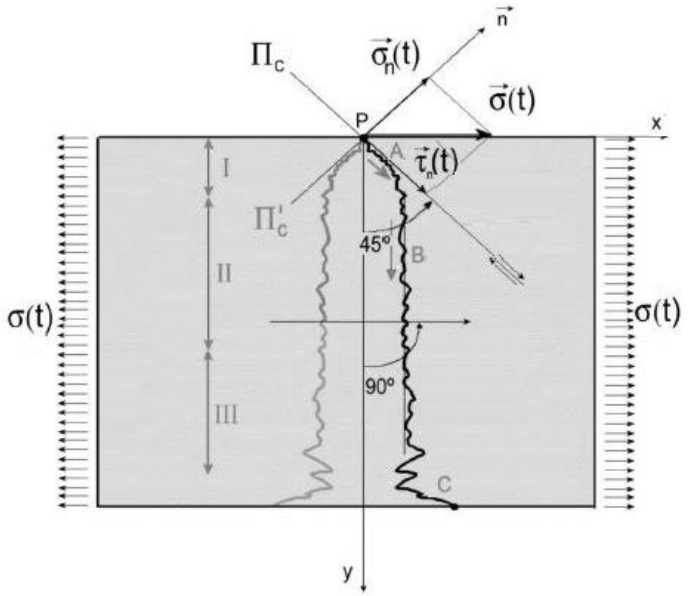

Figura 1.6. Dirección de la iniciación y crecimiento de grieta.

A través de un cumulo de deslizamientos de planos cristalinos se forman las microgrietas. Éstas a su vez, como consecuencia de la carga variable que deben seguir soportando, crecen y se unen con otras microgrietas formando así las denominadas macrogrietas. La duración de esta primera fase es muy diversa ya que el número de ciclos que durará depende de multitud de factores como, por ejemplo, las características de la carga o del material.

Una vez formada la macrogrieta, comienza la segunda fase del fallo por fatiga, la propagación. En esta zona, a diferencia de la zona de iniciación, la grieta continúa avanzando debido a las tensiones normales. Es por ello que la dirección de la grieta pasa de estar a 45º respecto a la carga, a estar a 90º o lo que es lo mismo, en dirección perpendicular a la carga.

En cuanto a la duración de esta segunda fase, normalmente se trata de un tiempo inferior al de la fase de iniciación. Esto se debe en gran medida a que la velocidad de propagación de la grieta va a aumentando ciclo a ciclo como consecuencia de que la grieta se comporta como un concentrador de tensiones y, además, como el área resistente de la pieza va disminuyendo, las tensiones cada vez son mayores.

La grieta continúa propagándose hasta que finalmente, llega un instante en el que la sección neta de la pieza no es capaz de resistir la carga de un ciclo y se produce la rotura

estática. Tras la rotura, en función del tamaño de esta última fase se podrá obtener información relevante de la forma de trabajo del componente. Así pues, tal y como se refleja en la Figura 1.7 se pueden distinguir dos posibles casos.

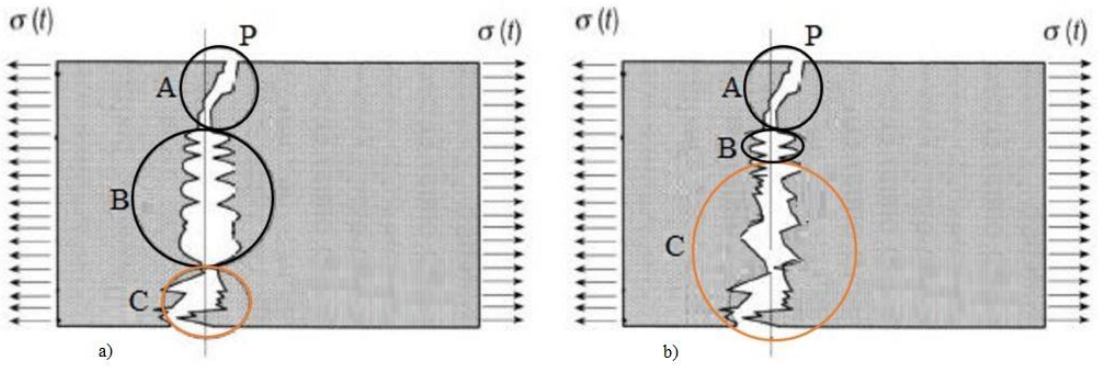

Figura 1.7. Casos característicos de la zona C: a) tensiones reducidas, b) tensiones elevadas.

En la Figura 1.7 a) se puede apreciar como la zona C es pequeña en comparación con el tamaño de la grieta. Esto indica que el componente ha estado sometido a unas tensiones reducidas y, por tanto, se ha tenido que disminuir de forma notable el área resistente hasta que al final, se ha llegado a la rotura estática. En este caso, bastaría con sustituir la pieza.

Por su parte, en la Figura 1.7 b) se tiene el caso opuesto ya que la zona C abarca gran porcentaje de la grieta. Esto indica que las tensiones de trabajo han sido elevadas y por tanto, con una ligera disminución de la sección neta se ha llegado a la rotura estática. En este caso, interesaría estudiar con mayor detalle el fallo ya que podría resultar un fallo de diseño del componente.

1.4 Métodos de fatiga para ciclos altos (HCF)

Tras introducir los aspectos generales con los que se caracteriza el fenómeno de la fatiga, lo siguiente de lo que se va a tratar va a ser sobre los métodos de fatiga de ciclos altos (HCF). Así pues, una de las posibles clasificaciones que se puede realizar es de acuerdo a la naturaleza de las tensiones que deberá soportar el componente; de forma que se pueden distinguir dos grandes grupos: fatiga uniaxial y fatiga multiaxial.

1.4.1 Fatiga uniaxial

1.4.1.1 Aspectos generales

Cuando se hace referencia al caso de fatiga uniaxial, se está indicando que sobre la pieza actúa un estado uniaxial de tensiones. Esto significa que, si se toma un diferencial

de volumen del punto más crítico, únicamente va a disponer de tensiones en una de las tres direcciones principales.

En cuanto a la naturaleza de dichas tensiones, cabe decir que no tienen por qué ser únicamente cargas alternas (variables) en la dirección en cuestión, sino que también pueden disponer de una componente media. Así pues, para poder visualizar todo esto gráficamente, se adjunta la Figura 1.8.

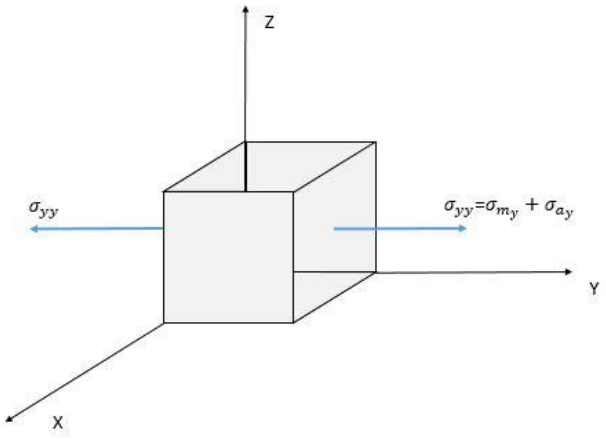

Figura 1.8. Estado tensional uniaxial.

Así pues, el tensor de tensiones que define el estado tensional de ese determinado punto de la pieza queda como se muestra en la ecuación (1.2) si se expresa en coordenadas cartesianas (x, y, z) o como en la expresión (1.3) si se emplean tensiones principales.

$$[\sigma(t)] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \sigma_{yy}(t) & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{1.2}$$

$$[\sigma(t)] = \begin{bmatrix} \sigma_1(t) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{1.3}$$

Tal y como se ha mencionado, a pesar de que en la Figura 1.8 y en las expresiones *(1.2)* y (1.3) se haya representado el caso uniaxial de tensiones en la dirección Y, el concepto es extrapolable a las direcciones X y Z.

1.4.1.2 Ensayos

Tras presentar la base de la fatiga uniaxial, a continuación, se va a tratar sobre los posibles ensayos a realizar que, básicamente se dividen en dos grupos: de probeta y de prototipo. Los primeros están enfocados a caracterizar el comportamiento del material a fatiga para después realizar los análisis pertinentes mientras que los segundos, están

destinados a tareas como la validación de componentes, por lo tanto, en adelante se va a tratar sobre los ensayos de probeta.

Dentro de los diferentes tipos de ensayos de probeta que existen en la actualidad, uno de los más empleados es la probeta rotatoria de Moore [4], también conocida como máquina de flexión alterna. Esto se debe a su forma de trabajo ya que, al estar biapoyada en sus extremos y rotando con respecto a su eje de revolución, se encuentra sometida a flexión alterna como se explicará más adelante.

La logística necesaria para llevar a cabo este ensayo, básicamente consta de los siguientes componentes: contador de ciclos, motor, acoplamiento flexible, soportes y la propia probeta, como se muestra en la Figura 1.9.

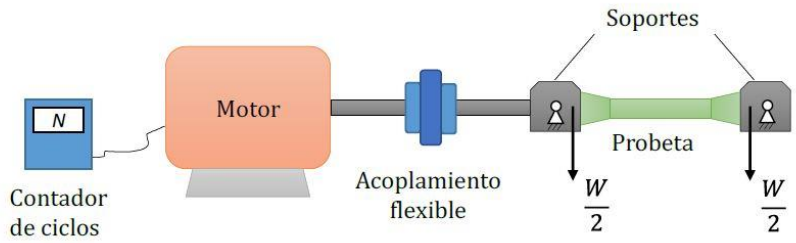

Figura 1.9. Esquema del ensayo con probeta rotatoria de Moore.

Así como se puede intuir, la sección crítica de la probeta es una sección circular que se encuentra en el cilindro central a la misma distancia de los dos extremos longitudinales ya que, es donde se tiene la tensión máxima. En ella se halla el que se denomina como punto P por ser el más solicitado y es por ello que ahí se iniciará a grieta.

En cuanto al principio de funcionamiento, como se ha adelantado previamente, el punto P estará sometido a un estado tensional de flexión alterna. Esto es, al girar el motor se hace rotar la probeta y por tanto el punto P pasa de tracción a compresión y viceversa (ver Figura 1.10) hasta que finalmente, al cabo de N ciclos se produce la rotura.

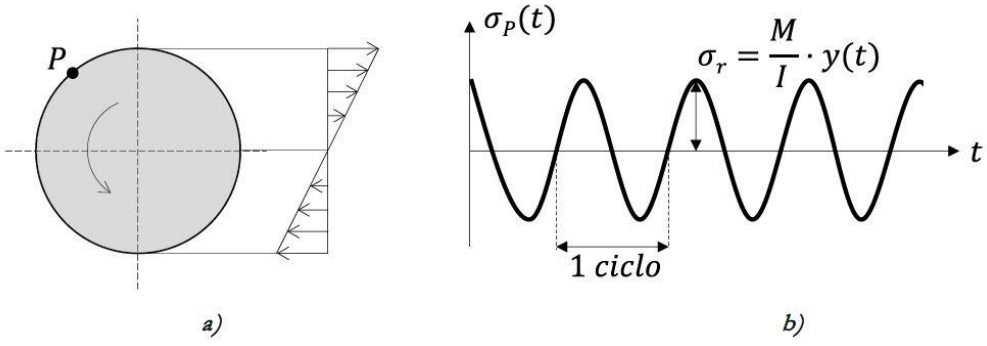

Figura 1.10. Tensiones en la probeta rotatoria de Moore: a) sección crítica, b) estado tensional.

Junto con esto, cabe remarcar dos aspectos del ensayo. Por un lado, las condiciones del ensayo están normalizadas por lo que características como las dimensiones o el acabado de la probeta (pulido a espejo) se deben respetar. Pese a ello, es muy probable que ocurra que dos probetas aparentemente idénticas, ensayadas en la misma máquina y bajo condiciones idénticas, fallen tras un número diferente de ciclos; lo cual se debe a la anteriormente mencionada dispersión del propio fenómeno.

Por otro lado, respecto al estado tensional, únicamente afecta la amplitud entre el máximo y el mínimo valor de la tensión entre los que se fluctúa; ya que la frecuencia y el periodo de cada ciclo no influyen. En lo único que afectarían estos últimos es en el tiempo que tardaría la pieza en llegar a la rotura ya que para alcanzar los N ciclos, cuanto menor sea el periodo más rápido se llegará.

Realizando varios ensayos de probeta rotatoria de Moore, y modificando de forma progresiva la carga a la que se somete a la probeta, se consiguen una serie de parejas de resultados de $\sigma_a - N$ que dan lugar a una banda de dispersión. Sin embargo, por comodidad se suele transformar a escala logarítmica o bien el eje de abscisas únicamente ($\sigma_a - \log N$) o ambos ejes ($\log \sigma_a - \log N$), siendo esta última forma la más empleada. Así pues, pasando las asíntotas resultantes a escala logarítmica, se consigue una representación como la de la Figura 1.11 cuyas zonas se van a explicar a continuación.

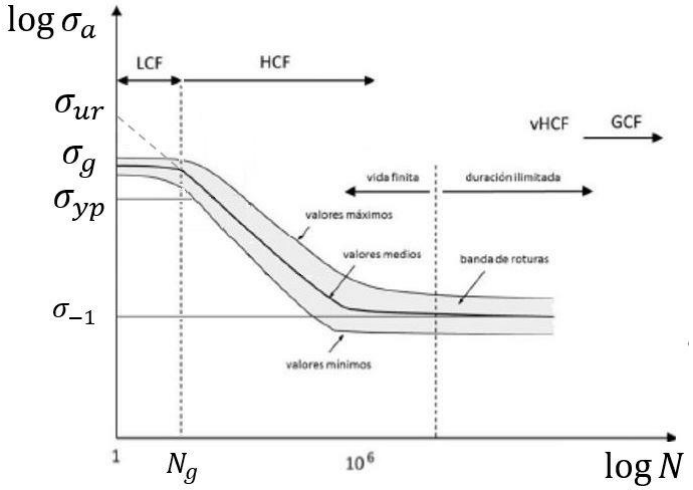

Figura 1.11. Curva de Basquin.

En primer lugar, se puede distinguir entre la zona de ciclos bajos (LCF, "Low Cycle Fatigue") y la zona de ciclos altos (HCF, "High Cycle Fatigue"). En la Figura 1.11 se ha expuesto como si dicha separación tuviese lugar a partir de un determinado número de ciclos; sin embargo, esto no es así ya se tiene una zona de transición entre ambos. Tras esto, se van a describir las características de ambas zonas.

Como se puede apreciar en la Figura 1.11, la zona de ciclos bajos corresponde a tensiones elevadas, cercanas e incluso mayores que la tensión de fluencia (σ_{yp}), lo cual hace que la duración de la pieza sea pequeña; en torno a unos pocos miles de ciclos. En el estudio de las piezas que trabajan en esta zona influye la plasticidad y esto se realiza mediante deformaciones; curva $\varepsilon - N$.

Por su parte, la zona de ciclos altos, como su propio nombre indica, considera tensiones menores a σ_{yp} que dan lugar a duraciones mayores; desde miles hasta millones de ciclos. La mayoría de los componentes de máquinas se diseñan para que su duración se encuentre sobre dicha zona; en la cual, a diferencia de LCF donde aparecía la plasticidad, el material trabaja en régimen elástico-lineal por lo que los análisis son más sencillos.

Por otro lado, en la representación se pueden observar tres curvas aproximadamente con la misma secuencia de geometría: recta horizontal, recta con pendiente negativa y recta horizontal. Las dos curvas de los extremos representan los límites de la banda lo cual acarrea una gran dispersión, sin embargo, con el fin de simplificar los cálculos, se utiliza la línea media de la banda que corresponde a una fiabilidad del 50%.

Para terminar con el análisis de la Figura 1.11, se van a mencionar las tensiones características de la curva. Por un lado, en el punto de corte que se obtiene de prolongar la recta inclinada con el eje de ordenadas, se aproxima a la tensión última real de rotura (σ_{ur}) mientras que, a la tensión que delimita las zonas de LCF y HCF se le denomina σ_g. Por último, en el extremo inferior de la curva, se tiene el límite de fatiga del material (σ_{-1}), el cual hace referencia a la tensión por debajo de la cual no se produce el fallo por fatiga, es decir, límite entre vida finita (superior) y vida infinita (inferior).

Tal y como se puede apreciar, en este caso, el límite de fatiga lleva asociado el subíndice "-1", lo cual está unido a la relación de carga R cuya expresión es la que sigue:

$$R = \frac{\sigma_{min}}{\sigma_{max}} \tag{1.4}$$

Donde:

$$\sigma_{min} = \sigma_m - \sigma_a \tag{1.5}$$

$$\sigma_{max} = \sigma_m + \sigma_a \tag{1.6}$$

Así pues, en función del estado tensional al que se someta a la pieza se pueden distinguir los cuatro casos que se muestran en la Figura 1.12. Como se puede apreciar, el primer caso (caso (a)) corresponde al estado tensional uniaxial alterno que, es el que se tiene al ensayar la probeta rotatoria de Moore, por lo que con dicho ensayo el límite de fatiga se escribe como σ_{-1}.

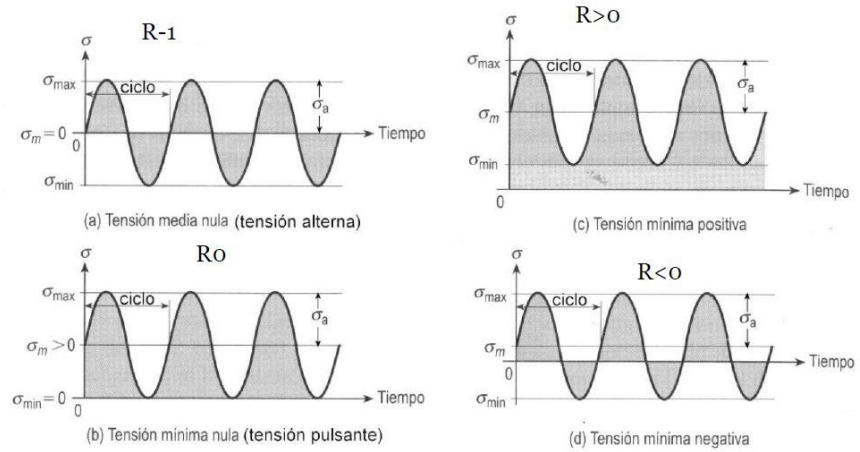

Figura 1.12. Casos de R en función del estado tensional [1].

Con el ensayo de probeta rotatoria de Moore únicamente se puede ensayar a R_{-1} y aunque en muchas ocasiones sea suficiente porque el estado tensional de la pieza lo permite, hay veces en las que es necesario realizar ensayos a otras R. Para ello, existen máquinas como la máquina de ensayos de tensión directa que permite ensayar probetas a cualquier R, pero tiene la desventaja de la complejidad y del coste económico. Por lo tanto, al trabajar con una curva SN de un material, se debe conocer tanto el tipo de carga (axial, flexión…) como la R con la que se ha realizado el ensayo.

Junto con esto último, cabe mencionar que, a pesar de que la mayoría de los aceros a temperatura ambiente tienen su límite de fatiga, existen excepciones como los aceros inoxidables o las aleaciones no férricas que no lo tienen (ver Figura 1.13). En estos casos, se determina una tensión denominada "endurance limit" que se suele encontrar entre 10^6 y 10^8 ciclos, aproximadamente. Así, la diferencia entre el límite de fatiga y el "endurance limit" está en que por debajo de dicha tensión en el primero no se produciría el fallo (vida infinita) mientras que, en el segundo sí.

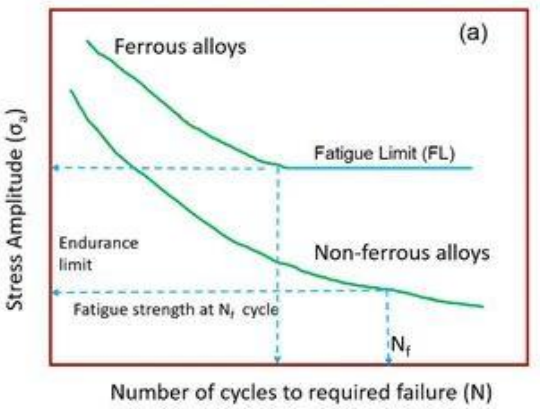

Figura 1.13. Diferencia entre material con límite de fatiga y con "endurance limit"[8].

Por otra parte, en el ensayo de probeta la curva SN que se obtiene corresponde al material; sin embargo, el componente de máquina que va a estar sometido a fatiga tiene sus propias peculiaridades por lo que para adaptar los resultados obtenidos de la probeta a dicha pieza se utilizan los coeficientes modificativos de Marin. Éstos influyen a los tres puntos que delimitan la curva (σ_g, N_g y σ_{-1}) como se indica a continuación.

$$\sigma_e = {C_{mod}}^e \cdot \sigma_{-1} \tag{1.7}$$

$${\sigma_g}_e = {C_{mod}}^g \cdot \sigma_g \tag{1.8}$$

$$N_e = {C_{mod}}^N \cdot N_g \tag{1.9}$$

Sin embargo, debido a la gran cantidad de información que se dispone sobre ${C_{mod}}^e$ y la escasa documentación que hay sobre ${C_{mod}}^g$ y ${C_{mod}}^N$, es habitual trabajar únicamente con la fórmula (1.7) y no considerar las expresiones (1.8) y (1.9). Así pues, a modo de ejemplo se van a mencionar algunos de los coeficientes modificativos más influyentes en ${C_{mod}}^e$: acabado superficial (c_s), dimensiones y geometría(c_d), forma de trabajo(c_t), fiabilidad(c_f), temperatura(c_T) e impactos(c_k).

Tras obtener la curva SN de la pieza que se desea estudiar a fatiga, a continuación, se muestra gráficamente sobre la Figura 1.14 la manera de proceder para obtener la duración del componente. Como se puede apreciar, se entra con la tensión alterna en el eje de ordenadas y en donde corta con la curva SN se traza una recta vertical que indica el número de ciclos que va a soportar la pieza. Como se puede intuir, para la resolución matemática, bastaría con realizar los cálculos mediante semejanza de triángulos.

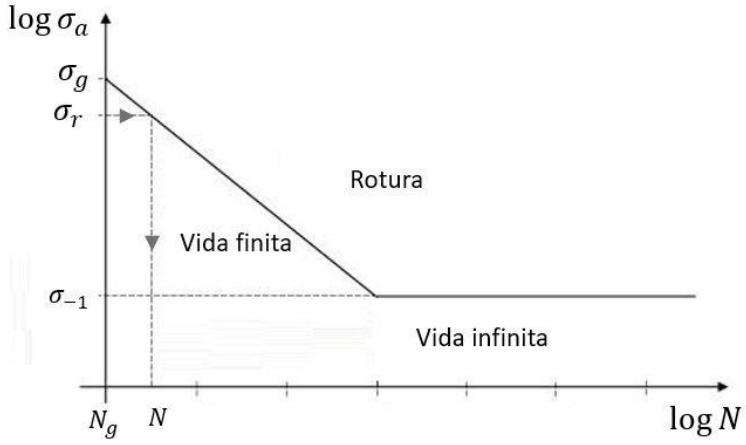

Figura 1.14. Procedimiento para obtener la duración a fatiga a R_{-1} en la curva SN.

Hasta ahora se ha explicado el caso en el que únicamente se tiene tensión alterna, pero, sin embargo, es habitual que sobre la pieza actúen también la tensión media. En este caso,

se ensayan probetas para diferentes casos de $\sigma_m - \sigma_a$ y se obtiene la duración N de modo que se puede representar los resultados en la gráfica $\sigma_m - \sigma_a - N$; también denominado diagrama de Haigh (ver Figura 1.15).

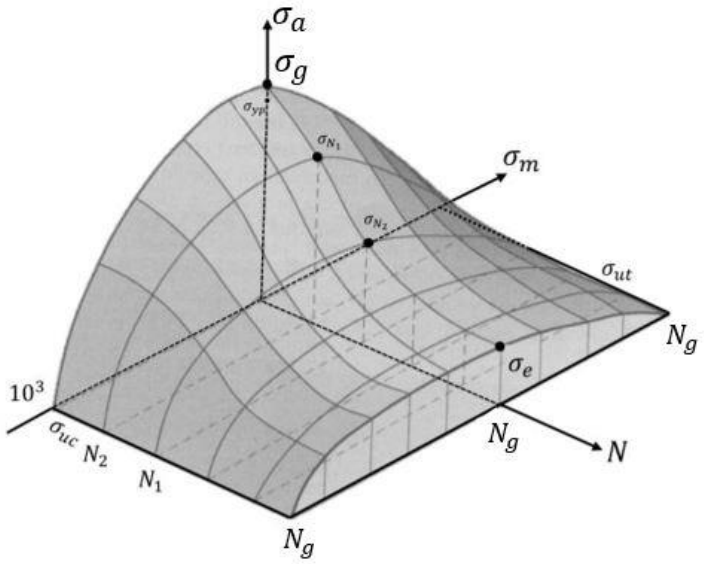

Figura 1.15. Diagrama de Haigh [4].

A la vista de la representación, y como era de esperar, cuando la tensión media es nula la curva coincide con la mostrada en la Figura 1.11 mientras que, cuando se introduce una tensión media no nula, los resultados son diferentes si ésta es de tracción o compresión. El primer caso corresponde a la situación más crítica porque tiende a "abrir" la grieta y cuando σ_m es de compresión ocurre lo contrario; esto es, tiende a "cerrar" la grieta y por tanto, la duración de la pieza es mayor.

Una vez que se tiene la curva de Haigh, la manera de proceder para la obtención de la duración del componente se puede considerar como una continuación de lo explicado para el caso de σ_m nula. Se comienza trabajando en el plano $\sigma_m - \sigma_a$ de forma que, se coloca el estado tensional al que se somete el componente y, trazando una recta desde el extremo de la curva que corta el eje de abscisas, se calcula una tensión alterna equivalente que genera el mismo daño. Con ella, se pasa al plano $\sigma_a - N$ y siguiendo lo explicado mediante la Figura 1.14 se obtiene la duración aproximada de la pieza, N.

1.4.1.3 Criterios de ajuste

Tal y como se ha explicado, cuando sobre la pieza además de la tensión alterna actúa también la tensión media, hay que considerarlo y para ello, se trabaja con curvas de ajuste. Así pues, existen multitud de tipos y variantes en función de si el material es dúctil o frágil; sin embargo, en adelante se va a tratar el caso de materiales dúctiles ya que además de ser los más empleados, son parte del presente trabajo.

Por lo tanto, se va a comenzar describiendo las diferentes curvas de ajuste que hay para el caso de tensión media de tracción y a continuación, se van a mencionar algunas consideraciones para el caso de compresión. Así pues, sobre la Figura 1.16 se muestran algunos de los criterios más empleados en materiales dúctiles con tensiones medias de tracción.

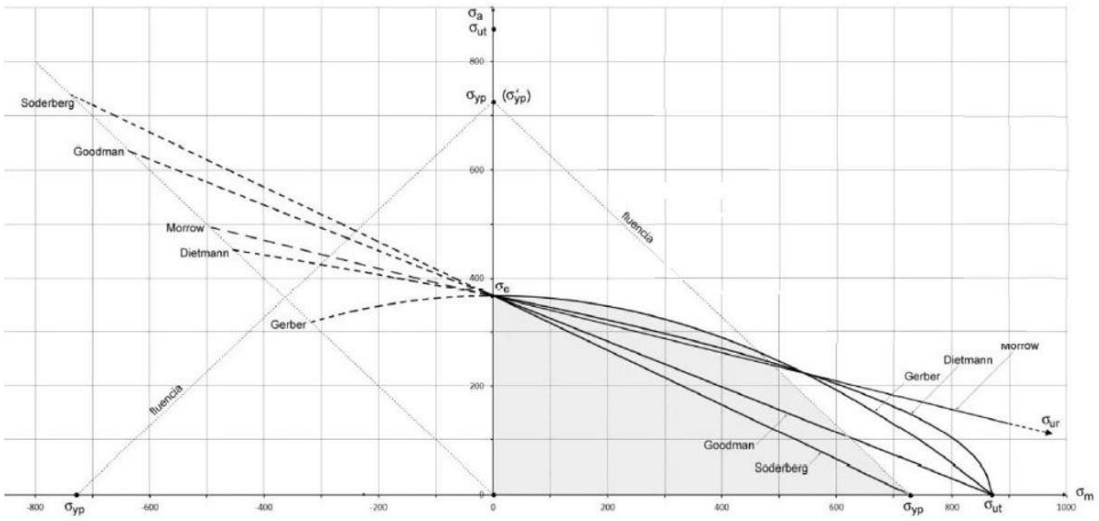

Figura 1.16. Criterios de ajuste a fatiga con tensión media en materiales dúctiles [1].

El criterio de Goodman es uno de los más empleados. Se suele utilizar en normas y de forma general, se puede considerar como un criterio conservador. La expresión que lo define es la siguiente.

$$\frac{\sigma_a}{\sigma_e} + \frac{\sigma_m}{\sigma_{ut}} = 1 \tag{1.10}$$

Un criterio menos utilizado es el de Soderberg por ser demasiado conservador. Su fórmula es:

$$\frac{\sigma_a}{\sigma_e} + \frac{\sigma_m}{\sigma_{yp}} = 1 \tag{1.11}$$

En cuanto al criterio de Morrow, se puede apreciar cómo es semejante al de Goodman pero empleando datos de la curva tensión-deformación real en lugar de la ingenieril; es decir, sustituyendo σ_{ut} por σ_{ur} como sigue.

$$\frac{\sigma_a}{\sigma_e} + \frac{\sigma_m}{\sigma_{ur}} = 1 \tag{1.12}$$

Los criterios mencionados hasta el momento corresponden con rectas en el plano $\sigma_m - \sigma_a$ pero también existen criterios que se representan por curvas como Dietmann, la parábola de Gerber o la elipse de Marin cuyas expresiones se van a presentar a continuación en las ecuaciones (1.13), (1.14) y (1.15) respectivamente.

$$\left(\frac{\sigma_a}{\sigma_e}\right)^2 + \frac{\sigma_m}{\sigma_{ut}} = 1 \tag{1.13}$$

$$\frac{\sigma_a}{\sigma_e} + \left(\frac{\sigma_m}{\sigma_{ut}}\right)^2 = 1 \tag{1.14}$$

$$\left(\frac{\sigma_a}{\sigma_e}\right)^2 + \left(\frac{\sigma_m}{\sigma_{ut}}\right)^2 = 1 \tag{1.15}$$

Por su parte, si se tiene una tensión media de compresión, de las diferentes maneras de trabajar que existen se van a destacar las dos siguientes. Una opción es considerar que la tensión media no influye a compresión trazando una recta horizontal por σ_e, lo cual es conservador ya que la σ_m de compresión es beneficiosa a fatiga. Otra posible opción es prolongar las curvas del caso de tracción como se muestra en la Figura 1.16, lo cual es lo que se va a realizar en este trabajo.

Para finalizar, se concluye diciendo que, como es lógico, el mejor criterio es aquel que mejor se ajusta al comportamiento real del componente y para escogerlo habría que realizar ensayos experimentales, pero como se ha mencionado previamente, esto es costoso desde el punto de vista tanto temporal como económico. Es por ello que, salvo en casos concretos como las piezas de gran responsabilidad, se trabaja con el criterio que indique la norma que se está siguiendo y, si no hay norma, con el criterio que se estime más oportuno.

1.4.2 Fatiga multiaxial

1.4.2.1 Aspectos generales

Tras explicar lo relevante respecto a la fatiga uniaxial, en este subapartado se va a tratar sobre la fatiga multiaxial que, se puede considerar como una extensión de la fatiga uniaxial en donde se tienen tensiones actuando en varias direcciones simultáneamente. Al igual que se ha realizado en el caso de fatiga uniaxial, en la Figura 1.17 a) se representa, sobre un diferencial de volumen, el caso genérico de un estado tensional multiaxial.

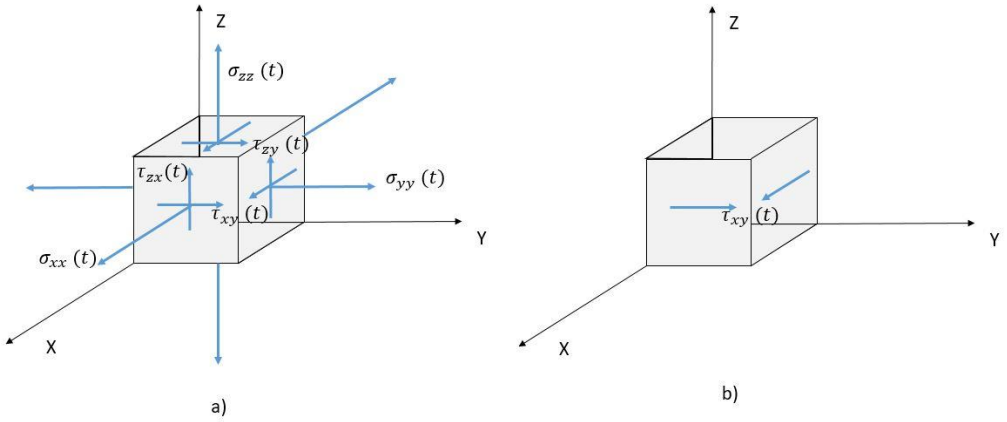

Figura 1.17. Estado tensional multiaxial: a) caso general, b) caso biaxial.

En la Figura 1.17 a) se ha representado el caso general en donde actúan tanto las tres componentes de tensión normal ($\sigma_{xx}, \sigma_{yy}, \sigma_{zz}$) como las tres componentes de tensión tangencial ($\tau_{xx}, \tau_{yy}, \tau_{zz}$). Pese a ello, para que el estado tensional sea multiaxial, basta con tener o bien una componente de torsión (ver Figura 1.17 b)) o como mínimo, dos de las seis componentes que se acaban de citar. Así, el tensor de tensiones que define el estado tensional de ese determinado punto de la pieza queda como se muestra en la ecuación (1.16) si se expresa en coordenadas cartesianas (x, y, z) o con las tensiones principales como en la expresión (1.17) junto con las direcciones principales.

$$[\sigma(t)] = \begin{bmatrix} \sigma_{xx}(t) & \tau_{xy}(t) & \tau_{xz}(t) \\ \tau_{xy}(t) & \sigma_{yy}(t) & \tau_{yz}(t) \\ \tau_{xz}(t) & \tau_{yz}(t) & \sigma_{zz}(t) \end{bmatrix} \tag{1.16}$$

$$[\sigma(t)] = \begin{bmatrix} \sigma_1(t) & 0 & 0 \\ 0 & \sigma_2(t) & 0 \\ 0 & 0 & \sigma_3(t) \end{bmatrix} \tag{1.17}$$

En este sentido, cabe destacar la importancia de los estados tensionales multiaxiales ya que, en la realidad lo más común es que los componentes de máquinas se encuentren sometidos a este tipo de tensiones. Un ejemplo típico de ello es un eje que se encuentra sometido a torsión, cuyo estado tensional corresponde con la Figura 1.17 b) y por tanto, está dentro de este grupo.

Por último, decir que independientemente de las tensiones que den lugar al estado de tensiones multiaxial, el análisis a fatiga con este tipo de tensiones es un tema de controversia y sobre el que se sigue investigando [9]. Esto se debe en buena medida a su complejidad y a la gran cantidad de aspectos que influyen (p.ej.: componentes de las

tensiones, desfases entre ellas o variación de las direcciones principales) a la hora de realizar los ensayos y desarrollar métodos de cálculo generalistas. Es por ello que continuamente están apareciendo diferentes métodos de cálculo y variaciones de los anteriores, pero actualmente no existe un método reconocido para cualquier aplicación.

1.4.2.2 Ensayos

Previamente se ha reflejado tanto la importancia de la fatiga multiaxial como la complejidad que conllevan sus ensayos. Es por ello que, hay empresas que deciden no realizar ensayos de fatiga multiaxial; de modo que trabajan con métodos reconocidos y en la última fase realizan ensayos de prototipos [1]. En el lado opuesto se encuentran las empresas que invierten cantidades económicas considerables en realizar ensayos a fatiga multiaxial, ya que las máquinas necesarias para introducir el estado tensional multiaxial sobre la probeta son más caras que las que se emplean para fatiga uniaxial.

Del mismo modo, las probetas que se ensayan en estas máquinas también difieren con respecto a las de fatiga uniaxial; siendo las más comunes las probetas cilíndricas (ver Figura 1.18 b)) y las probetas cruciformes (ver Figura 1.18 a)).

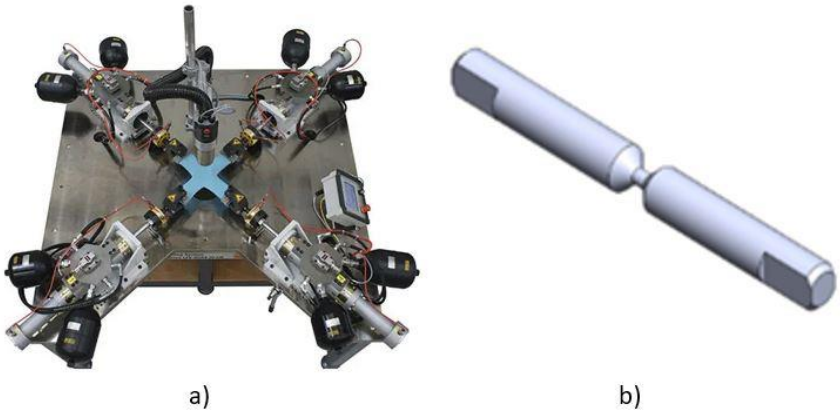

Figura 1.18. Probetas para el ensayo de fatiga multiaxial: a) Cruciforme [10], b) Cilíndrica [11].

Las probetas cruciformes se suelen someter a cargas biaxiales como se puede apreciar en la Figura 1.18 a), pero debido a la concentración de tensiones que se tiene en la propia cruceta hace que su uso sea menor que el de las probetas cilíndricas. Pese a ello, entre las posibles soluciones que se suelen emplear, se tiene aplicar un radio de acuerdo considerable o disminuir la sección de la probeta en la zona de la cruceta. Por último, en cuanto a las probetas cilíndricas como la de la Figura 1.18 b), destacar la necesidad de ensayarla bajo tres condiciones de carga (torsión pura, flexión junto con torsión, axial junto con torsión; siendo este último el más habitual) para conseguir representar el diagrama $\sigma_1 - \sigma_2$.

1.4.2.3 Métodos de fatiga multiaxial

Así como se ha adelantado en el subapartado 1.4.2.1 Aspectos generales, existen multitud de métodos de fatiga multiaxial pero, dado que ninguno de ellos es el método generalista aplicable a cualquier situación, cada uno está destinado a un rango de aplicación limitado. Asimismo, todos los métodos no tienen la misma base de partida por lo que, sobre la Figura 1.19 se propone una clasificación de los métodos de fatiga multiaxial.

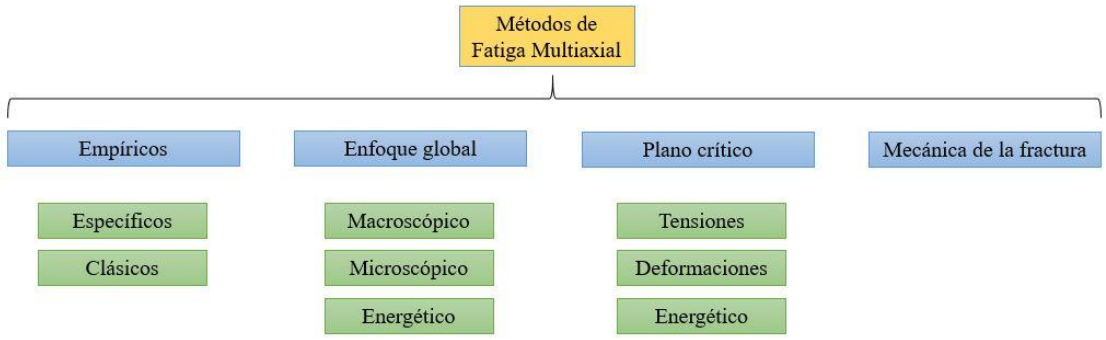

Figura 1.19. Clasificación de métodos de fatiga multiaxial.

Los primeros métodos que se desarrollaron fueron los empíricos y más concretamente, los denominados específicos ya que, se ajustan a los resultados de ensayos de probetas sometidas a fatiga multiaxial. Esto hace que sean válidos únicamente para estados tensionales concretos y por consiguiente, en la actualidad, no se empleen con frecuencia. Por su parte, los métodos clásicos se pueden considerar como la evolución de los métodos específicos ya que se amplía el campo de aplicación y su uso es más extendido. Sin embargo, dispone de una serie de inconvenientes que se van a recoger a continuación.

En primer lugar, dado que la base de los métodos clásicos son las teorías de fallo estático como, por ejemplo, Von Mises (ver ecuación (1.18)), se puede considerar que no tienen un fundamento teórico sólido para la fatiga multiaxial. Asimismo, no son adecuados cuando se tiene un estado tensional hidrostático, ya que se anula la expresión de Von Mises, ni consideran aspectos como la influencia de la tensión cortante media o la variación de las direcciones principales.

$$\sigma_{eq_{VM}} = \sqrt{{}^{1}/_{2} \cdot [(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2]} \qquad (1.18)$$

Pese a las citadas desventajas que presentan los métodos clásicos, debido a la sencillez que los caracteriza, aún se emplean en aplicaciones de poca responsabilidad o cálculos preliminares. Sin embargo, en componentes de gran responsabilidad cuyo

comportamiento es crítico, estos métodos no son admisibles y se deben emplear métodos avanzados que permitan tener en cuenta las consideraciones mencionadas.

Así pues, dentro de los métodos avanzados se encuentran los tres grupos principales que restaban atendiendo a la Figura 1.19; es decir, los métodos de enfoque global, plano crítico y mecánica de la fractura. Este último se trata de un método de inspección para determinar la vida útil restante de la pieza mientras que, los dos primeros son métodos de Vida Total para estimar el número de ciclos de su duración. Son grupos complementarios y en este caso, se va a trabajar con los métodos de Vida Total por lo que a continuación, se introducirán sus bases.

Los métodos de enfoque global permiten analizar componentes de máquinas sometidos a estados tensionales cuasi hidrostáticos y además son capaces de considerar la influencia de la tensión cortante media; lo cual no se puede realizar mediante los métodos clásicos. Sin embargo, dado que estudian la fatiga a nivel de punto en vez de a nivel de plano, no son capaces de considerar el efecto de la variación de las direcciones principales por lo que, en estos casos se trabajaría con los métodos de plano crítico. Entre los ejemplos de métodos pertenecientes al grupo de los métodos de enfoque global se pueden destacar los siguientes: Sines, GAM (Gonçalves, Araújo, Mamiya) y Crossland.

Por su parte, los métodos de plano crítico son los métodos más sofisticados ya que, además de las ventajas de los métodos de enfoque global, al estudiar la fatiga a nivel de plano, son capaces de considerar también la influencia de la variación de las direcciones principales. Sin embargo, tiene la desventaja de que para conseguir todo esto, el coste computacional aumenta considerablemente en comparación con el método anterior (se estudia un punto en lugar de un plano). Ejemplos típicos de los métodos de plano crítico son: Matake, Findley, Susmel, Papuga, Dang Van, Robert, McDiarmid, SPM (Spagnoli Modificado), QCP (Quadratic Critical Plane) y LM (Liu, Mahadevan).

Por último, tras contextualizar la importancia que tienen los estudios de fatiga multiaxial decir que, al igual que se ha mencionado previamente, para poder seleccionar el método más adecuado para cada aplicación lo mejor sería realizar ensayos experimentales e identificar el método que mejor se ajuste. Sin embargo, esto es excesivamente costoso en el caso de fatiga multiaxial por lo que, se hace notoria la necesidad de herramientas como la que se ha desarrollado en el presente trabajo que faciliten dicha selección.

2 Contexto

Este trabajo se encuentra estrechamente ligado a la asignatura de *Métodos de Análisis y Diseño para Fractura y Fatiga* que se imparte en el segundo cuatrimestre del Máster Universitario en Ingeniería Mecánica en la UPV/EHU. En ella, se desarrollan los aspectos fundamentales para comprender tanto el propio fenómeno de la fatiga como los diferentes criterios de diseño y análisis que existen en la actualidad.

El objetivo de dicha disciplina es dotar al alumno de las herramientas necesarias para que éste adquiera destreza en el análisis y diseño de componentes mecánicos que se encuentren sometidos a situaciones de fatiga o fractura [12]. Asimismo, se busca que el estudiante sea capaz de seleccionar los ensayos necesarios que se deben realizar en función de las características del problema que se desea resolver.

Junto con lo mencionado anteriormente, cabe destacar como, para el desarrollo del trabajo que se plantea en la asignatura se deben emplear softwares comerciales como Ansys y Matlab. Esto resulta de gran utilidad ya que, en muchas ocasiones, para la resolución de los problemas de fatiga los métodos requieren de ciclos iterativos cuya resolución tradicional (sin ordenador) resulta inviable.

Por último, destacar que este proyecto se ha llevado a cabo en el entorno docente del Departamento de Ingeniería Mecánica de la Escuela de Ingeniería de Bilbao. De esta forma, se ha podido elaborar el trabajo en un marco ingenieril donde, aplicando los diferentes conocimientos que se han ido adquiriendo durante el transcurso del Máster, se han podido solventar los infortunios que han ido sucediendo.

3 Objetivos y alcance del trabajo

3.1 Objetivo

El objetivo principal del presente proyecto es el desarrollo de una aplicación en donde se particularicen los métodos de fatiga multiaxial a los casos de fatiga uniaxial pura y de torsión pura. Para ello, se requiere un software que, además de soportar las expresiones matemáticas necesarias, permita al usuario interactuar con la app por lo que, todo ello se pretende implementar en el diseñador de aplicaciones "App Designer" del que dispone el software comercial Matlab.

Con el desarrollo de esta aplicación, se desea conseguir una herramienta que facilite la identificación del método de fatiga multiaxial más conveniente junto a sus parámetros característicos. Para conseguir esto, no se van a requerir ensayos de fatiga multiaxial que, como se ha mencionado en el apartado 1. Introducción, son costosos tanto desde el punto de vista temporal como desde el económico. Así pues, en lugar de trabajar con los resultados de ensayos de fatiga multiaxial, se van a emplear los resultados de ensayos de fatiga uniaxial pura y de torsión pura para tratar de deducir de forma intuitiva el método más conveniente.

Junto con esto, se aspira a desarrollar una aplicación que sirva como complemento de la asignatura de *Métodos de Análisis y Diseño para Fractura y Fatiga*. Una parte importante de la asignatura se enfoca primero hacia el método de fatiga uniaxial y después, se introduce en aspectos relacionados con la fatiga multiaxial como una continuación de los primeros. Sin embargo, no se particularizan los métodos de fatiga multiaxial como se realiza en este trabajo por lo que, resulta interesante para complementar los conceptos.

Por otra parte, dado que se va a tratar de una aplicación con la que el usuario va a tener que interactuar, otro objetivo a destacar es que la app disponga de características ventajosas para éste. En este sentido, se pueden remarcar aspectos como la sencillez de su manejo, la posibilidad de instalarla en ordenadores que no dispongan de Matlab o la capacidad de almacenamiento; para que datos como los puntos de fallo de materiales que no se encuentran en la librería de la app, se guarden una vez importados por el usuario.

En conclusión, los objetivos fundamentales que se persiguen con la elaboración de este trabajo son, la particularización de los métodos de fatiga multiaxial a los casos de fatiga uniaxial pura y de torsión pura y, su posterior programación desarrollando una herramienta gráfica con el diseñador de aplicaciones "App Designer" que ofrece Matlab.

3.2 Alcance

Tal y como se ha mencionado anteriormente, la aplicación está enfocada a la particularización de los métodos de fatiga multiaxial para los casos de fatiga uniaxial pura y de torsión pura. Esto se debe a dos razones principales, por un lado, la mayoría de los fallos por fatiga en la realidad son del tipo multiaxial lo cual, evidencia su interés. Y, por otro lado, el realizar ensayos en casos genéricos de fatiga multiaxial resulta más complejo y costoso que el hacer lo propio en los casos de fatiga uniaxial pura o de torsión pura.

Por otra parte, en cuanto a los métodos que se van a programar en la aplicación, se pueden dividir en dos grandes grupos: métodos de enfoque global y de plano crítico. Dentro de los métodos de enfoque global se van a introducir Sines y Crossland mientras que, pertenecientes a los métodos de plano crítico son: Matake, Findley, Dang Van y Papuga PCr. La elección de dichos métodos está basada principalmente en que todos ellos corresponden a métodos contrastados y conocidos dentro de dicho ámbito.

En cada uno de los métodos que se han mencionado se tienen dos parámetros característicos que dependen de los ensayos que se hayan empleado para su obtención. Así pues, excepto en el caso de Papuga PCr donde la alternativa de parámetros depende de si el material es dúctil o frágil, en el resto de métodos los resultados son fuertemente dependientes de la pareja de ensayos empleada. Así pues, en este trabajo se ha optado por trabajar con las tres siguientes parejas: $\sigma_{-1} - \tau_{-1}$, $\sigma_{-1} - \sigma_0$ y $\sigma_{-1} - \sigma_{ut}$ donde, σ_{-1}, τ_{-1} y σ_{ut} corresponden al fallo en el ensayo de flexión alterna, tracción pulsante y tracción estática, respectivamente. El empleo de dichos ensayos se fundamenta principalmente en que son casos de carga sencillos en los que se conocen los valores de fallo.

Todos estos métodos dan lugar a una serie de gráficas que se van a representar sobre dos gráficos normalizados: por un lado, ${\sigma_a}/{\sigma_{-1}} - {\sigma_m}/{\sigma_{ut}}$ y, por otro lado, ${\tau_a}/{\tau_{-1}} - {\tau_m}/{\tau_{ut}}$. Asimismo, además de las curvas correspondientes a los citados métodos de fatiga multiaxial particularizados, se va a trabajar con los criterios de ajuste más frecuentes en fatiga uniaxial con tensiones medias y materiales dúctiles: Goodman, Dietmann, Gerber y la elipse de Marin y, se van a extrapolar al estado tensional de torsión.

Por último, además de los datos de fallo correspondientes a los materiales que se van a incorporar por defecto en la propia aplicación, se pretende tener la opción de importar nuevos datos de otros materiales. Sin embargo, por razones de espacio en la interfaz de la aplicación, el número máximo de materiales que se podrán adjuntar simultáneamente será 22 pero, como el usuario va a tener la opción de restablecer los materiales, va a ser capaz de sustituir los datos previos por los nuevos que desee incorporar.

4 Beneficios

En los siguientes subapartados se muestran los beneficios que se consiguen mediante el desarrollo del presente proyecto. Éstos se dividen en tres grandes grupos: técnicos, económicos y sociales que, se van a explicar con mayor detalle a continuación.

4.1 Técnicos

Con el desarrollo de la presente aplicación, se pretende emplear los resultados obtenidos de los ensayos de fatiga uniaxial y de torsión pura con el fin de identificar el método de fatiga multiaxial que mejor se ajusta al estado tensional. Esto se hace para evitar tener que realizar los ensayos de fatiga multiaxial ya que, además de la complejidad que los caracteriza, sobre ellos influyen aspectos adicionales que aumentan la incertidumbre de los resultados; como la variación de las direcciones principales que, no interviene en los ensayos de fatiga uniaxial.

4.2 Económicos

Tal y como se ha mencionado, en lugar de trabajar con los ensayos de fatiga multiaxial, se van a emplear los resultados de fatiga uniaxial y de torsión pura para tratar de identificar el método de fatiga multiaxial más adecuado. Así, además de reducir la complejidad de los ensayos, se pueden conseguir importantes beneficios económicos por las dos siguientes razones.

Por un lado, si se comparan los métodos de fatiga multiaxial con los de fatiga uniaxial o de torsión pura, por ejemplo, en cuanto a logística del ensayo, se tratan de experimentos más caros (máquinas especiales, etcétera). Por otro lado, el tiempo y, por tanto, el coste económico que se emplea para llevar a cabo este tipo de ensayos también es considerablemente superior al de los ensayos de fatiga uniaxial y de torsión pura; de forma que, el evitarlos redunda en beneficios económicos.

Por último, se destaca también el carácter gratuito de la aplicación. Esto permite a todos aquellos usuarios que lo deseen, tanto personal relacionado con la docencia (profesores y alumnos) como cualquier otra persona particular, disponer de la herramienta en su propio ordenador para poder realizar los análisis que considere oportunos.

4.3 Sociales

Los beneficios sociales que se pueden extraer del trabajo están relacionados principalmente con la facilitación de una herramienta complementaria a los alumnos de la asignatura de *Métodos de Análisis y Diseño para Fractura y Fatiga*. Dentro del temario de ésta, es habitual comenzar por la fatiga uniaxial y después tratar la fatiga multiaxial como si fuese una continuación de lo anterior. Sin embargo, tras terminar con la fatiga multiaxial, es interesante hacer el inciso sobre la particularización de dichos métodos en la fatiga uniaxial y de torsión pura para que los alumnos asimilen, comprendan y relacionen mejor los conceptos que se han tratado.

En este sentido, mediante la presente aplicación se pone al alcance del profesorado una herramienta que facilite dicha docencia. Así, se va a conseguir que los alumnos, además de percibir los fundamentos teóricos, sean capaces de visualizar las diferentes representaciones que se pueden conseguir e interactuar con la app. Como consecuencia de todo ello, en muchas ocasiones se van a comprender y relacionar mejor los conceptos, lo cual hará que aumente su interés por la asignatura y esto redundará en una mejora de sus calificaciones.

5 Descripción de requerimientos y estado del arte

En este apartado, se van a exponer conjuntamente los requerimientos que deberá recoger la aplicación que se desea desarrollar y el estado del arte de los conceptos teóricos que se necesitan para poder llevarla a cabo. A continuación, se muestra en rasgos generales los asuntos que se van a tratar en cada una de las dos partes.

En primer lugar, en el subapartado 5.1. Requerimientos de diseño, se van a enumerar los requisitos con los que se desea identificar la aplicación. Dentro de dicho grupo se van a adjuntar aquellos aspectos que hagan de la aplicación una herramienta más práctica e intuitiva para el usuario.

Tras esto, se van a explicar con mayor detalle los conceptos teóricos que se desean introducir en la aplicación junto con su correspondiente estado del arte ya que, se consideran fundamentales para la correcta comprensión del desarrollo de la app. Así pues, en este subapartado se van a distinguir cuatro puntos: consideraciones generales, métodos de enfoque global, métodos de plano crítico y métodos de resolución.

5.1 Requerimientos de diseño

Independientemente del perfil del usuario que vaya a emplear la aplicación que se desea desarrollar, a fin de conseguir una herramienta útil y que sirva de apoyo, se pretende que disponga de las cualidades que se van a citar a continuación.

- **Conceptos teóricos:** Se pretende que la aplicación abarque los conceptos teóricos que se han mencionado en el subapartado 3.2. Alcance. Entre ellos, dado que constituyen el núcleo del trabajo, se presta especial interés en los métodos de fatiga multiaxial profundizando en ellos en el siguiente subapartado, 5.2. Conceptos teóricos.

- **Detección de errores:** Para que la herramienta que se presente resulte útil y válida no debe incluir errores. En este sentido, es conveniente que incorpore mensajes de alerta y limitantes que adviertan al usuario sobre sus acciones.

- **Facilidad de ejecución:** Se desea disponer de una aplicación capaz de ejecutarla en cualquier ordenador sin necesidad de tener instalado el programa Matlab.

- **Interfaz sencilla:** se busca que la app sea manejable y sencilla para el usuario; de forma que toda persona en disposición de utilizarla identifique la función de cada uno de los comandos sin que esto le suponga una dedicación excesiva de tiempo.

- **Desarrollo secuencial:** De esta forma, se pretende facilitar la cronología de las tareas que debe llevar a cabo el usuario. Esta característica adquiere especial interés cuando se comienza a manejar la herramienta ya que todavía no se está habituado con su interfaz.

- **Representaciones simultáneas:** Se trata de la característica principal de la aplicación. Con esto se busca ser capaz de representar tantos métodos como el usuario desee, junto con los puntos de fallo de los materiales que se seleccionen, para facilitar la elección del método de fatiga multiaxial que mejor se ajuste.

- **Almacenamiento de datos:** Además de los materiales que va a disponer la aplicación por defecto, se pretende que el usuario tenga la opción de importar nuevos materiales.

- **Fluidez:** Sin que el usuario precise de un ordenador de grandes prestaciones, se busca que los diferentes comandos se ejecuten con cierta rapidez para que se pueda interactuar con la herramienta de un modo dinámico.

5.2 Conceptos teóricos

Tal y como se ha comentado al comienzo del apartado, este punto se enfoca principalmente a la exposición de los conceptos teóricos referentes a los métodos avanzados de fatiga multiaxial que se incorporan en la aplicación. En este sentido, se va a comenzar tratando aspectos comunes a ambos tipos de métodos, se continuará con la descripción de los métodos de enfoque global y de plano crítico que se recogen en el alcance del trabajo y, por último, se presentarán los métodos de resolución de ecuaciones que se van a emplear.

5.2.1 Consideraciones generales

Entre las características que se pueden considerar comunes a ambos tipos de métodos de fatiga multiaxial, se considera que aquellas que resultan más relevantes para el seguimiento del trabajo son, por un lado, la influencia de la tensión cortante media y, por otro lado, la base de la función de daño empleada en los diferentes métodos. Es por ello que en adelante se va a tratar sobre éstos temas.

5.2.1.1 Influencia de la tensión cortante media

Al igual que ocurre con el estudio de la fatiga multiaxial en líneas generales, existe controversia sobre la influencia de la tensión cortante media (τ_m) en el comportamiento a fatiga [2]. Así pues, mientras que los métodos clásicos de fatiga multiaxial consideran por principio la τ_m junto con la tensión cortante alterna (τ_a), los métodos avanzados las diferencian para, en varios casos, despreciar su efecto.

Para poder afirmar con cierto grado de seguridad si la τ_m afecta sobre el material con el que se desea trabajar, al igual que se ha comentado previamente en el subapartado 1.4.2. Fatiga multiaxial, se deben ensayar probetas cilíndricas a torsión, torsión junto con axial o torsión junto con flexión. Sin embargo, por diferentes motivos esto no siempre se puede realizar, por lo que, para predecir dicha influencia, cada autor propone sus argumentos.

Por un lado, hay autores que defienden que mientras la tensión cortante alterna no supere el límite de fatiga a cortadura (τ_{-1}) la τ_m no afecta a la resistencia a fatiga del material. Un ejemplo práctico de esto se puede encontrar en los ensayos realizados por Sines; cuyo resultado se muestra sobre la Figura 5.1 y, como se puede apreciar, solamente cuando la τ_a supera a la τ_{-1} influye la τ_m.

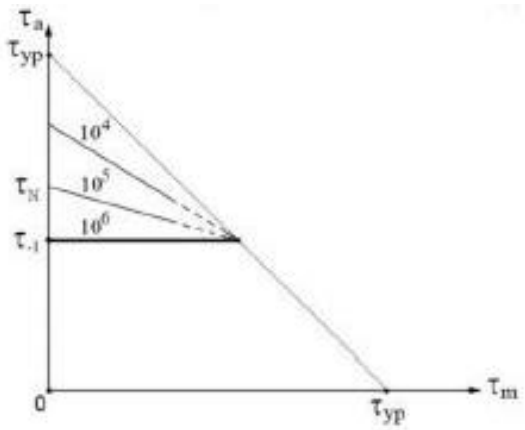

Figura 5.1. Influencia de la τ_m según Sines [1].

Asimismo, hay otros que también llegan a la misma conclusión que Sines, pero en base a otra premisa. Ésta se encuentra relacionada con la fase de iniciación de la grieta que se ha comentado en el subapartado 1.3.2. Fases del fallo por fatiga ya que como la grieta se inicia por el deslizamiento entre planos cristalinos debido a la τ_a, concluyen que el comportamiento del material es independiente de la τ_m.

Por otro lado, se tienen los autores que justifican como el efecto de la τ_m no se puede considerar despreciable. Sobre la Figura 5.2 se muestran los resultados obtenidos en los ensayos realizados por Findley para el Aluminio 76S-T61 y por Wang y Miller para un acero Cr-Ni-Mo en donde se demuestra que la τ_m hay que considerarla.

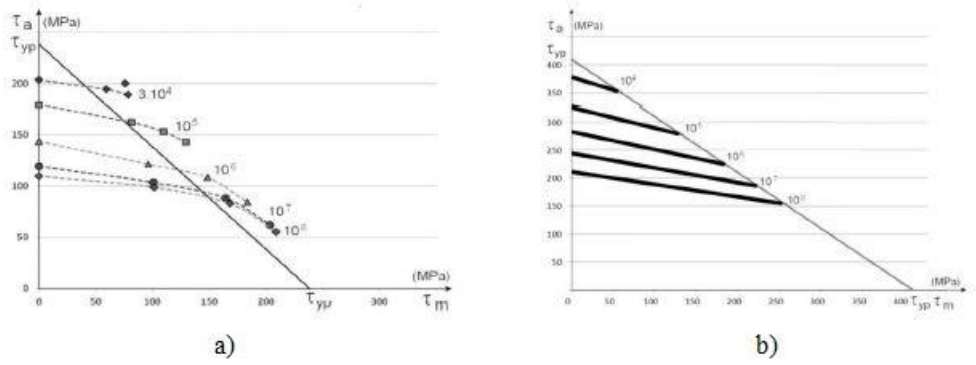

Figura 5.2. Influencia de la τ_m según: a) Findley, b) Wang y Miller [1].

Tras esto, se pueden resumir las siguientes conclusiones sobre el efecto de la τ_m en la resistencia del material. Por un lado, su influencia depende del material pero independientemente de éste, siempre predomina la σ_m frente a la τ_m ya que, como se puede apreciar en la Figura 5.2 la pendiente de las curvas que consideran su efecto no es muy elevada. Por otro lado, a la vista de los ensayos, τ_m siempre afecta a vida finita y cuanto mayor sea el estado tensional, mayor es su influencia ya que la pendiente de las curvas aumenta progresivamente.

Para finalizar, cabe decir que entre los métodos de fatiga multiaxial que se van a tratar en este trabajo y que no consideran el efecto de la τ_m se encuentran Sines, Crossland, Matake y Dang Van mientras que, los restantes (Findley y Papuga) sí lo consideran.

5.2.1.2 Funciones de daño

Después de tratar sobre la controversia que hay en torno a la influencia de la τ_m en la resistencia a fatiga, se va a proceder a explicar lo referente a las funciones de daño que emplean los métodos avanzados para determinar si la pieza va a tener vida infinita. Así pues, se define la función de daño como una combinación de las componentes del tensor de tensiones que proporciona el daño a fatiga (d) [2]. Por lo tanto, para que no se produzca el fallo de la pieza que se está analizando, se tiene que cumplir que dicho valor sea inferior al daño que produciría el fallo por fatiga (D), como se expresa en la ecuación (5.1).

$$f([\sigma(t)]) = d < D \tag{5.1}$$

Esto es válido conceptualmente, pero en la realidad no se conoce ni el valor de "d" ni el de "D" por lo que se trata de aproximar el comportamiento mediante funciones que se ajusten a resultados previos que se han obtenido de ensayos. Como se va a desarrollar con más detalle a continuación, el número de parámetros que se incluyen en la función de daño va a depender del tipo de método que se trate.

En el caso de los métodos de fatiga multiaxial que no consideran la influencia de la τ_m, la función de daño se suele definir como combinación lineal de un término relacionado con las tensiones normales ($f_1(\sigma(t))$) y de otro término ligado a las tensiones cortantes alternas ($f_2(\tau_a(t))$). Esto se puede expresar matemáticamente mediante la ecuación (5.2); donde α y β corresponden a parámetros del propio método.

$$\alpha \cdot f_1\big(\sigma(t)\big) + \beta \cdot f_2\big(\tau_a(t)\big) \leq 1 \tag{5.2}$$

Asimismo, también se puede expresar la función de daño del modo que se recoge en la ecuación (5.3). En ella se define una tensión equivalente que debe ser inferior a la tensión σ_{fallo} para que no se produzca el fallo del material.

$$\sigma_{eq} = \alpha \cdot f_1\big(\sigma(t)\big) + f_2\big(\tau_a(t)\big) \leq \sigma_{fallo} \tag{5.3}$$

Independientemente de la forma en la que se exprese la función de daño, en ambos casos se tienen dos parámetros propios del método como son α y β en la expresión (5.2) y, α y σ_{fallo} en la expresión (5.3). Los valores de dichos parámetros se obtienen de ensayos en los que se conocen los valores del límite de fatiga, siendo los más habituales el ensayo de tensión alterna R_{-1} (σ_{-1}, por ejemplo, flexión alterna con la probeta rotatoria de Moore), tensión pulsante (σ_0), torsión alterna (τ_{-1}) o el ensayo de tracción estática (σ_{ut}). Por lo tanto, sustituyendo, para cada ensayo, los resultados obtenidos en la expresión empleada y resolviendo el sistema de dos ecuaciones con dos incógnitas se despejan los valores de los parámetros del método.

Por otro lado, cabe remarcar que el objeto de estas funciones es determinar si la pieza tendrá duración infinita o si, por el contrario, se va a producir el fallo. Sin embargo, en este último supuesto, en ocasiones se emplea la tensión equivalente obtenida del método en cuestión para determinar el número de ciclos que durará, siguiendo un procedimiento semejante al de la curva SN que se ha explicado en el subapartado 1.4.1. Fatiga uniaxial. Pese a ello, se debe ser consciente de que conceptualmente no es correcto ya que como se ha explicado en el subapartado anterior (5.2.1.1. Influencia de la tensión cortante media), la τ_m influye en la zona de vida finita.

En cuanto a los métodos que consideran la influencia de la τ_m, también se pueden distinguir dos formas de expresar la función de daño. Por un lado, se encuentran aquellas expresiones que se formulan de manera semejante a la ecuación (5.2) pero teniendo en cuenta que el término de la tensión normal se está considerando el efecto de τ_m ya que se obtiene a partir de todas las componentes del tensor de tensiones. La otra opción, consta de introducir en la función de daño un tercer componente($f_3(\tau_m(t))$) referente a la tensión cortante media como se aprecia a continuación.

$$\alpha \cdot f_1\big(\sigma(t)\big) + \beta \cdot f_2\big(\tau_a(t)\big) + \gamma \cdot f_3\big(\tau_m(t)\big) \leq 1 \tag{5.4}$$

Dado que se necesitan tantos ensayos como parámetros se empleen en el método, en este caso se requieren los límites de fatiga de tres ensayos debido a que el método consta de los tres siguientes parámetros: α, β y γ. Junto con esto, cabe decir que en función de los ensayos empleados los resultados varían; de forma que la solución correcta es aquella que mejor se ajuste al comportamiento real del material en cuestión.

5.2.2 Métodos de enfoque global

Tras analizar las diferentes consideraciones referentes a los métodos avanzados de fatiga multiaxial, en este subapartado se va a tratar sobre los métodos de enfoque global. Así pues, se va a comenzar describiendo de forma general las características de dichos métodos y a continuación, se va a profundizar en los métodos de Sines y de Crossland ya que son los que se encuentran dentro del alcance del presente trabajo.

Tal y como se verá con el desarrollo de los métodos de Sines y Crossland, este grupo de métodos se caracteriza por estudiar la fatiga a nivel de punto utilizando las tensiones principales; de ahí que no consideran el efecto de la variación de las direcciones principales. Sin embargo, son válidos con tensiones cuasi hidrostáticas y hay algunos que tienen en cuenta el efecto de la tensión cortante media por lo que, su rango de aplicación es más amplio que el de los métodos clásicos.

5.2.2.1 Método de Sines

Uno de los métodos de fatiga multiaxial más conocido es el método de Sines que, pertenece al grupo de métodos de enfoque global. Según este, se define la función de daño como combinación lineal de la tensión octaédrica alterna (τ_{oct_a}) y la tensión hidrostática media (σ_{h_m}) como se aprecia en la siguiente expresión.

$$\sigma_{eq_S} = \tau_{oct_a} + \alpha_S \cdot \sigma_{h_m} \geq \beta_S \rightarrow (Fallo) \tag{5.5}$$

Donde:

$$\tau_{oct_a} = {}^1\!/_3 \cdot \sqrt{\left(\sigma_{1_a} - \sigma_{2_a}\right)^2 + \left(\sigma_{2_a} - \sigma_{3_a}\right)^2 + \left(\sigma_{3_a} - \sigma_{1_a}\right)^2} \tag{5.6}$$

$$\sigma_{h_m} = {}^1\!/_3 \cdot (\sigma_{1_m} + \sigma_{2_m} + \sigma_{3_m}) \tag{5.7}$$

A la vista de la ecuación (5.5) se deduce que el método de Sines no considera la tensión cortante media; lo cual concuerda con los resultados obtenidos de sus ensayos que, se han explicado en el subapartado 5.2.1.1. Influencia de la tensión cortante media. Asimismo, se puede apreciar como la ecuación que define la τ_{oct_a} es semejante a la de Von Mises que se ha presentado en la expresión (1.18) por lo que, en ocasiones se formula Sines en función de Von Mises [13] (ver ecuación (5.8)).

$$\sigma_{eq_S}{}^{VM} = \sigma_{eq_a}{}^{VM} + \alpha_S{}^{VM} \cdot \sigma_{h_m} \geq \beta_S{}^{VM} \rightarrow (Fallo) \tag{5.8}$$

Como se puede apreciar, la relación entre la ecuación (5.5) y la (5.8) no es más que multiplicar a la primera por $^{3}/_{\sqrt{2}}$ de modo que, las constantes del método también cumplen la misma relación.

$$\alpha_S{}^{VM} = {}^{3}/_{\sqrt{2}} \cdot \alpha_S \tag{5.9}$$

$$\beta_S{}^{VM} = {}^{3}/_{\sqrt{2}} \cdot \beta_S \tag{5.10}$$

Independientemente de la forma que se emplee para expresar la función de daño, se van a obtener los parámetros del método a partir de dos ensayos de probeta cuyos límites de fatiga sean conocidos. Habitualmente, se suelen emplear los que se citan a continuación:

- Ensayos de flexión con tensión alterna, Se obtiene σ_{-1}, por ejemplo, a partir del ensayo con la probeta rotatoria de Moore.
- Ensayo de torsión alterna. Se obtiene τ_{-1}.
- Ensayo con tracción pulsante axial. Se obtiene σ_0.
- Ensayo de tracción uniaxial estática. Se obtiene σ_{ut}.

Cabe mencionar como, a pesar de que se esté trabajando en fatiga, entre los ensayos que se pueden emplear se encuentra el de tracción uniaxial estática. Esto está relacionado principalmente con la coherencia del método respecto a la cual, se adelanta que Sines es coherente para los cuatro ensayos; es decir, independientemente de la pareja de ensayos empleada, las constantes del método no varían para materiales dúctiles. A continuación, se particularizan los ensayos que se han mencionado para obtener los parámetros del método.

Ensayo de flexión con tensión alterna

Se tiene un estado tensional como el de la Figura 5.3 a) cuyo fallo se produce si la tensión alterna alcanza el límite de fatiga σ_{-1}.

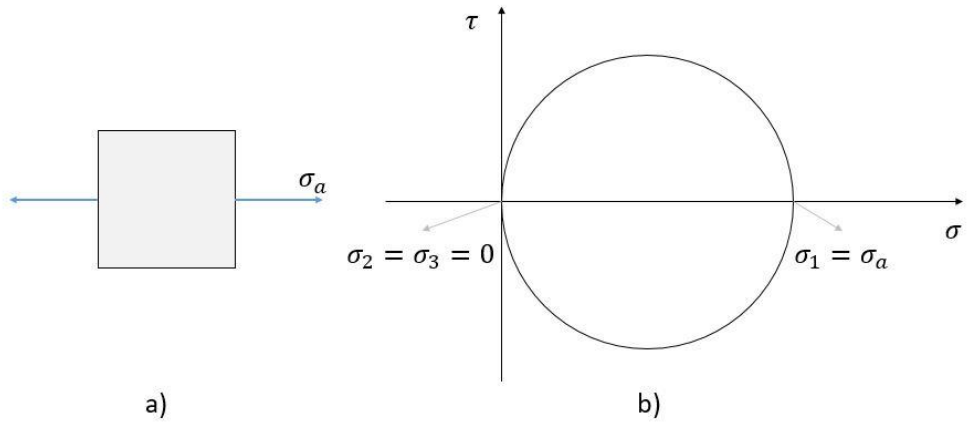

Figura 5.3. Ensayo de flexión alterna: a) Estado tensional, b) Círculo de Mohr.

Se plantea el círculo de Mohr como se muestra en la Figura 5.3 b) y a partir de las ecuaciones (5.6) y (5.7) se obtienen los valores de τ_{oct_a} y σ_{h_m}. Por último, sustituyendo los resultados obtenidos sobre la expresión inicial (ecuación (5.5)), se despeja la constante β_S.

$$\tau_{oct_a} = \frac{1}{3} \cdot \sqrt{(\sigma_a)^2 + (\sigma_a)^2} = \frac{\sqrt{2}}{3} \cdot \sigma_a \tag{5.11}$$

$$\sigma_{h_m} = \frac{1}{3} \cdot (0) = 0 \tag{5.12}$$

$$\frac{\sqrt{2}}{3} \cdot \sigma_{-1} + \alpha_S \cdot 0 = \beta_S \rightarrow \beta_S = \frac{\sqrt{2}}{3} \cdot \sigma_{-1} \tag{5.13}$$

Ensayo de torsión alterna

En este caso, el estado tensional es como el de la Figura 5.4 a) y el fallo se produce cuando la tensión cortante alterna alcanza τ_{-1}.

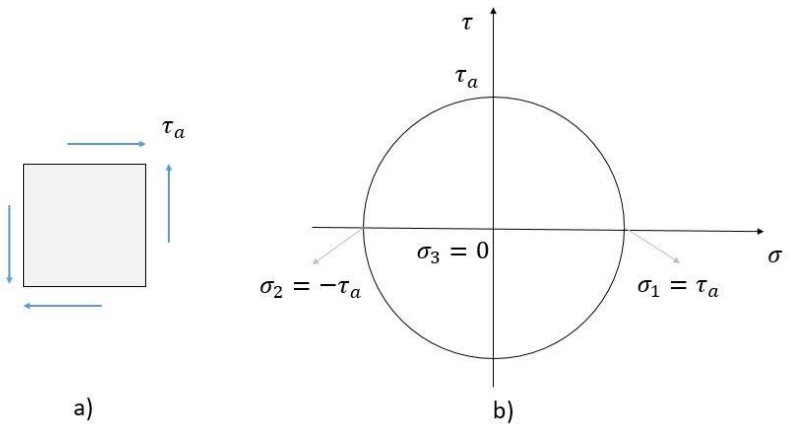

Figura 5.4. Ensayo de torsión alterna: a) Estado tensional, b) Círculo de Mohr.

Se realiza la misma secuencia de operaciones expuesta para el ensayo de flexión alterna pero en base al círculo de Mohr de la Figura 5.4 b).

$$\tau_{oct_a} = \frac{1}{3} \cdot \sqrt{(\tau_a - (-\tau_a))^2 + (\tau_a)^2 + (-\tau_a)^2} = \frac{\sqrt{6}}{3} \cdot \tau_a \tag{5.14}$$

$$\sigma_{h_m} = \frac{1}{3} \cdot (0) = 0 \tag{5.15}$$

$$\frac{\sqrt{6}}{3} \cdot \tau_{-1} + \alpha_S \cdot 0 = \beta_S \rightarrow \beta_S = \sqrt{\frac{2}{3}} \cdot \tau_{-1} \tag{5.16}$$

En principio, de acuerdo a las ecuaciones (5.13) y (5.16) se puede decir que el parámetro β_S adquiere valores diferentes dependiendo del ensayo que se emplee; sin embargo, a partir de la relación entre τ_{-1} y σ_{-1} que se puede deducir de la ecuación de Von Mises (ec. (1.18)) se demuestra cómo corresponden a la misma.

$${\sigma_{eq}}_a^{VM} = \sqrt{\frac{1}{2}[(\tau_a - (-\tau_a))^2 + (\tau_a)^2 + (-\tau_a)^2]} = \sqrt{3}\tau_{-1} \rightarrow \sigma_{-1} = \sqrt{3}\tau_{-1} \tag{5.17}$$

Sustituyendo en la expresión (5.16):

$$\beta_S = \sqrt{\frac{2}{3}} \cdot \tau_{-1} \rightarrow \beta_S = \sqrt{\frac{2}{3}} \cdot \frac{1}{\sqrt{3}} \sigma_{-1} \rightarrow \beta_S = \frac{\sqrt{2}}{3} \cdot \sigma_{-1} \tag{5.18}$$

Ensayo con tracción pulsante axial

El estado tensional corresponde al de la Figura 5.5 a) y el fallo se da cuando la tensión ($\sigma_m = \sigma_a = \sigma$) alcanza el límite de fatiga pulsante axial (σ_0).

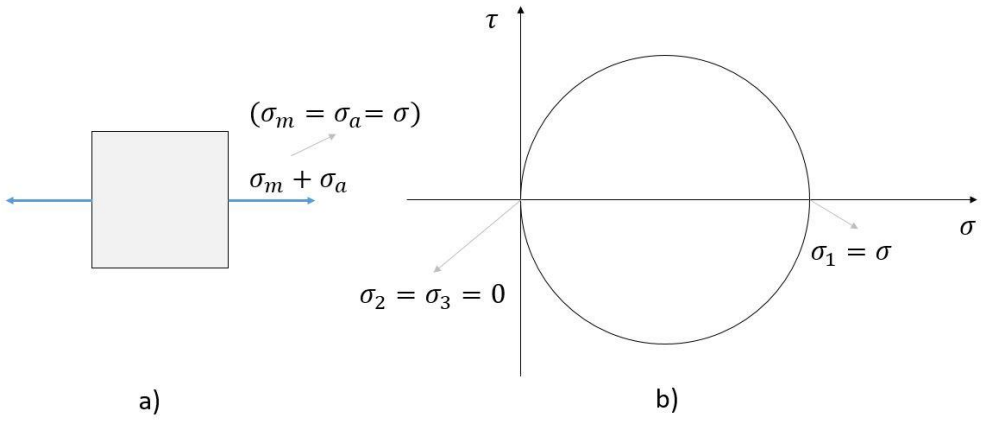

Figura 5.5. Ensayo de tracción pulsante: a) Estado tensional, b) Círculo de Mohr.

Siguiendo la misma sistemática que en los casos anteriores, se obtiene la relación entre las constantes α_S y β_S.

$$\tau_{oct_a} = \frac{1}{3} \cdot \sqrt{(\sigma)^2 + (-\sigma)^2} = \frac{\sqrt{2}}{3} \cdot \sigma \tag{5.19}$$

$$\sigma_{h_m} = \frac{1}{3} \cdot (\sigma) = \frac{\sigma}{3} \tag{5.20}$$

$$\frac{\sqrt{2}}{3} \cdot \sigma_0 + \alpha_S \cdot \frac{\sigma_0}{3} = \beta_S \tag{5.21}$$

Sustituyendo el valor de β_S sobre la ecuación (5.21) se despeja el valor de α_S en función del límite de fatiga de flexión alterna y del límite de fatiga de tracción pulsante axial.

$$\frac{\sqrt{2}}{3} \cdot \sigma_0 + \alpha_S \cdot \frac{\sigma_0}{3} = \frac{\sqrt{2}}{3} \cdot \sigma_{-1} \quad \rightarrow \quad \alpha_S = \frac{\sqrt{2} \cdot (\sigma_{-1} - \sigma_0)}{\sigma_0} \tag{5.22}$$

Ensayo de tracción uniaxial estática

En la Figura 5.6 a) se muestra el estado tensional del ensayo, cuyo fallo se produce cuando la tensión media alcanza la tensión última de tracción (σ_{ut}).

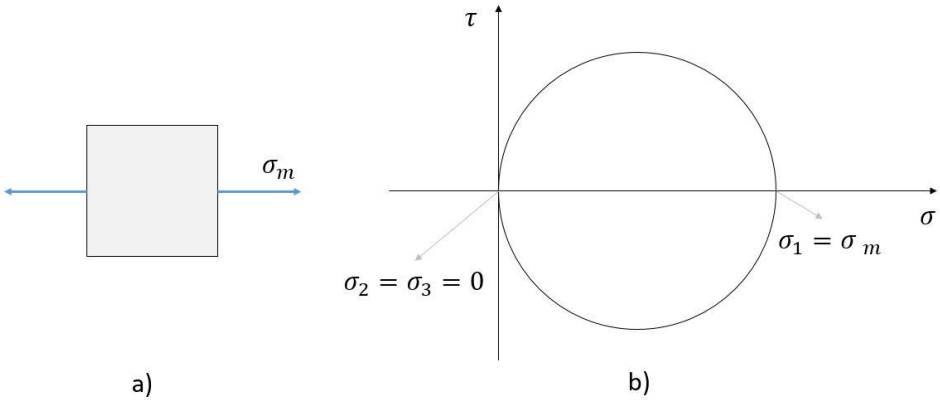

Figura 5.6. Ensayo de tracción uniaxial estática: a) Estado tensional, b) Círculo de Mohr.

Utilizando el mismo procedimiento descrito para los ensayos anteriores, se deduce el parámetro α_S previa sustitución de β_S.

$$\tau_{oct_a} = \frac{1}{3} \cdot \sqrt{0} = 0 \tag{5.23}$$

$$\sigma_{h_m} = \frac{1}{3} \cdot (\sigma) = \frac{\sigma}{3} \tag{5.24}$$

$$0 + \alpha_S \cdot \frac{\sigma_{ut}}{3} = \beta_S \rightarrow \alpha_S \cdot \frac{\sigma_{ut}}{3} = \frac{\sqrt{2}}{3} \cdot \sigma_{-1} \;\rightarrow\; \alpha_S = \frac{\sqrt{2} \cdot \sigma_{-1}}{\sigma_{ut}} \tag{5.25}$$

Una vez obtenidos los parámetros del método, se está en disposición de formular la expresión de la función de daño para esta pareja de ensayos ($\sigma_{-1} - \sigma_{ut}$). Empleando los resultados obtenidos de los ensayos de flexión alterna y de tracción uniaxial estática y sustituyendo en la ecuación (5.5), se llega a la siguiente expresión.

$$\tau_{oct_a} + \frac{\sqrt{2} \cdot \sigma_{-1}}{\sigma_{ut}} \cdot \sigma_{h_m} \geq \frac{\sqrt{2}}{3} \cdot \sigma_{-1} \;\rightarrow\; (Fallo) \tag{5.26}$$

Asimismo, haciendo lo propio en la forma semejante a Von Mises (ecuación (5.8)) se obtiene la función de daño equivalente.

$$\sigma_{eq_S}{}^{VM} = \sigma_{eq_a}{}^{VM} + \left(\frac{\sigma_{-1}}{\sigma_{ut}}\right) \cdot \sigma_{eq_m}{}^{VM} \geq \sigma_{-1} \rightarrow (Fallo) \tag{5.27}$$

Donde:

$$\sigma_{eq_m}{}^{VM} = 3 \cdot \sigma_{h_m} \tag{5.28}$$

Para finalizar, se va a describir el procedimiento que se emplea en ocasiones para determinar la duración a vida finita del componente, N. Sin embargo, lo primero que se debe decir es que conceptualmente es erróneo ya que, como se ha comentado previamente, la tensión cortante media sí influye a vida finita y, por tanto, habría que considerarlo, pero Sines no lo hace. Pese a ello, a veces se utiliza para realizar cálculos preliminares u orientativos aun sabiendo que no son correctos. Así pues, sobre la Figura 5.7 se muestra esquemáticamente el procedimiento a seguir para obtener N.

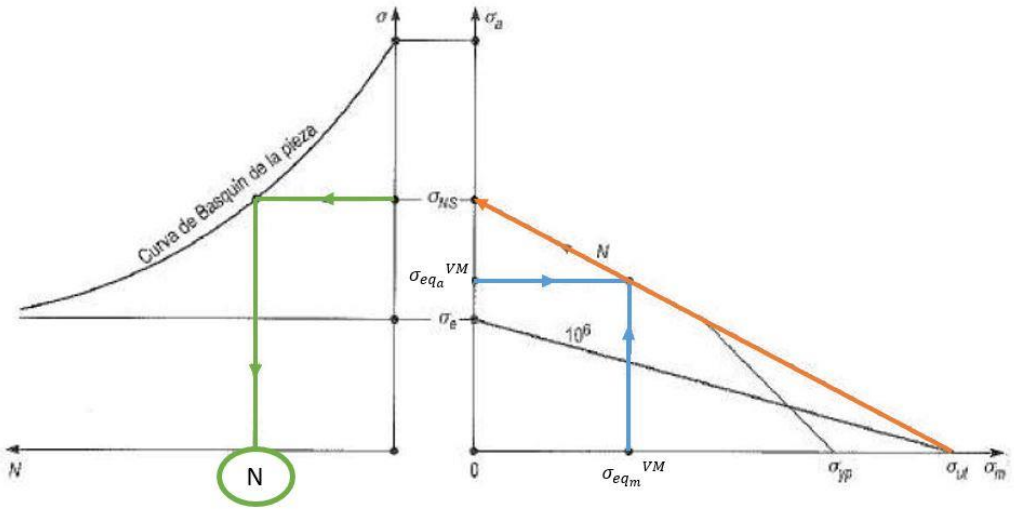

Figura 5.7. Obtención de la duración de la pieza partiendo del método de Sines.

5.2.2.2 Método de Crossland

Tras explicar el método de Sines, el segundo método de enfoque global que se va a presentar es el método de Crossland. Su procedimiento es semejante al de Sines, pero en lugar de utilizar la tensión hidrostática media emplea la máxima ($\sigma_{h_{max}}$). Así, la expresión matemática que representa la función de daño corresponde la que se muestra a continuación.

$$\sigma_{eq_C} = \tau_{oct_a} + \alpha_C \cdot \sigma_{h_{max}} \geq \beta_C \rightarrow (Fallo) \tag{5.29}$$

Donde:

$$\sigma_{h_{max}} = \sigma_{h_m} + \sigma_{h_a} \tag{5.30}$$

Además de la forma empleada en la ecuación (5.29), también existen otras maneras de formular la función de daño en el método de Crossland; como por ejemplo, utilizar el segundo invariante del tensor de tensiones (J_2) o por analogía con Von Mises. Sin embargo, en este proyecto se va a trabajar con la primera forma por lo que se hace hincapié en ella.

Para finalizar, cabe destacar como a diferencia del método de Sines, en el método de Crossland los valores de las constantes que intervienen en la función de daño varían dependiendo de la pareja de ensayos empleados. Así pues, tal y como se va a describir en el apartado 9. Cálculos, los resultados son fuertemente dependientes de los ensayos con los que se trabaje.

5.2.3 Métodos de plano crítico

Una vez finalizada la exposición referente a los métodos de enfoque global, se va a hacer lo propio con los métodos de plano crítico. Para ello, la sistemática que se va a seguir es la siguiente: se comienza describiendo la base y las características del método y a continuación, se exponen de forma particular los métodos de Matake, Findley, Papuga y Dang Van ya que son los que corresponden al alcance del presente trabajo.

La principal característica que distingue a los métodos de plano crítico frente al resto es la capacidad que tiene para considerar el efecto de la variación de las direcciones principales. Esto hace que el campo de aplicación de los métodos de fatiga de plano crítico sea mayor que el de los otros métodos, pero en contraposición, el coste de cálculo que acarrean es notablemente superior.

La base de los métodos de plano crítico se encuentra en que, para realizar el análisis del comportamiento a fatiga del material en un punto, hay que trabajar con el plano más desfavorable que contiene a dicho punto; es decir, con el denominado plano crítico. La definición de dicho plano no es directa y, como se verá más adelante, cada método emplea su criterio. Para ello, en las expresiones que se formulan se hace referencia a las componentes del vector de tensión por lo que, se va a comenzar deduciendo su obtención apoyándose en la Figura 5.8.

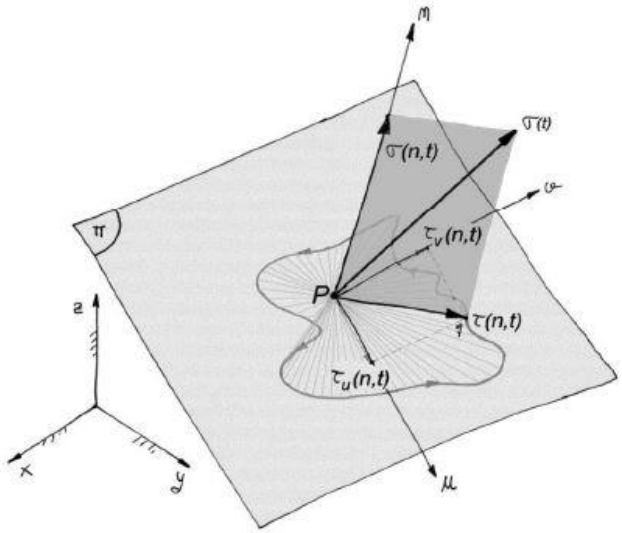

Figura 5.8. Tensiones en el punto P según el plano π [1].

Sobre la Figura 5.8 se representa un plano genérico, denominado π, que se va a considerar como el plano crítico del punto P. Así, se define el sistema de coordenadas (u, v, n) siendo "n" el eje normal al plano π y que pasa por el punto P y, "u" y "v" dos ejes ortogonales contenidos en dicho plano con dirección arbitraria y que también pasan por P. Dicho esto, se puede definir el vector de tensiones según el plano π como sigue.

$$\{\sigma(t)\} = [\sigma(t)]\{n\} \tag{5.31}$$

Considerando el caso genérico en el que el tensor de tensiones corresponde a la ecuación (1.16) y el vector normal $\{n\}$ se define por sus tres componentes cartesianas (n_x, n_y, n_z), se deduce el vector de tensiones como se muestra en la siguiente expresión.

$$\{\sigma(t)\} = \begin{bmatrix} \sigma_{xx}(t) & \tau_{xy}(t) & \tau_{xz}(t) \\ \tau_{xy}(t) & \sigma_{yy}(t) & \tau_{yz}(t) \\ \tau_{xz}(t) & \tau_{yz}(t) & \sigma_{zz}(t) \end{bmatrix} \begin{Bmatrix} n_x \\ n_y \\ n_z \end{Bmatrix} = \begin{Bmatrix} \sigma_{xx}n_x + \tau_{xy}n_y + \tau_{xz}n_z \\ \tau_{xy}n_x + \sigma_{yy}n_y + \tau_{yz}n_z \\ \tau_{xz}n_x + \tau_{yz}n_y + \sigma_{zz}n_z \end{Bmatrix} \tag{5.32}$$

Donde, las componentes normal y tangencial del vector de tensiones son:

$$\{\sigma(n,t)\} = (\{n\}^T\{\sigma(t)\})\{n\} \tag{5.33}$$

$$\{\tau(n,t)\} = \{\sigma(t)\} - \{\sigma(n,t)\} \tag{5.34}$$

Asimismo, los módulos de dichos vectores son los siguientes.

$$|\sigma(n,t)| = \{\sigma(t)\}^T\{n\} \tag{5.35}$$

$$|\tau_u(n,t)| = \{\sigma(t)\}^T\{u\} \tag{5.36}$$

$$|\tau_v(n,t)| = \{\sigma(t)\}^T\{v\} \tag{5.37}$$

Por lo tanto, mediante el teorema de Pitágoras, se calcula el módulo de la tensión tangencial.

$$|\tau(n,t)| = \sqrt{|\tau_u(n,t)|^2 + |\tau_v(n,t)|^2} \tag{5.38}$$

Dicho esto, el siguiente concepto que se introduce sobre la representación es el de ciclo de fatiga multiaxial en un caso genérico con tensiones que varían en tres dimensiones con el tiempo, tal y como se puede apreciar en la Figura 5.9.

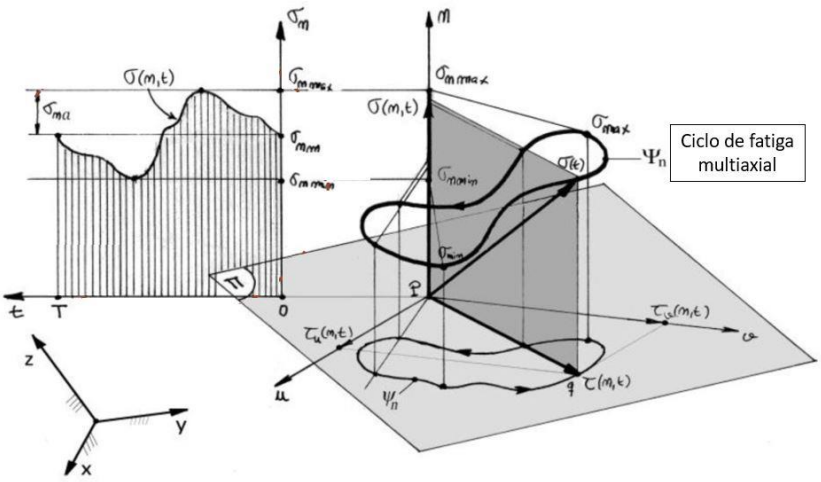

Figura 5.9. Ciclo de fatiga multiaxial [1].

A la vista de la Figura 5.9, el ciclo de fatiga multiaxial queda definido por el extremo del vector de tensión; de manera que se genera una curva cerrada tridimensional de periodo T. La obtención de la componente media y alterna de la tensión es directa tras proyectar la curva sobre el eje normal "n" ya que solamente hay que identificar los valores máximo y mínimo ($\sigma_{n_{max}}$ y $\sigma_{n_{min}}$) y operar como sigue.

$$\sigma_{n_m} = \frac{\sigma_{n_{max}} + \sigma_{n_{min}}}{2} \tag{5.39}$$

$$\sigma_{n_a} = \frac{\sigma_{n_{max}} - \sigma_{n_{min}}}{2} \tag{5.40}$$

Cabe decir que los términos σ_{n_m} y σ_{n_a} se pueden calcular como se ha expuesto en las ecuaciones (5.39) y (5.40) siempre y cuando se cumpla la definición de ciclo de fatiga multiaxial; es decir, trayectoria cerrada con un solo máximo y mínimo. De lo contrario, en lugar de utilizar este procedimiento se deben emplear métodos de daño acumulativo como Weber o Socie; lo cual queda fuera del alcance de este trabajo.

Tras identificar la componente media y alterna de la tensión normal, se hace lo propio con la tensión tangencial. Esto no es tan directo como en el caso anterior ya que la tensión cortante en el plano *u-v* describe una trayectoria (ver Figura 5.10 a)) y, por tanto, se requiere de un criterio para separar la componente media de la alterna.

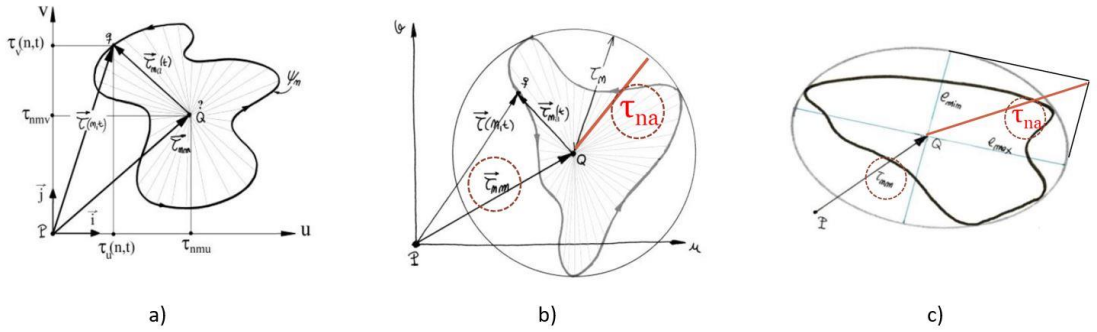

Figura 5.10. Obtención de la componente media y alterna de la tensión cortante: a) Caso general, b) Método de la circunferencia, c) Método de la elipse.

En el caso general que se muestra en la Figura 5.10 a), la componente media corresponde a la magnitud de $\overline{PQ}$ mientras que, la alterna varía instante a instante siendo su valor la magnitud que une el punto Q con el extremo de la curva. Sin embargo, para poder aplicar más adelante los métodos de plano crítico, se requiere de un valor constante por lo que, entre los métodos más empleados para deducirlo se tienen el método de la circunferencia y el de la elipse (casos a) y b) de la Figura 5.10 respectivamente).

El método de la circunferencia se basa en trazar la circunferencia de radio mínimo que circunscriba la trayectoria, definiendo τ_{n_m} como la magnitud de $\overline{PQ}$ y τ_{n_a} como el radio de la circunferencia. Por su parte, el método de la elipse es semejante al de la circunferencia pero empleando la elipse mínima circunscrita; de forma que la τ_{n_m} sigue siendo $\overline{PQ}$ pero la τ_{n_a} se obtiene con la raíz cuadrada de la suma de los semiejes.

Utilizando los conceptos que se acaban de explicar, se calcula la componente media y alterna tanto de la tensión normal como de la tangencial en tantos planos como se consideren candidatos a ser el denominado plano crítico. Una vez hecho esto y, antes de calcular el daño, se selecciona cuál de todos esos planos corresponde al plano crítico. Para ello, cada método emplea su propia formulación y, al igual que la función de daño, suele corresponder a una combinación lineal de las componentes σ_m, σ_a, τ_m y τ_a.

De forma general, se pueden distinguir dos grandes grupos de métodos de plano crítico: Maximum Shear Stress Range (MSSR) y Maximum Damage (MD). En cuanto a los primeros, MSSR, se caracterizan por emplear diferentes funciones para hallar el plano crítico y calcular el daño. Comúnmente, en HCF este conjunto de métodos suele emplear la máxima tensión cortante alterna para definir el plano crítico; lo cual se basa en que la iniciación de la grieta se produce por tensiones cortantes, como se ha comentado en el subapartado 1.3.2. Fases del fallo por fatiga. Por su parte, los métodos del tipo MD se caracterizan por tener la misma función tanto para encontrar el plano crítico como para calcular la función de daño.

Independientemente del grupo de métodos al que pertenezca el método de plano crítico con el que se desee trabajar, la sistemática que se debe seguir es común por lo que, antes de comenzar a explicar en detalle cada uno de los métodos particulares, se van a definir de forma cronológica los sucesivos pasos a realizar.

- Obtener el tensor de tensiones en los puntos críticos de la pieza. Para determinar los puntos de interés, se toma como apoyo los resultados obtenidos del MEF.
- Elegir un número "p" de planos para cada punto. Dicha elección tiene una notable transcendencia en el procedimiento ya que se debe elegir un número tal que sea suficiente como para identificar el plano crítico, pero no excesivo para no disponer de un coste desmedido.
- Calcular la componente media y alterna tanto de la tensión normal como de la tangencial para cada uno de los planos que se han definido.
- Identificar el plano crítico atendiendo a la función que define el método particular que se va a emplear.
- Calcular el daño mediante la función de daño que indica el método particular que desea utilizar.
- Determinar el posible fallo de la pieza. Como se puede intuir, si la función de daño augura un valor del daño superior al permitido se producirá el fallo mientras que, si es inferior, la pieza tendrá duración infinita.

Para finalizar, se van a destacar dos aspectos que se encuentran estrechamente ligados al procedimiento de cálculo mediante los métodos de plano crítico. Por un lado, remarcar la importancia de los ordenadores ya que, sin éstos no solo resultaría más compleja la puesta en práctica de los métodos, sino que la identificación de los puntos más solicitados

de la pieza sería dificultosa y en muchos casos, inviable. Por otro lado, como se ha mencionado al describir los pasos a realizar, hay que seleccionar de forma equilibrada el número de planos "p" que se van a analizar para alcanzar un diseño óptimo en cuanto a resultados y a coste de cálculo adjunto.

Con esto, se concluye con los fundamentos básicos y las características que describen a los métodos de plano crítico y, en adelante se va a tratar de forma particular cada uno de los métodos de plano crítico basados en tensiones que se recogen en el alcance del trabajo. A continuación, se comienza desarrollando el método de Matake.

5.2.3.1 Método de Matake

El método de Matake se desarrolló en 1977 y pertenece a los métodos del tipo MSSR ya que no coinciden la función que define el plano crítico con la función de daño. En este sentido, identifica el plano crítico como aquel en el que la tensión tangencial es máxima; lo cual matemáticamente se expresa como sigue.

$$max_{n=1}^{p} = (\tau_a) \;\rightarrow\; (Plano\ crítico) \tag{5.41}$$

Por su parte, la función de daño o lo que es equivalente, la comprobación a fatiga en el plano crítico, se define como se muestra en (5.42). En ella se puede apreciar cómo no se considera de ninguna forma los posibles desfases entre la tensión normal y tangencial.

$$\tau_{a_{eq}} = \tau_a + \alpha_M \cdot (\sigma_m + \sigma_a) \geq \beta_M \;\rightarrow\; (Fallo) \tag{5.42}$$

A la vista de la ecuación (5.42), en la función de daño intervienen dos parámetros (α y β) inherentes al propio método por lo que, al igual que con los métodos de enfoque global, se deben emplear dos ensayos de probeta con límite de fatiga conocidos. Así pues, a continuación se va a seguir un procedimiento similar al utilizado en el subapartado 5.2.2.1. Método de Sines para obtener las constantes del método.

Ensayo de flexión con tensión alterna

Antes de particularizar la ecuación (5.42), se debe localizar el plano crítico. Para ello, se representa el estado tensional sobre el círculo de Mohr y a continuación, se identifica aquella posición en la que se cumple la expresión (5.41) (ver Figura 5.11).

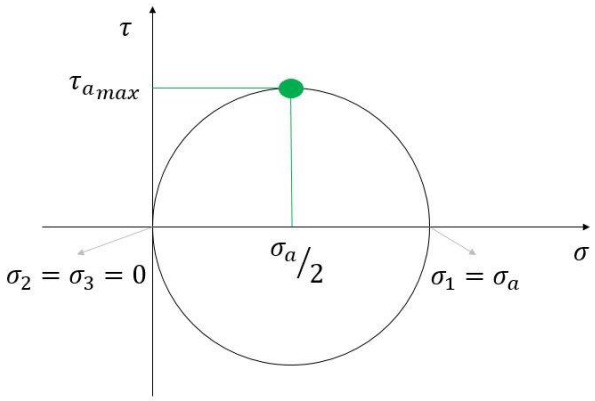

Figura 5.11. Identificación del plano crítico según el método de Matake en el ensayo de flexión alterna.

Sustituyendo sobre la ecuación (5.42):

$$\frac{\sigma_{-1}}{2} + \alpha_M \cdot (0 + \sigma_{-1}) = \beta_M \rightarrow \alpha_M = \frac{2\beta_M - \sigma_{-1}}{\sigma_{-1}} \tag{5.43}$$

Ensayo de torsión alterna

Como se ha visto, únicamente con la ecuación (5.43) no se puede obtener ninguno de los dos parámetros que intervienen en la función de daño por lo que, se necesita otra ecuación que posibilite la resolución mediante un sistema de dos ecuaciones con dos incógnitas. Así pues, en este caso se emplea el ensayo de torsión alterna donde el plano crítico se encuentra en el lugar remarcado sobre la Figura 5.12.

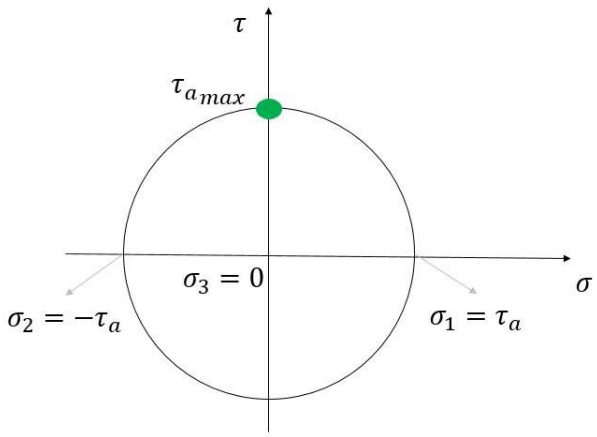

Figura 5.12. Identificación del plano crítico según el método de Matake en el ensayo de torsión alterna.

Sustituyendo en la ecuación (5.42):

$$\tau_{-1} + \alpha_M \cdot (0 + 0) = \beta_M \;\; \rightarrow \;\; \beta_M = \tau_{-1} \tag{5.44}$$

Introduciendo el resultado de (5.44) sobre la relación (5.43) se terminan de definir las dos constantes del método.

$$\alpha_M = \frac{2\tau_{-1} - \sigma_{-1}}{\sigma_{-1}} = 2\frac{\tau_{-1}}{\sigma_{-1}} - 1 \tag{5.45}$$

Cabe remarcar que los parámetros que se han obtenido (expresiones (5.44) y (5.45)) corresponden a la pareja de ensayos de $\sigma_{-1} - \tau_{-1}$ pero, tal y como se va a desarrollar en el apartado 9. Cálculos, también se pueden emplear otras parejas de ensayos.

5.2.3.2 Método de Findley

El método de Findley se desarrolló en 1957 y, a diferencia del método de Matake, se trata de un método del tipo MD ya que se emplea la misma función para definir el plano crítico y calcular el daño. Así pues, para identificar el plano crítico emplea la siguiente expresión.

$$max_{n=1}^{p} = \left(\tau_a + \alpha_F \cdot (\sigma_m + \sigma_a)\right) \;\; \rightarrow \;\; (Plano\ crítico) \tag{5.46}$$

Asimismo, la función de daño es la que se muestra en la expresión (5.47) y, al igual que ocurre con el método de Matake, no se contempla la posibilidad de que la tensión normal y la tangencial estén desfasadas. Además, cabe decir que este método solamente es válido cuando se cumple que la fracción entre τ_{-1} y σ_{-1} sea mayor que 0.5.

$$\tau_{a_{eq}} = \tau_a + \alpha_F \cdot (\sigma_m + \sigma_a) \geq \beta_F \;\; \rightarrow \;\; (Fallo) \tag{5.47}$$

Para la definición de los parámetros de Findley, en ocasiones se suelen emplear los mismos valores del método de Matake, pero en realidad, los resultados entre ambos métodos difieren. Es por ello que, a continuación, se va a proceder a su demostración mediante la pareja de ensayos de $\sigma_{-1} - \tau_{-1}$.

Ensayo de flexión con tensión alterna

Al igual que en el método de Matake, lo primero que se debe hacer es identificar el plano crítico, pero, sin embargo, en este caso no es tan directo ya que se trabaja con una función en la que intervienen más parámetros. Por lo tanto, aunque se podría pensar en hacerlo iterativamente, resulta menos costoso computacionalmente si se opera previamente de forma analítica y se obtiene la posición.

Así pues, para obtener el plano crítico de forma analítica, se realiza la derivada de la función que lo define respecto de θ y se iguala a cero. Esto se puede hacer así porque según la definición de fatiga, en cada ciclo se pasa del valor máximo al mínimo y viceversa (de hecho, en el círculo de Mohr esto se vería al ser simétrico respecto a la horizontal) por lo que, partiendo del caso general que se representa sobre la Figura 5.13 se va a deducir dónde se encuentra el plano crítico.

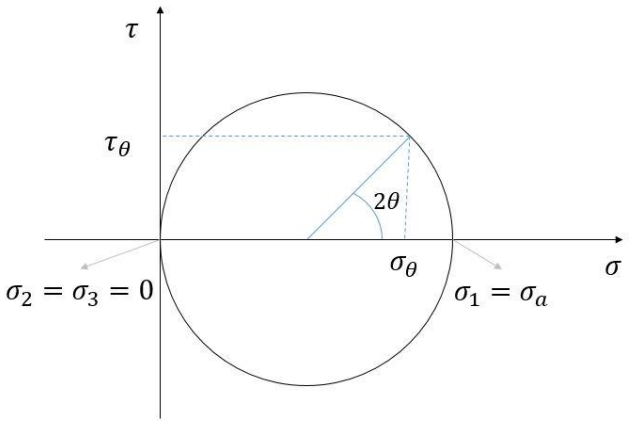

Figura 5.13. Identificación del plano crítico según el método de Findley en el ensayo de flexión alterna.

Considerando σ_{xx} a la tensión genérica que se aplica sobre la probeta, se definen la tensión tangencial y la tensión normal como sigue.

$$\tau_\theta = \frac{\sigma_{xx}}{2} \cdot \sin(2\theta) \tag{5.48}$$

$$\sigma_\theta = \frac{\sigma_{xx}}{2} \cdot (1 + \cos(2\theta)) \tag{5.49}$$

Introduciendo estas expresiones sobre la ecuación (5.46) y desarrollando la expresión que se plantea a continuación, se obtiene la posición del plano crítico.

$$\frac{d\left(\frac{\sigma_a}{2} \cdot \sin(2\theta) + \alpha_F \cdot \left(\frac{(\sigma_m + \sigma_a)}{2} \cdot (1 + \cos(2\theta))\right)\right)}{d\theta} = 0 \tag{5.50}$$

Se deriva según la regla de la cadena y se despeja el valor de $2\theta^*$ (plano crítico).

$$2\theta^* = arctg\left(\frac{1}{\alpha_F}\cdot\frac{\sigma_a}{(\sigma_m+\sigma_a)}\right) \rightarrow (Plano\ crítico) \tag{5.51}$$

Atendiendo a la representación de la Figura 5.13 y, dado que se está ante un caso de flexión alterna, se particularizan los términos que se deben emplear en las funciones del método de Findley de la siguiente forma.

$$\tau_{\theta\,a} = \frac{\sigma_{-1}}{2}\cdot\sin(2\theta) \tag{5.52}$$

$$\sigma_{\theta\,m} = 0 \tag{5.53}$$

$$\sigma_{\theta\,a} = \frac{\sigma_{-1}}{2}\cdot(1+\cos(2\theta)) \tag{5.54}$$

Una vez conocido tanto el plano crítico como las expresiones que se deben introducir sobre la función de daño((5.52), (5.53) y (5.54)) el siguiente paso es sustituir para obtener una ecuación que relacione los parámetros del método. Antes de ello, se muestran las relaciones trigonométricas que se necesita emplear para introducir la posición del plano crítico en las ecuaciones (5.52), (5.53) y (5.54).

$$\sin\big(arctg(x)\big) = \frac{x}{\sqrt{1+x^2}} \tag{5.55}$$

$$\cos\big(arctg(x)\big) = \frac{1}{\sqrt{1+x^2}} \tag{5.56}$$

Sustituyendo en (5.47).

$$\frac{\sigma_{-1}}{2}\cdot\left(\left(\frac{\frac{1}{\alpha_F}}{\sqrt{1+\left(\frac{1}{\alpha_F}\right)^2}}\right)+\alpha_F\cdot\left(1+\frac{1}{\sqrt{1+\left(\frac{1}{\alpha_F}\right)^2}}\right)\right) = \beta_F \tag{5.57}$$

Desarrollando y simplificando la expresión:

$$\frac{\sigma_{-1}}{2} \cdot \left(\sqrt{{\alpha_F}^2 + 1} + \alpha_F\right) = \beta_F \tag{5.58}$$

Hasta ahora, solamente se tiene una relación entre las dos constantes que intervienen en el método de Findley por lo que, para poder despejar sus valores, se requiere de un segundo ensayo; tomando en este caso, el ensayo de torsión.

Ensayo de torsión alterna

Del mismo modo que se ha realizado con el ensayo de flexión alterna, sobre la Figura 5.14 se muestra de forma genérica la posición donde se encuentra el plano crítico en el ensayo de torsión.

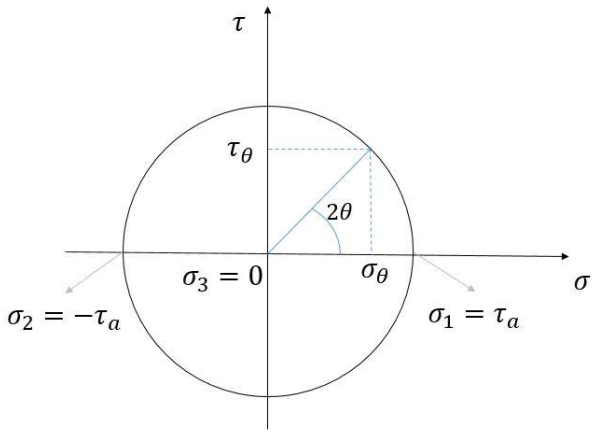

Figura 5.14. Identificación del plano crítico según el método de Findley en el ensayo de flexión alterna.

En este caso, considerando τ_{xy} como la tensión cortante genérica que debe soportar la probeta, las componentes que se deducen según la Figura 5.14 son los siguientes.

$$\tau_\theta = \tau_{xy} \cdot \sin(2\theta) \tag{5.59}$$

$$\sigma_\theta = \tau_{xy} \cdot \cos(2\theta) \tag{5.60}$$

Partiendo de las ecuaciones genéricas (5.59) y (5.60) se obtiene la expresión dónde se encuentra el plano crítico derivando de forma semejante al ensayo de flexión alterna.

$$\frac{d\left(\tau_a \cdot \sin(2\theta) + \alpha_F \cdot \left((\tau_m + \tau_a) \cdot \cos(2\theta)\right)\right)}{d\theta} = 0 \tag{5.61}$$

Derivando de acuerdo a la regla de la cadena y desarrollando, se obtiene la posición del plano crítico ($2\theta^*$).

$$2\theta^* = arctg\left(\frac{\tau_a}{\alpha_F \cdot (\tau_m + \tau_a)}\right) \rightarrow (Plano\ crítico) \tag{5.62}$$

Una vez conocida la expresión que define la posición del plano crítico (ec. *(5.62)*), se pueden particularizar los términos (5.59) y (5.60) para finalmente, introducirlos en la función de daño (ec. (5.47)) como se muestra a continuación.

$$\tau_{-1} \cdot \left(\left(\frac{\frac{1}{\alpha_F}}{\sqrt{1 + \left(\frac{1}{\alpha_F}\right)^2}} \right) + \alpha_F \cdot \left(\frac{1}{\sqrt{1 + \left(\frac{1}{\alpha_F}\right)^2}} \right) \right) = \beta_F \tag{5.63}$$

Desarrollando y simplificando la ecuación:

$$\tau_{-1} \cdot \sqrt{1 + {\alpha_F}^2} = \beta_F \tag{5.64}$$

Llegados a este punto, se han obtenido las dos ecuaciones que relacionan los parámetros del método (expresiones (5.58) y (5.64)) y por tanto, se está en disposición de deducirlas mediante la resolución de dos ecuaciones con dos incógnitas. Así pues, en uno de los múltiples procedimientos que se tienen para obtenerlas, se comienza dividendo la expresión (5.58) entre la (5.64).

$$\frac{\frac{\sigma_{-1}}{2} \cdot \left(\sqrt{{\alpha_F}^2 + 1} + \alpha_F\right)}{\tau_{-1} \cdot \sqrt{1 + {\alpha_F}^2}} = 1 \tag{5.65}$$

Operando y despejando la constante α_F se llega a la siguiente expresión.

$$\alpha_F = \frac{2 - \frac{\sigma_{-1}}{\tau_{-1}}}{2 \cdot \sqrt{\frac{\sigma_{-1}}{\tau_{-1}} - 1}} \tag{5.66}$$

Tras esto, se introduce el valor de α_F (ecuación(5.66)) sobre cualquiera de las dos expresiones de las que se ha partido ((5.58) y(5.64)) y se despeja el valor de β_F.

$$\beta_F = \frac{\sigma_{-1}}{2 \cdot \sqrt{\frac{\sigma_{-1}}{\tau_{-1}} - 1}} \tag{5.67}$$

Con esto se demuestra como los resultados de las constantes en el método de Findley difieren de los obtenidos en el método de Matake. Por otra parte, remarcar como a pesar de que aquí se haya empleado la pareja de ensayos $\sigma_{-1} - \tau_{-1}$, al igual que se ha mencionado en el resto de métodos, se puede utilizar otra pareja de ensayos diferente.

5.2.3.3 Método de Papuga (PCr)

De los tres métodos desarrollados por Papuga para el análisis de fatiga con estado tensional muliaxial, el único que pertenece al grupo de plano crítico es el método PCr (Papuga Critical Plane). Los otros dos, son métodos integrales; los cuales se caracterizan por evaluar la función de daño integrándola en todos los planos, en lugar de utilizar un único plano. Tal y como se puede intuir, esto hace que requieran un coste computacional muy superior al de los métodos de plano crítico y, además, sus resultados son comparables a los anteriores [14] por lo que, en adelante cuando se mencione el método de Papuga se está haciendo referencia al método PCr que se explica a continuación.

El método PCr se desarrolló en 2008 y pertenece al grupo de métodos MD ya que utiliza la misma función para identificar el plano crítico y para calcular el daño. A diferencia otros métodos de plano crítico como Matake o Findley, éste método requiere conocer tres propiedades del material como son los límites de fatiga σ_0, σ_{-1} y τ_{-1}.

$$max_{n=1}^{p} = \left(\sqrt{a_P \cdot {\tau_a}^2 + b_P \cdot \left(\sigma_a + \frac{\tau_{-1}}{2 \cdot \sigma_0} \sigma_m \right)} \right) \rightarrow (Plano\ crítico) \tag{5.68}$$

Por su parte, la función de daño se formula como sigue.

$$\sigma_{eq_P} = \sqrt{a_P \cdot {\tau_a}^2 + b_P \cdot \left(\sigma_a + \frac{\tau_{-1}}{2 \cdot \sigma_0} \sigma_m \right)} \geq \sigma_{-1} \rightarrow (Fallo) \tag{5.69}$$

Para la definición de las constantes del método (a_P y b_P), Papuga distingue entre materiales frágiles $\left({}^{\sigma_{-1}}/_{\tau_{-1}} < {}^{4}/_{3}\right)$ y dúctiles $\left({}^{\sigma_{-1}}/_{\tau_{-1}} > {}^{4}/_{3}\right)$. A pesar de que lo referente a los materiales frágiles quede fuera del alcance del trabajo, a continuación, se van a presentar las fórmulas que emplea en cada caso para demostrar la diferencia que existe entre ambos.

Material frágil:

$$a_P = \frac{1}{2} \cdot \left(\left(\frac{\sigma_{-1}}{\tau_{-1}} \right)^2 + \sqrt{\left(\frac{\sigma_{-1}}{\tau_{-1}} \right)^4 - \left(\frac{\sigma_{-1}}{\tau_{-1}} \right)^2} \right) \tag{5.70}$$

$$b_P = \sigma_{-1} \tag{5.71}$$

Material dúctil

$$a_P = \left(\frac{4 \cdot \left(\frac{\sigma_{-1}}{\tau_{-1}} \right)^2}{4 + \left(\frac{\sigma_{-1}}{\tau_{-1}} \right)^2} \right)^2 \tag{5.72}$$

$$b_P = \frac{8 \cdot \sigma_{-1} \cdot \left(\frac{\sigma_{-1}}{\tau_{-1}} \right)^2 \cdot \left(4 - \left(\frac{\sigma_{-1}}{\tau_{-1}} \right)^2 \right)}{\left(4 + \left(\frac{\sigma_{-1}}{\tau_{-1}} \right)^2 \right)^2} \tag{5.73}$$

5.2.3.4 Método de Dang Van

El método de Dang Van, aunque pertenece a los métodos del tipo MD por emplear la misma función tanto para identificar el plano crítico como para calcular el daño, realiza el análisis a fatiga bajo un concepto diferente a los anteriores. Esto se debe a que estudia las tensiones a nivel microscópico y después, a partir de las hipótesis de Lin-Taylor [15], [16], se relacionan con las tensiones macroscópicas. Cabe decir que, pese a que conceptualmente lo correcto sería utilizar las tensiones microscópicas, suele ser habitual trabajar directamente con las tensiones macroscópicas.

Otra diferencia que mantiene el método de Dang Van respecto a los anteriores se encuentra en que la evaluación de las tensiones se realiza instante a instante de forma que la definición del plano crítico y el correspondiente cálculo de daño varían continuamente. Esto conlleva un incremento del coste computacional en torno a un orden de magnitud ya que, además de estudiar varios planos en varios puntos de la pieza, hay que hacerlo instante a instante. Sin embargo, con ello se consideran posibles desfases entre la tensión normal y la cortante; lo cual no es posible mediante métodos los métodos anteriores.

La identificación del plano crítico se realiza mediante la siguiente expresión.

$$max_{n=1}^{p} = (\tau_a(t) + \alpha_{DV} \cdot \sigma_h(t)) \;\; \rightarrow \;\; (Plano\ crítico) \tag{5.74}$$

Donde la tensión hidrostática en cada instante de tiempo ($\sigma_h(t)$) viene dada como sigue.

$$\sigma_h(t) = \frac{\sigma_1(t) + \sigma_2(t) + \sigma_3(t)}{3} \tag{5.75}$$

Por su parte, la función de daño corresponde a la ecuación (5.76) y, al igual que ocurre con otros métodos como, por ejemplo, Findley, únicamente se puede emplear este método si se cumple que ${\sigma_{-1}}/{\tau_{-1}} > 0{,}5$.

$$\tau_{a_{eq}} = max(\tau_a(t) + \alpha_{DV} \cdot \sigma_h(t)) \geq \beta_{DV} \;\; \rightarrow \;\; (Fallo) \tag{5.76}$$

Al igual que el resto de métodos descritos previamente, el método de Dang Van dispone de dos parámetros particulares (α_{DV} y β_{DV}) que, se obtienen a partir de dos ensayos con límite de fatiga conocido. Así pues, en este caso se va a realizar empleando la pareja de ensayos $\sigma_{-1} - \tau_{-1}$.

Ensayo de flexión con tensión alterna

De la misma manera que se ha realizado con otros métodos de plano crítico, se comienza localizando el plano crítico. Para ello, se parte de una situación análoga a la representada sobre la Figura 5.13 con el método de Findley y a partir de la expresión (5.74) se plantea lo siguiente.

$$\frac{d\left(\frac{\sigma_a}{2} \cdot \sin(2\theta) + \alpha_{DV} \cdot \left(\frac{\sigma_1(t) + \sigma_2(t) + \sigma_3(t)}{3}\right)\right)}{d\theta} = 0 \tag{5.77}$$

Se deriva según la regla de la cadena y se despeja el valor de $2\theta^*$ (plano crítico).

$$2\frac{\sigma_a}{2}\cos(2\theta) = 0 \;\rightarrow\; \cos(2\theta) = 0 \;\rightarrow\; 2\theta^* = 90° \rightarrow\; (Plano\ crítico) \tag{5.78}$$

Por lo tanto, el plano crítico se encuentra en la perpendicular desde el centro del círculo de Mohr; consiguiendo un caso semejante al representado sobre la Figura 5.11 para el método de Matake. Pese a que dicha representación sea significativa para ambos métodos, hay que tener presente que los criterios que se emplean en cada caso son diferentes.

Sustituyendo en (5.76).

$$\frac{\sigma_{-1}}{2} + \alpha_{DV} \cdot \frac{\sigma_{-1}}{3} = \beta_{DV} \tag{5.79}$$

Con esto se tiene una primera relación entre las dos constantes del método por lo que, mediante un segundo ensayo, en este caso el de torsión alterna, se obtiene la segunda ecuación que permite despejar sus valores resolviendo el sistema.

Ensayo de torsión alterna

Al igual que ha ocurrido en el ensayo de flexión con tensión alterna, la posición del plano crítico para el ensayo de torsión alterna coincide en el método de Matake y en el de Dang Van (ver Figura 5.12); tal y como se demuestra a continuación.

$$\frac{d\left(\tau_{-1} \cdot \sin(2\theta) + \alpha_{DV} \cdot \left(\frac{\tau_{-1} - \tau_{-1} + 0}{3}\right)\right)}{d\theta} = 0 \tag{5.80}$$

$$2\tau_{-1}\cos(2\theta) = 0 \;\rightarrow\; \cos(2\theta) = 0 \;\rightarrow\; 2\theta^{*} = 90° \rightarrow\; (Plano\ crítico) \tag{5.81}$$

Por lo tanto, sustituyendo en (5.76).

$$\tau_{-1} + \alpha_{DV} \cdot \left(\frac{\tau_{-1} - \tau_{-1} + 0}{3}\right) = \beta_{DV} \;\rightarrow\; \beta_{DV} = \tau_{-1} \tag{5.82}$$

Introduciendo el valor de β_{DV} en la ecuación (5.79), se despeja α_{DV}.

$$\frac{\sigma_{-1}}{2} + \alpha_{DV} \cdot \frac{\sigma_{-1}}{3} = \tau_{-1} \;\rightarrow\; \alpha_{DV} = 3 \cdot \left(\frac{\tau_{-1}}{\sigma_{-1}} - \frac{1}{2}\right) \tag{5.83}$$

Tras deducir los parámetros de Dang Van, se concluye la explicación del método delimitando la zona donde se produce el fallo con la zona de vida infinita. Para que el material falle a fatiga, se debe cumplir la expresión (5.76), donde el término $\tau_a(t)$ se obtiene instante a instante desde el centro de la circunferencia hasta el punto de la trayectoria (ver 5.2.3. Métodos de plano crítico). Así pues, considerando que dicho término siempre tiene un valor positivo (únicamente varía su magnitud) y que la $\sigma_h(t)$ adquiere su signo correspondiente, la frontera entre la vida finita e infinita se define como una recta (ver Figura 5.15).

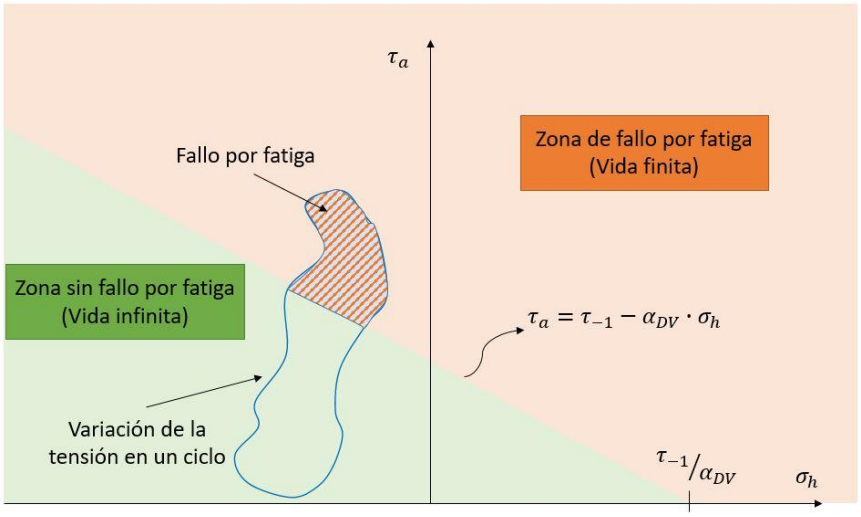

Figura 5.15. Delimitación de las zonas de fallo según el método de Dang Van.

Tal y como se puede apreciar, en la zona no sombreada de la Figura 5.15 se prevé que el componente tenga vida infinita mientras que, en la zona sombreada se producirá el fallo. Por último, se observa también como la recta que delimita ambas zonas tiene una pendiente negativa; lo cual concuerda con lo comentado previamente respecto a la propagación de la grieta (las grietas sometidas a tensiones de compresión tienen mejor comportamiento a fatiga que las que se encuentran sometidas a tracción ya que la compresión tiende a "cerrar" la grieta).

5.2.4 Métodos de resolución de ecuaciones

Tras describir los diferentes métodos de fatiga multiaxial que se van a recoger en la aplicación y, aunque todavía no se hayan obtenido las ecuaciones correspondientes a la particularización de éstas a los casos uniaxial y de torsión pura, se van a presentar los métodos que se van a requerir para su resolución. En este sentido, se adelanta que se va a trabajar con dos técnicas de resolución conjuntamente: por un lado, el método de bisección y, por otro lado, el método de Newton-Raphson.

Las ecuaciones que se van a tener que resolver pueden ser tanto lineales como no-lineales por lo que, empleando los métodos que se han citado, se constata su capacidad de resolución; independientemente del tipo de ecuación que se tenga. Así pues, se va a comenzar exponiendo lo referente al método de bisección y tras esto, se hará lo propio con el método de Newton-Raphson.

5.2.4.1 Método de bisección

El método de bisección; también denominado método de Bolzano, es un método numérico iterativo del tipo cerrado que se emplea para la resolución de raíces de

ecuaciones no lineales. Esto es, se trata de determinar el valor de la variable de la función "x" tal que se cumpla la siguiente expresión.

$$f(x) = 0 \tag{5.84}$$

Al tratarse de un método cerrado, el punto de partida es un intervalo que contiene a la raíz "x" definido por el límite inferior "a" y el límite superior "b" como sigue.

$$x \in [a, b] \;\; ; \;\; \begin{cases} f(a) < 0 \\ f(b) > 0 \end{cases} \tag{5.85}$$

Si se cumple la expresión *(5.85)*, según el Teorema del Valor Intermedio para funciones continuas, va a existir al menos una raíz "x" que verifique la igualdad (5.84). Una vez definido el intervalo donde se encuentra la raíz que se desea obtener, el método de Bolzano establece que lo siguiente es dividir dicho intervalo en dos fracciones iguales quedándose con la fracción en la que se tiene la raíz; es decir, donde f cambia de signo. Para poder visualizar este procedimiento, sobre la Figura 5.16 se muestra un ejemplo sobre el que cabe decir que, a pesar de representar tres iteraciones, el proceso continua hasta cumplir con el criterio de parada.

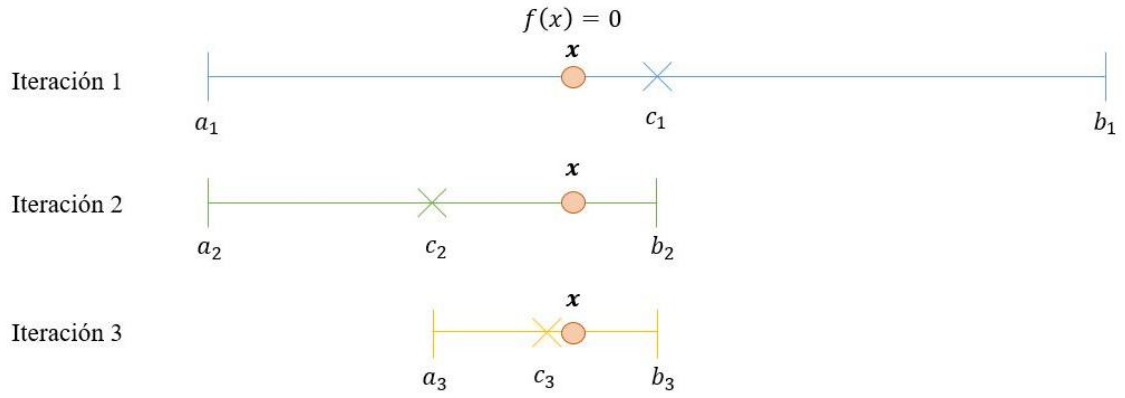

Figura 5.16. Identificación del intervalo que contiene a la raíz en el método de bisección.

Donde c_j es el valor intermedio de la iteración j.

$$c_j = \frac{a_j + b_j}{2} \tag{5.86}$$

Computacionalmente, para identificar el intervalo que se debe retener se procede según lo mostrado en la siguiente operación.

$$\begin{cases} Si\ f(c_j)\cdot f(a_j) < 0 \begin{cases} a_{j+1} = a_j \\ b_{j+1} = c_j \end{cases} \\ Si\ f(c_j)\cdot f(b_j) < 0 \begin{cases} a_{j+1} = c_j \\ b_{j+1} = b_j \end{cases} \end{cases} \tag{5.87}$$

Tal y como se ha mencionado previamente, se trata de un proceso iterativo que finaliza cuando se cumple alguno de los criterios de parada establecidos en su programación. En este sentido, es habitual emplear dos criterios simultáneamente; uno de tolerancia y otro restringiendo el número máximo de iteraciones.

Mediante el criterio de tolerancia, se va a comparar la solución obtenida del ciclo iterativo que se acaba de realizar con una tolerancia "ε_{tol}" para decidir si es lo suficientemente buena como para aceptarla. Existen varios criterios para hacerlo (error relativo, error relativo porcentual, error absoluto…) siendo el criterio de la longitud el que se va a presentar a continuación, por ser el que se va a emplear en este trabajo.

$$lon|b_j - a_j| \leq \varepsilon_{tol} \tag{5.88}$$

Por su parte, el segundo criterio que se suele incluir, se puede considerar como complemento del anterior y su función es restringir el número de iteraciones a realizar. De esta forma, en los casos en los que se tenga un ciclo iterativo que tienda a infinito, se detiene el proceso al llegar al número máximo de iteraciones que se ha predefinido.

Para finalizar, cabe decir que pese a la ventaja que ofrece el método de bisección de converger siempre a una solución, también dispone de desventajas; siendo la principal, la lentitud de convergencia. Esto se debe a que tiene convergencia lineal y, por tanto, es habitual emplearlo como paso previo de otros métodos con convergencia de mayor orden como, por ejemplo, el método de Newton-Raphson (convergencia cuadrática) que se va a explicar a continuación.

5.2.4.2 Método de Newton-Raphson

Dentro de los diferentes métodos numéricos que existen, el método de Newton-Raphson se puede considerar como uno de los más conocidos y empleados para la resolución de ecuaciones no lineales. Al igual que el método de bisección, se trata de un método iterativo, pero en este caso, pertenece al tipo de métodos abiertos ya que no se requiere tener un intervalo que contenga a la raíz; sino una aproximación inicial.

Respecto a su procedimiento, se comienza asignando una solución aproximada (x_p) a la ecuación (5.84) cuya solución real es x_s. Así pues, a través de la recta tangente a la ecuación en x_p se obtiene una nueva solución aproximada x_{p+1} cuyo resultado se aproxima más a la solución real; es decir, se cumple que $|f(x_{p+1})| < |f(x_p)|$. Esto se puede apreciar mejor a partir del gráfico que se muestra en la Figura 5.17.

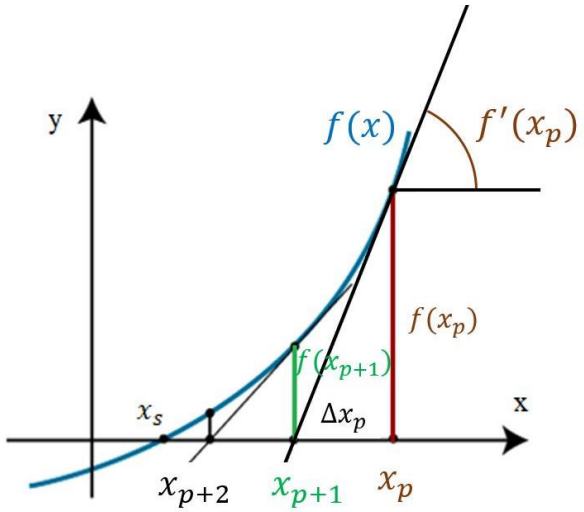

Figura 5.17. Método de Newton-Raphson.

$$f'(x_p) = \tan\alpha = \frac{f(x_p)}{\Delta x_p} \tag{5.89}$$

$$\Delta x_p = x_p - x_{p+1} \;\rightarrow\; x_{p+1} = x_p - \Delta x_p \tag{5.90}$$

Como se puede apreciar, se trata de un proceso iterativo donde x_p cada vez adquiere valores más próximos a la solución real (x_s). Por lo tanto, dicho proceso se considerará finalizado cuando se cumpla alguno de los dos criterios de parada que se establezcan: tolerancia o número máximo de iteraciones (al igual que ocurría en el método de bisección). En este caso, de las múltiples técnicas que existen para definir el criterio de parada de tolerancia se va a emplear el del error relativo porcentual que se muestra en la ecuación *(5.91)*.

$$e_{rel} = \frac{|x_{p+1} - x_p|}{|x_{p+1}|} \leq \varepsilon_{rel} \tag{5.91}$$

Por último, cabe remarcar como la principal ventaja del método de Newton-Raphson es su convergencia cuadrática; lo cual hace que el método sea notablemente más rápido que el método de bisección. Sin embargo, tiene la desventaja de que en algunas situaciones particulares el método puede no converger; por ejemplo, si se parte de una solución aproximada que dista mucho de la solución real.

Para evitar la citada desventaja del método de Newton-Raphson, en este trabajo se va a tomar como solución aproximada el valor intermedio del último intervalo que contiene a la raíz que se ha calculado en el método de bisección. De esta forma, se comienza la resolución mediante el método de bisección, que asegura la convergencia, y después, se aproxima la solución aplicando un método de mayor rapidez y eficiencia computacional como es el método de Newton-Raphson.

6 Análisis de alternativas y solución adoptada

En este apartado se van a analizar las diferentes alternativas que se podrían haber utilizado para realizar el proyecto y, a continuación, se va a argumentar la solución adoptada [17]. Para ello, esta sección se va a dividir en dos partes; en la primera, se estudian algunos de los métodos de fatiga multiaxial que podrían haber sido objeto del proyecto y, en la segunda, se van a analizar las posibles herramientas con las que se podría haber trabajado.

Con el objetivo de seleccionar las mejores alternativas, tanto de los métodos de fatiga multiaxial como de la herramienta que se va a utilizar para programarlos, se utilizará el método del valor técnico ponderado. Este método se basa en los siguientes seis pasos:

1) Se definen los *n* criterios que se van a tener en cuenta.
2) Se dota a cada criterio un peso ponderado, p_j.
3) Se analiza cada alternativa, asignándole (para cada criterio) una calificación, x_{ij}.
4) Para cada alternativa se hace el sumatorio de los productos de sus calificaciones por las ponderaciones de sus criterios correspondientes.

$$Alternativa\ (i) = \sum_{j=1}^{n} p_j \cdot x_{ij} \tag{6.1}$$

5) Se divide la suma anterior entre el producto de la puntuación máxima por la suma de todos los pesos.
6) Se escoge la alternativa con mayor puntuación.

6.1 Elección de los métodos de fatiga multiaxial

Antes de comenzar a estudiar los posibles métodos de fatiga multiaxial que podrían constituir el objeto del trabajo, se van a enumerar los diferentes criterios y pesos ponderados que se han utilizado para su posterior selección.

- **Utilidad actual $(\boldsymbol{p_1 = 3})$:** Se va a tener en cuenta si los métodos que se desea implementar, hoy en día se suelen emplear o si por el contrario, se encuentran en desuso.

- **Rango de aplicación ($p_2 = 3$):** Los métodos que abarquen mayor campo de aplicación se calificarán con mejores puntuaciones. Dentro de este punto se van a tener en cuenta aspectos como la capacidad de considerar el efecto de la τ_m o la variación de las direcciones principales.

- **Coste de cálculo asociado ($p_3 = 2$):** En este punto se va a tener en cuenta la relación entre las características de los resultados y su coste de cálculo asociado. Así, cuanto menor sea el coste de cálculo que se requiere para conseguir unos resultados aceptables, mejor se evaluará.

- **Objeto de estudio de la asignatura Métodos de Análisis y Diseño para Fractura y Fatiga ($p_3 = 2$):** Se valorará positivamente que los métodos sean ejemplos comunes en los que se profundice en dicha asignatura.

A continuación, se procede al análisis de cada uno de los métodos considerados asignando a cada alternativa, en función del grado de cumplimiento del criterio, una puntuación que oscilará desde los 0 puntos (mínima) hasta los 10 puntos (máxima).

Métodos empíricos específicos

Dentro de este grupo se está haciendo referencia a los métodos de Hohenemser y Prager y, Gough y Pollard. El primero se desarrolló en 1933 para el caso de carga axial constante y torsión alterna y por su parte, el segundo se desarrolló en 1935 para el caso particular de flexión alterna y torsión alterna.

Tal y como se puede deducir, se trata de métodos específicos para unos casos de carga concretos por lo que, su rango de aplicación queda limitado a ellos. Esto hace que en la actualidad se encuentren prácticamente en desuso y que, en su lugar se empleen otros métodos más generalistas y con mayor rango de aplicación.

En cuanto a su influencia dentro de la asignatura en la que se contextualiza el presente trabajo, se puede decir que no adquieren gran relevancia. Esto se debe a que, a pesar de que se mencionan por haber sido los primeros que se desarrollaron, su limitado campo de aplicación y el ajustado tiempo disponible para impartir el temario, premia el profundizar en otros métodos en lugar de en éstos.

Teniendo en cuenta lo mencionado en los párrafos anteriores, sobre la Tabla 6.1 se recogen las calificaciones asignadas a los métodos empíricos específicos.

Tabla 6.1. Calificaciones de los métodos empíricos específicos.

CRITERIO	Métodos empíricos específicos
Utilidad actual	4
Rango de aplicación	6
Coste de cálculo asociado	10
Objeto de estudio de MAyDFF	3

Métodos empíricos clásicos

Con el grupo de los métodos empíricos clásicos se hace referencia a los métodos de Von Mises y de Tresca. Estos métodos se basan en las teorías de fallo estático por lo que se puede considerar que las hipótesis de partida empleadas carecen de rigurosidad. Pese a ello, en ocasiones donde se desean analizar piezas de poca responsabilidad o realizar cálculos preliminares, se suelen emplear debido a la sencillez de sus cálculos.

Junto con esto, cabe mencionar cómo, aunque tengan un campo de aplicación mayor al de los métodos empíricos específicos, disponen de diversas limitaciones que hacen que su utilidad sea menor a la de los métodos avanzados. Entre ellas, se destaca el tratamiento de estados tensionales hidrostáticos y la falta de consideración tanto de la tensión tangencial media como de la variación de las direcciones principales.

Por todo ello, se suele dedicar una parte del tiempo de la asignatura de MAyDFF a la explicación de estos métodos, pero, debido a que tienen una versatilidad menor que algunos métodos avanzados, el nivel de profundización es menor. Así pues, sobre la Tabla 6.2 se muestran las calificaciones correspondientes.

Tabla 6.2. Calificaciones de los métodos empíricos clásicos.

CRITERIO	Métodos empíricos clásicos
Utilidad actual	7
Rango de aplicación	7
Coste de cálculo asociado	8
Objeto de estudio de MAyDFF	8

Métodos de enfoque global “grupo 1”

Dentro del “grupo 1” de los métodos de enfoque global se recogen los métodos de Sines y de Crossland que, corresponden a los criterios más empleados dentro de los métodos de enfoque global. A diferencia de los métodos empíricos, estos métodos parten de una base rigurosa y trabajan con una función de daño que permite trabajar con estados tensionales hidrostáticos.

Así pues, en ocasiones, cuando no se pueden emplear métodos básicos se puede trabajar con éstos sin necesidad de recurrir a métodos más sofisticados como los de plano crítico. Sin embargo, al no considerar el efecto de la variación de las direcciones principales, hay algunas situaciones en las que no son válidos y, hay que utilizar los métodos de plano crítico. Esto se debe a que el estudio de fatiga se realiza a nivel de punto mientras que, en los de plano crítico, se hace a nivel de plano.

Por último, en cuanto a la influencia de los métodos de Sines y de Crossland dentro de la asignatura de MAyDFF se puede decir que abarcan prácticamente la totalidad de la fracción de tiempo dedicada a los métodos de enfoque global; en buena parte, debido a su utilidad y su amplio campo de aplicación. Por todo esto, las calificaciones asignadas a estos métodos son las que se indican sobre la Tabla 6.3.

Tabla 6.3. Calificaciones de los métodos de enfoque global "grupo 1".

CRITERIO	Métodos de enfoque global "grupo 1"
Utilidad actual	8
Rango de aplicación	9
Coste de cálculo asociado	10
Objeto de estudio de MAyDFF	10

Métodos de enfoque global "grupo 2"

El "grupo 2" correspondiente a los métodos de enfoque global hace referencia al método GAM (Gonçalves, Araújo, Mamiya) [18]. A pesar de que se base en un planteamiento sólido, donde se define la función de daño a partir de tensor desviador de tensiones y de la tensión principal máxima, su utilidad actual es inferior a la de los métodos de Sines o Crossland ya que se enfoca principalmente para casos sencillos de fatiga multiaxial.

En este sentido, cuando se tienen estados tensionales en los que intervienen las componentes medias, los resultados que se obtienen tienden a ser más conservadores que los que se consiguen con otros métodos. Por otra parte, cuando el estado tensional es biaxial los resultados a los que se llega son poco conservadores y, por tanto, hay que tenerlo en cuenta.

Por otro lado, la influencia de este método dentro de las horas lectivas de la asignatura de MAyDFF es prácticamente insignificante ya que, como se ha mencionado previamente, la mayor parte del tiempo se dedica a los métodos de Sines y de Crossland por considerarse más relevantes. Con todo ello, se asignan las calificaciones correspondientes al método sobre la Tabla 6.4.

Tabla 6.4. Calificaciones de los métodos de enfoque global "grupo 2".

CRITERIO	Métodos de enfoque global "grupo 2"
Utilidad actual	6
Rango de aplicación	7
Coste de cálculo asociado	9
Objeto de estudio de MAyDFF	3

Métodos de plano crítico "grupo 1"

Dentro del "grupo 1" de los métodos de plano crítico se agrupan cuatro de los métodos de plano crítico más empleados: Matake, Findley, Papuga PCr (Papuga Critical Plane) y Dang Van. Este tipo de métodos, a diferencia de los anteriores, son capaces de estudiar también el efecto de la variación de las direcciones principales, lo cual vence la limitación de los métodos de enfoque global y hace que su campo de aplicación sea mayor.

Junto con esto, cabe remarcar cómo, aunque para poder considerar la variación de las direcciones principales aumente el coste de cálculo, en rasgos generales se trata de los métodos más completos. Esto se debe a que se estudia la fatiga a nivel de plano en lugar de a nivel de punto, como ocurre en los métodos de enfoque global.

Así pues, su influencia en la asignatura de MAyDFF es importante y prueba de ello es que, unido al denominado "grupo 1" de los métodos de enfoque global, abarcan la mayor parte del tiempo destinado a la docencia de los métodos de análisis de fatiga con tensiones multiaxiales. Teniendo todo esto en cuenta, sobre la Tabla 6.5 se presentan las calificaciones que se le han asignado.

Tabla 6.5. Calificaciones de los métodos de plano crítico "grupo 1".

CRITERIO	Métodos de plano crítico "grupo 1"
Utilidad actual	9
Rango de aplicación	10
Coste de cálculo asociado	10
Objeto de estudio de MAyDFF	9

Métodos de plano crítico "grupo 2"

El "grupo 2" de los métodos de plano crítico recoge los métodos de Susmel, Robert, McDiarmid, SPM (Spagnoli modificado) y QCP (Quadratic Critical Plane). Dado que pertenecen al tipo de métodos de plano crítico, se pueden considerar métodos versátiles, pero, en comparación con los métodos del denominado "grupo 1" de métodos de plano crítico, se utilizan en menor medida.

Como consecuencia de su menor utilidad y de la limitación de tiempo que se dispone a la hora de impartir la asignatura de MAyDFF, la gran parte del tiempo que se dedica a los métodos de plano crítico se destina a los métodos de plano crítico "grupo 1". Así pues, en base a lo explicado se han asignado las calificaciones que se muestran en la Tabla 6.6.

Tabla 6.6. Calificaciones de los métodos de plano crítico "grupo 2".

CRITERIO	Métodos de plano crítico "grupo 2"
Utilidad actual	6
Rango de aplicación	10
Coste de cálculo asociado	10
Objeto de estudio de MAyDFF	4

Métodos de plano crítico integrales

Dentro de los métodos de plano crítico integrales se agrupan los métodos de Papuga PI (Papuga Integral), Fogue y LZ (Liu y Zenner). Éstos métodos se pueden considerar como una continuación de los métodos de plano crítico ya que, en lugar de identificar el plano crítico y aplicar la función de daño en él, integran los resultados obtenidos de calcular la función de daño en todos los planos.

En teoría, con esto se pretende conseguir que los resultados se asemejen más a la realidad ya que se está considerando el conjunto de planos estudiados y no solamente el más crítico. Sin embargo, en la realidad, esto supone que el coste de cálculo asociado se incremente de forma notable, y los resultados obtenidos se asemejan a los que se consiguen con los métodos de plano crítico no integrales [14].

Todo ello, unido a lo acotadas que se encuentran las horas lectivas de la asignatura de MAyDFF, hace que su tratamiento sea introductorio; dedicando más tiempo a otros métodos que se consideran más relevantes. Así pues, considerando los diferentes aspectos que se han mencionado, sobre la Tabla 6.7 se recogen sus calificaciones.

Tabla 6.7. Calificaciones de los métodos de plano crítico integrales.

CRITERIO	Métodos de plano crítico integrales
Utilidad actual	4
Rango de aplicación	10
Coste de cálculo asociado	3
Objeto de estudio de MAyDFF	3

Una vez analizados los criterios de los diferentes grupos de métodos de análisis de fatiga multiaxial, y tras asignar a cada uno de ellos sus correspondientes calificaciones,

sobre la Tabla 6.8 se calcula el Valor Técnico Ponderado para poder así escoger los métodos más convenientes.

Tabla 6.8. Resultados de la valoración de los diferentes grupos de métodos.

	CRITERIO	*Utilidad actual*	*Rango de aplicación*	*Coste de cálculo asociado*	*Objeto de estudio de MAyDFF*	**Valor Técnico Ponderado**
	PESO	**3**	**3**	**2**	**2**	
GRUPOS DE MÉTODOS	Empíricos específicos	4	6	10	3	5,6
	Empíricos clásicos	7	7	8	8	7,4
	Enfoque global "grupo 1"	8	9	10	10	9,1
	Enfoque global "grupo 2"	6	7	9	3	6,3
	Plano crítico "grupo 1"	9	10	10	9	9,5
	Plano crítico "grupo 2"	6	10	10	4	7,6
	Plano crítico integrales	4	10	3	3	5,4

A la vista de los resultados obtenidos en la Tabla 6.8, los métodos de enfoque global del "grupo 1" y los métodos de plano crítico de "grupo 1" son las opciones con mayor puntuación. De hecho, el Valor Técnico Ponderado de ambos métodos es de sobresaliente (9,1 y 9,5 puntos) mientras que, la mayor calificación del resto de alternativas es de 7,6 puntos por lo que, se considera que estos dos grupos de métodos son los más ventajosos.

Finalmente, cabe destacar también como las alternativas escogidas cumplen, con puntuaciones remarcables, todos y cada uno de los criterios analizados. Así pues, estudiada la transcendencia de dichos métodos, va a ser sobre ellos en torno a los que se va a desarrollar la aplicación.

6.2 Elección de la herramienta software

Una vez escogidos los métodos de fatiga multiaxial que se desean particularizar en la aplicación, en el siguiente paso se trata de seleccionar la herramienta a emplear para el desarrollo de la interfaz gráfica. Para ello, se va a comenzar, al igual que se ha hecho para la elección de los métodos de fatiga multiaxial, definiendo los criterios y ponderaciones que se van a utilizar en el análisis de las diferentes alternativas.

- **Facilidad de uso ($\boldsymbol{p_1 = 1}$):** Se valorará que los comandos que posee el software sean sencillos e intuitivos.

- **Flexibilidad $(p_2 = 1)$:** Se tendrá en cuenta la facilidad con la que se puedan implementar cambios o variaciones sobre la interfaz.
- **Tiempos de simulación $(p_2 = 2)$:** Cuanto menor sea este parámetro, mayor ahorro de tiempo supondrá y por tanto, mejor se valorará.
- **Posibilidad de desarrollar una aplicación $(p_4 = 4)$:** Corresponde a una parte importante del trabajo por lo que, aquellos softwares con esta cualidad serán considerados positivamente.
- **Coste de la licencia $(p_5 = 2)$:** Cuanto menor sea su coste, más accesible será y por tanto, mejor se valorará.

A continuación, al igual que se ha hecho en el subapartado anterior con el análisis de alternativas de los métodos de fatiga multiaxial, se va a realizar un estudio de las herramientas disponibles. Asimismo, se va a asignar a cada una de las opciones una calificación que oscilará entre 0 puntos (mínima) y 10 puntos (máxima) en función del grado de cumplimiento de cada criterio y se va a decidir cuál de ellas es más conveniente para abordar el trabajo.

Visual Basic

Se trata de un lenguaje de programación situado en el entorno de Microsoft. Su objetivo es proporcionar una herramienta que permita el diseño de aplicaciones con seguridad tanto para ordenadores como para móviles. Así pues, a continuación, se van a analizar las principales ventajas y desventajas de este software.

En primer lugar, se debe mencionar que es un software libre tanto para usuarios individuales como para organizaciones que cumplan una serie de requisitos. Asimismo, otra de sus ventajas es la facilidad de uso de los comandos que incorpora; lo cual resulta ventajoso ya que no requiere demasiado tiempo para su aprendizaje [19].

Además de lo anterior, incorpora un entorno de desarrollo integrado en el que se permite editar interfaces gráficas de usuario y a continuación, compilarlas. Finalmente, cabe remarcar entre sus virtudes la capacidad para personalizar tablas de datos y realizar diversas operaciones con ellos.

Por otro lado, en cuanto a las desventajas de Visual Basic se pueden destacar su falta de flexibilidad y sus poco optimizados tiempos de simulación; en comparación con otras herramientas que se expondrán más adelante. Por último, pero no menos importantes, son los diferentes problemas que surgen al crear aplicaciones con varias interfaces.

A partir de todo lo anterior, en la Tabla 6.9 se recogen todas las calificaciones que se asignan a esta herramienta software.

Tabla 6.9. Calificaciones de la herramienta software Visual Basic.

CRITERIO	Visual Basic
Facilidad de uso	9
Flexibilidad	6
Tiempos de simulación	6
Posibilidad de desarrollar una aplicación	5
Coste de la licencia	4

Python

Python es uno de los lenguajes de programación más importantes dentro del grupo de softwares libres. Se encuentra administrado por la compañía *Python Software Foundation* y el principal objetivo que se persigue es la facilidad tanto de lectura como de diseño de la programación [20]. A continuación, se analizarán las ventajas y desventajas que lo describen.

En primer lugar, dispone de un lenguaje simplificado y ordenado que permite al usuario adoptar las nociones básicas para poder comenzar a programar. Asimismo, se trata de un lenguaje flexible en el que se incorporan varias herramientas para facilitar al usuario la programación.

Por otro lado, en este software se tiene la oportunidad de desarrollar aplicaciones a través de *WXPython* y de *Python Tkinter*. Para el empleo de la primera herramienta se requiere de un know-how superior al de *Python Tkinter*; lo cual no interesa para el desarrollo del presente trabajo por lo que, se va a tratar sobre la segunda.

Entre sus ventajas, destaca la extensa documentación que existe y la simplicidad de los comandos que la conforman [21]. Sin embargo, presenta varios inconvenientes entre los que se destacan la falta de elementos gráficos, tiempos de ejecución poco optimizados y la escasa versatilidad. Asimismo, hay varias librerías que no se instalan por defecto con el software y, por tanto, hay que buscarlas e introducirlas según la necesidad.

Por último, otra de las desventajas que se debe recalcar sobre Python, es el aumento de complejidad que representa la curva de aprendizaje a medida que aumenta el tiempo empleado. Así pues, a partir de todo lo anterior, se completa la Tabla 6.10.

Tabla 6.10. Calificaciones de la herramienta software Python.

CRITERIO	Python
Facilidad de uso	9
Flexibilidad	6
Tiempos de simulación	6
Posibilidad de desarrollar una aplicación	7
Coste de la licencia	10

Matlab

Se trata de un software perteneciente a la compañía MathWorks en el que se combinan el cálculo numérico, la visualización y la programación de un modo práctico. Su objetivo es proporcionar un acceso fácil al software matricial para llevar a cabo diferentes fines como pueden ser el análisis de datos o la creación de interfaces gráficas de usuario. A continuación, se analizarán las ventajas y desventajas que posee.

En primer lugar, dispone de un entorno de trabajo sencillo de manejar proporcionando una serie de comandos intuitivos para facilitar el trabajo del usuario. Asimismo, tiene una elevada flexibilidad; lo cual permite la implementación de diferentes modificaciones sin que ello suponga una tarea excesivamente laboriosa [22].

Del mismo modo, otra de las principales características de este software es la posibilidad de crear, aplicaciones. Para ello, contiene dos desarrolladores de aplicaciones: GUIDE (el primero que se desarrolló) y App Designer, siendo este último el entorno con mayor potencial [23]. En esta misma línea, cabe destacar que, estos softwares disponen de varias optimizaciones para poder así, ejecutar las funciones de una manera rápida y precisa.

Por otro lado, a pesar de no ser de uso libre, si se utiliza para una práctica académica o de investigación, su licencia es abierta. De ahí, que en instituciones como las Universidades se puedan llevar a cabo proyectos con dicha herramienta [24]. Así pues, teniendo en cuenta todo lo anterior, en la Tabla 6.11 se recogen todas las calificaciones que se asignan a esta herramienta software.

Tabla 6.11. Calificaciones de la herramienta software Matlab.

CRITERIO	**Matlab**
Facilidad de uso	9
Flexibilidad	9
Tiempos de simulación	9
Posibilidad de desarrollar una aplicación	10
Coste de la licencia	4

Octave

Octave es una de las alternativas más importantes para competir con Matlab. Esto se debe a que, a diferencia de Matlab, Octave es una herramienta de software libre para todos los usuarios. Así pues, a pesar de que ambos utilizan un lenguaje de programación semejante, existen una serie de diferencias que se va a proceder a mencionar junto con las características de Octave.

En primer lugar, la sintaxis utilizada es similar a la que emplea Matlab; de modo que se dispone de un entorno de trabajo sencillo de utilizar para el usuario. Junto con esto,

debido a las semejanzas que ambos softwares presentan, los tiempos de simulación que se requieren también se encuentran en torno a los mismos valores.

Sin embargo, Octave carece de varias de las herramientas de las que posee Matlab; lo que reduce su capacidad para llevar a cabo ciertas tareas. En esta línea, se puede encontrar la posibilidad de generar interfaces gráficas de usuario; donde los resultados obtenidos con Octave son notablemente inferiores a los que se consiguen mediante herramientas como App Designer en Matlab.

Así pues, a partir de todo lo mencionado anteriormente, sobre la Tabla 6.12 se recogen las calificaciones correspondientes a la herramienta software Octave.

Tabla 6.12. Calificaciones de la herramienta software Octave.

CRITERIO	**Octave**
Facilidad de uso	9
Flexibilidad	9
Tiempos de simulación	9
Posibilidad de desarrollar una aplicación	5
Coste de la licencia	10

LabVIEW

Se trata de una herramienta software que se enmarca en un entorno de desarrollo gráfico. Pertenece al desarrollador National Instruments y su principal objetivo es el desarrollo de sistemas de diseño, control y pruebas [25]. A continuación, se mostrarán las ventajas y desventajas que caracterizan a dicho software.

Para comenzar, entre sus ventajas se destacan el intuitivo lenguaje de programación y la facilidad de uso que presenta; en buena medida debido a que se encuentra enfocada a la visualización de cada aspecto de la herramienta. Asimismo, otra de sus virtudes es el reducido tiempo que requiere para llevar a cabo las simulaciones.

Por otro lado, este software permite el desarrollo de programas con una complejidad remarcable en los ámbitos de la adquisición de datos, análisis de medidas y presentación de datos. Es por eso que, otra de sus características es la posibilidad que ofrece para la extracción de información significativa a partir del análisis de datos [25].

Respecto a su licencia de uso, dispone de licencias temporales libres para aquellas personas que se encuentren en el ámbito académico o de la investigación mientras que, para el resto de organizaciones será necesario el pago de la misma. Tras esto y, teniendo en cuenta todo lo comentado previamente, sobre la Tabla 6.13 se van a recoger las calificaciones asignadas.

Tabla 6.13. Calificaciones de la herramienta software LabView.

CRITERIO	**LabVIEW**
Facilidad de utilización	9
Flexibilidad	8
Tiempos de simulación	9
Posibilidad de desarrollar una aplicación	7
Coste de la licencia	2

Una vez estudiadas cada una de las alternativas que se consideran candidatas a ser la herramienta software en la que se desarrolle la aplicación, lo siguiente es calcula el Valor Técnico Ponderado de todas ellas (ver Tabla 6.14) para poder elegir, en base a un criterio definido, la mejor opción.

Tabla 6.14. Resultados de la valoración de la herramienta software.

CRITERIO	*Facilidad de uso*	*Flexibilidad*	*Tiempos de simulación*	*Posibilidad de desarrollar una aplicación*	*Coste de la licencia*	**Valor Técnico Ponderado**
PESO	**1**	**1**	**2**	**4**	**2**	
Visual Basic	9	6	6	5	4	5,5
Python	9	6	6	7	10	7,5
Matlab	9	9	9	10	4	8,4
Octave	9	9	9	5	10	7,6
LabView	9	8	9	7	2	6,7

A la vista de los resultados obtenidos mediante la Tabla 6.14, se puede concluir que la herramienta software más adecuada para el desarrollo de la aplicación es Matlab, y más concretamente, su herramienta de App Designer. Esto, se debe en gran medida a las posibilidades que ofrece para el desarrollo de aplicaciones; lo cual corresponde con una parte importante de este trabajo.

Además de lo anterior, se puede deducir también, como la principal desventaja que presenta Matlab es el coste de su licencia ya que no se trata de un software libre. Pese a ello, el resto de características positivas que posee superan a este punto negativo y, además, cabe decir que, como el presente trabajo se enmarca en el contexto universitario de la UPV/EHU, los alumnos disponen licencia de uso educativo.

Por lo tanto, tras el análisis de alternativas que se ha llevado a cabo, se ha optado por realizar una aplicación en el entorno de Matlab donde se particularicen los siguientes métodos de fatiga multiaxial: Sines, Crossland, Matake, Findley, Papuga PCr y Dang Van. Sobre las características de cada uno de estos métodos se ha tratado en el apartado 5. Descripción de requerimientos y estado del arte por lo que, conviene hacer lo propio

con las principales cualidades de la herramienta software que se va a emplear: Matlab y, más concretamente, App Designer.

Matlab es un software creado principalmente para realizar cálculos matemáticos, especialmente con matrices; de ahí su nombre "MATrix LABoratory". Sin embargo, además de ser capaz de ejecutar de forma eficaz dichos cálculos, puede crear gráficos tanto bidimensionales como tridimensionales, trabajar con números complejos, logaritmos, funciones trigonométricas…En definitiva, permite realizar las mismas funciones de una calculadora científica, y muchas más.

Junto con esto, cabe destacar la posibilidad que ofrece Matlab desde sus primeras versiones para poder programar. Así, se permite combinar las funciones previamente mencionadas con la secuencia de código deseada para que el programa ejecute de forma secuencial una sucesión de comandos. Del mismo modo, el usuario puede introducir tantas funciones como desee sobre el programa principal.

Como consecuencia de todas las mejoras introducidas a lo largo de las diferentes versiones de la herramienta, Matlab se ha convertido en una herramienta estándar para ingenieros y científicos. En virtud de ello, en diversas disciplinas de estos ámbitos, se ha sustituido la programación tradicional por un programa de computación matemático como Matlab.

Por otro lado, en la versión que se va a emplear de Matlab (R2021a) se incluyen herramientas adicionales como es el caso de App Designer para el desarrollo de aplicaciones. Se trata de un amplio entorno de desarrollo que permite la creación de apps útiles a usuarios que no tienen por qué ser desarrolladores de software profesional.

Esta herramienta dispone de numerosas opciones que facilitan la creación de las interfaces gráficas de usuario. Dentro de ellas, destacan los componentes interactivos, el administrador de diseño de cuadrículas y la opción de reordenación automática para que la aplicación se adecúe a los cambios en el tamaño de pantalla [23].

Con el objetivo de simplificar la labor del usuario, las vistas de diseño y código se encuentran estrechamente relacionadas; de manera que, las modificaciones que se hacen sobre una de ellas afectan inmediatamente en la otra. Así pues, el propio software se encarga de generar el código correspondiente a medida que en la ventana de diseño se van añadiendo elementos a la interfaz.

Finalmente, cabe remarcar la posibilidad que existe de empaquetar la aplicación desarrollada mediante el compilador de App Designer. De esta manera, se puede generar un archivo ejecutable que se pueda utilizar en cualquier ordenador; independientemente de si en él se encuentra instalado Matlab o no.

II. METODOLOGÍA SEGUIDA EN EL DESARROLLO DEL TRABAJO

7 Descripción de tareas

En este apartado se van a describir de manera cronológica las diferentes tareas en las que se ha clasificado el presente proyecto. Además, junto con la explicación de cada tarea, se va a mencionar la unión que tiene con el resto de ellas.

Tarea 1 (T1): Seguimiento del trabajo con el director

Para poder verificar que el proyecto se realiza correctamente se dispone del apoyo del director del TFM, Mikel Abasolo Bilbao. En las reuniones que se mantienen con él, además de resolver las diferentes dudas que puedan surgir, se decide cuál es la mejor trayectoria que se debe seguir por lo que, esta tarea se va a prolongar hasta el final del trabajo.

Tarea 2 (T2): Definición del trabajo

Corresponde a la primera tarea y consiste en decidir que el tema principal en torno al que se va a realizar el trabajo es el desarrollo de una aplicación para la particularización de los métodos de fatiga multiaxial a los casos de fatiga uniaxial y torsión pura. Además, se definirán los requisitos y las especificaciones que se desean incluir en dicha aplicación y que el software que se va a utilizar para su desarrollo sea Matlab.

Tarea 3 (T3): Búsqueda de información

Tras la definición del proyecto, comienza la recopilación de información en los diferentes ámbitos que intervienen en el proceder del trabajo como son principalmente: los métodos de fatiga multiaxial y el software Matlab. Así pues, esta tarea se subdivide en las tareas T3.1 y T3.2 que se explican a continuación, y se tratará de una labor que durará tanto tiempo como el que se emplee en desarrollar la propia app.

Tarea 3.1 (T3.1): Fatiga multiaxial

Dado que los conceptos teóricos que se requieren para llevar a cabo el trabajo se encuentran recogidos en los apartados correspondientes de la asignatura MAyDFF, se comienza por repasarlos. Tras esto, también se recopila información de diferentes fuentes que sirva como complemento de los conceptos con los que se va a trabajar.

Tarea 3.2 (T3.2): Matlab y App Designer

Se trata de coger destreza con el software Matlab y de familiarizarse con la interfaz de App Designer; la herramienta en la que se va a desarrollar la aplicación. Para ello, además de consultar diferentes referencias bibliográficas, se realizarán sencillos

ejercicios que permitan comprobar si se están asimilando los comandos, funciones… que dispone el software.

Tarea 4 (T4): Diseño de la aplicación

Consiste en realizar un diseño preliminar del aspecto que adoptará la aplicación que se desea desarrollar. Así pues, en esta fase se va a decidir sobre la mejor opción para poder distribuir de forma ordenada toda la información que contiene la app.

Tarea 5 (T5): Desarrollo de la aplicación

Una vez organizadas las diferentes partes que constituirán la aplicación (T4), se está en disposición de comenzar con la siguiente tarea. En ésta, el núcleo consta de realizar conjuntamente las dos siguientes labores: por un lado, particularizar los métodos de fatiga multiaxial recogidos en el subapartado 3.2. Alcance a los casos de fatiga uniaxial y de torsión pura y por otro lado, implementarlos en App Designer.

Pese a no corresponder al núcleo principal, también es importante lo que se incluye en el intervalo "Aspectos adicionales"; ya que en él se engloban otras características que enriquecen el trabajo. Entre ellas, destaca la posibilidad de adjuntar materiales y la programación de los criterios de ajuste de fatiga uniaxial; junto con su extrapolación al caso de torsión. Así pues, se subdividirán las tareas de la siguiente manera.

Tarea 5.1 (T5.1): Métodos de enfoque global

Tarea 5.1.1 (T5.1.1): Método de Sines

Tarea 5.1.2 (T5.1.2): Método de Crossland

Tarea 5.2 (T5.2): Métodos de plano crítico

Tarea 5.2.1 (T5.2.1): Método de Matake

Tarea 5.2.2 (T5.2.2): Método de Findley

Tarea 5.2.3 (T5.2.3): Método de Dang Van

Tarea 5.2.4 (T5.2.4): Método de Papuga PCr

Tarea 5.3 (T5.3): Aspectos adicionales

Tarea 6 (T6): Validación y depuración del código

Después de finalizar con el desarrollo de la aplicación, se analizará para verificar que los resultados que proporciona son lógicos y corresponden a lo establecido. Tras esto, se depurará el código para que adquiera un aspecto más homogéneo y se introducirán algunos comentarios adicionales que faciliten su lectura.

Tarea 7 (T7): Documentación del trabajo

Finalmente, una vez completada la validación y la depuración del código, se va a redactar el informe del trabajo y a continuación, se va a preparar una presentación para su posterior Defensa. Por lo tanto, esta tarea a su vez, se divide en las dos que siguen.

Tarea 7.1 (T7.1): Redacción del trabajo

Se trata de documentar en un informe escrito el trabajo que se ha desarrollado, exponiendo, entre otros, los cálculos y el código que se han empleado.

Tarea 7.2 (T7.2): Preparación de la presentación

Consiste en preparar una presentación para su posterior exposición oral en la Defensa del Trabajo de Fin de Máster ante el tribunal.

8 Diagrama de Gantt

Después de describir las diferentes tareas en las que se ha dividido el trabajo, se realiza el diagrama de Gantt de la Figura 8.1 a partir de los periodos de tiempo que se muestran en la Tabla 8.1.

Tabla 8.1. Duración de las tareas que constituyen el Diagrama de Gantt.

Tareas	Fecha de inicio	Duración (días)	Fecha de finalización
T1. Seguimiento del trabajo con el director	16/09/2021	259	02/06/2022
T2. Definición del trabajo	16/09/2021	7	23/09/2021
T3. Búsqueda de información	23/09/2021	176	18/03/2022
T3.1. Fatiga multiaxial	23/09/2021	176	18/03/2022
T3.2. Matlab y App Designer	30/10/2021	139	18/03/2022
T4. Diseño de la aplicación	10/11/2021	12	22/11/2021
T5. Desarrollo de la aplicación	22/11/2021	116	18/03/2022
T5.1. Métodos de enfoque global	22/11/2021	25	17/12/2021
T5.1.1. Método de Sines	22/11/2021	18	10/12/2021
T5.1.2. Método de Crossland	10/12/2021	7	17/12/2021
T5.2. Métodos de plano crítico	04/02/2022	31	07/03/2022
T5.2.1. Método de Matake	04/02/2022	10	14/02/2022
T5.2.2. Método de Findley	14/02/2022	6	20/02/2022
T5.2.3. Método de Dang Van	20/02/2022	5	25/02/2022
T5.2.4. Método de Papuga PCr	25/02/2022	10	07/03/2022
T5.3. Aspectos adicionales	07/03/2022	11	18/03/2022
T6. Validación y depuración del código	18/03/2022	10	28/03/2022
T7. Documentación del trabajo	28/03/2022	66	02/06/2022
T7.1. Redacción del trabajo	28/03/2022	51	18/05/2022
T7.2. Preparación de la presentación	18/05/2022	15	02/06/2022

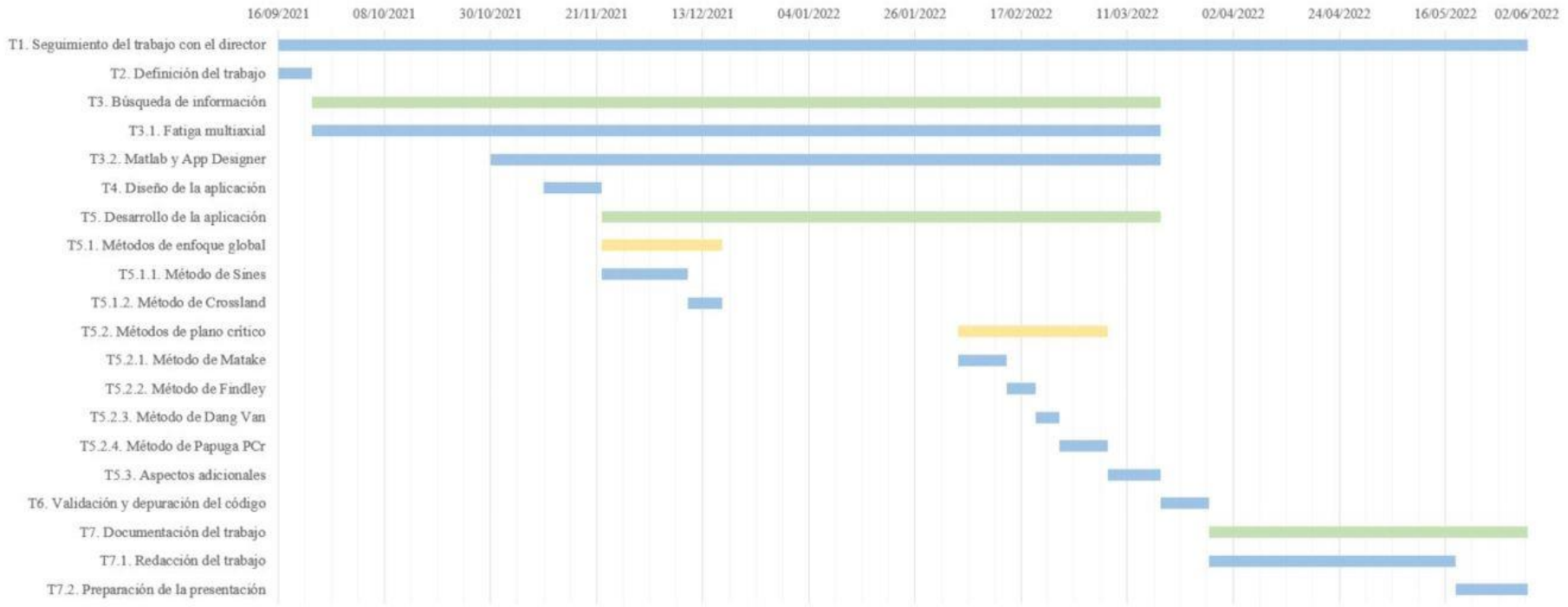

Figura 8.1. Diagrama de Gantt.

9 Cálculos

En el desarrollo de este trabajo, se han llevado a cabo una serie de cálculos que se pretenden exponer en el presente apartado. Para ello, esta sección se va a dividir en dos grandes bloques: por un lado, obtención de los parámetros de los métodos de fatiga multiaxial y, por otro lado, particularización de los propios métodos a los casos de fatiga uniaxial y de torsión pura.

9.1 Parámetros de los métodos

En el subapartado 5.2. Conceptos teóricos, junto con la descripción de cada método, se han deducido algunos de los parámetros α y β de los métodos de fatiga multiaxial con los que se va a trabajar. Sin embargo, para particularizar cada uno de los métodos según las tres parejas de ensayos que se recogen en el alcance del trabajo ($\sigma_{-1} - \tau_{-1}$, $\sigma_{-1} - \sigma_0$, $\sigma_{-1} - \sigma_{ut}$), es necesario complementar los cálculos. Así pues, siguiendo el mismo orden del subapartado 5.2. Conceptos teóricos se van a deducir las expresiones de dichas constantes en cada caso y al final de cada método se agrupan a modo resumen.

9.1.1 Método de Sines

En este caso, las operaciones necesarias para obtener las constantes del método se han presentado en el subapartado 5.2.2.1. Método de Sines, por lo que únicamente conviene agrupar las expresiones de cada pareja de ensayos a modo resumen (ver Tabla 9.1) y destacar el siguiente aspecto. Tal y como se puede apreciar, el parámetro α_S con la pareja de ensayos $\sigma_{-1} - \tau_{-1}$ queda indefinido; lo cual se debe a que tanto del ensayo de flexión con tensión alterna como del ensayo de torsión, la única constante que se puede despejar es β_S.

Tabla 9.1. Parámetros del método de Sines en función de la pareja de ensayos.

Pareja de ensayos	$\boldsymbol{\alpha_S}$	$\boldsymbol{\beta_S}$
$\sigma_{-1} - \tau_{-1}$	α_S	$\frac{\sqrt{2}}{3} \cdot \sigma_{-1}$
$\sigma_{-1} - \sigma_0$	$\frac{\sqrt{2} \cdot (\sigma_{-1} - \sigma_0)}{\sigma_0}$	$\frac{\sqrt{2}}{3} \cdot \sigma_{-1}$
$\sigma_{-1} - \sigma_{ut}$	$\alpha_S = \frac{\sqrt{2} \cdot \sigma_{-1}}{\sigma_{ut}}$	$\frac{\sqrt{2}}{3} \cdot \sigma_{-1}$

9.1.2 Método de Crossland

A diferencia del método de Sines, en el subapartado 5.2.2.2. Método de Crossland no se deducen las expresiones de las constantes del método. Por lo tanto, en este punto se van a desarrollar las operaciones necesarias para obtener las expresiones de α_C y β_C a través de una metodología semejante a la empleada en el método de Sines,

Ensayo de flexión con tensión alterna

En la Figura 5.3 se representan tanto el estado tensional (a) como el círculo de Mohr (b) correspondientes a este caso. Basándose en esto, se particularizan los términos que componen la ecuación de Crossland mediante las fórmulas (5.6) y (5.30) y, a continuación, se sustituyen en la propia expresión que predice el fallo (ec. (5.29)).

$$\tau_{oct_a} = {}^1\!/_3 \cdot \sqrt{(\sigma_{-1} - 0)^2 + 0 + (0 - \sigma_{-1})^2} = \frac{\sqrt{2}}{3} \cdot \sigma_{-1} \tag{9.1}$$

$$\sigma_{h_{max}} = \frac{1}{3} \cdot (0 + \sigma_{-1}) = \frac{\sigma_{-1}}{3} \tag{9.2}$$

$$\frac{\sqrt{2}}{3} \cdot \sigma_{-1} + \alpha_C \cdot \frac{\sigma_{-1}}{3} = \beta_C \tag{9.3}$$

Ensayo de torsión alterna

Siguiendo la misma sistemática que en el caso del ensayo de flexión con tensión alterna pero, en este caso, basándose en el círculo de Mohr representado sobre la Figura 5.4 b), se plantea lo que sigue.

$$\tau_{oct_a} = {}^{1}/_{3} \cdot \sqrt{(\tau_{-1} - (-\tau_{-1}))^2 + (\tau_{-1} - 0)^2 + (-\tau_{-1} - 0)^2} = \frac{\sqrt{6}}{3} \cdot \tau_{-1} \tag{9.4}$$

$$\sigma_{h_{max}} = \frac{1}{3} \cdot (0 + 0) = 0 \tag{9.5}$$

$$\frac{\sqrt{6}}{3} \cdot \tau_{-1} + \alpha_C \cdot 0 = \beta_C \rightarrow \beta_C = \frac{\sqrt{6}}{3} \cdot \tau_{-1} \tag{9.6}$$

Una vez conocida la constante β_C, se puede despejar α_C sustituyendo la expresión (9.6) en la (9.3).

$$\frac{\sqrt{2}}{3} \cdot \sigma_{-1} + \alpha_C \cdot \frac{\sigma_{-1}}{3} = \frac{\sqrt{6}}{3} \cdot \tau_{-1} \rightarrow \alpha_C = \sqrt{2} \cdot \left(\sqrt{3} \cdot \frac{\tau_{-1}}{\sigma_{-1}} - 1\right) \tag{9.7}$$

Ensayo con tracción pulsante axial

Tomando como referencia la representación mostrada sobre la Figura 5.5 y mediante un procedimiento análogo al empleado con los ensayos anteriores, se obtienen las expresiones adyacentes.

$$\tau_{oct_a} = {}^{1}/_{3} \cdot \sqrt{(\sigma_0 - 0)^2 + (0)^2 + (0 - \sigma_0)^2} = \frac{\sqrt{2}}{3} \cdot \sigma_0 \tag{9.8}$$

$$\sigma_{h_{max}} = \frac{1}{3} \cdot (\sigma_0 + \sigma_0) = \frac{2}{3} \cdot \sigma_0 \tag{9.9}$$

$$\frac{\sqrt{2}}{3} \cdot \sigma_0 + \alpha_C \cdot \frac{2}{3} \cdot \sigma_0 = \beta_C \tag{9.10}$$

Igualando las expresiones (9.3) y (9.10), se deduce la constante α_C.

$$\frac{\sqrt{2}}{3} \cdot \sigma_{-1} + \alpha_C \cdot \frac{\sigma_{-1}}{3} = \frac{\sqrt{2}}{3} \cdot \sigma_0 + \alpha_C \cdot \frac{2}{3} \cdot \sigma_0 \rightarrow \alpha_C = \frac{\sqrt{2} \cdot (\sigma_0 - \sigma_{-1})}{\sigma_{-1} - 2 \cdot \sigma_0} \tag{9.11}$$

Tras conseguir α_C, el obtener β_C es directo sustituyendo la expresión (9.11) en la relación (9.3) o (9.10). En este caso, se sustituye sobre la (9.3).

$$\frac{\sqrt{2}}{3}\sigma_{-1} + \frac{\sqrt{2}\cdot(\sigma_0 - \sigma_{-1})}{\sigma_{-1} - 2\cdot\sigma_0}\frac{\sigma_{-1}}{3} = \beta_C \;\rightarrow\; \beta_C = \frac{\sqrt{2}}{3}\sigma_{-1}\left(\frac{-\sigma_0}{\sigma_{-1} - 2\cdot\sigma_0}\right) \tag{9.12}$$

Ensayo de tracción uniaxial estática

El último ensayo para el que se particulariza el método de Crossland es el ensayo de tracción uniaxial estática y, se trabaja del mismo modo que se ha realizado con los ensayos anteriores, pero, en base a la representación de la Figura 5.6.

$$\tau_{oct_a} = {}^1\!/_3 \cdot \sqrt{(0)^2 + (0)^2 + (0)^2} = 0 \tag{9.13}$$

$$\sigma_{h_{max}} = \frac{1}{3}\cdot(\sigma_{ut} + 0) = \frac{1}{3}\cdot\sigma_{ut} \tag{9.14}$$

$$0 + \alpha_C \cdot \frac{1}{3}\cdot\sigma_{ut} = \beta_C \tag{9.15}$$

Se resuelve por sustitución el sistema de dos ecuaciones con dos incógnitas que se forma a partir de las ecuaciones (9.3) y (9.15).

$$\frac{\sqrt{2}}{3}\cdot\sigma_{-1} + \alpha_C \cdot \frac{\sigma_{-1}}{3} = \alpha_C \cdot \frac{1}{3}\cdot\sigma_{ut} \;\rightarrow\; \alpha_C = \frac{\sqrt{2}\cdot\sigma_{-1}}{\sigma_{ut} - \sigma_{-1}} \tag{9.16}$$

Introduciendo la expresión *(9.16)* sobre la (9.3), se despeja β_C.

$$\frac{\sqrt{2}}{3}\cdot\sigma_{-1} + \frac{\sqrt{2}\cdot\sigma_{-1}}{\sigma_{ut} - \sigma_{-1}}\cdot\frac{\sigma_{-1}}{3} = \beta_C \;\rightarrow\; \beta_C = \frac{\sqrt{2}}{3}\sigma_{-1}\left(\frac{\sigma_{ut}}{\sigma_{ut} - \sigma_{-1}}\right) \tag{9.17}$$

Para finalizar, sobre la Tabla 9.2 se presentan a modo resumen las expresiones de los parámetros α_C y β_C a emplear con el método de Crossland en función de la pareja de ensayos que se utilice.

Tabla 9.2. Parámetros del método de Crossland en función de la pareja de ensayos.

Pareja de ensayos	$\boldsymbol{\alpha_C}$	$\boldsymbol{\beta_C}$
$\sigma_{-1} - \tau_{-1}$	$\sqrt{2} \cdot \left(\sqrt{3} \cdot \frac{\tau_{-1}}{\sigma_{-1}} - 1\right)$	$\frac{\sqrt{6}}{3} \cdot \tau_{-1}$
$\sigma_{-1} - \sigma_0$	$\frac{\sqrt{2} \cdot (\sigma_0 - \sigma_{-1})}{\sigma_{-1} - 2 \cdot \sigma_0}$	$\frac{\sqrt{2}}{3}\sigma_{-1}\left(\frac{-\sigma_0}{\sigma_{-1} - 2 \cdot \sigma_0}\right)$
$\sigma_{-1} - \sigma_{ut}$	$\frac{\sqrt{2} \cdot \sigma_{-1}}{\sigma_{ut} - \sigma_{-1}}$	$\frac{\sqrt{2}}{3}\sigma_{-1}\left(\frac{\sigma_{ut}}{\sigma_{ut} - \sigma_{-1}}\right)$

9.1.3 Método de Matake

Tras deducir las constantes de los métodos de enfoque global (Sines y Crossland), se va a hacer lo propio con los métodos de plano crítico. En este punto en concreto, se van a deducir los parámetros del método de Matake (α_M y β_M) para los ensayos que no se han desarrollado en el subapartado 5.2.3.1. Método de Matake; estos son: ensayo con tracción pulsante axial y ensayo de tracción uniaxial estática y, a continuación, se agruparán las expresiones de las constantes en función de la pareja de ensayos empleada.

Ensayo con tracción pulsante axial

Del mismo modo que se ha realizado con el ensayo de flexión con tensión alterna y el ensayo de torsión en el subapartado 5.2.3.1. Método de Matake, lo primero que se debe hacer es identificar el plano crítico. Para ello, se representa el estado tensional en el círculo de Mohr y mediante la expresión (5.41) se localiza el plano crítico como se muestra en la Figura 9.1.

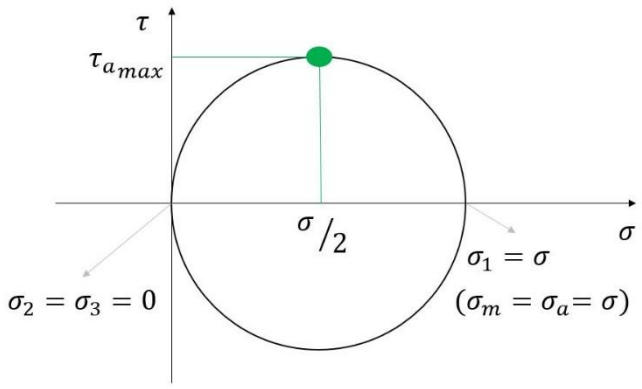

Figura 9.1. Identificación del plano crítico según el método de Matake en el ensayo de tracción pulsante axial.

Una vez identificado el plano crítico, se sustituyen los valores conocidos en la función de daño de Matake (ec. *(5.42)*) y se obtiene la relación que existe entre las constantes del método.

$$\frac{\sigma_0}{2} + \alpha_M \cdot \left(\frac{\sigma_0}{2} + \frac{\sigma_0}{2}\right) = \beta_M \quad \rightarrow \quad \alpha_M = \frac{2\beta_M - \sigma_0}{2\sigma_0} \tag{9.18}$$

Mediante las relaciones *(9.18)* y (5.43) se pueden obtener, por igualación, las constantes del método para la pareja de ensayos $\sigma_{-1} - \sigma_0$.

$$\frac{2\beta_M - \sigma_{-1}}{\sigma_{-1}} = \frac{2\beta_M - \sigma_0}{2\sigma_0} \quad \rightarrow \quad \beta_M = \frac{\sigma_{-1}}{2}\left(\frac{-\sigma_0}{\sigma_{-1} - 2\sigma_0}\right) \tag{9.19}$$

Finalmente, introduciendo la expresión de β_M sobre la ecuación (5.43) (también se puede hacer con la ecuación (9.18)), se obtiene la constante α_M.

$$\alpha_M = \frac{2\frac{\sigma_{-1}}{2}\left(\frac{-\sigma_0}{\sigma_{-1} - 2\sigma_0}\right) - \sigma_{-1}}{\sigma_{-1}} \quad \rightarrow \quad \alpha_M = \frac{\sigma_0 - \sigma_{-1}}{\sigma_{-1} - 2\sigma_0} \tag{9.20}$$

Ensayo de tracción uniaxial estática

Para poder expresar las constantes del método de Matake en función de los datos obtenidos con la pareja de ensayos $\sigma_{-1} - \sigma_{ut}$, se necesita establecer su relación a partir de la situación del ensayo de tracción uniaxial estático. Por lo tanto, del mismo modo que se ha realizado en los ensayos anteriores, se comienza identificando el plano crítico en el círculo de Mohr de acuerdo a la expresión (5.41) (ver Figura 9.2).

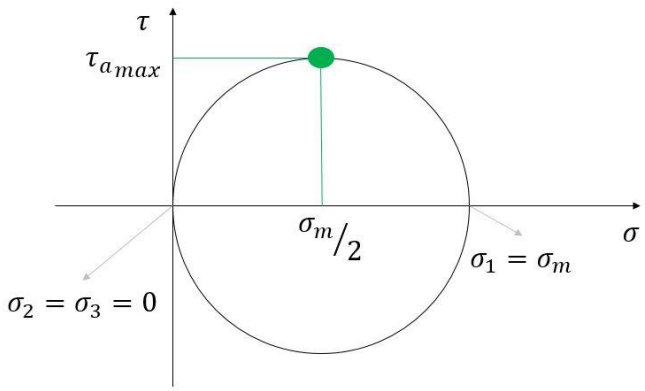

Figura 9.2. Identificación del plano crítico según el método de Matake en el ensayo de tracción uniaxial estática.

En base a la representación mostrada en la Figura 9.2, se sustituye sobre la función de daño de Matake (ec. (5.42)).

$$0 + \alpha_M \cdot \left(\frac{\sigma_{ut}}{2} + 0\right) = \beta_M \;\rightarrow\; \alpha_M = 2\frac{\beta_M}{\sigma_{ut}} \tag{9.21}$$

Resolviendo el sistema de ecuaciones lineales formado por las expresiones (5.43) y *(9.21)*, se obtienen las constantes del método de Matake para la pareja de ensayos $\sigma_{-1} - \sigma_{ut}$. En este caso, se emplea la técnica de resolución por sustitución.

$$\frac{2\beta_M - \sigma_{-1}}{\sigma_{-1}} = 2\frac{\beta_M}{\sigma_{ut}} \;\rightarrow\; \beta_M = \frac{-\sigma_{-1} \cdot \sigma_{ut}}{2 \cdot (\sigma_{-1} - \sigma_{ut})} \tag{9.22}$$

Sustituyendo la expresión *(9.22)* en (9.21) se deduce α_M.

$$\alpha_M = 2\frac{\frac{-\sigma_{-1} \cdot \sigma_{ut}}{2 \cdot (\sigma_{-1} - \sigma_{ut})}}{\sigma_{ut}} \;\rightarrow\; \alpha_M = \frac{-\sigma_{-1}}{\sigma_{-1} - \sigma_{ut}} \tag{9.23}$$

Tras deducir las constantes del método de Matake para cada una de las tres parejas de ensayos, sobre la Tabla 9.3 se recogen de forma resumida sus expresiones.

Tabla 9.3. Parámetros del método de Matake en función de la pareja de ensayos.

Pareja de ensayos	$\boldsymbol{\alpha_M}$	$\boldsymbol{\beta_M}$
$\sigma_{-1} - \tau_{-1}$	$2\frac{\tau_{-1}}{\sigma_{-1}} - 1$	τ_{-1}
$\sigma_{-1} - \sigma_0$	$\frac{\sigma_0 - \sigma_{-1}}{\sigma_{-1} - 2\sigma_0}$	$\frac{\sigma_{-1}}{2}\left(\frac{-\sigma_0}{\sigma_{-1} - 2\sigma_0}\right)$
$\sigma_{-1} - \sigma_{ut}$	$\frac{-\sigma_{-1}}{\sigma_{-1} - \sigma_{ut}}$	$\frac{-\sigma_{-1} \cdot \sigma_{ut}}{2 \cdot (\sigma_{-1} - \sigma_{ut})}$

9.1.4 Método de Findley

Otro de los métodos de plano crítico es el método de Findley. En este caso, al igual que ocurría con el método de Matake, del subapartado 5.2.3.2. Método de Findley

únicamente se obtienen las constantes del método para el ensayo de flexión alterna y de torsión alterna por lo que, se va a hacer lo propio con los ensayos de tracción pulsante axial y de tracción uniaxial estática.

Ensayo con tracción pulsante axial

Al igual que se ha realizado en otros métodos de plano crítico, el primer paso corresponde a identificar el plano crítico. Para ello, tal y como se puede apreciar sobre la Figura 9.3 se parte de una ubicación general en el círculo de Mohr del ensayo con tracción pulsante.

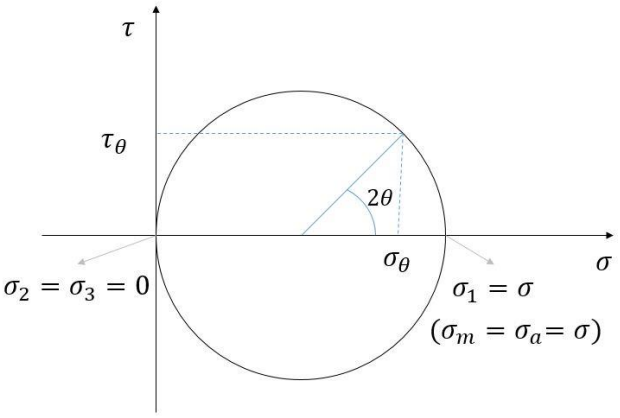

Figura 9.3. Identificación del plano crítico según el método de Findley en el ensayo de tracción pulsante axial.

En base a la Figura 9.3, se pueden definir los términos τ_θ y σ_θ como en las ecuaciones (5.48) y (5.49) y, derivando la ecuación que define el plano crítico (ec. (5.46)), se llega a una expresión general que determina su posición en el círculo de Mohr ($2\theta^*$). En este caso, al definir la expresión de forma genérica, coincide con la obtenida en el ensayo de flexión con tensión alterna (ec. (5.51)) pero, como es de esperar, al particularizarla para cada método su expresión varía.

Tras determinar el plano crítico, se definen los términos τ_{θ_a}, σ_{θ_m} y σ_{θ_a} como se muestra a continuación, para después, sustituirlos en la función de daño.

$$\tau_{\theta_a} = \frac{\sigma_0}{2} \cdot \sin(2\theta) \tag{9.24}$$

$$\sigma_{\theta_m} = \frac{\sigma_0}{2} \cdot (1 + \cos(2\theta)) \tag{9.25}$$

$$\sigma_{\theta\,a} = \frac{\sigma_0}{2} \cdot (1 + \cos(2\theta)) \tag{9.26}$$

Introduciendo los términos (9.24), (9.25) y *(9.26)* sobre la (5.47).

$$\frac{\sigma_0}{2} \cdot \sin(2\theta) + \alpha_F \cdot \left(\frac{\sigma_0}{2} \cdot (1 + \cos(2\theta)) + \frac{\sigma_0}{2} \cdot (1 + \cos(2\theta)) \right) = \beta_F \tag{9.27}$$

Sustituyendo en *(9.27)* el ángulo 2θ por la posición que se ha calculado para el plano crítico ($2\theta^*$) y empleando las relaciones (5.55) y (5.56).

$$\frac{\sigma_0}{2} \left(\left(\frac{\frac{\sigma_0}{\alpha_F(\sigma_0 + \sigma_0)}}{\sqrt{1 + \left(\frac{\sigma_0}{\alpha_F(\sigma_0 + \sigma_0)} \right)^2}} \right) + \alpha_F \sigma_0 \left(1 + \frac{1}{\sqrt{1 + \left(\frac{\sigma_0}{\alpha_F(\sigma_0 + \sigma_0)} \right)^2}} \right) \right) = \beta_F \tag{9.28}$$

Desarrollando *(9.28)*.

$$\frac{\sigma_0}{2} \left(\frac{1 + 2\alpha_F \sqrt{1 + 4{\alpha_F}^2} + 4{\alpha_F}^2}{2\sqrt{1 + 4{\alpha_F}^2}} \right) = \beta_F \tag{9.29}$$

Multiplicando en el numerador y en el denominador por $\sqrt{1 + 4{\alpha_F}^2}$ y simplificando, se llega a la expresión *(9.30)* que, establece una relación entre las constantes α_F y β_F.

$$\frac{\sigma_0}{2} \left(\sqrt{1 + 4{\alpha_F}^2} + 2\alpha_F \right) = \beta_F \tag{9.30}$$

Mediante las ecuaciones (5.58) y *(9.30)*, se puede plantear una ecuación de una única incógnita; por ejemplo, utilizando la técnica de sustitución como sigue.

$$\frac{\sigma_{-1}}{2} \left(\sqrt{{\alpha_F}^2 + 1} + \alpha_F \right) = \frac{\sigma_0}{2} \left(\sqrt{1 + 4{\alpha_F}^2} + 2\alpha_F \right) \tag{9.31}$$

$$\frac{\sigma_0}{\sigma_{-1}} = \frac{\sqrt{{\alpha_F}^2 + 1} + \alpha_F}{\sqrt{1 + 4{\alpha_F}^2} + 2\alpha_F} \tag{9.32}$$

$$\alpha_F(\sigma_{-1} - 2\sigma_0) - \sigma_0\sqrt{1 + 4{\alpha_F}^2} + \sigma_{-1}\sqrt{{\alpha_F}^2 + 1} = 0 \tag{9.33}$$

Tal y como se puede apreciar en la ecuación *(9.33)*, se trata de una ecuación no lineal cuya resolución se realiza mediante métodos numéricos; en concreto, con el método de Newton-Raphson. Tras esto, se conoce el valor de α_F y por tanto, solamente hay que sustituirlo en una de las dos ecuaciones de partida ((5.58) y (9.30)) para obtener el valor de la constante β_F.

Ensayo de tracción uniaxial estática

El último de los ensayos con los que se deducen los parámetros de Findley corresponde al ensayo de tracción uniaxial estática. Al igual que se ha realizado con el ensayo de tracción pulsante axial, se comienza definiendo el plano crítico y para ello, se representa sobre el círculo de Mohr de la Figura 9.4 un caso genérico.

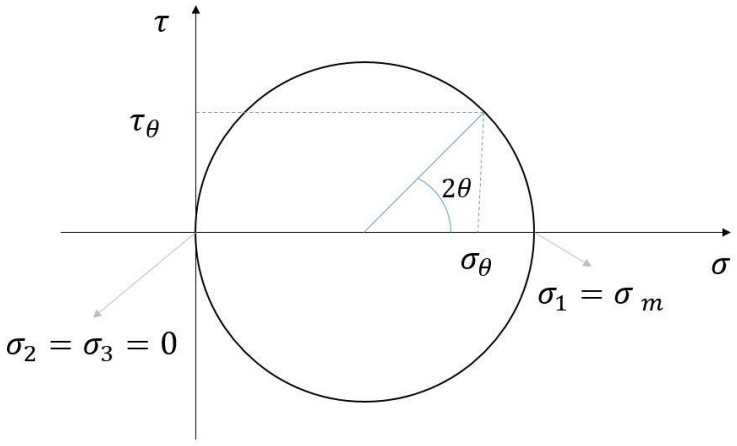

Figura 9.4. Identificación del plano crítico según el método de Findley en el ensayo de tracción uniaxial estática.

Empleando los términos genéricos que aparecen en la Figura 9.4 y haciendo uso de la derivada de la ecuación que define el plano crítico, se obtiene la expresión de $2\theta^*$. Del mismo modo que ocurría con el ensayo de flexión pulsante axial, ésta coincide con la ecuación *(5.51)* y al sustituir los términos que intervienen, se va a particularizar. Así pues, lo siguiente es definir los términos $\tau_{\theta\,a}$, $\sigma_{\theta\,m}$ y $\sigma_{\theta\,a}$.

$$\tau_{\theta\,a} = 0 \tag{9.34}$$

$$\sigma_{\theta\,m} = \frac{\sigma_{ut}}{2} \cdot (1 + \cos(2\theta)) \tag{9.35}$$

$$\sigma_{\theta\,a} = 0 \tag{9.36}$$

Una vez conocidas las expresiones de los términos que intervienen en la función de daño (ec. (5.47)), se sustituye para obtener la relación entre los parámetros del método.

$$0 + \alpha_F \cdot \left(\frac{\sigma_{ut}}{2} \cdot (1 + \cos(2\theta)) + 0\right) = \beta_F \tag{9.37}$$

Introduciendo la expresión particularizada del plano crítico *(5.51)*.

$$\alpha_F \cdot \frac{\sigma_{ut}}{2} \cdot \left(1 + \frac{1}{\sqrt{1 + \left(\frac{0}{\alpha_F(\sigma_{ut} + 0)}\right)^2}}\right) = \beta_F \tag{9.38}$$

$$\alpha_F \cdot \frac{\sigma_{ut}}{2} \cdot \left(1 + \frac{1}{1}\right) = \beta_F \;\; \rightarrow \;\; \beta_F = \alpha_F \cdot \sigma_{ut} \tag{9.39}$$

Se resuelve por sustitución el sistema de ecuaciones formado por (5.58) y *(9.39)*.

$$\frac{\sigma_{-1}}{2} \cdot \left(\sqrt{{\alpha_F}^2 + 1} + \alpha_F\right) = \alpha_F \cdot \sigma_{ut} \tag{9.40}$$

$$\sigma_{-1} \cdot \sqrt{{\alpha_F}^2 + 1} = \alpha_F \cdot (2\sigma_{ut} - \sigma_{-1}) \tag{9.41}$$

Elevando al cuadrado a ambos lados de la ecuación *(9.41)*.

$${\alpha_F}^2 \cdot ({\sigma_{-1}}^2 - (2\sigma_{ut} - \sigma_{-1})^2) + {\sigma_{-1}}^2 = 0 \tag{9.42}$$

Desarrollando como sigue, se despeja α_F.

$$\alpha_F = \sqrt{\frac{-{\sigma_{-1}}^2}{{\sigma_{-1}}^2 - (2\sigma_{ut} - \sigma_{-1})^2}} \rightarrow \alpha_F = \sqrt{\frac{{\sigma_{-1}}^2}{4\sigma_{ut} \cdot (\sigma_{ut} - \sigma_{-1})}} \tag{9.43}$$

$$\alpha_F = \frac{\sigma_{-1}}{2} \cdot \frac{1}{\sqrt{\sigma_{ut} \cdot (\sigma_{ut} - \sigma_{-1})}} \tag{9.44}$$

Introduciendo α_F en la expresión (5.58) o (9.39) (en este caso, en la (9.39)(9.31)) se despeja β_F.

$$\beta_F = \frac{\sigma_{-1}}{2} \cdot \frac{\sigma_{ut}}{\sqrt{\sigma_{ut} \cdot (\sigma_{ut} - \sigma_{-1})}} \tag{9.45}$$

Una vez finalizada la obtención de los parámetros del método de Findley para las tres parejas de ensayos que constituyen el alcance del presente trabajo, sobre la Tabla 9.4 se recogen sus expresiones de forma resumida.

Tabla 9.4. Parámetros del método de Findley en función de la pareja de ensayos.

Pareja de ensayos	$\boldsymbol{\alpha_F}$	$\boldsymbol{\beta_F}$
$\sigma_{-1} - \tau_{-1}$	$\frac{2 - \frac{\sigma_{-1}}{\tau_{-1}}}{2 \cdot \sqrt{\frac{\sigma_{-1}}{\tau_{-1}} - 1}}$	$\frac{\sigma_{-1}}{2 \cdot \sqrt{\frac{\sigma_{-1}}{\tau_{-1}} - 1}}$
$\sigma_{-1} - \sigma_0$	Resolución numérica de la ecuación (9.33)	$\frac{\sigma_0}{2}\left(\sqrt{1 + 4{\alpha_F}^2} + 2\alpha_F\right)$
$\sigma_{-1} - \sigma_{ut}$	$\frac{\sigma_{-1}}{2} \cdot \frac{1}{\sqrt{\sigma_{ut} \cdot (\sigma_{ut} - \sigma_{-1})}}$	$\frac{\sigma_{-1}}{2} \cdot \frac{\sigma_{ut}}{\sqrt{\sigma_{ut} \cdot (\sigma_{ut} - \sigma_{-1})}}$

9.1.5 Método de Papuga PCr

Después obtener las constantes del método de Findley, en este punto se va a tratar sobre el método de Papuga PCr. En los métodos de plano crítico que se han expuesto previamente, se obtienen las constantes del método a partir de una pareja de ensayos con límite de fatiga conocido; sin embargo, en el método de Papuga PCr se emplea la pareja de ensayos $\sigma_{-1} - \tau_{-1}$ y la distinción se hace dependiendo del tipo de material, tal y como se ha explicado en el subapartado 5.2.3.3. Método de Papuga (PCr).

En este sentido, dado que el alcance del presente trabajo abarca los materiales dúctiles, se va a trabajar con las expresiones (5.72) y (5.73). Así pues, con el objetivo de recogerlas en una tabla resumen, al igual que se ha realizado para los métodos anteriores se presenta la Tabla 9.5.

Tabla 9.5. Parámetros del método de Papuga PCr en función de la pareja de ensayos.

Pareja de ensayos	$\boldsymbol{a_P}$	$\boldsymbol{b_P}$
$\sigma_{-1} - \tau_{-1}$ (Material dúctil)	$\left(\frac{4 \cdot \left(\frac{\sigma_{-1}}{\tau_{-1}}\right)^2}{4 + \left(\frac{\sigma_{-1}}{\tau_{-1}}\right)^2} \right)^2$	$\frac{8 \cdot \sigma_{-1} \cdot \left(\frac{\sigma_{-1}}{\tau_{-1}}\right)^2 \cdot \left(4 - \left(\frac{\sigma_{-1}}{\tau_{-1}}\right)^2\right)}{\left(4 + \left(\frac{\sigma_{-1}}{\tau_{-1}}\right)^2\right)^2}$

9.1.6 Método de Dang Van

El último de los métodos de plano crítico del que se van a obtener sus constantes es el método de Dang Van. Así pues, en este punto se va a complementar lo referente al método de Dang Van expuesto en el subapartado 5.2.3.4. Método de Dang Van (ensayos de flexión con tensión alterna y ensayo de torsión alterna) añadiendo los ensayos de tracción pulsante axial y tracción uniaxial estática.

Ensayo con tracción pulsante axial

Como se ha realizado en los ensayos de plano crítico precedentes, se comienza identificando el plano crítico. Para ello, se toma como base el caso genérico del estado tensional en el círculo de Mohr (ver Figura 9.3) y derivando la expresión que define el plano crítico (ec. *(5.74)*) se determina que $2\theta^*$ se encuentra a 90º tal y como se ha deducido en la ecuación *(5.78)*.

Tras identificar la posición del plano crítico, en base a ellos se definen los términos que intervienen en la función de daño del método de Dang Van.

$$\tau_a(t) = \frac{\sigma_0}{2} \cdot \sin(2\theta^*) = \frac{\sigma_0}{2} \sin(90) = \frac{\sigma_0}{2} \tag{9.46}$$

$$\sigma_h(t) = \frac{\sigma_0 + \sigma_0 + 0 + 0}{3} = \frac{2\sigma_0}{3} \tag{9.47}$$

Introduciendo los términos (9.46) y *(9.47)* en la función de daño de Dang Van (5.76).

$$max(\tau_a(t) + \alpha_{DV} \cdot \sigma_h(t)) = \beta_{DV} \;\rightarrow\; \frac{\sigma_0}{2} + \alpha_{DV} \cdot \frac{2\sigma_0}{3} = \beta_{DV} \tag{9.48}$$

Llegados a este punto, se dispone de dos ecuaciones independientes que relacionan las constantes α_{DV} y β_{DV} para la pareja de ensayos $\sigma_{-1} - \tau_{-1}$ (ec. (5.79) y *(9.48)*) por lo que se pueden despejar sus valores.

$$\frac{\sigma_{-1}}{2} + \alpha_{DV} \cdot \frac{\sigma_{-1}}{3} = \frac{\sigma_0}{2} + \alpha_{DV} \cdot \frac{2\sigma_0}{3} \tag{9.49}$$

$$\alpha_{DV} \cdot \left(\frac{\sigma_{-1}}{3} - \frac{2\sigma_0}{3}\right) = \frac{\sigma_0 - \sigma_{-1}}{2} \;\rightarrow\; \alpha_{DV} = \frac{3(\sigma_0 - \sigma_{-1})}{2(\sigma_{-1} - 2\sigma_0)} \tag{9.50}$$

Por último, se introduce α_{DV} en la ecuación (9.48) y se obtiene β_{DV}.

$$\beta_{DV} = \frac{\sigma_0}{2} + \frac{3(\sigma_0 - \sigma_{-1})}{2(\sigma_{-1} - 2\sigma_0)} \cdot \frac{2\sigma_0}{3} \;\rightarrow\; \beta_{DV} = \sigma_0 \left(\frac{-\sigma_{-1}}{2(\sigma_{-1} - 2\sigma_0)}\right) \tag{9.51}$$

Ensayo de tracción uniaxial estática

La segunda relación entre las constantes α_{DV} y β_{DV} para la pareja de ensayos $\sigma_{-1} - \sigma_{ut}$ se obtiene particularizando la función de daño de Dang Van al ensayo de tracción uniaxial estática. Así pues, se comienza identificando el plano crítico del mismo modo que se ha hecho con el ensayo de tracción pulsante axial, pero tomando como referencia el caso genérico del círculo de Mohr representado en la Figura 9.4.

Siguiendo un procedimiento análogo al del ensayo de tracción pulsante axial, se deduce que el plano crítico se encuentra en la posición $2\theta^*$ de 90º. Por lo tanto, conocido esto se pueden particularizar los términos que intervienen en la función de daño como se muestra a continuación.

$$\tau_a(t) = \frac{0}{2} \cdot \sin(90) = 0 \tag{9.52}$$

$$\sigma_h(t) = \frac{\sigma_{ut} + 0 + 0}{3} = \frac{\sigma_{ut}}{3} \tag{9.53}$$

Se sustituyen los términos de las expresiones (9.52) y (9.53) en la función de daño del método de Dang Van (ec. (5.76)).

$$0 + \alpha_{DV} \cdot \frac{\sigma_{ut}}{3} = \beta_{DV} \tag{9.54}$$

Una vez obtenida la segunda relación de los parámetros del método, combinando las expresiones (5.79) y *(9.54)* y empleando la técnica de sustitución, se obtiene

$$\frac{\sigma_{-1}}{2} + \alpha_{DV} \cdot \frac{\sigma_{-1}}{3} = \alpha_{DV} \cdot \frac{\sigma_{ut}}{3} \;\rightarrow\; \alpha_{DV} = \frac{3}{2}\left(\frac{-\sigma_{-1}}{\sigma_{-1} - \sigma_{ut}}\right) \tag{9.55}$$

Se introduce la expresión de α_{DV} en la ecuación *(9.54)* y se obtiene β_{DV}.

$$\beta_{DV} = \frac{3}{2}\left(\frac{-\sigma_{-1}}{\sigma_{-1} - \sigma_{ut}}\right) \cdot \frac{\sigma_{ut}}{3} \;\rightarrow\; \beta_{DV} = \frac{\sigma_{ut}}{2}\left(\frac{-\sigma_{-1}}{\sigma_{-1} - \sigma_{ut}}\right) \tag{9.56}$$

Para finalizar, del mismo modo que se ha realizado con los métodos anteriores, sobre la Tabla 9.6 se muestran de forma agrupada las expresiones de las constantes del método de Dang Van en función de la pareja de ensayos empleada.

Tabla 9.6. Parámetros del método de Dang Van en función de la pareja de ensayos.

Pareja de ensayos	$\boldsymbol{\alpha_{DV}}$	$\boldsymbol{\beta_{DV}}$
$\sigma_{-1} - \tau_{-1}$	$3 \cdot \left(\frac{\tau_{-1}}{\sigma_{-1}} - \frac{1}{2}\right)$	τ_{-1}
$\sigma_{-1} - \sigma_0$	$\frac{3(\sigma_0 - \sigma_{-1})}{2(\sigma_{-1} - 2\sigma_0)}$	$\sigma_0\left(\frac{-\sigma_{-1}}{2(\sigma_{-1} - 2\sigma_0)}\right)$
$\sigma_{-1} - \sigma_{ut}$	$\frac{3}{2}\left(\frac{-\sigma_{-1}}{\sigma_{-1} - \sigma_{ut}}\right)$	$\frac{\sigma_{ut}}{2}\left(\frac{-\sigma_{-1}}{\sigma_{-1} - \sigma_{ut}}\right)$

9.2 Particularización de los métodos

Tras obtener los parámetros α y β de los métodos, en este segundo bloque del apartado 9. Cálculos se van a exponer las operaciones que se han empleado para particularizar los

métodos de fatiga multiaxial a los casos de fatiga uniaxial y de torsión pura. Así pues, se va a comenzar exponiendo lo referente a la particularización para el caso de fatiga uniaxial y a continuación, se hará lo propio para el caso de torsión pura.

9.2.1 Caso I: fatiga uniaxial

En este subapartado se van a particularizar los métodos de fatiga multiaxial que abarca el alcance del trabajo al caso concreto del estado tensional uniaxial. Se va a comenzar por los métodos de enfoque global y después, se continuará con los métodos de plano crítico pero, antes de ello, sobre la Figura 9.5 se presenta el círculo de Mohr del estado tensional uniaxial (ver Figura 1.8) ya que corresponde la base del procedimiento que se va a seguir.

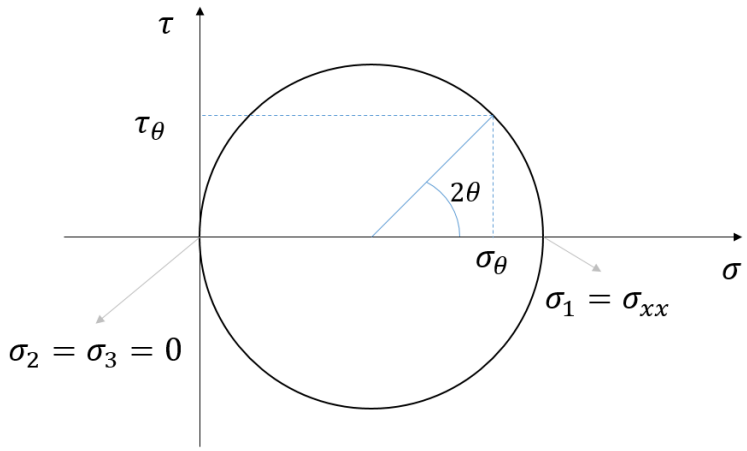

Figura 9.5. Círculo de Mohr genérico del estado tensional uniaxial.

9.2.1.1 Método de Sines

Para la particularización del método de Sines al caso de fatiga uniaxial, lo primero que se debe hacer es definir los términos de la función de daño (ec. *(5.5)*) tomando como referencia el estado tensional de la Figura 9.5.

$$\tau_{oct_a} = \frac{1}{3} \cdot \sqrt{\left(\sigma_{xx_a} - 0\right)^2 + (0)^2 + \left(-\sigma_{xx_a}\right)^2} = \frac{\sqrt{2}}{3}\sigma_{xx_a} \tag{9.57}$$

$$\sigma_{h_m} = \frac{1}{3} \cdot \left(\sigma_{xx_m} + 0 + 0\right) = \frac{1}{3}\sigma_{xx_m} \tag{9.58}$$

Tras esto, se sustituye sobre la ecuación (5.5) y la expresión que se obtiene (ec. (9.59)) corresponde a la particularización del método de Sines para el caso de fatiga uniaxial. En ella, las constantes del método son conocidas (obtenidas en el subapartado 9.1.1. Método

de Sines) por lo que, se trata de una función de dos variables que se puede representar sobre las coordenadas $\sigma_m - \sigma_a$.

$$\frac{\sqrt{2}}{3}\sigma_{xx_a} + \alpha_S \cdot \frac{1}{3}\sigma_{xx_m} - \beta_S = 0 \tag{9.59}$$

Para su resolución, se aplicará lo explicado en el subapartado 5.2.4. Métodos de resolución de ecuaciones. Esto es, conocido el valor de σ_{xx_m} se comenzará aplicando el método de bisección para buscar un intervalo donde se encuentre σ_{xx_a} y tomando el valor promedio como aproximación inicial, se obtendrá una solución más cercana a la exacta mediante Newton-Raphson. Cabe decir que, en este caso, al ser una función lineal se podría resolver sin aplicar métodos numéricos, pero, dado que no aporta desventajas significativas y, por analogía con otros métodos, se resuelve de esta forma.

9.2.1.2 Método de Crossland

La particularización del método de Crossland al caso de fatiga uniaxial se realiza de forma semejante a la del método de Sines. Por lo tanto, lo primero que se hace es particularizar los términos de la función de daño de Crossland a la situación mostrada sobre la Figura 9.5. De hecho, la τ_{oct_a} coincide con la expresión (9.57) y el único término a presentar es el de la $\sigma_{h_{max}}$.

$$\sigma_{h_{max}} = \frac{\sigma_{xx_m}}{3} + \frac{\sigma_{xx_a}}{3} \tag{9.60}$$

Introduciendo las expresiones (9.57) y *(9.60)* sobre la función de daño de Crossland (ec. (5.29)), se obtiene la particularización del método al caso de fatiga uniaxial (ec. (9.61)). En ella, las constantes del método vienen dadas según lo explicado en el subapartado 9.1.2. Método de Crossland y, en cuanto a su resolución, se trata de un caso similar al de Sines y por tanto, se lleva a cabo de manera análoga.

$$\frac{\sqrt{2}}{3}\sigma_{xx_a} + \alpha_C \cdot \left(\frac{\sigma_{xx_m} + \sigma_{xx_a}}{3}\right) - \beta_C = 0 \tag{9.61}$$

9.2.1.3 Método de Matake

A diferencia de los métodos de enfoque global tratados previamente, el procedimiento para particularizar los métodos de plano crítico, como Matake, al caso de fatiga uniaxial es más extenso. Esto se debe en gran medida a la necesidad de identificar el plano crítico tal y como se podrá apreciar en adelante.

El primer paso para particularizar la función de daño de Matake al caso uniaxial, trata de obtener dónde se encuentra el plano crítico en el círculo de Mohr de la Figura 9.5. Para ello, en base a la ecuación que define el plano crítico en el método de Matake (ec. (5.41)) se deduce que se encuentra a 90º respecto de la horizontal y, partiendo desde el centro del círculo (situación análoga a la de la Figura 5.11). Conocido esto, se determinan los términos de la función de daño como sigue.

$$\tau_a = \frac{\sigma_{xx_a}}{2} \tag{9.62}$$

$$\sigma_m = \frac{\sigma_{xx_m}}{2} \tag{9.63}$$

$$\sigma_a = \frac{\sigma_{xx_a}}{2} \tag{9.64}$$

Introduciendo estos términos sobre la función de daño de Matake (ec. (5.42)) se llega a la expresión de Matake particularizada para el caso de fatiga con un estado tensional uniaxial. Por último, decir que su resolución se realiza de una manera similar a la del método de Sines que se ha expuesto en el subapartado 9.2.1.1. Método de Sines.

$$\frac{\sigma_{xx_a}}{2} + \alpha_M \cdot \left(\frac{\sigma_{xx_m}}{2} + \frac{\sigma_{xx_a}}{2}\right) - \beta_M = 0 \tag{9.65}$$

9.2.1.4 Método de Findley

Según el método de Findley, las expresiones que definen el plano crítico y la función de daño son la (5.46) y la (5.47), respectivamente. Al igual que se ha procedido con la particularización del método de Matake, lo primero es definir la posición del plano crítico para el estado tensional representado en el círculo de Mohr de la Figura 9.5. Para ello, se sigue un procedimiento análogo al explicado en el subapartado 5.2.3.2. Método de Findley para el caso del ensayo de flexión con tensión alterna, llegando a la siguiente expresión.

$$2\theta^* = arctg\left(\frac{1}{\alpha_F} \cdot \frac{\sigma_{xx_a}}{\left(\sigma_{xx_m} + \sigma_{xx_a}\right)}\right) \;\rightarrow\; (Plano\ crítico) \tag{9.66}$$

Una vez conocida la posición del plano crítico ($2\theta^*$), se pueden particularizar los términos que aparecen en la función de daño de Findley (ec. (5.47)) del siguiente modo.

$$\tau_a = \frac{\sigma_{xx_a}}{2} \cdot \sin(2\theta^*) = \frac{\sigma_{xx_a}}{2} \cdot \sin\left(arctg\left(\frac{1}{\alpha_F} \cdot \frac{\sigma_{xx_a}}{\left(\sigma_{xx_m} + \sigma_{xx_a}\right)}\right)\right) \tag{9.67}$$

Aplicando la expresión (5.55).

$$\tau_a = \frac{\sigma_{xx_a}}{2} \cdot \frac{\frac{1}{\alpha_F} \cdot \frac{\sigma_{xx_a}}{\left(\sigma_{xx_m} + \sigma_{xx_a}\right)}}{\sqrt{1 + \left(\frac{1}{\alpha_F} \cdot \frac{\sigma_{xx_a}}{\left(\sigma_{xx_m} + \sigma_{xx_a}\right)}\right)^2}} \tag{9.68}$$

Por su parte, los términos referentes a la tensión normal quedan como sigue.

$$\sigma_m = \frac{\sigma_{xx_m}}{2}(1 + \cos(2\theta^*)) = \frac{\sigma_{xx_m}}{2}\left(1 + \cos\left(arctg\left(\frac{1}{\alpha_F}\frac{\sigma_{xx_a}}{\left(\sigma_{xx_m} + \sigma_{xx_a}\right)}\right)\right)\right) \tag{9.69}$$

Aplicando la expresión (5.56).

$$\sigma_m = \frac{\sigma_{xx_m}}{2}\left(1 + \frac{1}{\sqrt{1 + \left(\frac{1}{\alpha_F} \cdot \frac{\sigma_{xx_a}}{\left(\sigma_{xx_m} + \sigma_{xx_a}\right)}\right)^2}}\right) \tag{9.70}$$

Análogamente para el término de σ_a.

$$\sigma_a = \frac{\sigma_{xx_a}}{2}\left(1 + \frac{1}{\sqrt{1 + \left(\frac{1}{\alpha_F} \cdot \frac{\sigma_{xx_a}}{\left(\sigma_{xx_m} + \sigma_{xx_a}\right)}\right)^2}}\right) \tag{9.71}$$

Se introducen las expresiones (9.68), (9.70) y (9.71) en la función de daño de Findley (ec. (5.47)) y se desarrolla.

$$\tau_{a_{eq}} = \frac{\sigma_{xx_a}}{2} \frac{\frac{1}{\alpha_F} \cdot \frac{\sigma_{xx_a}}{(\sigma_{xx_m} + \sigma_{xx_a})}}{\sqrt{1 + \left(\frac{1}{\alpha_F} \cdot \frac{\sigma_{xx_a}}{(\sigma_{xx_m} + \sigma_{xx_a})}\right)^2}} \cdots$$

$$\cdots + \alpha_F \left(\frac{\sigma_{xx_m} + \sigma_{xx_a}}{2}\right) \left(1 + \frac{1}{\sqrt{1 + \left(\frac{1}{\alpha_F} \cdot \frac{\sigma_{xx_a}}{(\sigma_{xx_m} + \sigma_{xx_a})}\right)^2}}\right) \geq \beta_F \quad \rightarrow \quad (Fallo) \tag{9.72}$$

Se elimina la raíz de los términos del denominador, multiplicando y dividiendo por dicho término.

$$\tau_{a_{eq}} = \frac{\sigma_{xx_a} \sqrt{{\alpha_F}^2 \left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2 + {\sigma_{xx_a}}^2} \left(\sigma_{xx_a} {\alpha_F}^2 \left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2\right)}{2\sqrt{{\alpha_F}^2 \left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2} \left(\alpha_F \left(\sigma_{xx_m} + \sigma_{xx_a}\right)\right) \left({\alpha_F}^2 \left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2 + {\sigma_{xx_a}}^2\right)} \cdots$$

$$\cdots + \alpha_F \left(\frac{\sigma_{xx_m} + \sigma_{xx_a}}{2}\right) \left(1 + \frac{\sqrt{{\alpha_F}^2 \left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2 + {\sigma_{xx_a}}^2} \left({\alpha_F}^2 \left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2\right)}{\sqrt{{\alpha_F}^2 \left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2} \left({\alpha_F}^2 \left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2 + {\sigma_{xx_a}}^2\right)}\right) \geq \beta_F \tag{9.73}$$

Simplificando.

$$\tau_{a_{eq}} = \frac{{\sigma_{xx_a}}^2 \sqrt{{\alpha_F}^2 \left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2 + {\sigma_{xx_a}}^2}}{2\left({\alpha_F}^2 \left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2 + {\sigma_{xx_a}}^2\right)} + \alpha_F \left(\frac{\sigma_{xx_m} + \sigma_{xx_a}}{2}\right) \cdot \quad \cdots$$

$$\cdots \cdot \left(1 + \frac{\sqrt{{\alpha_F}^2 \left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2 + {\sigma_{xx_a}}^2} \left(\alpha_F \left(\sigma_{xx_m} + \sigma_{xx_a}\right)\right)}{\left({\alpha_F}^2 \left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2 + {\sigma_{xx_a}}^2\right)}\right) - \beta_F = 0 \tag{9.74}$$

Desarrollando la expresión y sacando el término de la raíz como factor común.

$$\tau_{a_{eq}} = \sqrt{{\alpha_F}^2\left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2 + {\sigma_{xx_a}}^2} \cdot \left(\frac{{\sigma_{xx_a}}^2}{2\left({\alpha_F}^2\left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2 + {\sigma_{xx_a}}^2\right)} \cdots \right.$$
$$\left. \cdots + \frac{{\alpha_F}^2\left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2}{2\left({\alpha_F}^2\left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2 + {\sigma_{xx_a}}^2\right)} \right) + \alpha_F\left(\frac{\sigma_{xx_m} + \sigma_{xx_a}}{2}\right) - \beta_F = 0 \quad (9.75)$$

Simplificando y reordenando la expresión *(9.75)*.

$$\sqrt{{\alpha_F}^2\left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2 + {\sigma_{xx_a}}^2} \cdot \frac{1}{2} = \beta_F - \alpha_F\left(\frac{\sigma_{xx_m} + \sigma_{xx_a}}{2}\right) \quad (9.76)$$

Multiplicando por 2 a ambos lados de la ecuación *(9.76)* y elevando al cuadrado.

$${\alpha_F}^2\left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2 + {\sigma_{xx_a}}^2 = \left(2\beta_F - \alpha_F\left(\sigma_{xx_m} + \sigma_{xx_a}\right)\right)^2 \quad (9.77)$$

Desarrollando las potencias de la expresión (9.77).

$${\alpha_F}^2\left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2 + {\sigma_{xx_a}}^2 = 4{\beta_F}^2 + {\alpha_F}^2\left(\sigma_{xx_m} + \sigma_{xx_a}\right)^2 - 4\beta_F\alpha_F\left(\sigma_{xx_m} + \sigma_{xx_a}\right) \quad (9.78)$$

Por último, simplificando y reordenando los términos de (9.78), se llega a la ecuación que particulariza el método de Findley para el caso uniaxial. En este caso, a diferencia de los anteriores, se tiene una ecuación no lineal (en σ_{xx_a}) por lo que, para su resolución se debe aplicar lo explicado en el subapartado 5.2.4. Métodos de resolución de ecuaciones.

$${\sigma_{xx_a}}^2 + 4\beta_F\alpha_F\left(\sigma_{xx_m} + \sigma_{xx_a}\right) - 4{\beta_F}^2 = 0 \quad (9.79)$$

9.2.1.5 Método de Papuga PCr

La particularización del método de Papuga PCr al caso uniaxial se plantea de una forma diferente a la empleada hasta ahora en el resto de métodos de plano crítico. Esto se debe a que, al introducir la posición del plano crítico obtenida analíticamente en la función de daño, se llega a una ecuación no lineal de orden cuatro; lo cual, no es viable resolverlo

como se ha venido haciendo hasta ahora por la variedad de resultados que se pueden obtener y el excesivo coste que acarrea.

Como alternativa a la resolución analítica que se ha empleado en los métodos anteriores, se podría trabajar de dos formas diferentes. La primera ("Forma general") corresponde al método general de aplicación de los métodos de plano crítico mientras que, la segunda ("Forma alternativa") es una variante que se ha deducido a partir de los conceptos generales y que ofrece la ventaja de disminuir de manera notable el coste computacional de los cálculos. A continuación, se van a presentar ambos procedimientos.

Forma general

En principio, corresponde con el procedimiento general que se emplea para aplicar los métodos de plano crítico. Según este, se debe llevar a cabo una doble iteración entre la ecuación que define el plano crítico (ec. (5.68)) y la que define la función de daño (ec. (5.69)) tal y como se va a explicar a continuación.

Para una determinada tensión media, se comienza asignando un valor a la tensión alterna (por ejemplo, se parte del valor de σ_{-1}). A partir de esta pareja de valores, se puede identificar la posición del plano crítico ($2\theta^*$) operando con la derivada de la ecuación que lo define (ec. (5.68)).

$$\frac{d\left(\sqrt{a_P\left(\frac{\sigma_{xx_a}}{2}\sin(2\theta)\right)^2 + b_P\left(\frac{\sigma_{xx_a}}{2}(1+\cos(2\theta)) + \frac{\tau_{-1}}{2\sigma_0}\frac{\sigma_{xx_m}}{2}(1+\cos(2\theta))\right)}\right)}{d\theta} = 0 \quad (9.80)$$

Derivando según la regla de la cadena.

$$\frac{1}{2}\frac{\left(a_P\left(\frac{\sigma_{xx_a}}{2}\sin(2\theta)\right)^2 + b_P\left(\frac{\sigma_{xx_a}}{2}(1+\cos(2\theta)) + \frac{\tau_{-1}}{2\sigma_0}\frac{\sigma_{xx_m}}{2}(1+\cos(2\theta))\right)\right)'}{\sqrt{a_P\left(\frac{\sigma_{xx_a}}{2}\sin(2\theta)\right)^2 + b_P\left(\frac{\sigma_{xx_a}}{2}(1+\cos(2\theta)) + \frac{\tau_{-1}}{2\sigma_0}\frac{\sigma_{xx_m}}{2}(1+\cos(2\theta))\right)}} = 0 \quad (9.81)$$

$$a_P\frac{{\sigma_{xx_a}}^2}{4}2\sin(2\theta^*)\,2\cos(2\theta^*) + b_P\left(\frac{\sigma_{xx_a}}{2}(-2\sin(2\theta^*))\cdots\right.$$
$$\left.\cdots + \frac{\tau_{-1}}{2\sigma_0}\frac{\sigma_{xx_m}}{2}(-2\sin(2\theta^*))\right) = 0 \quad (9.82)$$

Dividiendo entre el término "$\sin(2\theta^*)$" y simplificando.

$$a_P {\sigma_{xx}}_a{}^2 \cos(2\theta^*) + b_P\left(-{\sigma_{xx}}_a - \frac{\tau_{-1}\sigma_{xx_m}}{2\sigma_0}\right) = 0 \tag{9.83}$$

Despejando la posición del plano crítico ($2\theta^*$).

$$2\theta^* = arccos\left(\frac{b_P}{a_P {\sigma_{xx}}_a{}^2}\left({\sigma_{xx}}_a + \frac{\tau_{-1}\sigma_{xx_m}}{2\sigma_0}\right)\right) \tag{9.84}$$

Por lo tanto, introduciendo en la expresión (9.84) los valores iniciales de σ_{xx_m} y de σ_{xx_a}, se llega a una solución como la mostrada en la Figura 9.6 a).

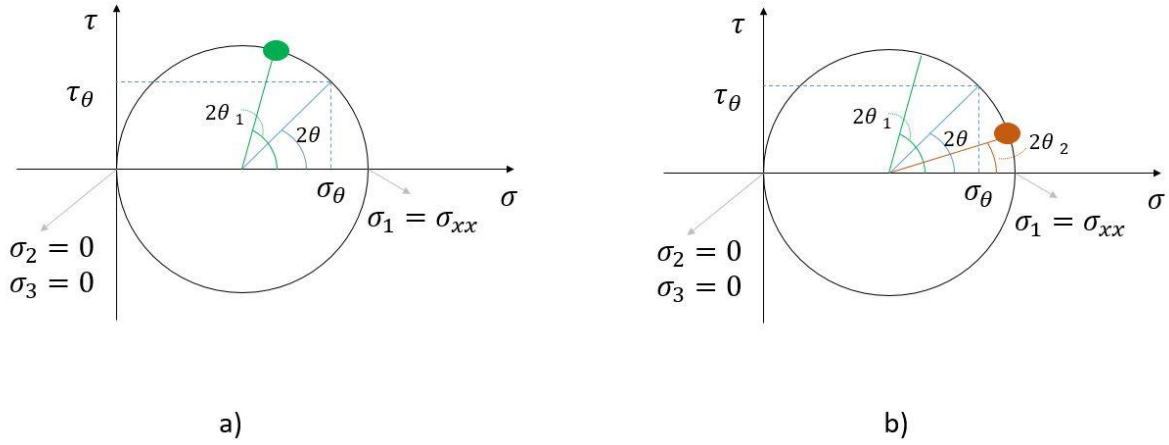

Figura 9.6. Posiciones del plano crítico en: a) iteración 1, b) iteración 2.

Tras obtener la posición del plano crítico, se está en disposición de calcular la tensión alterna que da lugar al fallo según Papuga. Para ello, se presenta la función de daño Papuga (ec. (5.69)) particularizada al caso uniaxial (ec. *(9.85)*).

$$a_P\left(\frac{\sigma_{xx_a}}{2}\sin(2\theta^*)\right)^2 + b_P\left(\frac{\sigma_{xx_a}}{2}(1+\cos(2\theta^*)) + \frac{\tau_{-1}}{2\sigma_0}\frac{\sigma_{xx_m}}{2}(1+\cos(2\theta^*))\right) - {\sigma_{-1}}^2 = 0 \tag{9.85}$$

Si al despejar la σ_{xx_a} su valor corresponde al de la iteración inicial, finaliza el proceso; de lo contrario, se comienza un nuevo ciclo iterativo calculando la nueva posición del plano crítico. Por último, cabe decir que, al tratarse de un proceso iterativo, se deben definir los criterios de parada que se van a emplear. En este sentido, comúnmente se suelen utilizar una tolerancia que represente el error máximo permitido entre dos iteraciones sucesivas y un número máximo de iteraciones.

Forma alternativa

Como se ha podido intuir, la resolución del método de Papuga PCr a partir de la "Forma general" resulta un proceso extenso que, además acarrea un coste computacional considerable. Así pues, como alternativa para su resolución se va a trabajar con la denominada "Forma alternativa" que se plantea a continuación.

En este caso, al igual que se ha realizado en la "Forma general" se va a hacer un barrido de planos, pero con un enfoque diferente. Para ello, se parte de la función de daño particularizada del método (ec.*(9.85)*); donde todos los términos son constantes y conocidos (características del propio material) excepto los que se citan posteriormente.

- σ_{xx_m}: Se define inicialmente para cada posición a lo largo del eje de abscisas.
- 2θ: Se trata de un término que va a tomar tantos valores discretos como fracciones en las que se distribuya el barrido de planos.
- σ_{xx_a}: Corresponde a la incógnita que se desea resolver para cada posible plano crítico, 2θ.

Por lo tanto, se constituyen dos vectores del mismo tamaño "n": uno para los valores discretos de 2θ y el otro para recoger las soluciones de la σ_{xx_a} asociada. Así pues, el plano crítico se definirá como aquel en el que se necesite una tensión alterna más pequeña para producir el fallo del material. Para poder comprender este procedimiento de una forma más visual, se muestra la Figura 9.7.

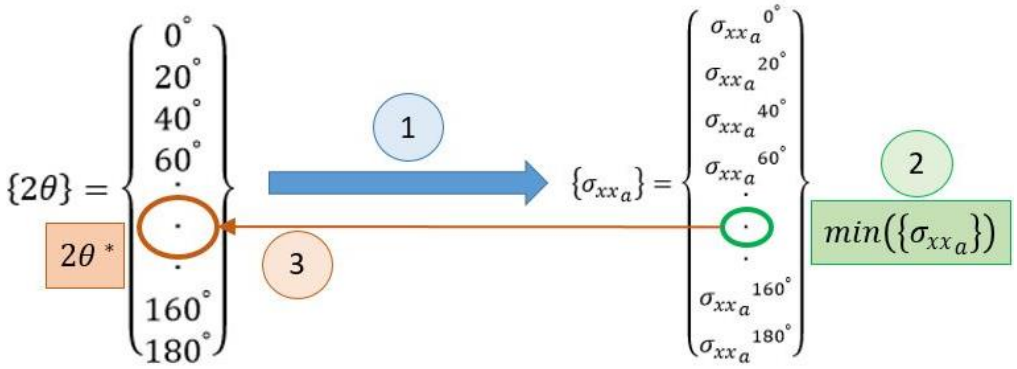

Figura 9.7. Esquema del procedimiento de resolución del método de Papuga PCr mediante la "Forma alternativa".

Junto con esto, se deben mencionar dos aspectos. Por un lado, como consecuencia de la simetría del círculo de Mohr respecto a la horizontal, al definir el vector de 2θ basta con hacer el barrido de planos desde 0º hasta 180º; ya que desde los 180º hasta los 360º corresponden a una situación idéntica en términos de fatiga. Un ejemplo de esto es cuando

2θ está en la posición de 340º que, desde el punto de vista del daño a fatiga, se trata del mismo caso que si está a 20º.

Por otro lado, a la vista de la Figura 9.7, se ha discretizado el vector de los planos candidatos a plano crítico con un incremento de ángulo de 20º. Si se hubiese tomado un valor mayor, disminuiría el coste ya que se realizaría un menor número de operaciones, pero muy probablemente, no se identificaría con suficiente calidad la posición del plano crítico y daría lugar a resultados poco precisos. Por su parte, si se redujese el incremento del ángulo, se conseguiría la posición del plano crítico con mayor nitidez, pero en contraposición, el coste de cálculo que acarrearía aumentaría de forma exponencial. Por lo tanto, para conseguir un compromiso entre la calidad de los resultados y su coste asociado, se ha optado por trabajar con un incremento de ángulo de 20º.

Para finalizar, sobre la Figura 9.8 se presenta de forma resumida otra posible perspectiva para entender la denominada "Forma alternativa" de resolución del método de Papuga PCr. Tal y como se puede apreciar, para un valor de la σ_{xx_m} conocido, se calculan las diferentes σ_{xx_a} asociadas a cada posible plano crítico (2θ); siendo el plano crítico aquel en el que se requiere la mínima σ_{xx_a} para que se produzca el fallo.

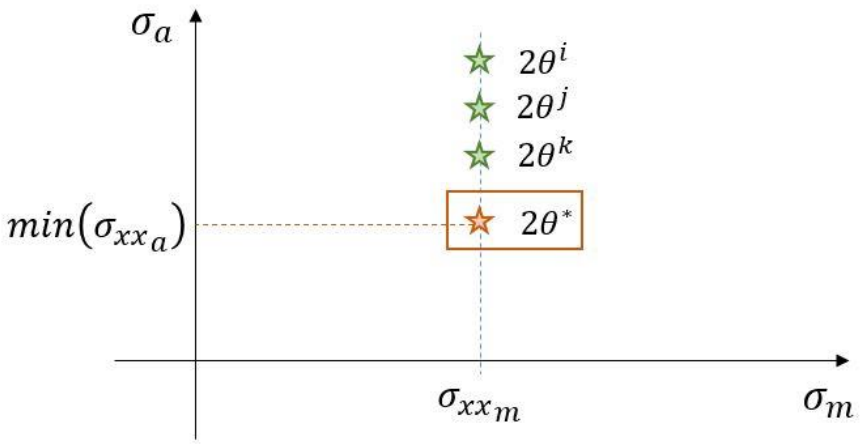

Figura 9.8. Representación simplificada en ejes cartesianos de la resolución del método de Papuga PCr mediante la "Forma alternativa"

9.2.1.6 Método de Dang Van

El último de los métodos de fatiga multiaxial que se va a particularizar al caso uniaxial es el método de Dang Van. Al igual que con otros métodos de plano crítico como por ejemplo Matake, lo primero que se debe hacer es identificar el plano crítico sobre el círculo de Mohr de la Figura 9.5. Esto se hace derivando la ecuación que define el plano crítico en el método de Dang Van (ec. (5.74)). Así, según lo explicado en el subapartado 5.2.3.4. Método de Dang Van, el plano crítico se encuentra a 90º desde el centro del círculo y en sentido antihorario (ver Figura 5.11).

Tras identificar dónde se encuentra el plano crítico, lo siguiente es particularizar los términos que intervienen en la función de daño de Dang Van; cuyas expresiones, en este caso, coinciden con las deducidas para el método de Matake (ec. (9.62), (9.63) y (9.64)). Finalmente, sustituyendo dichos términos sobre la propia función de daño (ec. (5.76)), se deduce la ecuación que particulariza el método de Dang Van al caso uniaxial (ec. (9.86)).

$$\frac{\sigma_{xx_a}}{2} + \alpha_{DV} \cdot \left(\frac{\sigma_{xx_m} + \sigma_{xx_a}}{3}\right) - \beta_{DV} = 0 \tag{9.86}$$

9.2.2 Caso II: torsión pura

Después de exponer la particularización de los métodos de fatiga multiaxial al caso uniaxial, en este subapartado se va a hacer lo propio para el caso de torsión pura. Al igual que en el caso anterior, se comenzará particularizando los métodos de enfoque global y tras esto, se continuará con los métodos de plano crítico. Para ello, se va a tomar como base el círculo de Mohr del estado tensional de torsión pura (ver Figura 1.17 b)) que se representa sobre la Figura 9.9.

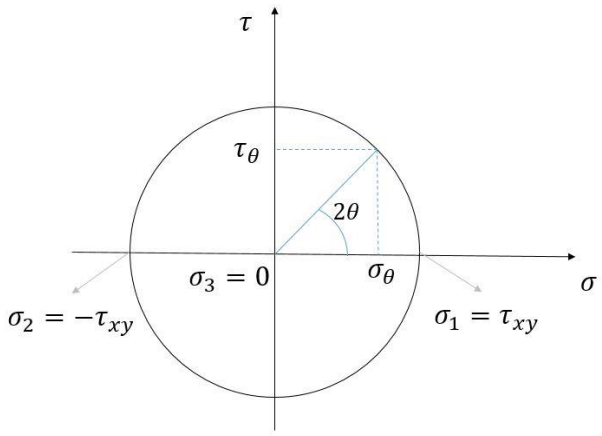

Figura 9.9. Círculo de Mohr genérico del estado tensional de torsión pura.

9.2.2.1 Método de Sines

El primer paso para particularizar el método de Sines al caso de torsión pura es particularizar los términos que constituyen la función de daño (ec. *(5.5)*) ajustándolos a la situación mostrada en la Figura 9.9.

$$\tau_{oct_a} = \frac{1}{3} \cdot \sqrt{\left(\tau_{xy_a} - \left(-\tau_{xy_a}\right)\right)^2 + \left(\tau_{xy_a}\right)^2 + \left(-\tau_{xy_a}\right)^2} = \frac{\sqrt{6}}{3}\tau_{xy_a} \tag{9.87}$$

$$\sigma_{h_m} = \frac{1}{3} \cdot \left(\tau_{xy_m} - \tau_{xy_m} + 0\right) = 0 \tag{9.88}$$

Introduciendo estos términos en las posiciones correspondientes de la ecuación (5.5), se llega a la expresión que particulariza el método de Sines para el caso de torsión pura.

$$\frac{\sqrt{6}}{3}\tau_{xy_a} + \alpha_S \cdot 0 - \beta_S = 0 \;\rightarrow\; \tau_{xy_a} - \frac{3}{\sqrt{6}}\beta_S = 0 \tag{9.89}$$

Tal y como se puede apreciar en la ecuación (9.89), la expresión que particulariza el método de Sines al caso de torsión pura es independiente de la tensión cortante media y únicamente depende del parámetro β_S.

9.2.2.2 Método de Crossland

El segundo de los métodos de enfoque global que se particulariza para torsión pura es el método de Crossland. Al igual que se ha realizado con el método de Sines, se comienza por determinar los términos que se incluyen en la función de daño (ec. (5.29)); donde la τ_{oct_a} coincide con la ecuación (9.87) y la $\sigma_{h_{max}}$ se anula de acuerdo a la expresión que se muestra a continuación.

$$\sigma_{h_{max}} = \frac{\tau_{xy_m} - \tau_{xy_m}}{3} + \frac{\tau_{xy_a} - \tau_{xy_a}}{3} = 0 \tag{9.90}$$

Por lo tanto, sustituyendo sobre la función de daño (ec. (5.29)) se obtiene la expresión (9.91) que particulariza el método de Crossland al caso de torsión pura.

$$\frac{\sqrt{6}}{3}\tau_{xy_a} + \alpha_C \cdot (0) - \beta_C = 0 \;\rightarrow\; \tau_{xy_a} - \frac{3}{\sqrt{6}}\beta_C = 0 \tag{9.91}$$

De la misma forma que ha ocurrido en el método de Sines, atendiendo a la expresión (9.91) no se considera la tensión cortante media y el único parámetro que influye es β_C.

9.2.2.3 Método de Matake

Una vez finalizada la particularización de los métodos de enfoque global, se va a hacer lo propio con los métodos de plano crítico; comenzando en este punto por el método de Matake. Para ello, del mismo modo que se ha procedido al particularizarlos al caso uniaxial, el primer paso es determinar la posición del plano crítico en el círculo de Mohr de la Figura 9.9. Así pues, según la expresión que define el plano crítico para el método de Matake (ec *(5.41)*) se deduce que éste se encentra en la intersección entre el círculo y la parte positiva del eje de ordenadas; dando lugar a las siguientes expresiones.

$$\tau_a = \tau_{xy_a} \tag{9.92}$$

$$\sigma_m = 0 \tag{9.93}$$

$$\sigma_a = 0 \tag{9.94}$$

Introduciendo las expresiones (9.92), (9.93) y (9.94) sobre la función de daño (ec. *(5.42)*) se llega a la ecuación (9.95) que, particulariza el método de Matake al caso de torsión pura.

$$\tau_{xy_a} + \alpha_M \cdot (0 + 0) - \beta_M = 0 \;\rightarrow\; \tau_{xy_a} - \beta_M = 0 \tag{9.95}$$

Según la ecuación (9.95), la tensión cortante alterna solamente varía en función del parámetro β_M; siendo independiente de la componente media de la tensión tangencial.

9.2.2.4 Método de Findley

De la misma forma que se ha realizado la particularización en el método de Matake, el primer paso para particularizar el método de Findley al caso de torsión pura es identificar el plano crítico en el círculo de Mohr de la Figura 9.5.Haciendo esto de acuerdo a lo explicado en el subapartado 5.2.3.2. Método de Findley para el ensayo de torsión alterna, se llega a la expresión que se indica a continuación.

$$2\theta^* = arctg\left(\frac{\tau_{xy_a}}{\alpha_F \cdot \left(\tau_{xy_m} + \tau_{xy_a}\right)}\right) \;\rightarrow\; (Plano\ crítico) \tag{9.96}$$

Tras conocer la expresión de $2\theta^*$, lo siguiente es particularizar los términos que constituyen la función de daño de Findley (ec. (5.47)).

$$\tau_a = \tau_{xy_a} \cdot \sin(2\theta^*) = \tau_{xy_a} \cdot \sin\left(arctg\left(\frac{\tau_{xy_a}}{\alpha_F \cdot \left(\tau_{xy_m} + \tau_{xy_a}\right)}\right)\right) \tag{9.97}$$

Empleando la relación *(5.55)*.

$$\tau_a = \tau_{xy_a} \cdot \frac{\frac{\tau_{xy_a}}{\alpha_F \cdot \left(\tau_{xy_m} + \tau_{xy_a}\right)}}{\sqrt{1 + \left(\frac{\tau_{xy_a}}{\alpha_F \cdot \left(\tau_{xy_m} + \tau_{xy_a}\right)}\right)^2}} \tag{9.98}$$

Por su parte, los términos correspondientes a la tensión normal son los siguientes.

$$\sigma_m = \tau_{xy_m} \cos(2\theta^*) = \tau_{xy_m} \cos\left(arctg\left(\frac{\tau_{xy_a}}{\alpha_F \cdot \left(\tau_{xy_m} + \tau_{xy_a}\right)}\right)\right) \tag{9.99}$$

Empleando la relación *(5.56)*.

$$\sigma_m = \tau_{xy_m} \left(\frac{1}{\sqrt{1 + \left(\frac{\tau_{xy_a}}{\alpha_F \cdot \left(\tau_{xy_m} + \tau_{xy_a}\right)}\right)^2}}\right) \tag{9.100}$$

Análogamente para σ_a.

$$\sigma_a = \tau_{xy_a}\left(\frac{1}{\sqrt{1+\left(\frac{\tau_{xy_a}}{\alpha_F\cdot\left(\tau_{xy_m}+\tau_{xy_a}\right)}\right)^2}}\right) \tag{9.101}$$

Tras esto, lo siguiente es introducir las expresiones (9.98), *(9.100)* y (9.101) en la función de daño de Findley (ec. (5.47)).

$$\tau_{a_{eq}} = \tau_{xy_a}\frac{\frac{\tau_{xy_a}}{\alpha_F\cdot\left(\tau_{xy_m}+\tau_{xy_a}\right)}}{\sqrt{1+\left(\frac{\tau_{xy_a}}{\alpha_F\cdot\left(\tau_{xy_m}+\tau_{xy_a}\right)}\right)^2}}\cdots$$
$$\cdots+\alpha_F\left(\tau_{xy_m}+\tau_{xy_a}\right)\left(\frac{1}{\sqrt{1+\left(\frac{\tau_{xy_a}}{\alpha_F\cdot\left(\tau_{xy_m}+\tau_{xy_a}\right)}\right)^2}}\right)\geq\beta_F\ \rightarrow\ (Fallo) \tag{9.102}$$

Se desarrolla el término que se encuentra dentro de la raíz y se reescribe la ecuación.

$$\tau_{xy_a}\cdot\frac{\frac{\tau_{xy_a}}{\alpha_F\left(\tau_{xy_m}+\tau_{xy_a}\right)}}{\sqrt{\frac{\alpha_F{}^2\left(\tau_{xy_m}+\tau_{xy_a}\right)^2+\tau_{xy_a}{}^2}{\alpha_F{}^2\left(\tau_{xy_m}+\tau_{xy_a}\right)^2}}}+\frac{\alpha_F\left(\tau_{xy_m}+\tau_{xy_a}\right)}{\sqrt{\frac{\alpha_F{}^2\left(\tau_{xy_m}+\tau_{xy_a}\right)^2+\tau_{xy_a}{}^2}{\alpha_F{}^2\left(\tau_{xy_m}+\tau_{xy_a}\right)^2}}}=\beta_F \tag{9.103}$$

Simplificando.

$$\tau_{xy_a} \cdot \frac{\tau_{xy_a}}{\sqrt{{\alpha_F}^2\left(\tau_{xy_m}+\tau_{xy_a}\right)^2+{\tau_{xy_a}}^2}}+\frac{{\alpha_F}^2\left(\tau_{xy_m}+\tau_{xy_a}\right)^2}{\sqrt{{\alpha_F}^2\left(\tau_{xy_m}+\tau_{xy_a}\right)^2+{\tau_{xy_a}}^2}}=\beta_F \qquad (9.104)$$

Se elimina la raíz del denominador multiplicando y dividiendo por dicho término y tras esto, se saca como factor común.

$$\frac{\sqrt{{\alpha_F}^2\left(\tau_{xy_m}+\tau_{xy_a}\right)^2+{\tau_{xy_a}}^2}}{{\alpha_F}^2\left(\tau_{xy_m}+\tau_{xy_a}\right)^2+{\tau_{xy_a}}^2}\left({\tau_{xy_a}}^2+{\alpha_F}^2\left(\tau_{xy_m}+\tau_{xy_a}\right)^2\right)=\beta_F \qquad (9.105)$$

Simplificando.

$$\sqrt{{\alpha_F}^2\left(\tau_{xy_m}+\tau_{xy_a}\right)^2+{\tau_{xy_a}}^2}=\beta_F \qquad (9.106)$$

Elevando al cuadrado a ambos lados de la expresión (9.106) y reordenando.

$${\tau_{xy_a}}^2({\alpha_F}^2+1)+\tau_{xy_a}\left(2\tau_{xy_m}{\alpha_F}^2\right)+{\tau_{xy_m}}^2{\alpha_F}^2-{\beta_F}^2=0 \qquad (9.107)$$

En la ecuación (9.107) se muestra la particularización del método de Findley para el estado tensional de torsión pura. Tal y como se puede apreciar, se trata de una ecuación no lineal en τ_{xy_a} por lo que su resolución se debe llevar a cabo mediante lo explicado en el subapartado 5.2.4. Métodos de resolución de ecuaciones. Básicamente, esto se puede resumir en que partiendo de un valor dado de la τ_{xy_m}, con el método de bisección se va a obtener una aproximación inicial de la τ_{xy_a} y tomándola como aproximación inicial, se va a buscar una solución que se aproxime más a la real mediante el método de Newton-Raphson.

9.2.2.5 Método de Papuga PCr

Para particularizar el método de Papuga PCr al caso de torsión pura, al igual que ocurría en el caso uniaxial, se emplea un procedimiento que difiere del que se ha venido siguiendo en los métodos anteriores. Éste corresponde a la extrapolación de la

denominada "Forma alternativa" expuesta en el subapartado 9.2.1.5. Método de Papuga PCr del caso uniaxial al de torsión pura.

Pese a que el procedimiento seguido en ambas particularizaciones del método de Papuga PCr sea idéntico, el estado tensional varía y por tanto, el plano crítico ($2\theta^*$) y la función de daño tienen otras expresiones. Sin embargo, para emplear la "forma alternativa", únicamente se necesita la función de daño particularizada al caso de torsión pura por lo que, se comienza obteniendo los términos que la definen y a continuación, se sustituyen en la propia ecuación.

$$\tau_a = \tau_{xy_a} \sin(2\theta) \tag{9.108}$$

$$\sigma_a = \tau_{xy_a} \cos(2\theta) \tag{9.109}$$

$$\sigma_m = \tau_{xy_m} \cos(2\theta) \tag{9.110}$$

Introduciendo las expresiones (9.108), (9.109) y *(9.110)* sobre la función de daño (ec. (5.69)), se define la ecuación que particulariza el método de Papuga PCr al caso de torsión pura.

$$a_P \left(\tau_{xy_a} \sin(2\theta^*)\right)^2 + b_P \left(\tau_{xy_a} \cos(2\theta^*) + \frac{\tau_{-1}}{2\sigma_0} \tau_{xy_m} \cos(2\theta^*)\right) - {\sigma_{-1}}^2 = 0 \tag{9.111}$$

Tal y como se ha mencionado previamente, tras obtener la función de daño del método de Papuga PCr particularizada al caso de torsión pura, el procedimiento que se sigue para su resolución es idéntico y totalmente extrapolable de lo explicado en el punto 9.2.1.5. Método de Papuga PCr ("Forma alternativa").

9.2.2.6 Método de Dang Van

El último método de plano crítico que se particulariza al caso de torsión pura es el método de Dang Van. Para ello, de la misma manera que se ha realizado en otros métodos de este tipo, se comienza identificando la posición del plano crítico sobre el círculo de Mohr representado en la Figura 9.9. Derivando la ecuación que define el plano crítico (ec. (5.74)) según lo explicado en el punto 5.2.3.4. Método de Dang Van, se deduce que éste se encuentra en la intersección entre la parte positiva del eje de ordenadas y el propio círculo.

Una vez conocida la posición del plano crítico, se determinan las expresiones de los términos que constituyen la función de daño de Dang Van. En este caso, pese a que la

expresión del método de Dang Van difiere de la de Matake, los términos que aparecen en ambas son los mismos por lo que, sus expresiones son comunes (ecuaciones (9.92), (9.93) y (9.94)). Por lo tanto, introduciéndolas en la función de daño (ec.(5.76)) se llega a la expresión que particulariza el método de Dang Van al caso de torsión pura (ec. (9.113)).

$$\tau_{xy_a} + \alpha_{DV} \cdot (0) - \beta_{DV} = 0 \tag{9.112}$$

$$\tau_{xy_a} - \beta_{DV} = 0 \tag{9.113}$$

10 Desarrollo de la aplicación

Una vez explicados los diferentes cálculos que se han realizado, se implementan en la herramienta App Designer de Matlab y se desarrolla la interfaz gráfica que se muestra a continuación. Cabe mencionar que este subapartado se va a exponer de manera genérica ya que, para conocer con mayor detalle el funcionamiento de cada comando que incorpora la aplicación se adjunta el *VII. ANEXO II: Manual de usuario*.

Sobre la Figura 10.1 se muestra la vista general de la interfaz que se ha desarrollado. Tal y como se puede apreciar, se ha diseñado en una única página que dispone de tres pestañas o "*tabs*" ya que, de esta forma el usuario va a ser capaz de tener localizado todo el contenido que abarca la app en un marco inmóvil.

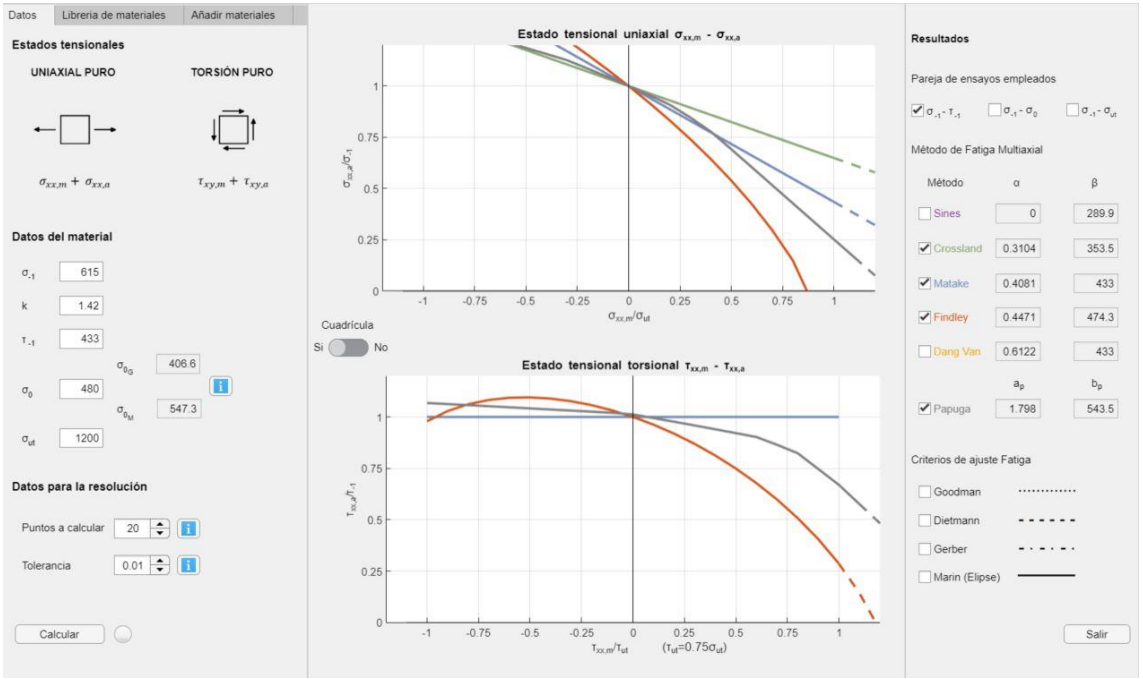

Figura 10.1. Vista de la aplicación (pestaña "Datos").

A la vista de la Figura 10.1, la aplicación dispone de tres zonas claramente diferenciables: izquierda ("Datos", "Librería de materiales" y "Añadir materiales"), central y derecha cuya finalidad se va a exponer a continuación. Para ello, se va a seguir un orden lógico de actuación; es decir, se va a comenzar por la pestaña "Datos" de la parte izquierda, se continúa con la parte central y derecha y, por último, se vuelve a las pestañas "librería de materiales y "Añadir materiales" de la parte izquierda.

En primer lugar, en la pestaña "Datos" de la parte izquierda de la interfaz, se deben introducir los datos del material con los que se desea realizar la particularización de los métodos de fatiga multiaxial a los casos uniaxial y torsión pura. Además de esto, en dicha zona se tiene la opción de variar los datos que se incluyen por defecto para la resolución de las ecuaciones que se han particularizado. Tras definir todos estos parámetros, se pulsa sobre el botón "Calcular" y el software realiza las operaciones pertinentes aplicando lo explicado en el apartado 9. Cálculos.

Una vez realizados los cálculos, en la parte central se dispone de dos gráficas correspondientes al estado tensional uniaxial y torsional respectivamente, donde se van a representar los resultados que el usuario elija de la parte derecha. Junto con esto, cabe mencionar que los ejes de ambas representaciones se encuentran normalizados; tal y como se puede apreciar en la Figura 10.1.

Por su parte, en la zona derecha de la aplicación se dispone de una serie de casillas de selección o "*checkboxes*" que permiten al usuario representar en la parte central solamente aquellos métodos que se deseen. Asimismo, se adjuntan los valores numéricos de los parámetros característicos de cada uno de los métodos de fatiga multiaxial; independientemente de si se encuentran seleccionados para representarlos o no.

Después de introducir los datos correspondientes al material con el que se desea trabajar y seleccionar los resultados a visualizar, se tiene la opción de representar sobre las mismas gráficas puntos de fallo obtenidos experimentalmente. Para ello, en la parte izquierda (al lado del *tab* "Datos") se dispone de las pestañas "Librería de materiales" y "Añadir materiales".

Si se selecciona el *tab* "Librería de materiales" (ver Figura 10.2), se dispone de una serie de materiales cuyos valores de fallo se han obtenido de bibliografía tal y como se explica en VI. ANEXO I. Materiales de bibliografía. En este caso, al igual que con los resultados de los métodos, se disponen de unos *checkboxes* que permiten seleccionar tantos materiales como se deseen. A modo de ejemplo, sobre la Figura 10.2 se puede apreciar cómo se han representado todos los materiales disponibles para el caso uniaxial mientras que, del caso de torsión pura no se ha mostrado ningún material.

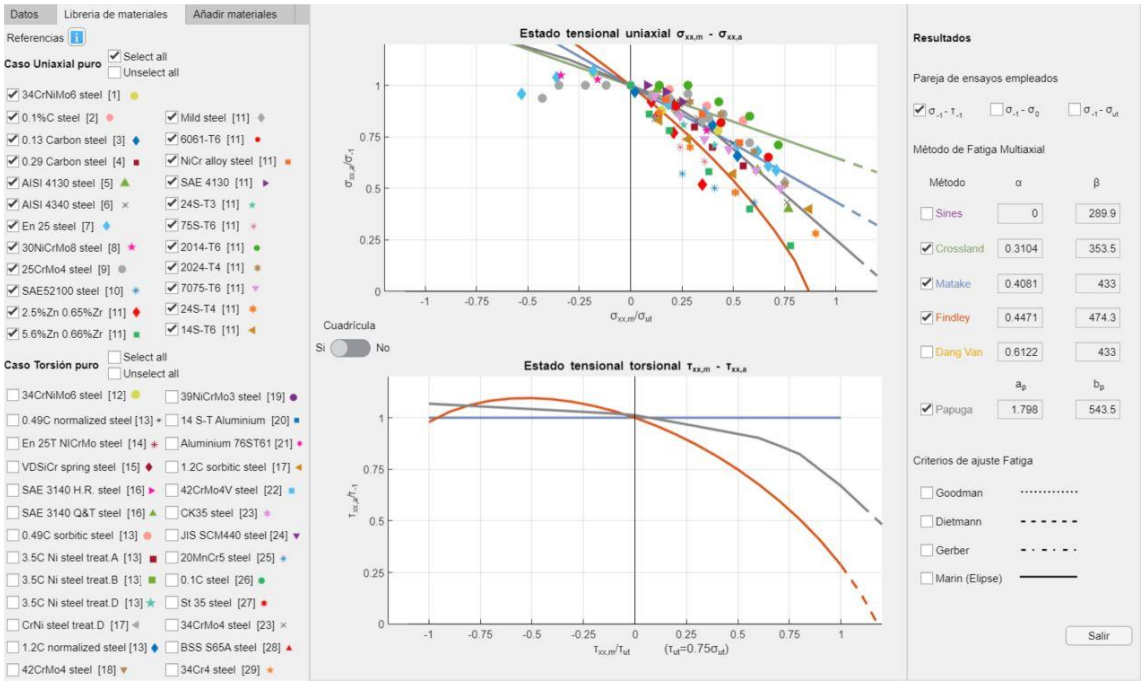

Figura 10.2. Vista de la aplicación (pestaña "Librería de materiales").

Por su parte, mediante la pestaña "Añadir materiales" (ver Figura 10.3) se tiene la opción de importar, desde una hoja Excel, materiales adicionales a los que se recogen en la librería; siendo su interfaz idéntica a la del *tab* "Librería de materiales". Así pues, pese a que se trata de pestañas independientes, se pueden considerar como complementarias ya que la finalidad de ambas es representar puntos de fallo obtenidos de ensayos experimentales sobre las gráficas centrales.

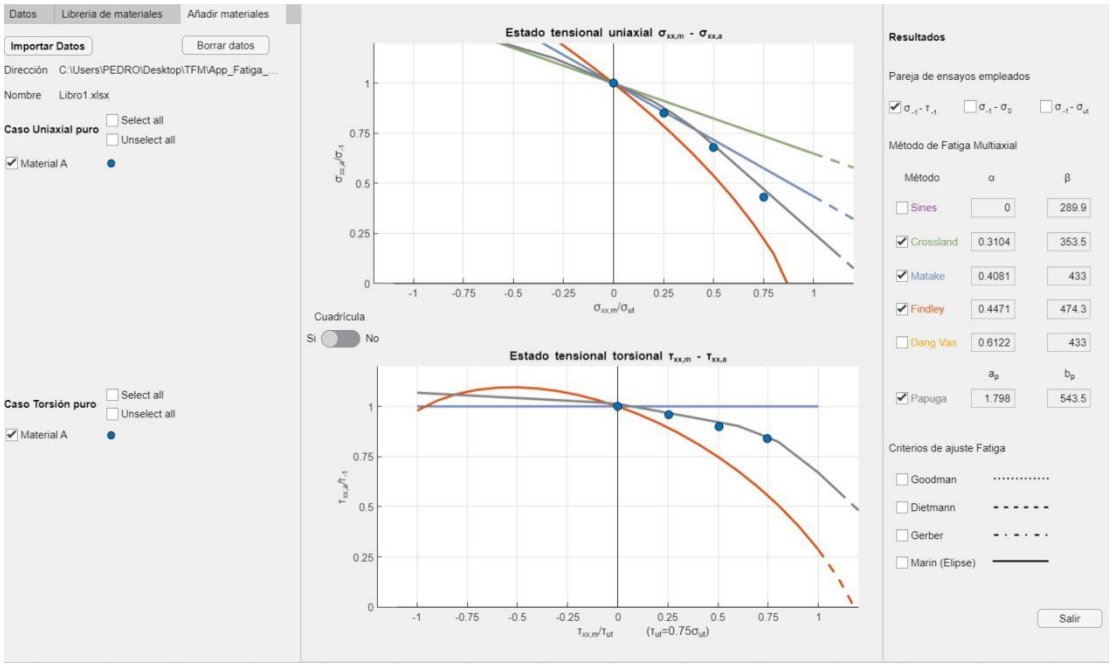

Figura 10.3. Vista de la aplicación (pestaña "Añadir materiales").

Para finalizar, unido a la pestaña de "Añadir materiales", cabe remarcar que la hoja Excel que se debe emplear en la importación de los datos debe concordar con el formato del "Libro1" que se adjunta con la propia aplicación (ver Figura 10.4). De lo contrario, la app no va a ser capaz de identificar los valores como corresponde y, por tanto, no se tendrá opción de adjuntar los datos de fallo de nuevos materiales.

Para llegar a los resultados de la Figura 10.3, la hoja Excel empleada corresponde a la de la Figura 10.4; donde se deben destacar dos aspectos. Por un lado, pese a que en ella solamente se muestre lo referente a los tres primeros materiales, el resto sigue el mismo formato y se determinan de forma análoga. Por otro lado, como se puede apreciar en la Figura 10.4 las características del material están definidas con valores unitarios; lo cual atiende a que estos datos se han definido a modo de ejemplo y no representan a ningún material real en concreto ya que, se han introducido con el único objetivo de visualizar la funcionalidad de la aplicación.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Número de materiales	1	NOTA: Máximo se pueden importar 22 materiales. De ahí en adelante, no se podrán añadir más materiales.									
2	Datos de cada material											
3	1	Material 1			2	Material 2			3	Material 3		
4	Nombre	Material A			Nombre	Material B			Nombre	Material C		
5	σ-1	1			σ-1				σ-1			
6	τ-1	1			τ-1				τ-1			
7	σ ut	1			σ ut				σ ut			
8	σ-m	σ-a	τ-m	τ-a	σ-m	σ-a	τ-m	τ-a	σ-m	σ-a	τ-m	τ-a
9	0	1	0	1								
10	0,25	0,85	0,19	0,96								
11	0,5	0,68	0,38	0,9								
12	0,75	0,43	0,56	0,84								
13												
14												

Figura 10.4. Ejemplo del formato Excel necesario para importar datos.

11 Descripción de los resultados

Con el desarrollo de las diferentes etapas que constituyen el presente trabajo, se pueden deducir los resultados que se van a exponer en el presente apartado. Para ello, se van a diferenciar tres grupos principales; donde los dos primeros tendrán un enfoque general y estarán relacionados con la obtención de los parámetros de los métodos y la particularización de sus funciones de daño, respectivamente. Por su parte, el tercer punto se va a destinar a estudiar, a modo de ejemplo, el caso concreto del acero 34CrNiMo6 y en base a los resultados obtenidos seleccionar el método de fatiga multiaxial más adecuado.

11.1 Parámetros de los métodos de fatiga multiaxial

De la obtención de las constantes de los métodos de fatiga multiaxial que se ha explicado en el subapartado 9.1. Parámetros de los métodos, se pueden destacar dos aspectos principalmente. Tal y como se va a exponer a continuación, el primero de ellos está relacionado con el método de Sines mientras que el segundo, se puede asociar al método de Findley.

Lo primero en cuanto a los parámetros del método de Sines es que, independientemente de la pareja de ensayos empleada para particularizar sus constantes, el valor de β_S siempre adquiere la misma expresión (ver Tabla 9.1). Esto se debe a que, en las tres parejas de ensayos que se recogen en el alcance del trabajo ($\sigma_{-1} - \tau_{-1}$, $\sigma_{-1} - \sigma_0$, $\sigma_{-1} - \sigma_{ut}$) se encuentra el ensayo de flexión alterna y al particularizar la función de daño de Sines a dicho caso, como la σ_{h_m} es cero, el término de α_S se anula y se llega a una expresión en la que β_S únicamente depende de σ_{-1}.

Junto con esto, otro aspecto a remarcar es la imposibilidad de determinar el parámetro α_S a partir de la pareja de ensayos de flexión con tensión alterna y de torsión alterna. De acuerdo a lo expuesto en el subapartado 5.2.2.1. Método de Sines, las expresiones que se obtienen al particularizar la función de daño a cada ensayo (ec. (5.13) y (5.16)) solamente involucran al término β_S y en principio, son diferentes. Sin embargo, atendiendo a la relación que se establece entre σ_{-1} y τ_{-1} al aplicar Von Mises al caso de torsión pura (ec. (5.17)) se puede asumir que ambas expresiones son semejantes.

Por último, respecto a la obtención de las constantes del método de Findley, decir que a diferencia del resto de métodos, en el caso concreto de la pareja de ensayos $\sigma_{-1} - \sigma_0$ la expresión que define al parámetro α_F es no lineal (ec. (9.33)). Esto implica que para su

resolución es necesario el empleo de un método de resolución numérico como el de Newton-Raphson.

11.2 Particularización de las funciones de daño

Tras analizar lo relativo a la determinación de los parámetros de los métodos de fatiga multiaxial, en este punto se van a recoger los principales resultados que se han obtenido al particularizar las funciones de daño. Para ello, se va a comenzar por lo referente a las expresiones obtenidas para el caso uniaxial puro y después, se hará lo propio para el caso de torsión pura.

A la vista de la Figura 11.1, lo primero que se puede deducir de las expresiones obtenidas al particularizar los métodos de Sines, Crossland, Matake y Dang Van al caso uniaxial (ecuaciones(9.59), (9.61), (9.65) y (9.86)), es que se trata de funciones de primer orden con pendiente negativa; es decir, rectas descendentes. Por su parte, las expresiones de Findley (ec. (9.79)) y de Papuga PCr (ec. (9.85)) dan lugar a funciones de orden dos y cuatro, respectivamente.

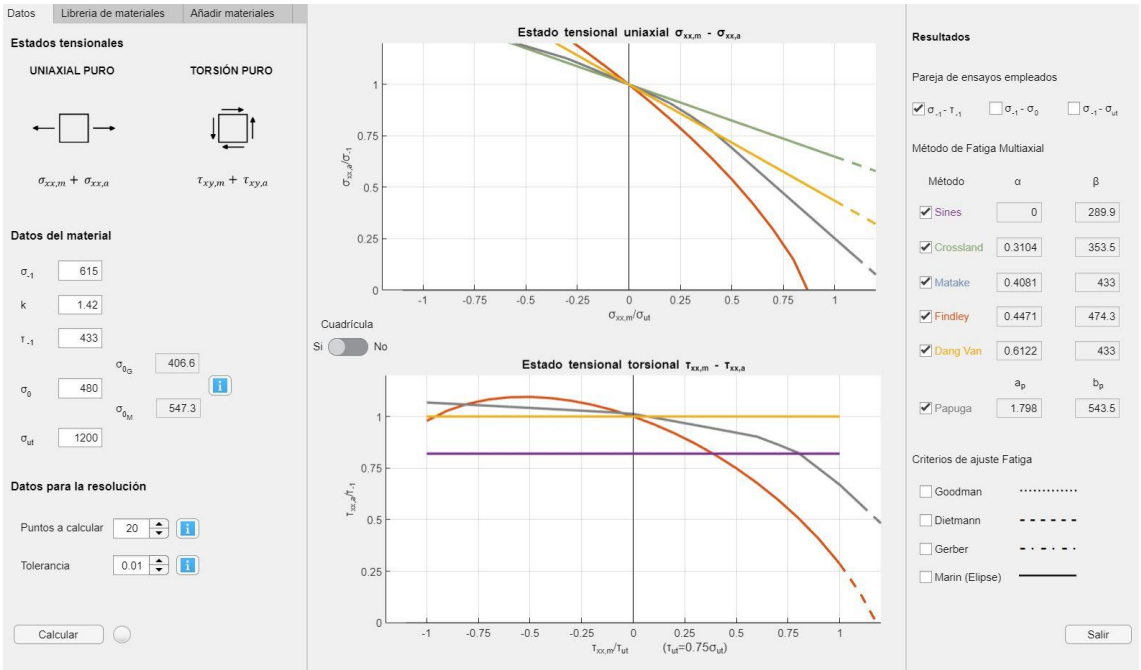

Figura 11.1. Resultados genéricos al particularizar las funciones de daño para la pareja de ensayos $\sigma_{-1} - \tau_{-1}$.

Asimismo, sobre la representación del caso uniaxial que se muestra en la Figura 11.1 también se aprecia como para el caso concreto de la pareja de ensayos $\sigma_{-1} - \tau_{-1}$ las funciones resultantes de particularizar los métodos de Matake y de Dang Van son coincidentes. Pese a que las expresiones que se han presentado para cada uno de los

métodos sean diferentes (ec. (9.65) y (9.86)), introduciendo en ellas los parámetros particulares de Matake (ver Tabla 9.3) y de Dang Van (ver Tabla 9.6), se demuestra matemáticamente que ambas son análogas.

$$\begin{cases} Matake\ (M) \rightarrow \dfrac{\sigma_{xx_a}}{2} + \left(2\dfrac{\tau_{-1}}{\sigma_{-1}} - 1\right)\left(\dfrac{\sigma_{xx_m}}{2} + \dfrac{\sigma_{xx_a}}{2}\right) - \tau_{-1} = 0 \\ Dang\ Van\ (DV) \rightarrow \dfrac{\sigma_{xx_a}}{2} + 3\left(\dfrac{\tau_{-1}}{\sigma_{-1}} - \dfrac{1}{2}\right)\left(\dfrac{\sigma_{xx_m} + \sigma_{xx_a}}{3}\right) - \tau_{-1} = 0 \end{cases} \tag{11.1}$$

Simplificando:

$$\begin{cases} (M) \rightarrow \dfrac{\sigma_{xx_a}}{2} + \left(\dfrac{\tau_{-1}}{\sigma_{-1}} - \dfrac{1}{2}\right)\left(\sigma_{xx_m} + \sigma_{xx_a}\right) - \tau_{-1} = 0 \\ (DV) \rightarrow \dfrac{\sigma_{xx_a}}{2} + \left(\dfrac{\tau_{-1}}{\sigma_{-1}} - \dfrac{1}{2}\right)\left(\sigma_{xx_m} + \sigma_{xx_a}\right) - \tau_{-1} = 0 \end{cases} \tag{11.2}$$

En esta misma línea, para la pareja de ensayos $\sigma_{-1} - \sigma_0$ se está ante una situación semejante con los métodos de Sines, Crossland, Matake y Dang Van. Es por ello que, al representarlos al mismo tiempo sobre la app, se superponen las funciones y parece que solamente se ha presentado un método (el último que se ha seleccionado) pero, en realidad, los restantes están debajo de éste (ver Figura 11.2).

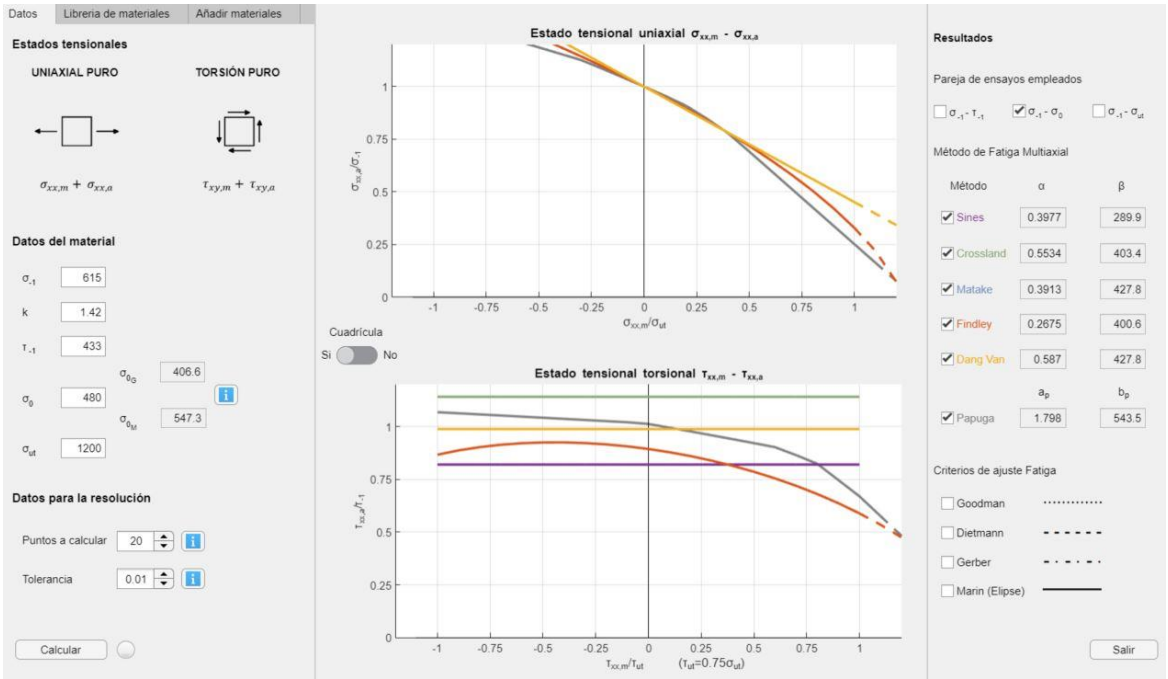

Figura 11.2. Resultados genéricos al particularizar las funciones de daño para la pareja de ensayos $\sigma_{-1} - \sigma_0$.

Al igual que se ha realizado para el caso de $\sigma_{-1} - \tau_{-1}$, matemáticamente se demuestra que sustituyendo en las expresiones de Sines, Crossland, Matake y Dang Van (ec. (9.59), (9.61), (9.65) y (9.86)) las constantes correspondientes a la pareja de ensayos $\sigma_{-1} - \sigma_0$, se llega a misma función.

$$\begin{cases} Sines\ (S) \\ \quad \frac{\sqrt{2}}{3}\sigma_{xx_a} + \frac{\sqrt{2}\cdot(\sigma_{-1}-\sigma_0)}{\sigma_0}\cdot\frac{1}{3}\sigma_{xx_m} - \frac{\sqrt{2}}{3}\cdot\sigma_{-1} = 0 \\ Crossland\ (C) \\ \quad \frac{\sqrt{2}}{3}\sigma_{xx_a} + \frac{\sqrt{2}\cdot(\sigma_0-\sigma_{-1})}{\sigma_{-1}-2\cdot\sigma_0}\left(\frac{\sigma_{xx_m}+\sigma_{xx_a}}{3}\right) - \frac{\sqrt{2}}{3}\sigma_{-1}\left(\frac{-\sigma_0}{\sigma_{-1}-2\cdot\sigma_0}\right) = 0 \\ Matake\ (M) \\ \quad \frac{\sigma_{xx_a}}{2} + \frac{\sigma_0-\sigma_{-1}}{\sigma_{-1}-2\sigma_0}\left(\frac{\sigma_{xx_m}}{2}+\frac{\sigma_{xx_a}}{2}\right) - \frac{\sigma_{-1}}{2}\left(\frac{-\sigma_0}{\sigma_{-1}-2\sigma_0}\right) = 0 \\ Dang\ Van\ (DV) \\ \quad \frac{\sigma_{xx_a}}{2} + \frac{3(\sigma_0-\sigma_{-1})}{2(\sigma_{-1}-2\sigma_0)}\left(\frac{\sigma_{xx_m}+\sigma_{xx_a}}{3}\right) - \sigma_0\left(\frac{-\sigma_{-1}}{2(\sigma_{-1}-2\sigma_0)}\right) = 0 \end{cases} \tag{11.3}$$

Simplificando:

$$\begin{cases} (S) \rightarrow \sigma_{xx_a} + \frac{(\sigma_{-1}-\sigma_0)}{\sigma_0}\sigma_{xx_m} - \sigma_{-1} = 0 \\ (C) \rightarrow \sigma_{xx_a} + \frac{(\sigma_0-\sigma_{-1})}{\sigma_{-1}-2\cdot\sigma_0}\cdot\left(\sigma_{xx_m}+\sigma_{xx_a}\right) - \sigma_{-1}\left(\frac{-\sigma_0}{\sigma_{-1}-2\sigma_0}\right) = 0 \\ (M) \rightarrow \sigma_{xx_a} + \frac{\sigma_0-\sigma_{-1}}{\sigma_{-1}-2\sigma_0}\cdot\left(\sigma_{xx_m}+\sigma_{xx_a}\right) - \sigma_{-1}\left(\frac{-\sigma_0}{\sigma_{-1}-2\sigma_0}\right) = 0 \\ (DV) \rightarrow \sigma_{xx_a} + \frac{(\sigma_0-\sigma_{-1})}{(\sigma_{-1}-2\sigma_0)}\cdot\left(\sigma_{xx_m}+\sigma_{xx_a}\right) - \sigma_{-1}\left(\frac{-\sigma_0}{\sigma_{-1}-2\sigma_0}\right) = 0 \end{cases} \tag{11.4}$$

En este punto, según (11.4) las expresiones de Crossland (C), Matake (M) y Dang Van (DV) son equivalentes; sin embargo, no es directo que la de Sines (S) lo sea. Por lo tanto, para poder comprobarlo, se va a desarrollar la expresión común a los tres últimos métodos como sigue.

$$\sigma_{xx_a} + \frac{1}{\sigma_{-1}-2\cdot\sigma_0}\cdot\left((\sigma_0-\sigma_{-1})(\sigma_{xx_m}+\sigma_{xx_a}) - \sigma_{-1}\sigma_0\right) = 0 \tag{11.5}$$

$$-\sigma_{xx_a}\sigma_{-1} + 2\sigma_{xx_a}\sigma_0 = \sigma_0\sigma_{xx_m} + \sigma_0\sigma_{xx_a} - \sigma_{-1}\sigma_{xx_m} - \sigma_{-1}\sigma_{xx_a} - \sigma_{-1}\sigma_0 \tag{11.6}$$

Finalmente, por un lado, simplificando y dividiendo entre σ_0 la ecuación (11.6) y, por otro lado, desarrollando la fracción de la expresión de Sines (expresión (S) de (11.4)), se verifica que los cuatro métodos se pueden representar mediante la ecuación (11.7).

$$\sigma_{xx_a} + \frac{\sigma_{-1}}{\sigma_0}\sigma_{xx_m} - \sigma_{xx_m} - \sigma_{-1} = 0 \qquad (11.7)$$

Por último, del mismo modo que se ha realizado con las parejas de ensayos $\sigma_{-1} - \tau_{-1}$ y $\sigma_{-1} - \sigma_0$, en la Figura 11.3 se muestran los resultados obtenidos para el caso de $\sigma_{-1} - \sigma_{ut}$. Atendiendo a la representación de las particularizaciones al caso uniaxial, se puede apreciar como los métodos de Sines, Crossland, Matake y Dang Van en este caso también coinciden.

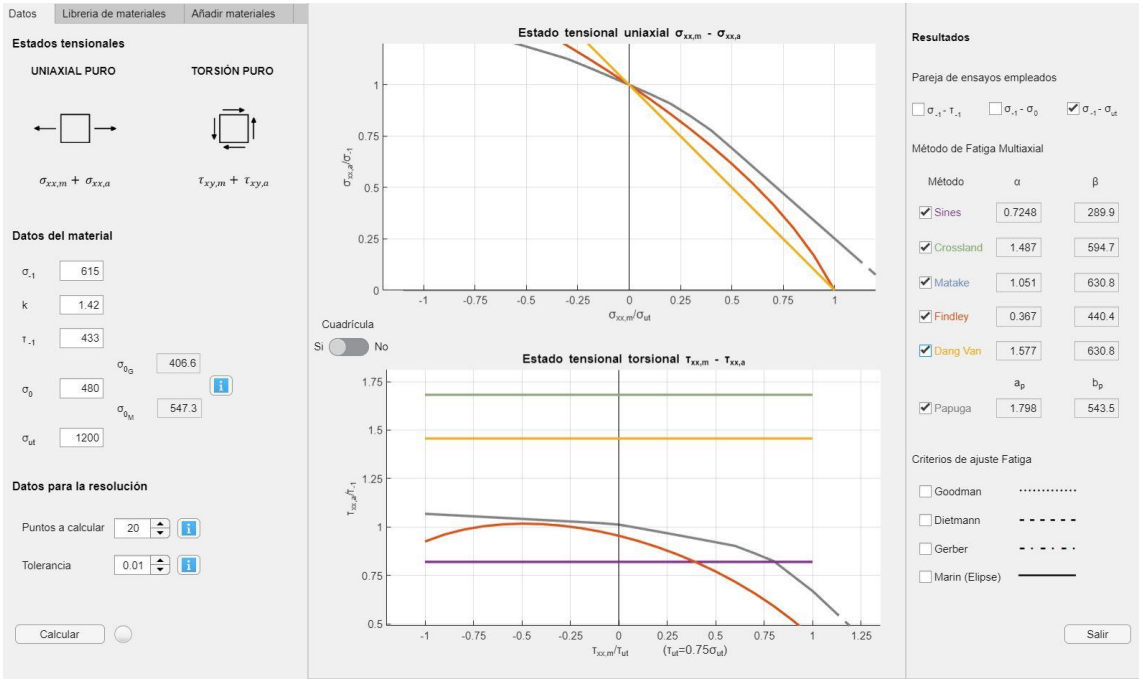

Figura 11.3. Resultados genéricos al particularizar las funciones de daño para la pareja de ensayos $\sigma_{-1} - \sigma_{ut}$.

Con el objetivo de demostrar cómo las funciones de los cuatro métodos que se han citado son análogas independientemente de los valores introducidos, al igual que se ha realizado para el caso de $\sigma_{-1} - \sigma_0$, se sustituyen las expresiones de los parámetros del método en las funciones de daño como sigue.

$$\begin{cases} Sines\ (S) \\ \quad \frac{\sqrt{2}}{3}\sigma_{xx_a} + \frac{\sqrt{2}\cdot\sigma_{-1}}{\sigma_{ut}}\cdot\frac{1}{3}\sigma_{xx_m} - \frac{\sqrt{2}}{3}\cdot\sigma_{-1} = 0 \\ Crossland\ (C) \\ \quad \frac{\sqrt{2}}{3}\sigma_{xx_a} + \frac{\sqrt{2}\cdot\sigma_{-1}}{\sigma_{ut}-\sigma_{-1}}\cdot\left(\frac{\sigma_{xx_m}+\sigma_{xx_a}}{3}\right) - \frac{\sqrt{2}}{3}\sigma_{-1}\left(\frac{\sigma_{ut}}{\sigma_{ut}-\sigma_{-1}}\right) = 0 \\ Matake\ (M) \\ \quad \frac{\sigma_{xx_a}}{2} + \frac{-\sigma_{-1}}{\sigma_{-1}-\sigma_{ut}}\cdot\left(\frac{\sigma_{xx_m}}{2}+\frac{\sigma_{xx_a}}{2}\right) - \frac{-\sigma_{-1}\sigma_{ut}}{2\cdot(\sigma_{-1}-\sigma_{ut})} = 0 \\ Dang\ Van\ (DV) \\ \quad \frac{\sigma_{xx_a}}{2} + \frac{3}{2}\left(\frac{-\sigma_{-1}}{\sigma_{-1}-\sigma_{ut}}\right)\cdot\left(\frac{\sigma_{xx_m}+\sigma_{xx_a}}{3}\right) - \frac{\sigma_{ut}}{2}\left(\frac{-\sigma_{-1}}{\sigma_{-1}-\sigma_{ut}}\right) = 0 \end{cases} \tag{11.8}$$

Simplificando:

$$\begin{cases} (S) \rightarrow \sigma_{xx_a} + \frac{\sigma_{-1}}{\sigma_{ut}}\sigma_{xx_m} - \sigma_{-1} = 0 \\ (C) \rightarrow \sigma_{xx_a} + \frac{1}{\sigma_{ut}-\sigma_{-1}}\cdot\left(\sigma_{-1}\left(\sigma_{xx_m}+\sigma_{xx_a}\right) - \sigma_{-1}\sigma_{ut}\right) = 0 \\ (M) \rightarrow \sigma_{xx_a} + \frac{1}{\sigma_{-1}-\sigma_{ut}}\cdot\left(-\sigma_{-1}\left(\sigma_{xx_m}+\sigma_{xx_a}\right) + \sigma_{-1}\sigma_{ut}\right) = 0 \\ (DV) \rightarrow \sigma_{xx_a} + \frac{1}{\sigma_{-1}-\sigma_{ut}}\cdot\left(-\sigma_{-1}\left(\sigma_{xx_m}+\sigma_{xx_a}\right) + \sigma_{-1}\sigma_{ut}\right) = 0 \end{cases} \tag{11.9}$$

Para verificar que las expresiones de Crossland (C), Matake (M) y Dang Van (DV) de la expresión (11.9) concuerdan con la de Sines (S), hay que desarrollarlas y simplificarlas. Como se puede apreciar, la única diferencia de la expresión de Crossland con la de Matake o Dang Van está en la manera de exponer el segundo término de la ecuación; correspondiendo ambas formulaciones a la misma función. Así pues, se va a partir de la expresión (M) y se va a desarrollar como sigue.

$$-\sigma_{xx_a}\sigma_{-1} + \sigma_{xx_a}\sigma_{ut} = -\sigma_{-1}\sigma_{xx_m} - \sigma_{-1}\sigma_{xx_a} + \sigma_{-1}\sigma_{ut} \tag{11.10}$$

Dividiendo entre σ_{ut} y reordenando, se demuestra matemáticamente que la función que define la particularización de los métodos de Sines, Crossland, Matake y Dang Van para el caso uniaxial (empleando la pareja de ensayos $\sigma_{-1} - \sigma_{ut}$ para determinar las constantes de los métodos) es la siguiente.

$$\sigma_{xx_a} + \frac{\sigma_{-1}}{\sigma_{ut}}\sigma_{xx_m} - \sigma_{-1} = 0 \tag{11.11}$$

Para concluir con los resultados referentes a la particularización de las funciones de daño, a continuación, se va a tratar sobre los resultados del caso de torsión pura. Así pues, por analogía con el caso uniaxial, se va a exponer lo correspondiente a las parejas de ensayos $\sigma_{-1} - \tau_{-1}$, $\sigma_{-1} - \sigma_0$ y $\sigma_{-1} - \sigma_{ut}$; en dicho orden.

A partir de las expresiones obtenidas al particularizar los métodos de fatiga multiaxial al caso de torsión pura se puede deducir la forma que va a adquirir cada uno de ellos. A diferencia de los métodos de Findley y Papuga PCr (ec. (9.107) y (9.111)) que dan lugar a funciones de orden 2 y 4 respectivamente, en el caso de Sines (ec. (9.89)), Crossland (ec. (9.91)), Matake (ec. (9.95)) y Dang Van (ec. (9.113)) la particularización de la función corresponde a rectas de pendiente nula; es decir, rectas horizontales.

Junto con esto, cabe remarcar la similitud entre las expresiones de Sines y de Crossland por un lado y, de Matake y Dang Van por otro lado. Pese a que todas ellas corresponden a rectas horizontales, sus ecuaciones son dependientes de las constantes del método por lo que, en función de la pareja de ensayos empleada su expresión varía. Es por eso que en adelante se van a indicar los métodos cuyas funciones coinciden para cada pareja de ensayos.

Como se puede intuir en la Figura 11.1, para la pareja de ensayos $\sigma_{-1} - \tau_{-1}$ los métodos de Crossland, Matake y Dang Van vienen dados por una función común. En principio, la expresión de Crossland (ec. (9.91)) no corresponde a las de Matake (ec. (9.95)) y Dang Van (ec. (9.113)) pero, al introducir en cada una de ellas sus parámetros correspondientes (ver Tabla 9.2, Tabla 9.3 y Tabla 9.6), matemáticamente se demuestra que se trata de la misma expresión.

$$\begin{cases} Crossland\ (C) \rightarrow \tau_{xy_a} - \dfrac{3}{\sqrt{6}}\dfrac{\sqrt{6}}{3}\tau_{-1} = 0 \rightarrow \tau_{xy_a} - \tau_{-1} = 0 \\ Matake\ (M) \rightarrow \tau_{xy_a} - \tau_{-1} = 0 \\ Dang\ Van(DV) \rightarrow \tau_{xy_a} - \tau_{-1} = 0 \end{cases} \tag{11.12}$$

Por su parte, si se emplea la pareja de ensayos $\sigma_{-1} - \sigma_0$, los únicos métodos cuya particularización al caso de torsión pura dan lugar a funciones coincidentes son Matake y Dang Van (ver Figura 11.2). A continuación, al igual que se ha realizado con los casos anteriores, se demuestra matemáticamente su expresión común sustituyendo en la ecuación de Matake (ec. (9.95)) y en la de Dang Van (ec. (9.113)) las constantes correspondientes (ver Tabla 9.3 y Tabla 9.6).

$$\begin{cases} Matake\ (M) \rightarrow \tau_{xy_a} - \dfrac{\sigma_{-1}}{2}\left(\dfrac{-\sigma_0}{\sigma_{-1} - 2\sigma_0}\right) = 0 \\ Dang\ Van(DV) \rightarrow \tau_{xy_a} - \dfrac{\sigma_0}{2}\left(\dfrac{-\sigma_{-1}}{\sigma_{-1} - 2\sigma_0}\right) = 0 \end{cases} \tag{11.13}$$

Por último, para la pareja de ensayos $\sigma_{-1} - \sigma_{ut}$, del mismo modo que ocurre con la pareja de ensayos $\sigma_{-1} - \sigma_0$, los únicos métodos cuya particularización al caso de torsión pura da lugar a la misma función son Matake y Dang Van. La demostración matemática de esto se obtiene introduciendo en las ecuaciones de Matake (ec. (9.95)) y Dang Van (ec. (9.113)) las constantes que se recogen en la Tabla 9.3 y Tabla 9.6 respectivamente.

$$\begin{cases} Matake\ (M) \rightarrow \tau_{xy_a} + \dfrac{\sigma_{-1}\sigma_{ut}}{2(\sigma_{-1} - \sigma_{ut})} = 0 \\ Dang\ Van(DV) \rightarrow \tau_{xy_a} + \dfrac{\sigma_{ut}}{2}\left(\dfrac{\sigma_{-1}}{\sigma_{-1} - \sigma_{ut}}\right) = 0 \end{cases} \tag{11.14}$$

11.3 Caso particular del acero 34CrNiMo6

Una vez finalizado con los resultados genéricos correspondientes a la particularización de las funciones de daño de los métodos de fatiga multiaxial, en este subapartado se va a realizar un ejemplo para el caso particular del acero 34CrNiMo6. Así pues, conocidos los puntos de fallo tanto del caso uniaxial como de torsión pura y los datos característicos del material (ver Tabla 11.1), se va a identificar el método que mejor se ajusta a este material.

Tabla 11.1. Datos característicos del acero 34CrNiMo6 [26].

Material	$\boldsymbol{\sigma_{-1}}$	$\boldsymbol{\tau_{-1}}$	$\boldsymbol{\sigma_0}$	$\boldsymbol{\sigma_{ut}}$
34CrNiMo6	615 MPa	433 MPa	480 MPa	1200 MPa

Tras introducir en la aplicación estos datos característicos, se selecciona dicho acero de la "Librería de materiales" y se comienza el proceso de identificación visual del método que mejor se ajusta a los puntos de fallo que se disponen. Con este fin, entre las posibles técnicas que se pueden seguir, la que se va a emplear consiste en seleccionar los seis métodos candidatos y para cada pareja de ensayos elegir la alternativa que mejor se ajusta. Después de esto, se tienen tres opciones (una para cada pareja de ensayos) entre las que se elige el método que mejor se ajusta.

A la vista de la Figura 11.4, no hay ningún método completamente adecuado para los resultados del caso uniaxial ya que todos ellos disponen de algún punto de fallo por debajo de su curva; lo cual indica que se está produciendo el fallo y el método no lo predice. Pese a ello, en una primera selección de los métodos que mejor se ajustan al caso uniaxial, se elige a Dang Van (o Matake, ya que la función es análoga a la de Dang Van), Papuga PCr y Findley. Por su parte, en el caso de torsión pura el único método aceptable es el de Findley ya que el resto son poco conservadores. De ahí que el método seleccionado si se emplease la pareja de ensayos $\sigma_{-1} - \tau_{-1}$ es el de Findley.

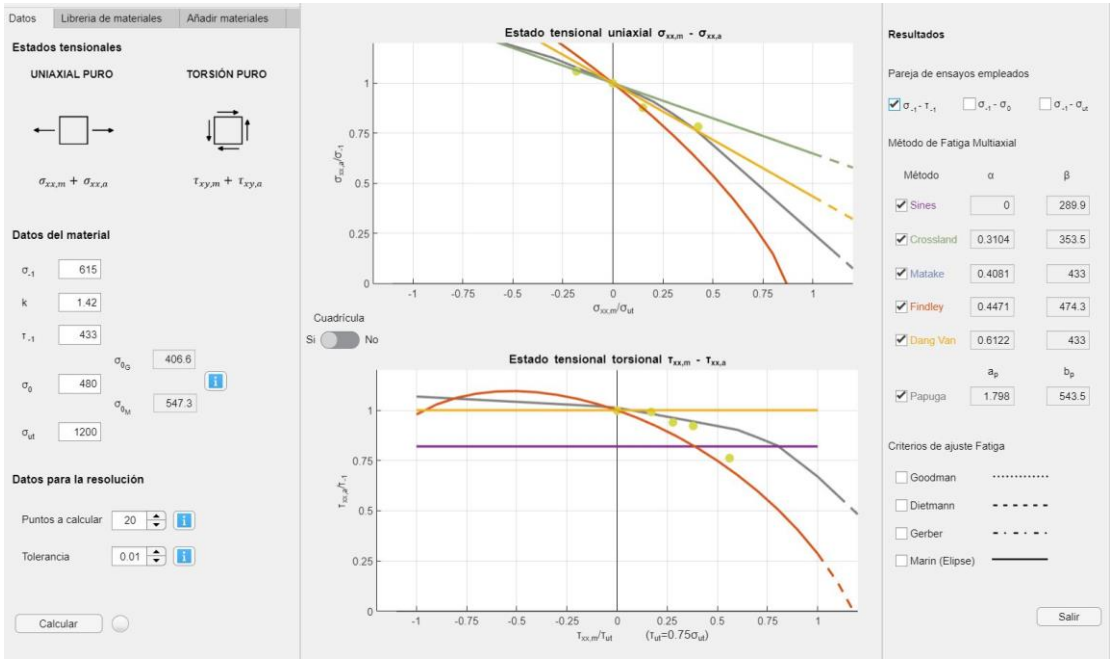

Figura 11.4. Caso particular del acero 34CrNiMo6 con la pareja de ensayos $\sigma_{-1} - \tau_{-1}$.

Por otro lado, trabajando con la pareja de ensayos $\sigma_{-1} - \sigma_0$ los resultados que se obtienen son los que se muestran sobre la Figura 11.5. En el caso uniaxial, no hay ningún método a destacar, pero en el caso de torsión pura, el único que se encuentra por el lado de la seguridad es el de Findley por lo que, se considera ésta la mejor opción.

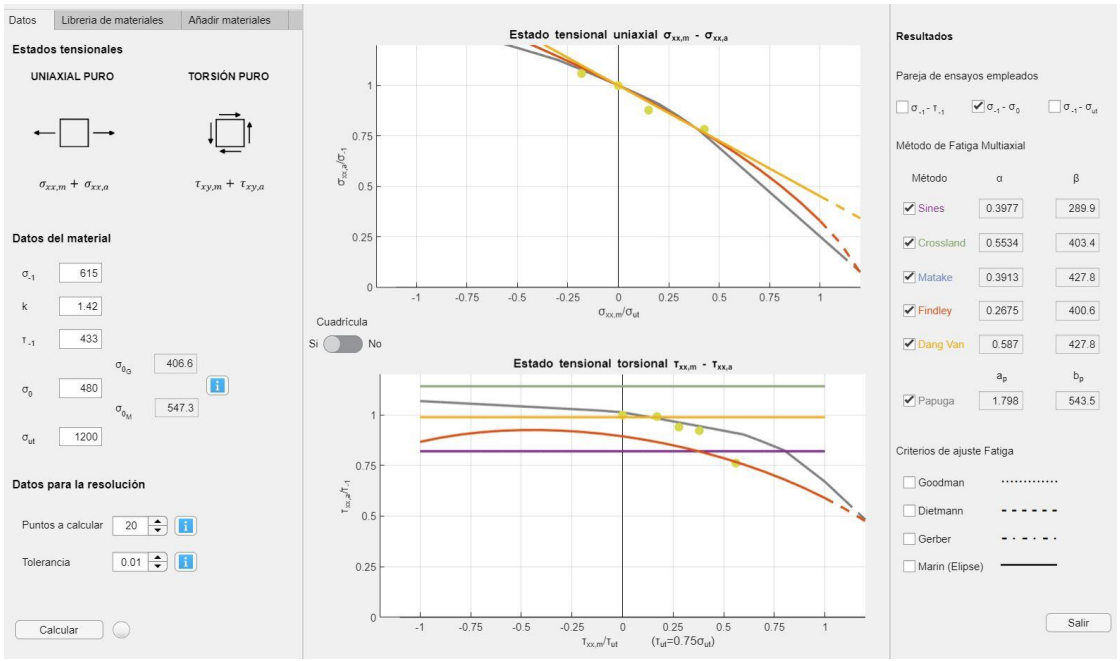

Figura 11.5. Caso particular del acero 34CrNiMo6 con la pareja de ensayos $\sigma_{-1} - \sigma_0$.

Con la tercera pareja de ensayos, $\sigma_{-1} - \sigma_{ut}$, se obtienen los resultados que de la Figura 11.6. En este caso, atendiendo a la representación del caso uniaxial se deduce que los métodos a emplear podrían ser tanto Dang Van (o Sines, Crossland o Matake ya que la función es análoga a la de Dang Van) o Findley. Sin embargo, de acuerdo al caso de torsión pura, el único método conservador es el de Findley ya que en el resto los puntos de fallo se encuentran en la zona de vida infinita. Por lo tanto, el método que se destaca de la pareja de ensayos, $\sigma_{-1} - \sigma_{ut}$ es el de Findley.

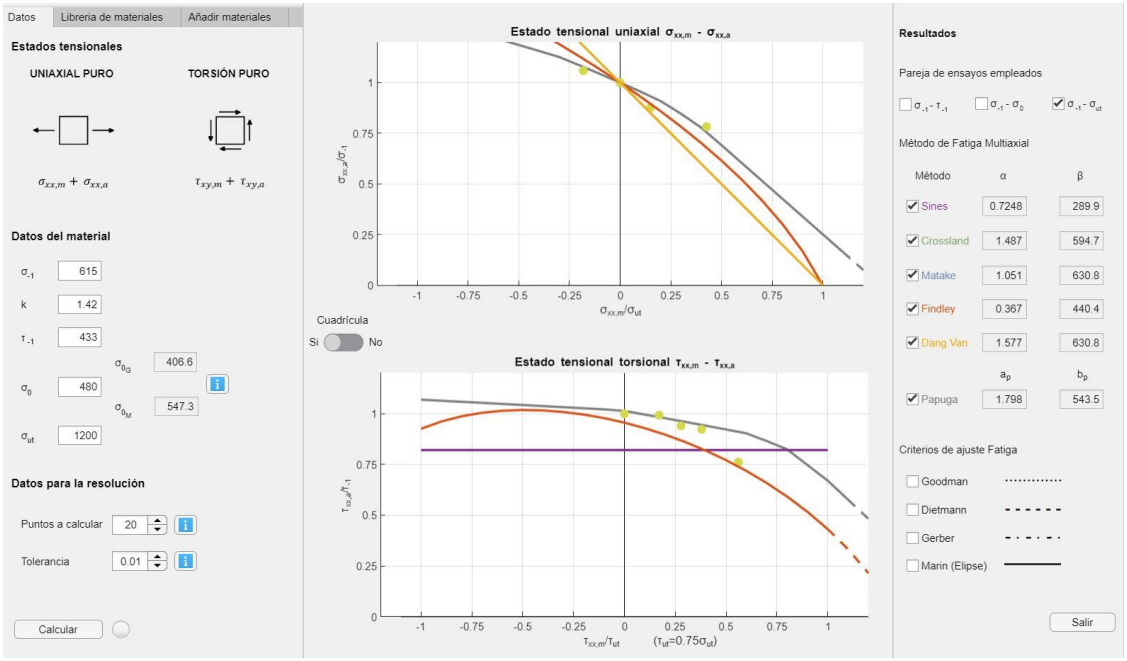

Figura 11.6. Caso particular del acero 34CrNiMo6 con la pareja de ensayos $\sigma_{-1} - \sigma_{ut}$.

Tras seleccionar una alternativa por cada pareja de ensayos, lo siguiente es elegir cuál de ellas constituye la mejor opción. En este sentido, la elección se va a basar en un criterio conservador dónde prevalece que el método en cuestión se ajuste lo mejor posible a los puntos de fallo y disponga del menor número de fallos en la zona de vida infinita. Así pues, de los tres métodos candidatos se deduce que el método que más se adecúa a dichas características y por tanto, mejor se ajusta al acero 34CrNiMo6 es el de Findley particularizado a la pareja de ensayos $\sigma_{-1} - \tau_{-1}$. Por último, en la Tabla 11.2 se recogen los valores de los parámetros del método que proporciona la app (ver Figura 11.4).

Tabla 11.2. Parámetros de Findley con la pareja de ensayos $\sigma_{-1} - \tau_{-1}$ para el ejemplo del acero 34CrNiMo6.

Parámetro	-
α	0,4471
β	474,3

III. ASPECTOS ECONÓMICOS

12 Descargo de gastos

El descargo de gastos que aquí se presenta, se engloba principalmente en tres partidas diferentes. Por un lado, se tienen los recursos humanos donde se consideran las horas internas que se han empleado en la elaboración del trabajo. Por otro lado, en la partida correspondiente a las amortizaciones, se contabiliza el gasto derivado del equipo y las licencias utilizadas. Finalmente, para recoger todos aquellos costes que no se pueden asociar a los gastos anteriores; como pueden ser la electricidad o los desplazamientos a la Escuela, se realiza la tercera partida; costes indirectos.

A pesar de que los datos que se van a utilizar para llevar a cabo cada una de estas partidas se van a mostrar explícitamente en su correspondiente tabla, cabe remarcar la estrecha relación que tiene la partida de costes indirectos con las dos anteriores. Esto se debe a que, para su cálculo, se ha considerado el 2% del subtotal de las dos primeras partidas: recursos humanos y amortizaciones.

Así pues, sobre la Tabla 12.1, la Tabla 12.2 y la Tabla 12.3 que se muestran a continuación, se recogen cada una de las partidas previamente mencionadas.

Tabla 12.1. Partida de costes de recursos humanos.

RECURSOS HUMANOS			
Concepto	**Tiempo (h)**	**Precio (€/h)**	**Total (€)**
Director del trabajo	50	50	2.500
Ingeniero junior	600	30	18.000
SUBTOTAL			20.500

Tabla 12.2. Partida de costes de amortizaciones.

AMORTIZACIONES				
Concepto	**Coste total (€)**	**Tiempo utilizado (h)**	**Tiempo útil (h)**	**Total (€)**
Ordenador portátil HP 15-eg1001ns	1000	600	8.000	75
Licencia anual Matlab R2021a [28]	800	400	1.000 (200 días; 5h/día)	320
Licencia Microsoft Office 2016	70	70	2.500	1,96
SUBTOTAL				396,96

Tabla 12.3. Partida de costes indirectos.

COSTES INDIRECTOS			
Concepto	**Subtotal (€)**	**Porcentaje**	**Total (€)**
Recursos humanos	20.500	2%	410
Amortizaciones	396,96	2%	7,94
SUBTOTAL			417,94

Por último, sobre la Tabla 12.4 se recoge la partida de costes total del trabajo.

Tabla 12.4. Partida de costes totales.

COSTE TOTAL	
Concepto	**Total (€)**
Recursos humanos	20.500
Amortizaciones	396,96
Costes indirectos	417,94
TOTAL	21.314,9

El coste total del presente Trajo de Fin de Máster asciende a la cantidad de VEINTIÚN MIL TRESCIENTOS CATORCE EUROS CON NOVENTA CÉNTIMOS.

IV. CONCLUSIONES Y LÍNEAS FUTURAS

13 Conclusiones y líneas futuras

13.1 Conclusiones

Con el desarrollo del presente trabajo, se pueden deducir numerosas conclusiones; de entre las cuales, se van a destacar las que se exponen a continuación.

En primer lugar, en todos los métodos de fatiga multiaxial que se han particularizado al caso uniaxial de tensiones se ha apreciado una tendencia descendente de las funciones. Esto indica que a medida que aumenta la tensión normal media, la tensión normal alterna que se requiere para que se produzca el fallo es menor; lo cual es lógico ya que la tensión media de tracción tiende a "abrir" la grieta por la que se produce el fallo por fatiga y por tanto, su comportamiento es más desfavorable.

Siguiendo con la particularización de los métodos al caso uniaxial, destacar como dependiendo de la pareja de ensayos empleada para la obtención de los parámetros del método, en algunos casos la función que los define es común. Esto ocurre con los métodos de Matake y Dang Van si se emplea la pareja de ensayos $\sigma_{-1} - \tau_{-1}$ y con Sines, Crossland, Matake y Dang Van si se trabaja con $\sigma_{-1} - \sigma_0$ o $\sigma_{-1} - \sigma_{ut}$.

Por otro lado, al particularizar los métodos de fatiga multiaxial al caso de torsión pura, se demuestra cómo la influencia de la tensión cortante media es un tema de controversia donde cada autor decide si la considera o no. En este sentido, los métodos de Sines, Crossland, Matake y Dang Van consideran despreciable su efecto y los únicos que la tienen en cuenta son Findley y Papuga. Es por ello que, la representación de los primeros viene dada por rectas horizontales mientras que la de éstos últimos, da lugar a curvas cuya tendencia es descendente a medida que aumenta la tensión cortante media.

Junto con esto, al igual que ha ocurrido con la particularización del caso uniaxial, dependiendo de la pareja de ensayos empleada para la obtención de las constantes del método, hay casos donde la función que los caracteriza es común. Éstos son Crossland, Matake y Dang Van para la pareja de ensayos $\sigma_{-1} - \tau_{-1}$ y Matake y Dang Van si se trabaja con $\sigma_{-1} - \sigma_0$ o $\sigma_{-1} - \sigma_{ut}$.

Además de todo esto, en cuanto a las características destacables de los métodos que se particularizan en este trabajo, se deben remarcar dos aspectos referentes a Sines y a Papuga. Respecto a Sines, mencionar la imposibilidad de representar su función particularizada al caso uniaxial mediante la pareja de ensayos $\sigma_{-1} - \tau_{-1}$ debido a que la constante α_S no se puede despejar de dichos ensayos. Por su parte, en cuanto a Papuga PCr destacar como para su definición se requieren tres límites de fatiga (σ_{-1}, τ_{-1} y σ_0);

lo cual se puede considerar una desventaja si se compara con el resto de métodos donde, es suficiente con conocer dos de ellos.

Por último, de forma general y a la vista de los resultados obtenidos al particularizar los métodos de fatiga multiaxial tanto al caso uniaxial como al de torsión pura, se concluye que los métodos son poco conservadores; o lo que es lo mismo, que en líneas generales no están por el lado de la seguridad. Esto se ha deducido en base a que varios de los puntos de fallo que se recogen en bibliografía se encuentran por debajo de las funciones resultantes al particularizar los métodos; es decir, están en lo que se presupone como la zona de vida infinita.

13.2 Líneas Futuras

A pesar de haber obtenido una aplicación con numerosos rasgos de gran interés, se considera que se podría aumentar su potencial si adicionalmente, se incluyesen otros aspectos que se van a mencionar a continuación.

En primer lugar, para poder concluir con mayor claridad si las funciones obtenidas tras la particularización de los métodos son conservadoras o no, convendría realizar ensayos experimentales tanto de fatiga uniaxial como de torsión pura e introduciendo en la app los datos característicos del material en cuestión, observar los resultados. Si la representación de las funciones queda por debajo de los puntos de fallo, los métodos estarán del lado de la seguridad; de lo contrario, al producirse el fallo en la zona donde no se prevé, se afirmaría que son poco conservadores.

Además de la realización de una campaña experimental, con el objetivo de aumentar la versatilidad de la aplicación se pueden implementar las mejoras que se citan en adelante. Por un lado, aunque disponga de varios de los métodos de fatiga multiaxial más destacados, se podrían incluir otros métodos para conseguir así, abarcar un mayor campo de aplicación que aumente la utilidad de la herramienta.

Por otro lado, pese a que los tiempos de ejecución que presenta la aplicación se pueden considerar correctos, es posible mejorarlos, por ejemplo, mediante técnicas de optimización de código. De esta forma, se obtendría una aplicación con mayor agilidad; especialmente al llevar a cabo las operaciones necesarias para la resolución de las particularizaciones de los métodos.

Finalmente, sería interesante también que el usuario fuese capaz de modificar ciertos aspectos de las representaciones como, por ejemplo, el color o el tipo de línea de las curvas que se muestran. Con esto, se conseguiría una aplicación que diese mayor libertad al usuario permitiéndole dotarla de rasgos personalizados a su criterio.

V. BIBLIOGRAFÍA

Bibliografía

[1] Abasolo, M. (2021/2022). Métodos de Análisis y Diseño para Fractura y Fatiga. *Departamento de Ingeniería Mecánica, Universidad del País Vasco (UPV/EHU)*.

[2] Avilés, R. (2015). *Métodos de cálculo de fatiga para ingeniería.* Ediciones Paraninfo.

[3] BBC News. (30 de Noviembre de 2019). Last square-windowed Comet moved to new de Havilland Museum hangar. *BBC* .

[4] Abasolo, M., Navalpotro, S., Iriondo, E., & Corral, J. (2017/2018). Diseño de Máquinas. *Departamento de Ingeniería Mecánica, Universidad del País Vasco*.

[5] Hilsz-Lothian, A. (15 de Mayo de 2019). *SAM CHUI aviation & travel*. Recuperado en 2022, de https://samchui.com/2019/05/15/boeing-prepares-777x-for-fatigue-testing/#.YhPt4OjMKUk

[6] *De Máquinas y Herramientas*. (21 de Mayo de 2015). Obtenido de https://www.demaquinasyherramientas.com/herramientas-de-medicion/caracterizacion-del-mecanismo-de-falla-de-ciguenal

[7] Schijve, J. (2004*). Fatigue of Structures and Materials*. Kluwer Academic Publishers.

[8] Ashutosh Sharma, Min Chul Oh, & Byungmin Ahn. (2020). Recent Advances in Very High Cycle Fatigue Behavior of Metals and Alloys. *Metals*.

[9] Socie, D., Marquis, G. (1999). *Multiaxial Fatigue*. Society of Automotive Engineers.

[10] Virtual Expo Group. (s.f.). *Direct Industry*. Recuperado en 2022, de https://www.directindustry.es/prod/walterbaiag/product-222874-2364811.html

[11] Almaguer-Zaldivar, P., Martínez-Grave-de-Peralta, J., González-Utria, E., & Santiago-Cuenca, H. (2018). Evaluación de probetas cilíndricas solicitadas a torsión cíclica simétrica. *Ingeniería Mecánica, 21*(2).

[12] UPV/EHU. (s.f.). *Máster en Ingeniería Mecánica*. Recuperado en 2022, de https://www.ehu.eus/es/web/master/master-ingenieria-mecanica/materia?p_anyo_ofd=20210&p_anyo_pop=20140&p_cod_centro=345&p_cod_materia=6733&p_cod_asignatura=504240&p_tipo_asignatura=1

[13] Stephens, R., Fatemi, A., Stephens, R., Fuchs, H. (2001). *Metal Fatigue in Engineering*. A Wiley-Interscience Publication.

[14] Papuga, J. (2011). A survey on evaluating the fatigue limit under multiaxial loading. *International Journal of Fatigue*, 153-165.

[15] Lin, T. H. (1957). Analysis of elastic and plastic strains of a face-centred cubic crystal. *Journal of the Mechanics and Physics of Solids*, *5*(2), 143-149.

[16] Taylor, G. I. (1938). Plastic strain in metals. *Twenty-eight may lectureto the Institute of Metals*, 307-324.

[17] Apodaka, P., Coria, I. (2020). *Desarrollo de una aplicación en Matlab para el análisis de mecanismos.*

[18] Gonçalves, C., Araújo, J., & Mamiya, E. (2004). A simple multiaxial fatigue criterion for metals. *C. R. Mecanique*, 963-968.

[19] Jesuïtes educació. (17 de junio de 2019). *Visual Basic: ventajas y desventajas*. Recuperado en 2022, de https://fp.uoc.fje.edu/blog/visual-basic-ventajas-y-desventajas/

[20] COVANTEC. (2019). *Programación en Python, nivel básico*. Obtenido de https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion1/ventajas_desventajas.html

[21] Lozano, E. (s.f.). *Creación de GUI con Python*. Recuperado en 2022, de https://eduardolozanoprogramacionparalela2015b.wordpress.com/2015/09/01/creacion-de-gui-con-python/

[22] *Componentes Electrónicas*. (26 de enero de 2018). Recuperado en 2022, de https://www.compelect.com.co/2018/01/26/7-ventajas-de-usar-matlab/

[23] MathWorks. (s.f.). *App Building Components*. Obtenido de https://es.mathworks.com/help/matlab/creating_guis/choose-components-for-your-app-designer-app.html

[24] MathWorks. (s.f.). *Licencias de Matlab para su uso en todo el campus*. Obtenido de https://es.mathworks.com/products/matlab-campus.html

[25] Universidad de Cantabria. (s.f.). *Servicio de Informática*. Recuperado en 2022, de https://sdei.unican.es/Paginas/servicios/software/Labview.aspx

[26] Pallarés Santasmartas, L. (2021). *Development of two multiaxial fatigue analysis methods for ductile metals based on energy considerations.* Bilbao.

[27] Frost NE, M. K. (1974). Metal fatigue. *Oxford: Oxford University Press*.

[28] MathWorks. (s.f.). *Pricing and Licensing*. Recuperado en 2022, de https://es.mathworks.com/pricing-licensing.html?prodcode=ML&intendeduse=comm

[29] Pallarés Santasmartas, L., Albizuri Irigoyen, J., Avilés Ajuria, A., & Avilés González, R. (2018). Mean Stress Effect on the Axial Fatigue Strength of DIN 34CrNiMo6 Quenched and Tempered Steel. *Metals*, *4*(213), 8.

[30] Ukrainetz, P. (1960). The Effect of the Mean Stress on the Endurance Limit. *The University of Columbia*.

[31] Gough, H. (1924). The Fatigue of Metals.

[32] Forrest, P. (1962). Fatigue of Metals. *Pergamon Press Inc*.

[33] Grover, H., Bishop, S., & Jackson, L. (1951). Axial Load Fatigue Tests of Unnotched Sheet Specimens of 24S-T3 and 75S-T6 Aluminum Alloys and SAE 4130 Steels. *National Advisory Committee for Aeronautics*.

[34] Trapp, W., & Schwartz, R. (1953). Elevated Temperature Fatigue Properties of SAE 4340 Steel.

[35] O'Connor, H., & Morrison, J. (1956). The Effect of Mean Stress on the Push-Pull Fatigue Properties of an Alloy Steel. *In Proceedings of the International Conference on Fatigue of Metals*, 102-109.

[36] Grün, P., Troost, A., Akin, O., & Klubberg. (1991). Langzeitund Dauerschwingfestigkeit des Vergütungsstahls 25CrMo4 bei mehrachsiger Beanspruchung durch dreischwingende Lastspannungen. *Materialwissenschaft Werkstofftechnik*.

[37] Schäfer, H., Hempen, M., Klubberg, F., & Beiss, P. (2001). Mittelspannungsempfindlichkeit metallischer Werkstoffe bei schwingender Beanspruchung. *Aachen*.

[38] Bomas, H., Bacher-Hoechst, M., Kienzler, R., Kunow, S., Loewisch, G., Muehleder, & Schroeder, R. (2010). Crack initiation and endurance limit of a hard steel under multiaxial cyclic loads. *Fatigue Fract. Eng. Mater. Struct.*

[39] Pallarés-Santasmartas, L., Albizuri, J., Avilés, A., Saintier, N., & Merzeau, J. (2018). Influence of mean shear stress on the torsional fatigue behaviour of 34CrNiMo6 steel. *International Journal of Fatigue*.

[40] Jasper, T., & Moore, H. (1924). An investigation of the fatigue of metals. *University of Illinois Engineering Experiment Station*.

[41] Moore, H., & Jasper, T. (1925). An investigation of the fatigue of metals. Series of 1925. A report of the investigation. *University of Illinois Engineering Experiment Station*.

[42] JO, S. (1939). The effect of range of stress on the torsional fatigue strength of steel. *University of Illinois Engineering Experiment Station*.

[43] Sauer, J. (1948). A study of fatigue phenomena under combined stress. *Proceedings of the seventh international congress for applied mechanics*.

[44] Gough, H., Pollard, H., & Clenshaw, W. (1951). Some experiments on the resistance of metals to fatigue under combined stresses. *Aeronautical Research Council reports and memoranda*.

[45] Findley, W. (1953). Combined-stress fatigue strength of 76S–T61 Aluminum alloy with superimposed mean stresses and corrections for yielding.

[46] WT, C. (1956). Fatigue strength in shear of an alloy steel, with particular reference to the effect of mean stress and directional properties.

[47] Baier, F. (1970). Zeit- und Dauerfestigkeit bei überlagerter statischer und schwingender Zug-Druck- und Torsionbeanspruchung. *Stuttgart*.

[48] Issler, L. (1973). Festigkeitsverhalten metallischer Werkstoffe bei mehrachsiger phasenverschobener Schwingbeanspruchung. *Stuttgart*.

[49] Lempp, W. (1977). Festigkeitsverhalten von Stählen bei mehrachsiger Dauerschwingbeanspruchung durch Normalspannungen mit überlagerten phasengleichen und phasenverschobenen Schubspannungen. *University of Stuttgart*.

[50] Richter, H. (1984). Schubspannungsintensitätshypothese —weitere experimentelle und theoretische Untersuchungen. *Konstruktion*.

[51] T., B. (1986). Festigkeitsverhalten von Stählen unter mehrachsiger phasenverschobener Schwingbeanspruchung mit unterschiedlichen Schwingungsformen und Frequenzen. *Stuttgart*.

[52] The Society of Materials Science. (1996). JSMS Databook: Databook on fatigue strength of metallic materials.

[53] Lüpfert, H.-P., & Spies, H.-J. (2001). EinfluB von Druckvorspannungen auf die Dauerfestigkeit metallischer Werkstoffe bei ein-und mehrachsiger Beanspruchung. *Mat-wiss u Werkstofftech*.

[54] Davoli, P. (2003). Independence of the torsional fatigue limit upon a mean shear stress. *International Journal of Fatigue*.

[55] Mayer, H. (2015). Cyclic torsion very high cycle fatigue of VDSiCr spring steel at different load ratios,» International Journal of Fatigue.

VI. ANEXO I. Materiales de bibliografía

Entre los diferentes objetivos con los que se ha desarrollado la aplicación, se encuentra la posibilidad de identificar el método de fatiga multiaxial que mejor se ajusta a un determinado material empleando los puntos de fallo del ensayo uniaxial y de torsión pura en lugar de realizar ensayos de fatiga multiaxial. Así pues, pese a que lo ideal sería obtener los valores de dichos puntos de fallo a partir de ensayos del material particular con el que se va a trabajar, en ocasiones entre los datos de bibliografía se dispone de datos referentes a dicho material y es suficiente para decidir sobre el método que mejor se ajusta.

En este sentido, en la pestaña "Librería de materiales" que se dispone en la aplicación, se incluyen los datos de fallo de una serie de materiales recogidos en bibliografía. A continuación, se muestran con mayor detalle los puntos de fallo de cada uno de estos materiales dividiéndolos en dos grupos: uniaxial puro y torsión puro.

Caso Uniaxial puro

Sobre la Figura I. 1 se recogen algunos de los datos de fallo que se han incluido en la aplicación para el caso uniaxial puro. Mediante la leyenda que acompaña a la representación, se puede identificar el material al que pertenecen cada uno de los puntos de fallo y como se puede apreciar, los ejes se encuentran normalizados; es decir, en el eje de ordenadas se tiene σ_a/σ_{-1} y en el de abscisas $\sigma_m/\sigma_{\mathrm{ut}}$.

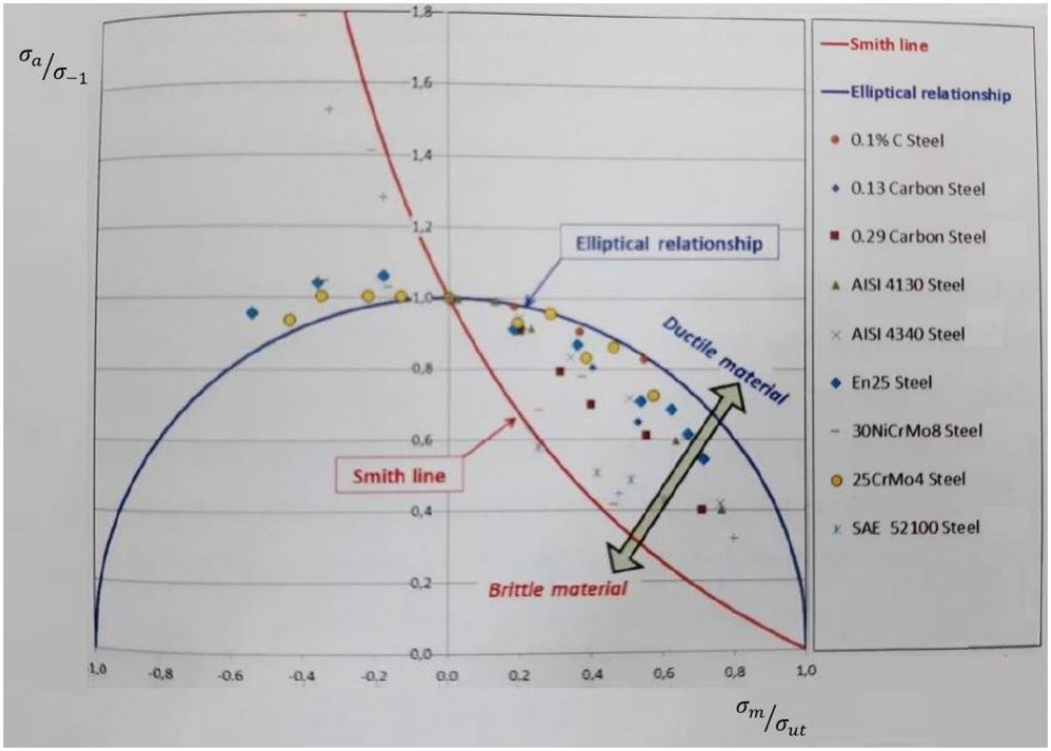

Figura I. 1. Grupo "1" de materiales en el diagrama de Haigh normalizado para el caso uniaxial [26].

Además de los materiales mostrados en la Figura I. 1, la aplicación también incluye, para el caso uniaxial, los puntos de fallo de los materiales de la Figura I. 2. Al igual que en el caso anterior, para identificar el material en cuestión se dispone de una leyenda en la parte derecha y los ejes están normalizados.

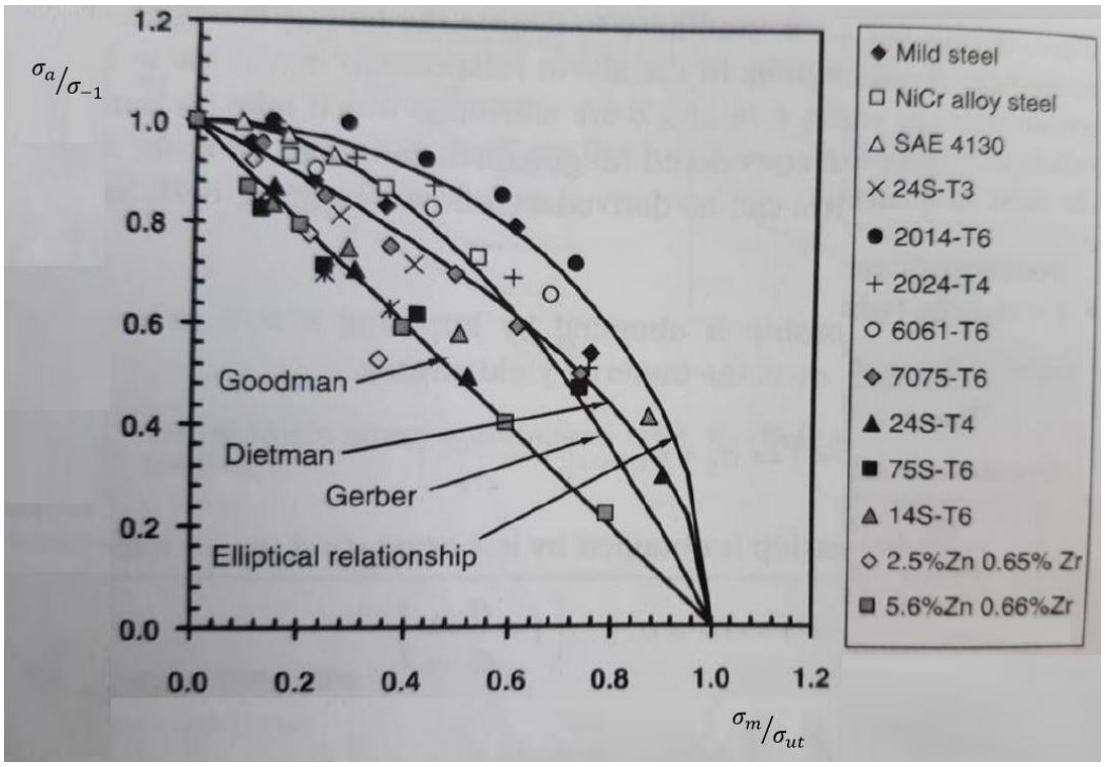

Figura I. 2. Grupo "2" de materiales en el diagrama de Haigh normalizado para el caso uniaxial [27].

A modo resumen, en el siguiente listado se recogen los materiales de bibliografía, referentes al caso uniaxial, que se incluyen en la app.

- 34CrNiMo6 steel [29]
- 0,1% C Steel [30]
- 0,13 Carbon Steel [31]
- 0,29 Carbon Steel [32]
- AISI 4130 Steel [33]
- AISI 4340 Steel [34]
- En 25 Steel [35]
- 30NiCrMo8 Steel [36]
- 25CrMo4 Steel [37]
- SAE 52100 Steel [38]
- Mild Steel [27]
- NiCr alloy Steel [27]
- SAE 4130 [27]
- 24S-T3 [27]
- 2014-T6 [27]
- 2024-T4 [27]
- 6061-T6 [27]
- 7075-T6 [27]
- 24S-T4 [27]
- 75S-T6 [27]
- 14S-T6 [27]
- 2,5%Zn 0,65%Zr [27]
- 5,6%Zn 0,66%Zr [27]

Caso Torsión puro

Por su parte, para el caso de torsión pura los puntos de fallo de los materiales se representan sobre la Figura I. 3. De la misma forma que para los materiales del caso uniaxial, cada material se puede identificar mediante la leyenda de la parte derecha y, los ejes del gráfico están normalizados; τ_a/τ_{-1} en el eje de ordenadas y τ_m/τ_{ut} en el de abscisas.

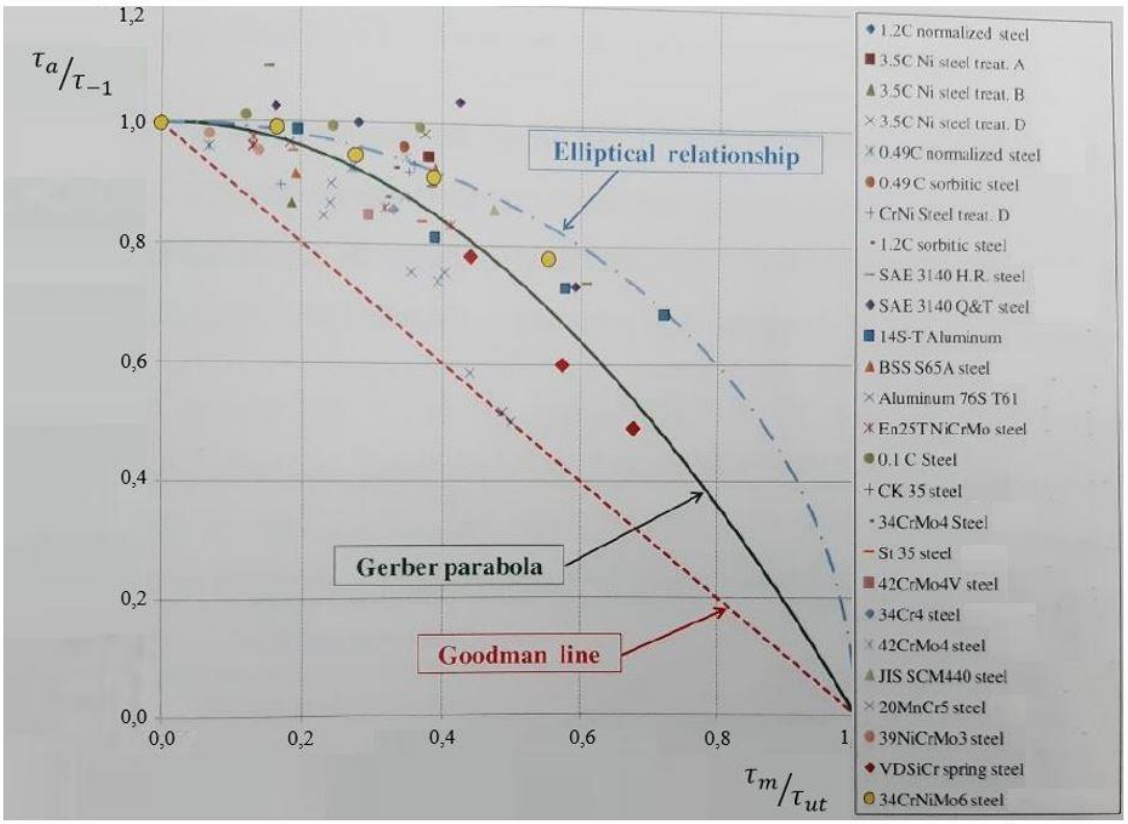

Figura I. 3 Materiales en el diagrama de Haigh normalizado para el caso de torsión pura [26].

Por analogía con el caso uniaxial, a continuación, se recogen en una lista los materiales de bibliografía del caso de torsión pura.

- 34CrNiMo6 steel [39]
- 1,2C normalized steel [40]
- 3,5C Ni steel treatment A [40]
- 3,5C Ni steel treatment B [40]
- 3,5C Ni steel treatment D [40]
- 0,49C normalized steel [40]
- 0,49C sorbitic steel [40]
- CrNi steel treatment D [41]
- 1,2C sorbitic steel [41]
- SAE 3140 H.R. steel [42]
- SAE 3140 Q&T steel [42]
- 14S-T Aluminium [43]
- BSS S65A steel [44]
- Aluminium 76S T61 [45]

- En 25T NICrMo steel [46]
- 0,1C steel [30]
- CK 35 steel [47]
- 34CrMo4 steel [47]
- St 35 steel [48]
- 42CrMo4V steel [49]
- 34Cr4 steel [50]
- 42CrMo4 steel [51]
- JIS SCM440 steel [52]
- 20MnCr5 steel [53]
- 39NiCrMo3 steel [54]
- VDSiCr spring steel [55]

VII. ANEXO II: Manual de usuario

A pesar del carácter intuitivo por el que se caracteriza la aplicación desarrollada para la particularización de los métodos de fatiga multiaxial a los casos uniaxial y torsión pura, se presenta una guía para que el usuario pueda consultar las dudas que le puedan surgir. Por lo tanto, se va a comenzar exponiendo los pasos que se deben seguir para su instalación y, a continuación, se explicarán los diferentes comandos que se incorporan.

Instalación de la aplicación

Tal y como se puede apreciar en la Figura II. 1, tras compilar la aplicación, Matlab genera varios archivos que contienen información acerca de la misma.

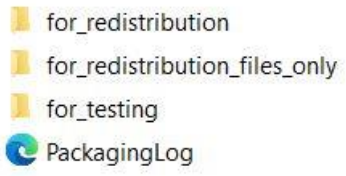

Figura II. 1. Archivos generados al compilar la aplicación.

De entre ellos, para poder disponer de la aplicación en cualquier ordenador (independientemente de si en él está instalado Matlab o no) se debe entrar en la carpeta *"for_redistribution"* y ejecutar el archivo *"MyAppInstaller_mcr"* que se muestra en la Figura II. 2.

Figura II. 2. Archivo MyAppInstaller.

Después de ejecutar el archivo *"MyAppInstaller_mcr"*, se abre la primera página del instalador de la aplicación (ver Figura II. 3) dónde se recogen los datos característicos de la app. Para continuar con la instalación, se debe pulsar sobre el botón remarcado *"Next"*.

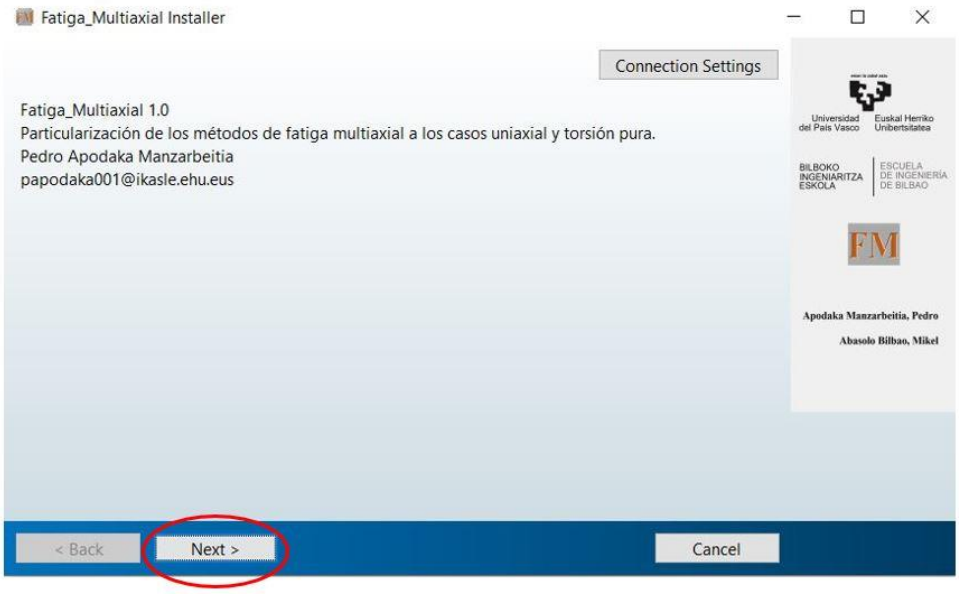

Figura II. 3. Datos característicos de la aplicación.

En la segunda página del instalador, se selecciona el destino dónde se desea guardar la carpeta con los archivos de la app (paso 1 de la Figura II. 4) y además, se tiene la opción de añadir un acceso directo en el escritorio (paso 2 de la Figura II. 4). Una vez completado este paso, para continuar se pulsa sobre el botón *"Next"* (paso 3 de la Figura II. 4).

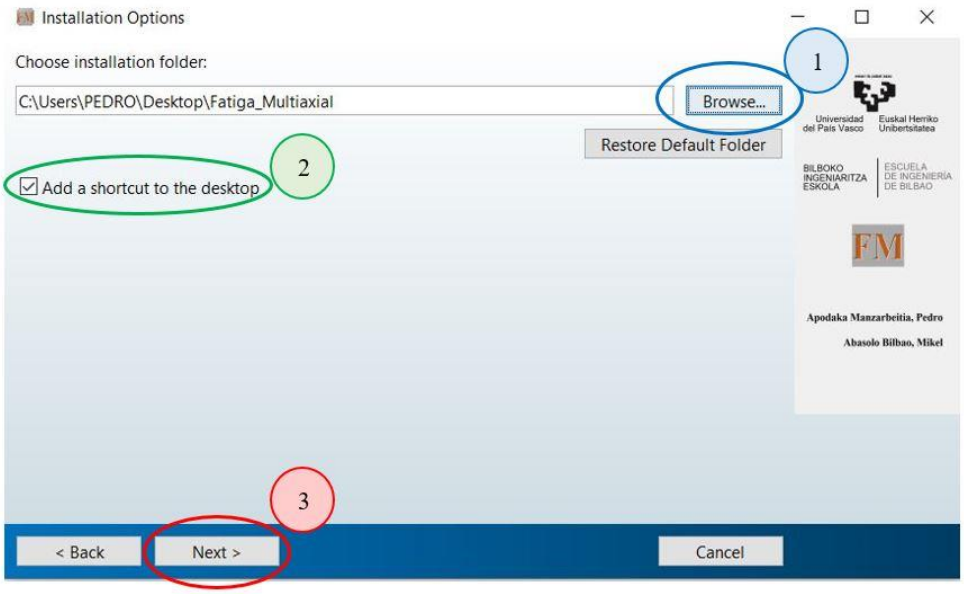

Figura II. 4. Selección del destino de instalación de la app.

Lo siguiente es determinar el destino de los archivos adicionales a la aplicación (paso 1 de la Figura II. 5) y continuar con la instalación clicando en *"Next"* (paso 2 de la Figura II. 5).

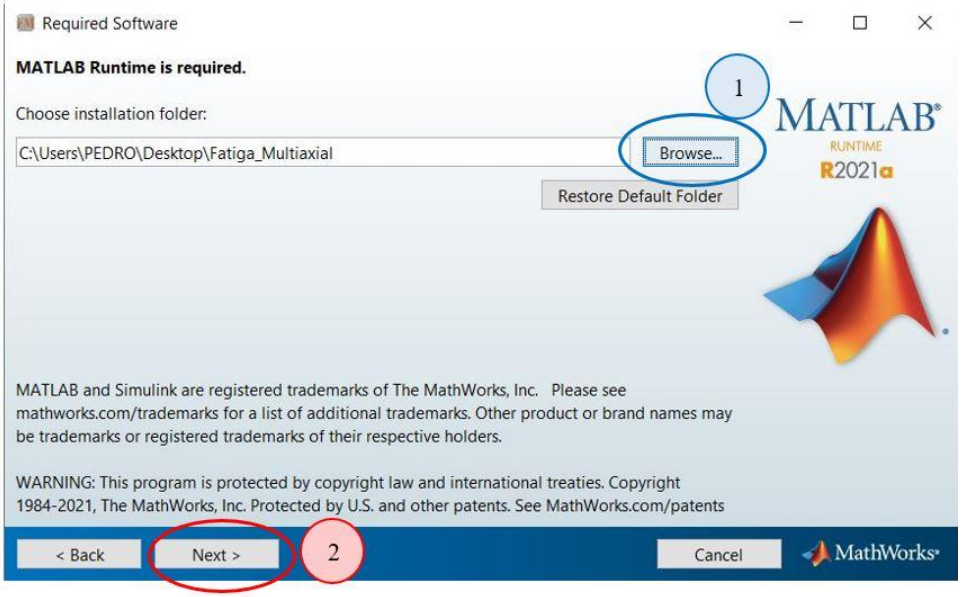

Figura II. 5. Selección del destino de los archivos complementarios de la app.

Una vez seleccionados los destinos dónde se desean guardar los archivos de la app, se deben aceptar los términos y condiciones de Matlab (paso 1 de la Figura II. 6) para poder continuar con la instalación pulsando en "Next" (paso 2 de la Figura II. 6).

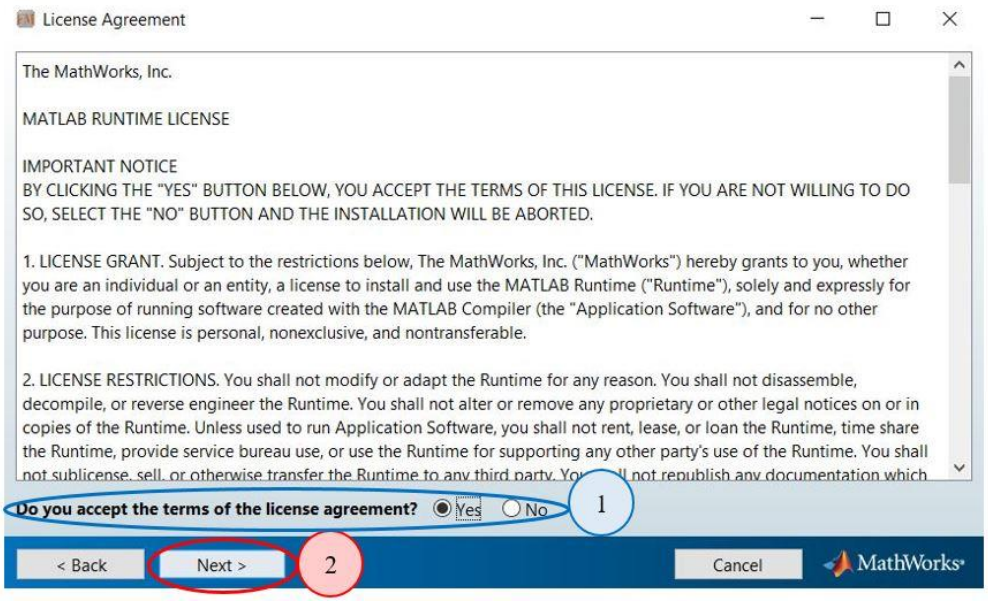

Figura II. 6. Términos y condiciones de Matlab.

Por último, antes de comenzar la instalación, se muestran los destinos que se han seleccionado para guardar los archivos de la aplicación (ver Figura II. 7). Esta página tiene un fin únicamente informativo ya que, si se desea modificar alguno de los directorios, es necesario volver a los pasos anteriores pulsando en los botones *"Back"*. En cambio, si se está de acuerdo con las rutas donde se guardarán los archivos, basta con pulsar sobre el botón *"Install"* para que comience la instalación.

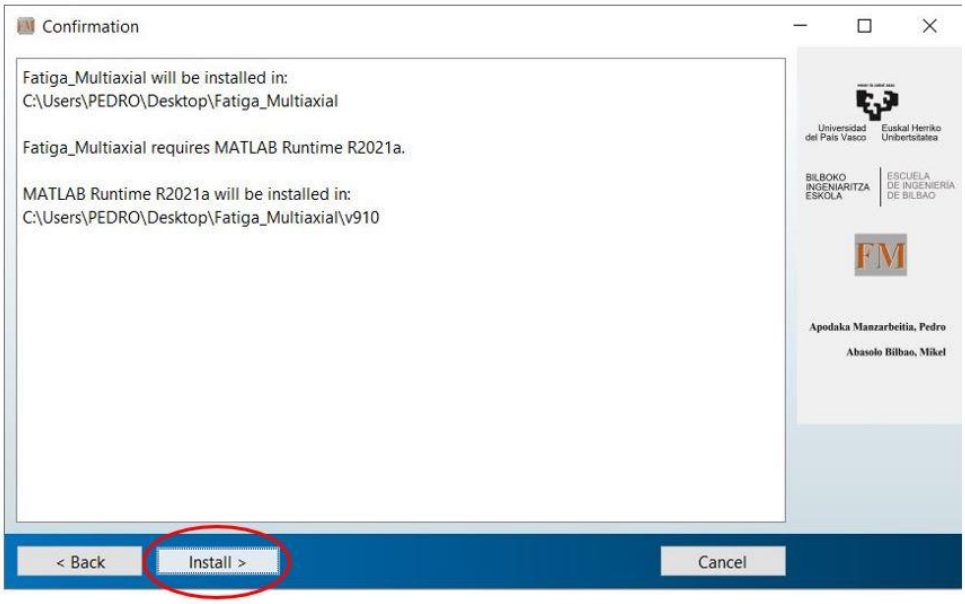

Figura II. 7. Confirmación de los destinos seleccionados para los archivos.

Una vez completada la instalación, aparecerá una página como la de la Figura II. 8 dónde, basta con clicar en el botón *"Finish"* para que concluya la instalación.

Figura II. 8. Fin de la instalación.

Por último, en el caso de haber seleccionado la opción de crear un acceso directo en el escritorio (paso 2 de la Figura II. 4), aparecerá un archivo como el que se muestra en la Figura II. 9 que permita abrir directamente la aplicación. En caso de que no se haya seleccionado dicha opción, para iniciar la aplicación, se deberá acceder desde la ubicación donde se encuentre guardada; es decir, el destino que se ha asignado en el paso 1 de la Figura II. 4.

Figura II. 9. Acceso directo generado en el escritorio.

Guía de la aplicación

Una vez completada la instalación de la aplicación, lo primero que aparece al acceder a ella es la imagen correspondiente a la portada que se muestra en la Figura II. 10. Como se puede apreciar, en la parte inferior izquierda se tiene el término *"Iniciando…"* para informar al usuario sobre el estado en el que se encuentra la app.

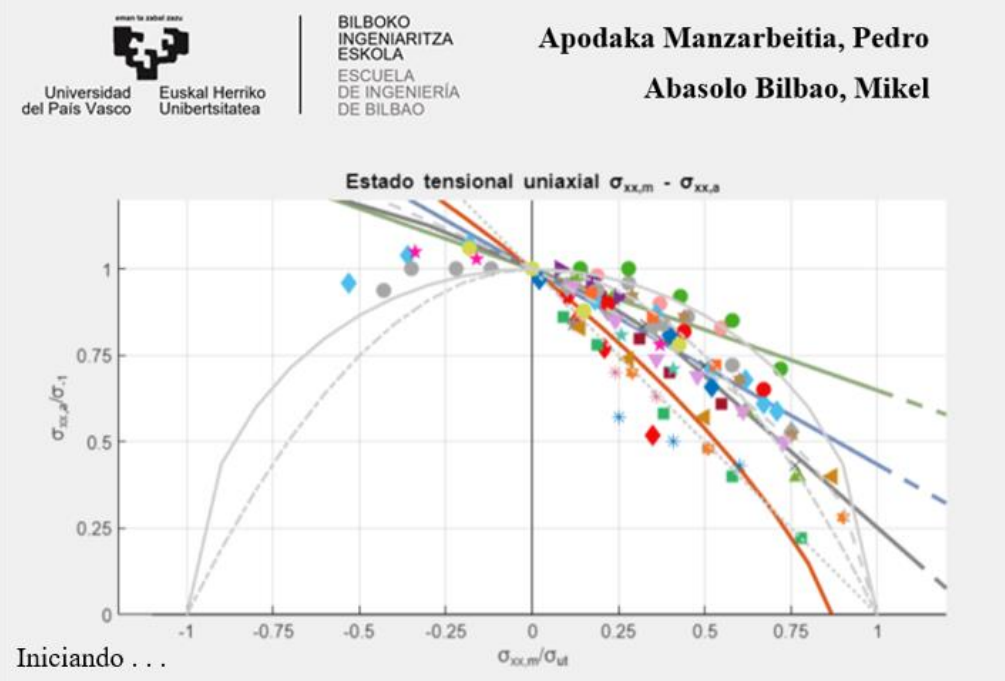

Figura II. 10. Portada de la aplicación.

Tras iniciarse la aplicación, se abre la interfaz que se muestra sobre la Figura II. 11. Tal y como se puede apreciar, la app dispone de tres zonas diferenciables que, en adelante, se denominarán como parte izquierda, parte central y parte derecha. Asimismo, la parte izquierda dispone de tres pestañas entre las que se puede alternar: Datos, Librería de Materiales y Añadir materiales. A continuación, se va a proceder a analizar cada una de dichas secciones siguiendo el orden lógico en el que se han citado.

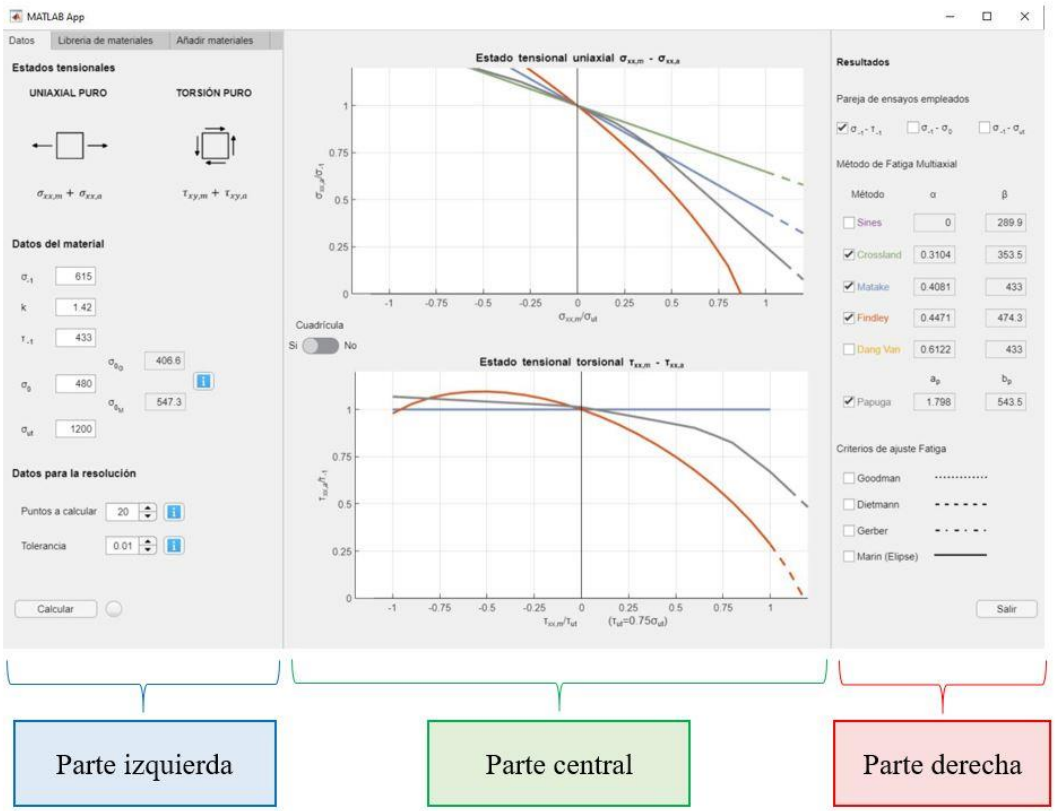

Figura II. 11. Interfaz de la aplicación.

Introducir los datos

En primer lugar, en la pestaña "Datos" de la parte izquierda se comienza mostrando las representaciones gráficas de los estados tensionales a los que se van a particularizar los métodos de fatiga multiaxial; es decir, el estado tensional uniaxial puro y de torsión puro. Tras esto y, dado que el objetivo de dicha pestaña es permitir al usuario introducir los datos con los que desee realizar los cálculos para la particularización de los métodos, se dispone de dos grupos de datos: los correspondientes al material y los requeridos para la resolución.

En cuanto a los datos del material, por defecto se dispone de los referentes al acero 34CrNiMo6, pero, éstos se pueden modificar de acurdo a los criterios que se recogen a continuación.

1) Los valores numéricos introducidos deben ser positivos. De lo contrario, la aplicación no guardará el valor introducido y mostrará un mensaje de error como el de la Figura II. 12.

Figura II. 12. Mensaje de error por no introducir datos numéricos positivos.

2) La relación entre el límite de fatiga de flexión con tensión alterna (σ_{-1}) y de torsión alterna (τ_{-1}) viene dada por el parámetro *k* (fracción entre σ_{-1} y τ_{-1}). Por lo tanto, únicamente se pueden definir dos de ellos ya que el tercero se obtiene automáticamente a partir de dicha relación.

3) Con el objetivo de emplear datos cercanos a la realidad, se fuerza a que el límite de fatiga del ensayo de tracción pulsante axial (σ_0) esté comprendido entre los valores correspondientes a los puntos de intersección de la recta a 45º con la recta de Goodman (σ_{0_G}) y la elipse de Marin(σ_{0_M}) (ver Figura II. 13). Por lo tanto, esto se expone mediante el botón de información y unido a ello, se adjuntan los valores límite del intervalo.

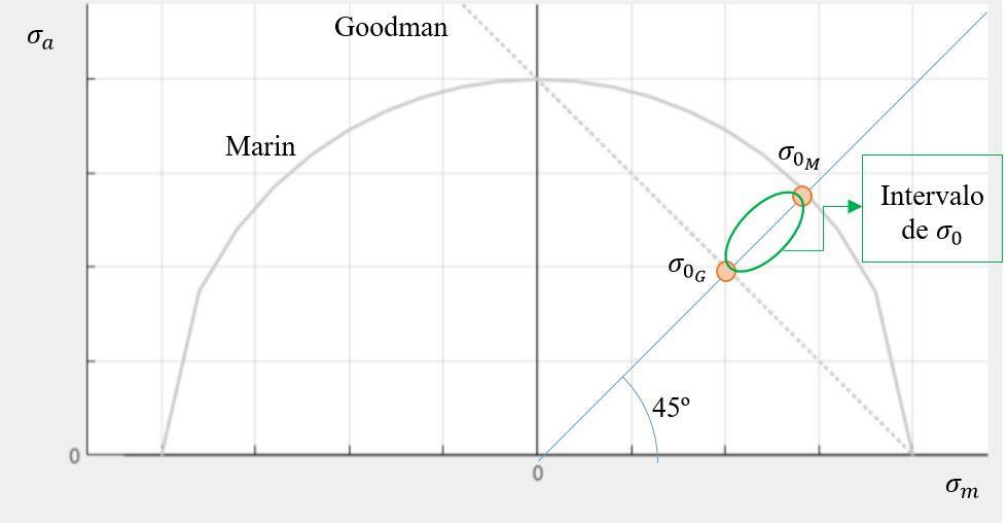

Figura II. 13. Intervalo donde se encuentra σ_0.

Por otro lado, para completar los datos necesarios se incluyen dos celdas relacionadas con la resolución de las operaciones, que se denominan: "Puntos a calcular" y "Tolerancia". Mediante la primera, "Puntos a calcular", se define el número de puntos en los que se discretiza el intervalo de la ecuación *(II.1)*. Dicho número de puntos debe estar comprendido entre 5 y 500 ya que un valor inferior en varios casos resulta poco representativo y uno superior da lugar a un coste excesivo sin conseguir una gran mejora en los resultados.

$$-1 \leq {}^{\sigma_m}/_{\sigma_{ut}} \leq 1 \qquad \text{(II.1)}$$

Por su parte, con la celda de "Tolerancia", se determina la precisión con la que se consideran aceptables los cálculos que se realizan numéricamente (método de Newton-Raphson). Inicialmente, estos valores son de 20 y 0,01 respectivamente, por su compromiso entre la calidad de los resultados y el coste computacional asociado, pero, al igual que los datos característicos del material, se pueden variar.

Tal y como se ha comentado, por defecto se dispone de los datos que aparecen en la Figura II. 11, siendo posible su modificación siempre que se cumplan los requisitos previamente mencionados. En caso de realizar algún cambio, la lámpara de la parte inferior se iluminará de color naranja (ver Figura II. 14 a)) lo cual indica que las operaciones necesarias para la particularización de los métodos están sin actualizar. Para actualizarlos, basta con pulsar sobre el botón "Calcular" y tras esto, como señal indicativa de que los cálculos han finalizado, la bombilla se iluminará de color verde; como se muestra en la Figura II. 14 b).

Figura II. 14. Colores de la lámpara: a) Naranja (cálculos sin actualizar), b) Verde (cálculos actualizados).

Resultados

Una vez introducidos los datos con los que se desean realizar los cálculos, en la parte central se van a representar los métodos que se seleccionen en la parte derecha. Así pues, en dicha parte se van a visualizar las representaciones correspondientes al caso uniaxial (gráfico superior) y al caso de torsión pura (gráfico inferior).

Junto con esto, en la parte central también se dispone de un botón denominado "Cuadrícula" con las opciones "Si" y "No". Su objetivo es que el usuario tenga la opción

de representar los resultados con una cuadrícula de fondo o sin ella; por lo que, esto no afecta de ningún modo a la propia particularización de los métodos. A modo de ejemplo, en la Figura II. 15 se muestra la misma particularización de la Figura II. 11 pero, sin la cuadrícula de fondo.

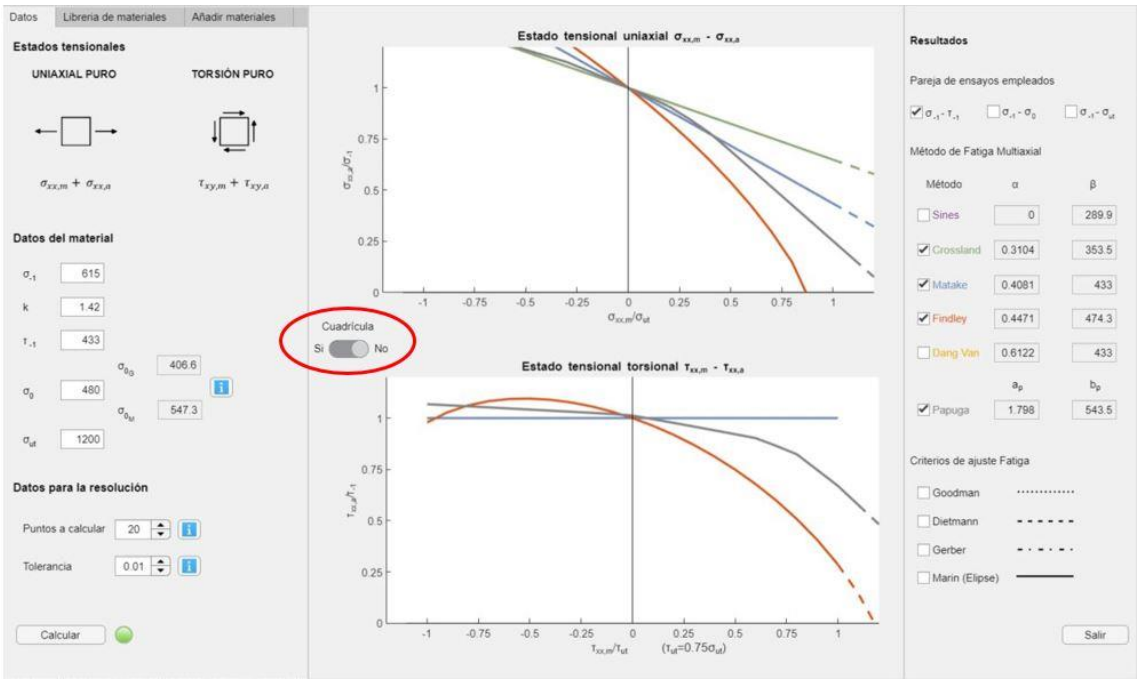

Figura II. 15. Representaciones sin cuadrícula de fondo.

En cuanto a la parte derecha de la interfaz, se pueden distinguir tres niveles de *checkboxes* o selección de casillas: pareja de ensayos empleados, método de Fatiga Multiaxial y criterios de ajuste Fatiga. Los dos primeros se basan en la fatiga multiaxial y están correlacionados mientras que, el tercer nivel hace referencia al caso uniaxial y por tanto, se puede considerar independiente de los anteriores.

En el primer nivel, se selecciona, de entre las tres opciones disponibles, la pareja de ensayos que se desea emplear para la particularización de los métodos. Por su parte, en el segundo nivel, Método de Fatiga Multiaxial, se eligen todos aquellos métodos que se desean representar en las gráficas de la parte central e independientemente de si se ha seleccionado o no, se adjuntan los parámetros propios de cada uno de ellos a su derecha. Por último, en el nivel inferior, Criterios de ajuste Fatiga, se tiene la opción de graficar los principales criterios de ajuste de fatiga con tensión media.

Junto con esto, cabe decir que en caso de tener seleccionada la pareja de ensayos $\sigma_{-1} - \tau_{-1}$, si se pulsa sobre el *checkbox* de Sines aparecerá un mensaje de advertencia como el de la Figura II. 16. Tal y como se indica en él, advierte de que no se puede particularizar

dicho método al caso uniaxial con la pareja de ensayos seleccionada; lo cual se debe a la imposibilidad de determinar el valor de α_S.

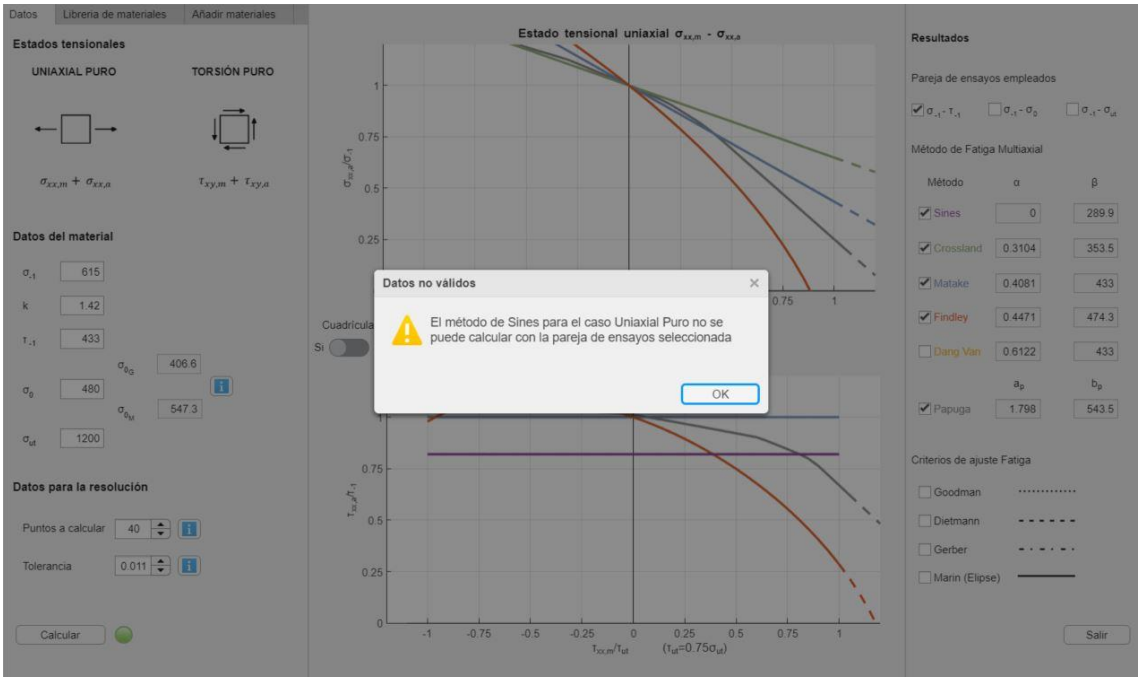

Figura II. 16. Advertencia al seleccionar Sines para la pareja de ensayos $\sigma_{-1} - \tau_{-1}$.

Por último, en la parte inferior de la parte derecha de la interfaz se tiene el botón "Salir" que, en caso de clicarlo, se abre un cuadro de dialogo como el de la Figura II. 17. Pulsando sobre "Si", se cierra la aplicación mientras que, clicando sobre "No" (opción predeterminada) se cancela la operación.

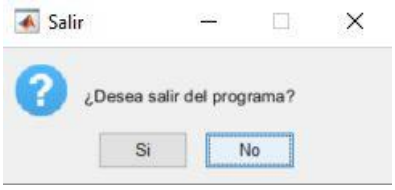

Figura II. 17. Cuadro de dialogo al pulsar sobre el botón "Salir".

Representación de puntos de fallo

Una vez seleccionados todos los métodos que se desean representar, se pueden adjuntar puntos de fallo de diferentes materiales; obviamente, obtenidos al realizar ensayos con un estado tensional uniaxial o torsional. Para ello, se dispone de las pestañas "Librería de materiales" y "Añadir materiales", ubicadas en la parte izquierda de la interfaz.

En la pestaña "Librería de materiales", se recogen una serie de puntos de fallo de diferentes materiales obtenidos de bibliografía; concretamente, 23 materiales para el caso uniaxial y 26 para el caso de torsión pura. Al igual que ocurre con los *checkboxes* de los métodos (parte derecha), se pueden seleccionar tantos materiales como se desee y, además, se tiene la opción "Select all" y "Unselect all" para seleccionar y deseleccionar todos ellos, respectivamente. Asimismo, tal y como se muestra en la Figura II. 18, cada material va acompañado de una referencia cuya explicación en detalle se puede extraer al pulsar en el botón de información "Referencias".

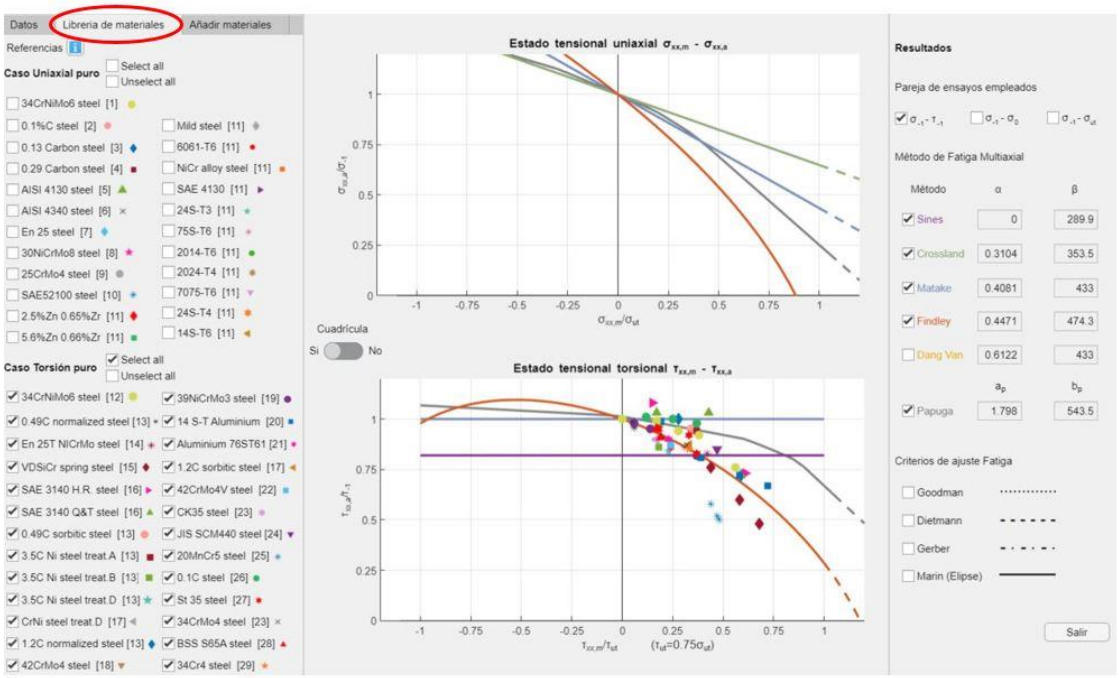

Figura II. 18. Pestaña "Librería de materiales".

Además de los materiales que se disponen en la "Librería de materiales", se tiene la opción de adjuntar nuevos materiales (máximo 22) a través de la pestaña "Añadir materiales" (ver Figura II. 19). Para ello, se debe pulsar sobre el botón "Importar Datos" y seleccionar el archivo Excel (con la plantilla que se expone más adelante) donde se tienen guardados dichos datos. Tras esto, en la parte superior se indican, a título informativo, tanto la dirección donde se encuentra el archivo como su nombre.

A la vista de la Figura II. 18 y la Figura II. 19, la interfaz de las pestañas "Librería de materiales" y "Añadir materiales" es análoga y por tanto, la manera de representar sobre las gráficas los materiales importados es semejante. Por otro lado, en caso de que se deseen eliminar los datos importados, basta con pulsar sobre el botón "Borrar datos" para retomar la posición inicial (sin materiales añadidos).

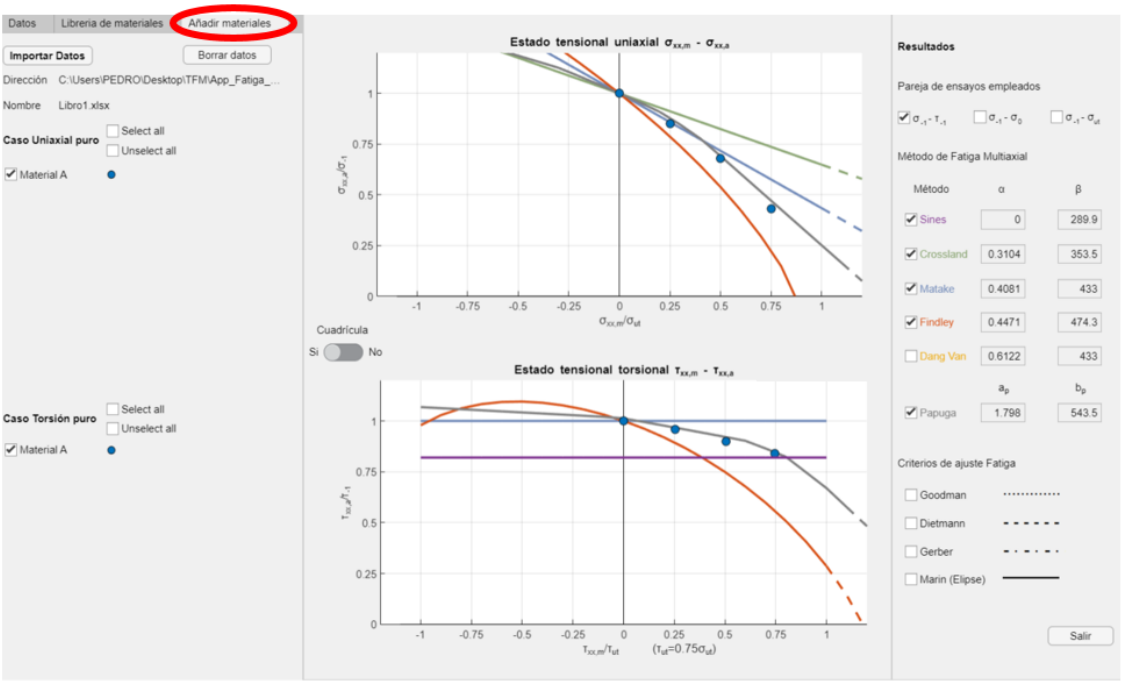

Figura II. 19. Pestaña "Añadir materiales".

Tal y como se ha comentado previamente, para poder importar los datos de fallo de nuevos materiales se debe trabajar con una plantilla predeterminada del archivo Excel. Así pues, sobre la Figura II. 20 se muestra el formato que se debe seguir para que la aplicación reconozca correctamente los datos introducidos.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Número de materiales		**NOTA:** Máximo se pueden importar 22 materiales. De ahí en adelante, no se podrán añadir más materiales.									
2	Datos de cada material											
3	**1**	Material 1			**2**	Material 2			**3**	Material 3		
4	Nombre	Material A			Nombre	Material B			Nombre	Material C		
5	σ-1				σ-1				σ-1			
6	τ-1				τ-1				τ-1			
7	σ ut				σ ut				σ ut			
8	σ-m	σ-a	τ-m	τ-a	σ-m	σ-a	τ-m	τ-a	σ-m	σ-a	τ-m	τ-a
9												
10												
11												
12												
13												
14												

Figura II. 20. Plantilla del archivo Excel.

Opciones de las representaciones

Una vez explicado lo referente al funcionamiento de la aplicación, lo siguiente va a ser detallar las diferentes opciones o comandos con los que se puede interactuar en las representaciones gráficas. Para ello, se van a tomar como referencia los cinco comandos que se remarcan sobre la representación del estado tensional torsional de la Figura II. 21.

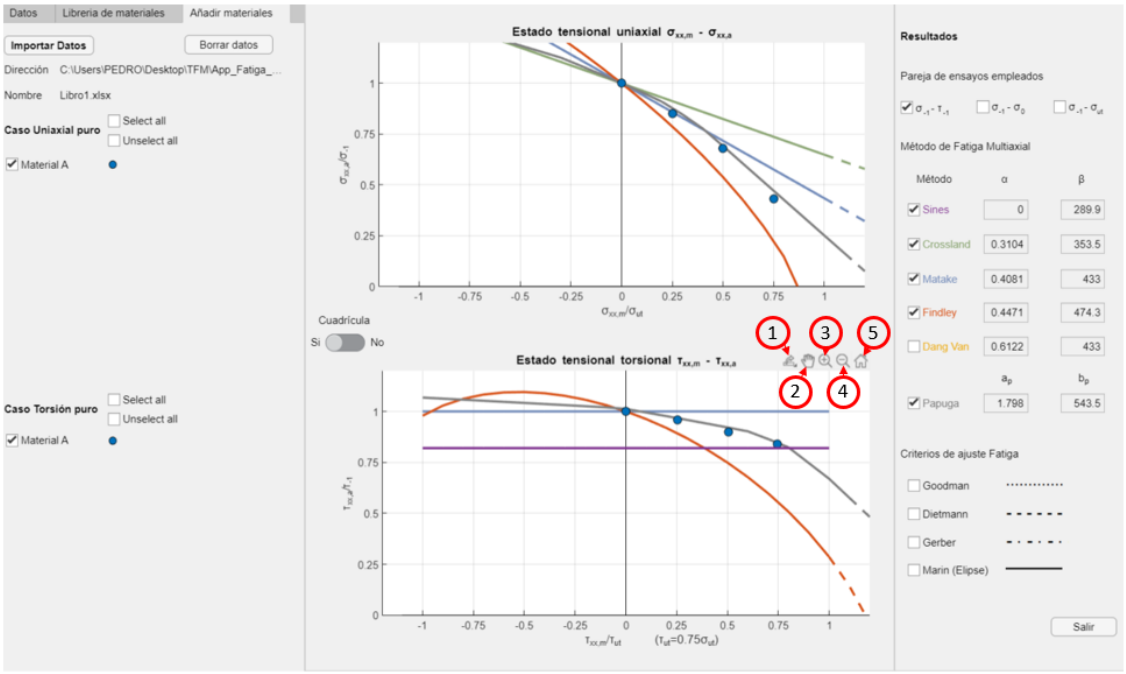

Figura II. 21. Opciones disponibles en las representaciones gráficas.

- Comando 1: "Exportar". Sirve para exportar una captura de la representación en los diferentes formatos que se muestran en la Figura II. 22.

Figura II. 22. Formatos disponibles para exportar las representaciones gráficas.

- Comando 2: "Mover". Permite mover la representación dentro del cuadro limitante.
- Comando 3: "Zoom positivo". Acerca la imagen de una determinada zona de la representación.
- Comando 4: "Zoom negativo". Aleja la imagen de la representación.
- Comando 5: "Home". Retorna a la posición inicial de la representación.

VIII. ANEXO III: Código

El objetivo de este Anexo es recoger el código que se ha empleado para la programación de la aplicación. Para ello, se van a diferenciar dos partes: el "Código principal" y las "Funciones auxiliares". En el "Código principal" se expone el núcleo de la programación en la que se basa la app mientras que, en las denominadas "funciones auxiliares" se presentan todas aquellas funciones secundarias a las que llama o se dirige el "Código principal".

Código principal

```
classdef pagina_principal < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        pag_principal                   matlab.ui.Figure
        GridLayout                      matlab.ui.container.GridLayout
        LeftPanel                       matlab.ui.container.Panel
        CenterPanel                     matlab.ui.container.Panel
        CuadrculaSwitch                 matlab.ui.control.Switch
        CuadrculaSwitchLabel            matlab.ui.control.Label
        Representacion_torsion          matlab.ui.control.UIAxes
        Representacion                  matlab.ui.control.UIAxes
        RightPanel                      matlab.ui.container.Panel
        utLabel_2                       matlab.ui.control.Label
        Label_19                        matlab.ui.control.Label
        Label_18                        matlab.ui.control.Label
        Label_17                        matlab.ui.control.Label
        Label_16                        matlab.ui.control.Label
        Label_15                        matlab.ui.control.Label
        pLabel_2                        matlab.ui.control.Label
        bLabel                          matlab.ui.control.Label
        pLabel                          matlab.ui.control.Label
        aLabel                          matlab.ui.control.Label
        b_p_Papuga                      matlab.ui.control.NumericEditField
        a_p_Papuga                      matlab.ui.control.NumericEditField
        PapugaCheckBox                  matlab.ui.control.CheckBox
        Image2_4                        matlab.ui.control.Image
        Image2_3                        matlab.ui.control.Image
        Image2_2                        matlab.ui.control.Image
        Image2                          matlab.ui.control.Image
        ResultadosLabel                 matlab.ui.control.Label
        CheckBox_3                      matlab.ui.control.CheckBox
        CheckBox_2                      matlab.ui.control.CheckBox
        ParejadeensayosempleadosLabel   matlab.ui.control.Label
        CheckBox                        matlab.ui.control.CheckBox
        Label_4                         matlab.ui.control.Label
        Label_3                         matlab.ui.control.Label
        MtodoLabel                      matlab.ui.control.Label
        BetaDangVan                     matlab.ui.control.NumericEditField
        AlfaDangVan                     matlab.ui.control.NumericEditField
        BetaFindley                     matlab.ui.control.NumericEditField
        AlfaFindley                     matlab.ui.control.NumericEditField
        BetaMatake                      matlab.ui.control.NumericEditField
        AlfaMatake                      matlab.ui.control.NumericEditField
        BetaCrossland                   matlab.ui.control.NumericEditField
        AlfaCrossland                   matlab.ui.control.NumericEditField
        BetaSines                       matlab.ui.control.NumericEditField
        AlfaSines                       matlab.ui.control.NumericEditField
        GerberCheckBox_2                matlab.ui.control.CheckBox
        SalirButton                     matlab.ui.control.Button
        CriteriosdeajusteFatigaLabel    matlab.ui.control.Label
        MtododeFatigaMultiaxialLabel    matlab.ui.control.Label
        MarinElipseCheckBox             matlab.ui.control.CheckBox
        DietmannCheckBox                matlab.ui.control.CheckBox
        GoodmanCheckBox                 matlab.ui.control.CheckBox
        DangVanCheckBox                 matlab.ui.control.CheckBox
        FindleyCheckBox                 matlab.ui.control.CheckBox
        MatakeCheckBox                  matlab.ui.control.CheckBox
        CrosslandCheckBox               matlab.ui.control.CheckBox
        SinesCheckBox                   matlab.ui.control.CheckBox
```

```
TabGroup                      matlab.ui.container.TabGroup
DatosTab                      matlab.ui.container.Tab
utLabel                       matlab.ui.control.Label
MLabel_2                      matlab.ui.control.Label
Label_13                      matlab.ui.control.Label
GLabel                        matlab.ui.control.Label
Label_12                      matlab.ui.control.Label
Label_10                      matlab.ui.control.Label
Label_8                       matlab.ui.control.Label
Label_6                       matlab.ui.control.Label
Button_4                      matlab.ui.control.Button
DatosparalaresolucinLabel     matlab.ui.control.Label
DatosdelmaterialLabel         matlab.ui.control.Label
EstadostensionalesLabel       matlab.ui.control.Label
Image_2                       matlab.ui.control.Image
TORSINPUROLabel               matlab.ui.control.Label
Image                         matlab.ui.control.Image
UNIAXIALPUROLabel             matlab.ui.control.Label
s_0_Marin                     matlab.ui.control.NumericEditField
Label_14                      matlab.ui.control.Label
s_0_Goodman                   matlab.ui.control.NumericEditField
Label_11                      matlab.ui.control.Label
EditField_5                   matlab.ui.control.NumericEditField
Label_9                       matlab.ui.control.Label
ToleranciaSpinner             matlab.ui.control.Spinner
ToleranciaSpinnerLabel        matlab.ui.control.Label
EditField_6                   matlab.ui.control.NumericEditField
EditField_6Label              matlab.ui.control.Label
EditField_4                   matlab.ui.control.NumericEditField
Label_7                       matlab.ui.control.Label
kEditField                    matlab.ui.control.NumericEditField
kEditFieldLabel               matlab.ui.control.Label
EditField_3                   matlab.ui.control.NumericEditField
Label_5                       matlab.ui.control.Label
PuntosacalcularSpinner        matlab.ui.control.Spinner
PuntosacalcularSpinnerLabel   matlab.ui.control.Label
Lamp_calcular                 matlab.ui.control.Lamp
Button_3                      matlab.ui.control.Button
Button_2                      matlab.ui.control.Button
CalcularButton                matlab.ui.control.Button
LibreriadematerialesTab       matlab.ui.container.Tab
Image5_49                     matlab.ui.control.Image
CrNiMo6steel1CheckBox         matlab.ui.control.CheckBox
Cnormalizedsteel13CheckBox_2  matlab.ui.control.CheckBox
Button                        matlab.ui.control.Button
ReferenciasLabel              matlab.ui.control.Label
Image5_48                     matlab.ui.control.Image
Image5_47                     matlab.ui.control.Image
Image5_46                     matlab.ui.control.Image
Image5_45                     matlab.ui.control.Image
Image5_44                     matlab.ui.control.Image
Image5_43                     matlab.ui.control.Image
Image5_42                     matlab.ui.control.Image
Image5_41                     matlab.ui.control.Image
Image5_40                     matlab.ui.control.Image
Image5_39                     matlab.ui.control.Image
Image5_38                     matlab.ui.control.Image
Image5_37                     matlab.ui.control.Image
Image5_36                     matlab.ui.control.Image
Image5_35                     matlab.ui.control.Image
Image5_34                     matlab.ui.control.Image
Image5_33                     matlab.ui.control.Image
Image5_32                     matlab.ui.control.Image
Image5_31                     matlab.ui.control.Image
Image5_30                     matlab.ui.control.Image
Image5_29                     matlab.ui.control.Image
Image5_28                     matlab.ui.control.Image
Image5_27                     matlab.ui.control.Image
Image5_26                     matlab.ui.control.Image
Image5_25                     matlab.ui.control.Image
Image5_24                     matlab.ui.control.Image
Image5_23                     matlab.ui.control.Image
Image5_22                     matlab.ui.control.Image
Image5_21                     matlab.ui.control.Image
Image5_20                     matlab.ui.control.Image
Image5_19                     matlab.ui.control.Image
```

```
Image5_18                       matlab.ui.control.Image
Image5_17                       matlab.ui.control.Image
Image5_16                       matlab.ui.control.Image
Image5_15                       matlab.ui.control.Image
Image5_14                       matlab.ui.control.Image
Image5_13                       matlab.ui.control.Image
Image5_12                       matlab.ui.control.Image
Image5_11                       matlab.ui.control.Image
Image5_10                       matlab.ui.control.Image
Image5_9                        matlab.ui.control.Image
Image5_8                        matlab.ui.control.Image
Image5_7                        matlab.ui.control.Image
Image5_6                        matlab.ui.control.Image
Image5_5                        matlab.ui.control.Image
Image5_4                        matlab.ui.control.Image
Image5_3                        matlab.ui.control.Image
Image5_2                        matlab.ui.control.Image
Image5                          matlab.ui.control.Image
Unselectall_torsionCheckBox     matlab.ui.control.CheckBox
Selectall_torsionCheckBox       matlab.ui.control.CheckBox
Unselectall_uniaxialCheckBox    matlab.ui.control.CheckBox
Selectall_uniaxialCheckBox      matlab.ui.control.CheckBox
CrMo4steel18CheckBox            matlab.ui.control.CheckBox
CK35steel23CheckBox             matlab.ui.control.CheckBox
CrNisteeltreatD17CheckBox       matlab.ui.control.CheckBox
CNisteeltreatD13CheckBox        matlab.ui.control.CheckBox
CNisteeltreatB13CheckBox        matlab.ui.control.CheckBox
CNisteeltreatA13CheckBox        matlab.ui.control.CheckBox
CrNiMo6steel12CheckBox          matlab.ui.control.CheckBox
VDSiCrspringsteel15CheckBox     matlab.ui.control.CheckBox
NiCrMo3steel19CheckBox          matlab.ui.control.CheckBox
MnCr5steel25CheckBox            matlab.ui.control.CheckBox
JISSCM440steel24CheckBox        matlab.ui.control.CheckBox
Cr4steel29CheckBox              matlab.ui.control.CheckBox
T611CheckBox_3                  matlab.ui.control.CheckBox
CrMo4steel9CheckBox             matlab.ui.control.CheckBox
NiCrMo8steel8CheckBox           matlab.ui.control.CheckBox
En25steel7CheckBox              matlab.ui.control.CheckBox
AISI4340steel6CheckBox          matlab.ui.control.CheckBox
AISI4130steel5CheckBox          matlab.ui.control.CheckBox
Carbonsteel4CheckBox            matlab.ui.control.CheckBox
Carbonsteel3CheckBox            matlab.ui.control.CheckBox
Csteel2CheckBox                 matlab.ui.control.CheckBox
CrMo4Vsteel22CheckBox           matlab.ui.control.CheckBox
St35steel27CheckBox             matlab.ui.control.CheckBox
CrMo4steel23CheckBox            matlab.ui.control.CheckBox
Csteel26CheckBox                matlab.ui.control.CheckBox
En25TNICrMosteel14CheckBox      matlab.ui.control.CheckBox
Aluminium76ST6121CheckBox       matlab.ui.control.CheckBox
BSSS65Asteel28CheckBox          matlab.ui.control.CheckBox
STAluminium20CheckBox           matlab.ui.control.CheckBox
SAE3140QTsteel16CheckBox        matlab.ui.control.CheckBox
SAE3140HRsteel16CheckBox        matlab.ui.control.CheckBox
Csorbiticsteel17CheckBox        matlab.ui.control.CheckBox
Csorbiticsteel13CheckBox        matlab.ui.control.CheckBox
Cnormalizedsteel13CheckBox      matlab.ui.control.CheckBox
CasoTorsinpuroLabel             matlab.ui.control.Label
CasoUniaxialpuroLabel           matlab.ui.control.Label
Zn066Zr11CheckBox               matlab.ui.control.CheckBox
Zn065Zr11CheckBox               matlab.ui.control.CheckBox
T411CheckBox                    matlab.ui.control.CheckBox
T611CheckBox_2                  matlab.ui.control.CheckBox
ST611CheckBox_2                 matlab.ui.control.CheckBox
ST311CheckBox                   matlab.ui.control.CheckBox
ST611CheckBox                   matlab.ui.control.CheckBox
ST411CheckBox                   matlab.ui.control.CheckBox
T611CheckBox                    matlab.ui.control.CheckBox
SAE413011CheckBox               matlab.ui.control.CheckBox
NiCralloysteel11CheckBox        matlab.ui.control.CheckBox
Mildsteel11CheckBox             matlab.ui.control.CheckBox
SAE52100steel10CheckBox         matlab.ui.control.CheckBox
AadirmaterialesTab              matlab.ui.container.Tab
BorrardatosButton               matlab.ui.control.Button
im_mat_14_tor                   matlab.ui.control.Image
im_mat_15_tor                   matlab.ui.control.Image
im_mat_16_tor                   matlab.ui.control.Image
```

```
im_mat_17_tor               matlab.ui.control.Image
im_mat_18_tor               matlab.ui.control.Image
im_mat_19_tor               matlab.ui.control.Image
im_mat_20_tor               matlab.ui.control.Image
im_mat_21_tor               matlab.ui.control.Image
im_mat_22_tor               matlab.ui.control.Image
im_mat_13_tor               matlab.ui.control.Image
im_mat_12_tor               matlab.ui.control.Image
im_mat_11_tor               matlab.ui.control.Image
im_mat_10_tor               matlab.ui.control.Image
im_mat_9_tor                matlab.ui.control.Image
im_mat_8_tor                matlab.ui.control.Image
im_mat_7_tor                matlab.ui.control.Image
im_mat_6_tor                matlab.ui.control.Image
im_mat_5_tor                matlab.ui.control.Image
im_mat_4_tor                matlab.ui.control.Image
im_mat_3_tor                matlab.ui.control.Image
im_mat_2_tor                matlab.ui.control.Image
im_mat_1_tor                matlab.ui.control.Image
Material13_torCheckBox      matlab.ui.control.CheckBox
Material8_torCheckBox       matlab.ui.control.CheckBox
Material7_torCheckBox       matlab.ui.control.CheckBox
Material6_torCheckBox       matlab.ui.control.CheckBox
Material5_torCheckBox       matlab.ui.control.CheckBox
Material4_torCheckBox       matlab.ui.control.CheckBox
Material3_torCheckBox       matlab.ui.control.CheckBox
Material2_torCheckBox       matlab.ui.control.CheckBox
Material1_torCheckBox       matlab.ui.control.CheckBox
Material11_torCheckBox      matlab.ui.control.CheckBox
Material10_torCheckBox      matlab.ui.control.CheckBox
Material19_torCheckBox      matlab.ui.control.CheckBox
Material18_torCheckBox      matlab.ui.control.CheckBox
Material17_torCheckBox      matlab.ui.control.CheckBox
Material16_torCheckBox      matlab.ui.control.CheckBox
Material22_torCheckBox      matlab.ui.control.CheckBox
Material21_torCheckBox      matlab.ui.control.CheckBox
Material20_torCheckBox      matlab.ui.control.CheckBox
Material15_torCheckBox      matlab.ui.control.CheckBox
Material14_torCheckBox      matlab.ui.control.CheckBox
Material12_torCheckBox      matlab.ui.control.CheckBox
Material9_torCheckBox       matlab.ui.control.CheckBox
nombre_excelLabel           matlab.ui.control.Label
NombreLabel                 matlab.ui.control.Label
direccion_excelLabel        matlab.ui.control.Label
DireccinLabel               matlab.ui.control.Label
CargandoLabel               matlab.ui.control.Label
ImportarDatosButton         matlab.ui.control.Button
im_mat_14_uni               matlab.ui.control.Image
im_mat_15_uni               matlab.ui.control.Image
im_mat_16_uni               matlab.ui.control.Image
im_mat_17_uni               matlab.ui.control.Image
im_mat_18_uni               matlab.ui.control.Image
im_mat_19_uni               matlab.ui.control.Image
im_mat_20_uni               matlab.ui.control.Image
im_mat_21_uni               matlab.ui.control.Image
im_mat_22_uni               matlab.ui.control.Image
im_mat_13_uni               matlab.ui.control.Image
im_mat_12_uni               matlab.ui.control.Image
im_mat_11_uni               matlab.ui.control.Image
im_mat_10_uni               matlab.ui.control.Image
im_mat_9_uni                matlab.ui.control.Image
im_mat_8_uni                matlab.ui.control.Image
im_mat_7_uni                matlab.ui.control.Image
im_mat_6_uni                matlab.ui.control.Image
im_mat_5_uni                matlab.ui.control.Image
im_mat_4_uni                matlab.ui.control.Image
im_mat_3_uni                matlab.ui.control.Image
im_mat_2_uni                matlab.ui.control.Image
im_mat_1_uni                matlab.ui.control.Image
Unsel_all_exc_tor           matlab.ui.control.CheckBox
Sel_all_exc_tor             matlab.ui.control.CheckBox
Unsel_all_exc_uni           matlab.ui.control.CheckBox
Sel_all_exc_uni             matlab.ui.control.CheckBox
Material13_uniCheckBox      matlab.ui.control.CheckBox
Material8_uniCheckBox       matlab.ui.control.CheckBox
Material7_uniCheckBox       matlab.ui.control.CheckBox
```

```
        Material6_uniCheckBox          matlab.ui.control.CheckBox
        Material5_uniCheckBox          matlab.ui.control.CheckBox
        Material4_uniCheckBox          matlab.ui.control.CheckBox
        Material3_uniCheckBox          matlab.ui.control.CheckBox
        Material2_uniCheckBox          matlab.ui.control.CheckBox
        Material1_uniCheckBox          matlab.ui.control.CheckBox
        CasoTorsinpuroLabel_2          matlab.ui.control.Label
        CasoUniaxialpuroLabel_2        matlab.ui.control.Label
        Material11_uniCheckBox         matlab.ui.control.CheckBox
        Material10_uniCheckBox         matlab.ui.control.CheckBox
        Material19_uniCheckBox         matlab.ui.control.CheckBox
        Material18_uniCheckBox         matlab.ui.control.CheckBox
        Material17_uniCheckBox         matlab.ui.control.CheckBox
        Material16_uniCheckBox         matlab.ui.control.CheckBox
        Material22_uniCheckBox         matlab.ui.control.CheckBox
        Material21_uniCheckBox         matlab.ui.control.CheckBox
        Material20_uniCheckBox         matlab.ui.control.CheckBox
        Material15_uniCheckBox         matlab.ui.control.CheckBox
        Material14_uniCheckBox         matlab.ui.control.CheckBox
        Material12_uniCheckBox         matlab.ui.control.CheckBox
        Material9_uniCheckBox          matlab.ui.control.CheckBox
    end

    % Properties that correspond to apps with auto-reflow
    properties (Access = private)
        onePanelWidth = 576;
        twoPanelWidth = 768;
    end

    % Callbacks that handle component events
    methods (Access = private)

        % Code that executes after component creation
       function startupFcn(app)
global lf_ax_alt
global k
global lf_tors_alt
global s_ut
global t_ut
global lf_tr_pul
global avance
global avance_t
global tolerancia
global mallado

global alfa
global beta
global s_m
global s_a
global t_m
global t_a
global puntos
global entra
global entra_t
global xmin
global xmin_t
global xmin_eje
global xmax
global xmax_t
global xmax_eje
global ymin
global ymin_t
global ymin_eje
global ymax
global ymax_t
global ymax_eje
global dib
global dib_t
global situacion
global calculo
global puntos_discretizacion
global s_0_m
global s_0_g
global x
global y
```

```
global papuga
global sit
global num_materiales

%Lectura de datos de los materiales

%34CrNiMo6 steel (uniaxial)
x.CrNiMo6_uniaxial=[-0.183;0;0.15;0.425];
y.CrNiMo6_uniaxial=[1.06;1;0.878;0.781];
%0.1% C steel
x.c_steel=[0.19;0.37;0.55];
y.c_steel=[0.98;0.9;0.83];
%0.13 carbon steel
x.carb_steel_013=[0.02;0.4;0.52];
y.carb_steel_013=[0.97;0.81;0.66];
%0.29 carbon steel
x.carb_steel_029=[0.2;0.31;0.4;0.55];
y.carb_steel_029=[0.92;0.8;0.7;0.61];
%AISI 4130 steel
x.AISI_4130=[0.13;0.23;0.62;0.77];
y.AISI_4130=[0.98;0.91;0.59;0.4];
%AISI 4340
x.AISI_4340=[0.33;0.5;0.76];
y.AISI_4340=[0.84;0.72;0.43];
%En 25 steel
x.En_25=[-0.53;-0.36;-0.18;0.18;0.36;0.52;0.62;0.67;0.71];
y.En_25=[0.96;1.04;1.07;0.91;0.87;0.71;0.68;0.61;0.59];
%30NiCrMo8 Steel
x.NiCrMo8=[-0.34;-0.16;0.37];
y.NiCrMo8=[1.05;1.03;0.78];
%25CrMo4 Steel
x.CrMo4=[-0.43;-0.35;-0.22;-0.12;0;0.19;0.28;0.38;0.45;0.58];
y.CrMo4=[0.94;1;1;1;1;0.94;0.96;0.84;0.86;0.72];
%SAE 52100 steel
x.SAE_52100=[0.25;0.41;0.51;0.6];
y.SAE_52100=[0.57;0.5;0.48;0.43];
%Mild Steel
x.Mild_Steel=[0.1;0.22;0.35;0.75];
y.Mild_Steel=[0.95;0.88;0.83;0.53];
%NiCr alloy Steel
x.NiCr_alloy=[0.17;0.35;0.53];
y.NiCr_alloy=[0.93;0.86;0.72];
%SAE 4130
x.SAE_4130=[0.08;0.17;0.25];
y.SAE_4130=[1;0.97;0.92];
%24S-T3
x.S24_T3=[0.12;0.26;0.41];
y.S24_T3=[0.84;0.81;0.71];
%75S-T6
x.S75_T6=[0.12;0.24;0.36];
y.S75_T6=[0.84;0.7;0.63];
%2014-T6
x.a_2014_T6=[0.14;0.28;0.43;0.58;0.72];
y.a_2014_T6=[1;1;0.92;0.85;0.71];
%2024-T4
x.a_2024_T4=[0.29;0.44;0.6;0.75];
y.a_2024_T4=[0.92;0.86;0.68;0.52];
%6061-T6
x.a_6061_T6=[0.22;0.44;0.67];
y.a_6061_T6=[0.9;0.82;0.65];
%7075-T6
x.a_7075_T6=[0.12;0.24;0.36;0.48;0.61;0.73];
y.a_7075_T6=[0.95;0.85;0.74;0.69;0.59;0.5];
%24S-T4
x.s_24_T4=[0.14;0.29;0.51;0.9];
y.s_24_T4=[0.87;0.7;0.48;0.28];
% %75S-T6
% x.s_75_T6=[0.12;0.24;0.42;0.73];
% y.s_75_T6=[0.83;0.71;0.62;0.47];
%14S-T6
x.s_14_T6=[0.14;0.28;0.5;0.87];
y.s_14_T6=[0.83;0.74;0.57;0.4];
%2.5%Zn 0.65%Zr
x.zn_25_zr65=[0.1;0.13;0.21;0.35];
```

```
y.zn_25_zr65=[0.92;0.85;0.77;0.52];
%5.6%Zn 0.66%Zr
x.zn_56_zr66=[0;0.09;0.19;0.38;0.58;0.78];
y.zn_56_zr66=[1;0.86;0.78;0.58;0.4;0.22];

%Materiales de torsion
%1.2C normalized steel
x.C_12norm=[0.28;0.59];
y.C_12norm=[1;0.72];
%0.49C normalized steel
x.C_49norm=0.06;
y.C_49norm=0.96;
%0.49C sorbitic steel
x.C_49sorb=0.34;
y.C_49sorb=0.95;
%1.2C sorbitic steel
x.C_12sorb=0.33;
y.C_12sorb=0.87;
%SAE 3140 H.R. steel
x.sae_3140_HR=[0.15;0.61];
y.sae_3140_HR=[1.08;0.73];
%SAE 3140 Q&T steel
x.sae_3140_QT=[0.17;0.43;0.59];
y.sae_3140_QT=[1.03;1.03;0.72];
%14 S-T Aluminium
x.st_14_Al=[0.19;0.39;0.58;0.72];
y.st_14_Al=[0.99;0.81;0.72;0.67];
%BSS S65 A steel
x.bss_s65=[0.17;0.19];
y.bss_s65=[0.95;0.91];
%Aluminium 75S T61
x.al_76=[0.23;0.24];
y.al_76=[0.9;0.86];
%En 25T NiCrMo steel
x.En_25T=[0.18;0.31;0.42];
y.En_25T=[0.96;0.86;0.83];
%0.1C steel
x.c_01=[0.12;0.25;0.37];
y.c_01=[1.01;1;0.98];
%34CrMo4 steel
x.CrMo4_steel=[0.05;0.32];
y.CrMo4_steel=[0.99;0.87];
%ST 35 steel
x.ST_35=[0.18;0.33;0.37];
y.ST_35=[0.95;0.92;0.83];
%42CrMo4V steel
x.CrMo4V=0.24;
y.CrMo4V=0.87;
%34Cr4 steel
x.Cr4_steel=0.33;
y.Cr4_steel=0.85;
%JIS SCM440 steel
x.JIS=0.47;
y.JIS=0.85;
%20MnCr5 steel
x.MnCr5=[0.44;0.47;0.48];
y.MnCr5=[0.58;0.52;0.5];
%39NiCrMo3 steel
x.NiCrMo3=[0.06;0.14;0.37];
y.NiCrMo3=[0.98;0.95;0.82];
%VDSiCr spring steel
x.VDSiCr=[0.44;0.58;0.68];
y.VDSiCr=[0.76;0.6;0.48];
%34CrNiMo6 steel
x.CrNiMo6_steel=[0.0;0.17;0.28;0.38;0.56];
y.CrNiMo6_steel=[1;0.99;0.94;0.92;0.76];
%3.5C Ni steel treat.A
x.C_35Ni_A=0.37;
y.C_35Ni_A=0.94;
%3.5C Ni steel treat.B
x.C_35Ni_B=0.18;
y.C_35Ni_B=0.86;
%3.5C Ni steel treat.D
x.C_35Ni_D=0.23;
```

```
y.C_35Ni_D=0.84;
%CrNi steel treat.D
x.CrNi_D=0.17;
y.CrNi_D=0.9;
%CK 35 steel
x.CK_35=0.17;
y.CK_35=0.9;
%42CrMo4 steel
x.Cr_Mo_42=0.18;
y.Cr_Mo_42=0.96;


%Uniaxial
situacion.CrNiMo6_uniaxial=0;
app.CrNiMo6steel1CheckBox.Value=0;
situacion.c_steel =0;
app.Csteel2CheckBox.Value=0;
situacion.carb_steel_013 =0;
app.Carbonsteel3CheckBox.Value=0;
situacion.carb_steel_029 =0;
app.Carbonsteel4CheckBox.Value=0;
situacion.AISI_4130 =0;
app.AISI4130steel5CheckBox.Value=0;
situacion.AISI_4340 =0;
app.AISI4340steel6CheckBox.Value=0;
situacion.En_25 =0;
app.En25steel7CheckBox.Value=0;
situacion.NiCrMo8 =0;
app.NiCrMo8steel8CheckBox.Value=0;
situacion.CrMo4 =0;
app.CrMo4steel9CheckBox.Value=0;
situacion.SAE_52100 =0;
app.SAE52100steel10CheckBox.Value=0;
situacion.zn_25_zr65 =0;
app.Zn065Zr11CheckBox.Value=0;
situacion.zn_56_zr66 =0;
app.Zn066Zr11CheckBox.Value=0;
situacion.Mild_Steel =0;
app.Mildsteel11CheckBox.Value=0;
situacion.a_6061_T6=0;
app.T611CheckBox_3.Value=0;
situacion.NiCr_alloy =0;
app.NiCralloysteel11CheckBox.Value=0;
situacion.SAE_4130 =0;
app.SAE413011CheckBox.Value=0;
situacion.S24_T3 =0;
app.ST311CheckBox.Value=0;
situacion.S75_T6 =0;
app.ST611CheckBox_2.Value=0;
situacion.a_2014_T6 =0;
app.T611CheckBox_2.Value=0;
situacion.a_2024_T4 =0;
app.T411CheckBox.Value=0;
situacion.a_7075_T6 =0;
app.T611CheckBox.Value=0;
situacion.s_24_T4 =0;
app.ST411CheckBox.Value=0;
situacion.s_14_T6 =0;
app.ST611CheckBox.Value=0;

%Torsion
situacion.C_12norm =0;
app.Cnormalizedsteel13CheckBox_2.Value=0;
situacion.C_49norm =0;
app.Cnormalizedsteel13CheckBox.Value=0;
situacion.En_25T =0;
app.En25TNICrMosteel14CheckBox.Value=0;
situacion.VDSiCr =0;
app.VDSiCrspringsteel15CheckBox.Value=0;
situacion.sae_3140_HR =0;
app.SAE3140HRsteel16CheckBox.Value=0;
situacion.sae_3140_QT =0;
app.SAE3140QTsteel16CheckBox.Value=0;
situacion.C_49sorb =0;
```

```
app.Csorbiticsteel13CheckBox.Value=0;
situacion.C_35Ni_A =0;
app.CNisteeltreatA13CheckBox.Value=0;
situacion.C_35Ni_B =0;
app.CNisteeltreatB13CheckBox.Value=0;
situacion.C_35Ni_D =0;
app.CNisteeltreatD13CheckBox.Value=0;
situacion.CrNi_D =0;
app.CrNisteeltreatD17CheckBox.Value=0;
situacion.CK_35 =0;
app.CK35steel23CheckBox.Value=0;
situacion.Cr_Mo_42 =0;
app.CrMo4steel18CheckBox.Value=0;
situacion.NiCrMo3 =0;
app.NiCrMo3steel19CheckBox.Value=0;
situacion.st_14_Al=0;
app.STAluminium20CheckBox.Value=0;
situacion.al_76 =0;
app.Aluminium76ST6121CheckBox.Value=0;
situacion.C_12sorb =0;
app.Csorbiticsteel17CheckBox.Value=0;
situacion.CrMo4V =0;
app.CrMo4Vsteel22CheckBox.Value=0;
situacion.CrNiMo6_steel =0;
app.CrNiMo6steel12CheckBox.Value=0;
situacion.JIS=0;
app.JISSCM440steel24CheckBox.Value=0;
situacion.MnCr5 =0;
app.MnCr5steel25CheckBox.Value=0;
situacion.c_01 =0;
app.Csteel26CheckBox.Value=0;
situacion.ST_35 =0;
app.St35steel27CheckBox.Value=0;
situacion.CrMo4_steel=0;
app.CrMo4steel23CheckBox.Value=0;
situacion.bss_s65 =0;
app.BSSS65Asteel28CheckBox.Value=0;
situacion.Cr4_steel=0;
app.Cr4steel29CheckBox.Value=0;

%Datos importados
sit.mat_1_uni=0;
app.Material1_uniCheckBox.Value=0;
sit.mat_2_uni=0;
app.Material2_uniCheckBox.Value=0;
sit.mat_3_uni=0;
app.Material3_uniCheckBox.Value=0;
sit.mat_4_uni=0;
app.Material4_uniCheckBox.Value
sit.mat_5_uni=0;
app.Material5_uniCheckBox.Value=0;
sit.mat_6_uni=0;
app.Material6_uniCheckBox.Value=0;
sit.mat_7_uni=0;
app.Material7_uniCheckBox.Value=0;
sit.mat_8_uni=0;
app.Material8_uniCheckBox.Value=0;
sit.mat_9_uni=0;
app.Material9_uniCheckBox.Value=0;
sit.mat_10_uni=0;
app.Material10_uniCheckBox.Value=0;
sit.mat_11_uni=0;
app.Material11_uniCheckBox.Value=0;
sit.mat_12_uni=0;
app.Material12_uniCheckBox.Value=0;
sit.mat_13_uni=0;
app.Material13_uniCheckBox.Value=0;
sit.mat_14_uni=0;
app.Material14_uniCheckBox.Value=0;
sit.mat_15_uni=0;
app.Material15_uniCheckBox.Value=0;
sit.mat_16_uni=0;
app.Material16_uniCheckBox.Value=0;
sit.mat_17_uni=0;
```

```
app.Material17_uniCheckBox.Value=0;
sit.mat_18_uni=0;
app.Material18_uniCheckBox.Value=0;
sit.mat_19_uni=0;
app.Material19_uniCheckBox.Value=0;
sit.mat_20_uni=0;
app.Material20_uniCheckBox.Value=0;
sit.mat_21_uni=0;
app.Material21_uniCheckBox.Value=0;
sit.mat_22_uni=0;
app.Material22_uniCheckBox.Value=0;
sit.mat_1_tor=0;
app.Material1_torCheckBox.Value=0;
sit.mat_2_tor=0;
app.Material2_torCheckBox.Value=0;
sit.mat_3_tor=0;
app.Material3_torCheckBox.Value=0;
sit.mat_4_tor=0;
app.Material4_torCheckBox.Value=0;
sit.mat_5_tor=0;
app.Material5_torCheckBox.Value=0;
sit.mat_6_tor=0;
app.Material6_torCheckBox.Value=0;
sit.mat_7_tor=0;
app.Material7_torCheckBox.Value=0;
sit.mat_8_tor=0;
app.Material8_torCheckBox.Value=0;
sit.mat_9_tor=0;
app.Material9_torCheckBox.Value=0;
sit.mat_10_tor=0;
app.Material10_torCheckBox.Value=0;
sit.mat_11_tor=0;
app.Material11_torCheckBox.Value=0;
sit.mat_12_tor=0;
app.Material12_torCheckBox.Value=0;
sit.mat_13_tor=0;
app.Material13_torCheckBox.Value=0;
sit.mat_14_tor=0;
app.Material14_torCheckBox.Value=0;
sit.mat_15_tor=0;
app.Material15_torCheckBox.Value=0;
sit.mat_16_tor=0;
app.Material16_torCheckBox.Value=0;
sit.mat_17_tor=0;
app.Material17_torCheckBox.Value=0;
sit.mat_18_tor=0;
app.Material18_torCheckBox.Value=0;
sit.mat_19_tor=0;
app.Material19_torCheckBox.Value=0;
sit.mat_20_tor=0;
app.Material20_torCheckBox.Value=0;
sit.mat_21_tor=0;
app.Material21_torCheckBox.Value=0;
sit.mat_22_tor=0;
app.Material22_torCheckBox.Value=0;

sit.select_all_exc_uni=0;
app.Sel_all_exc_uni.Value=0;
sit.unselect_all_exc_uni=0;
app.Unsel_all_exc_uni.Value=0;
sit.select_all_exc_tor=0;
app.Sel_all_exc_tor.Value=0;
sit.unselect_all_exc_tor=0;
app.Unsel_all_exc_tor.Value=0;

num_materiales=0;

    mallado='Si';
    app.CuadrculaSwitch.Value=mallado;
    %El grid se hace al final porque depede de xmax e ymax

    situacion.tau=1;
    app.CheckBox.Value=situacion.tau;
    situacion.zer=0;
```

```
app.CheckBox_2.Value=situacion.zer;
situacion.ut=0;
app.CheckBox_3.Value=situacion.ut;

situacion.sines=0;
app.SinesCheckBox.Value=situacion.sines;
situacion.crossland=1;
app.CrosslandCheckBox.Value=situacion.crossland;
situacion.matake=1;
app.MatakeCheckBox.Value=situacion.matake;
situacion.findley=1;
app.FindleyCheckBox.Value=situacion.findley;
situacion.dangvan=0;
app.DangVanCheckBox.Value=situacion.dangvan;
situacion.papuga=1;
app.PapugaCheckBox.Value=situacion.papuga;

situacion.goodman=0;
app.GoodmanCheckBox.Value=situacion.goodman;
situacion.dietmann=0;
calculo.dietmann=0;
app.DietmannCheckBox.Value=situacion.dietmann;
situacion.gerber=0;
calculo.gerber=0;
app.GerberCheckBox_2.Value=situacion.gerber;
situacion.marin=0;
calculo.marin=0;
app.MarinElipseCheckBox.Value=situacion.marin;


app.EditField_5.Value=615;
lf_ax_alt=615;
app.EditField_4.Value=433;
lf_tors_alt=433;
app.kEditField.Value=lf_ax_alt/lf_tors_alt; %sqrt(3);
k=lf_ax_alt/lf_tors_alt; %sqrt(3);
app.EditField_6.Value=1200;
s_ut=1200;
t_ut=0.75*s_ut;
app.EditField_5.Value=480;
lf_tr_pul =480;

s_0_m=sqrt((lf_ax_alt^2*s_ut^2)/(lf_ax_alt^2+s_ut^2));
app.s_0_Marin.Value=s_0_m;
s_0_g=(lf_ax_alt*s_ut)/(lf_ax_alt+s_ut);
app.s_0_Goodman.Value=s_0_g;


app.PuntosacalcularSpinner.Value=20;
puntos_discretizacion=20;
app.ToleranciaSpinner.Value=0.01;
tolerancia=0.01;

xmin_eje=-1.2;
xmin=xmin_eje;
xmax_eje=1.2;
xmax=xmax_eje;
ymin_eje=0;
ymin=ymin_eje;
ymax_eje=1.2;
ymax=ymax_eje;

xmin_t=xmin_eje;
xmax_t=xmax_eje;
ymin_t=ymin_eje;
ymax_t=ymax_eje;

%se calcula el avance
avance=round((2*s_ut)/puntos_discretizacion);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   UNIAXIAL PURO   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   SINES   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Parametros del metodo
%Ensayo s_-1--tau_-1
    %No se puede sacar alfa
beta.sines_tau=(sqrt(2)/3)*lf_ax_alt;
%Ensayo s_-1--s_0
alfa.sines_zer=sqrt(2)*((lf_ax_alt/lf_tr_pul)-1);
beta.sines_zer=sqrt(2)/3*lf_ax_alt;
%Ensayo s_-1--s_ut
alfa.sines_ut=sqrt(2)*lf_ax_alt/s_ut;
beta.sines_ut=sqrt(2)/3*lf_ax_alt;


    %Ensayo s_-1--tau_-1
        % No se puede sacar alfa
    %Ensayo s_-1--s_0
        [s_m.sines_zer,s_a.sines_zer,entra.sines_zer,puntos.sines_zer] =
calcula_sines_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.sines_zer,beta.sines_zer);
        if length(s_a.sines_zer)<=puntos_discretizacion
            entra.sines_zer=0;
        end
        %para los ejes
            if max(s_m.sines_zer(:,1))*1.2>xmax
                xmax=max(s_m.sines_zer(:,1))*1.2;
            end

            if max(s_a.sines_zer(:,1))*1.2>ymax
                ymax=max(s_a.sines_zer(:,1))*1.2;
            end
    %Ensayo s_-1--s_ut
        [s_m.sines_ut,s_a.sines_ut,entra.sines_ut,puntos.sines_ut] =
calcula_sines_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.sines_ut,beta.sines_ut);
        if length(s_a.sines_ut)<=puntos_discretizacion
            entra.sines_ut=0;
        end
        %para los ejes
            if max(s_m.sines_ut(:,1))*1.2>xmax
                xmax=max(s_m.sines_ut(:,1))*1.2;
            end

            if max(s_a.sines_ut(:,1))*1.2>ymax
                ymax=max(s_a.sines_ut(:,1))*1.2;
            end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   CROSSLAND   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Parametros del metodo
%Ensayo s_-1--tau_-1
alfa.crossland_tau=sqrt(2)*(sqrt(3)*(1/k)-1);
beta.crossland_tau=sqrt(2/3)*lf_tors_alt;
%Ensayo s_-1--s_0
alfa.crossland_zer=sqrt(2)*(lf_tr_pul-lf_ax_alt)/(lf_ax_alt-2*lf_tr_pul);
beta.crossland_zer=sqrt(2)/3*lf_ax_alt*(-lf_tr_pul)/(lf_ax_alt-2*lf_tr_pul);
%Ensayo s_-1--s_ut
alfa.crossland_ut=sqrt(2)*lf_ax_alt/(s_ut-lf_ax_alt);
beta.crossland_ut=sqrt(2)/3*(s_ut*lf_ax_alt)/(s_ut-lf_ax_alt);


    %Ensayo s_-1--tau_-1
        [s_m.crossland_tau,s_a.crossland_tau,entra.crossland_tau,puntos.crossland_tau] =
calcula_crossland_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.crossland_tau,beta.crossla
nd_tau);
        if length(s_a.crossland_tau)<=puntos_discretizacion
            entra.crossland_tau=0;
        end
        %para los ejes
            if max(s_m.crossland_tau(:,1))*1.2>xmax
                xmax=max(s_m.crossland_tau(:,1))*1.2;
            end

            if max(s_a.crossland_tau(:,1))*1.2>ymax
                ymax=max(s_a.crossland_tau(:,1))*1.2;
```

```
            end
        %Ensayo s_-1--s_0
            [s_m.crossland_zer,s_a.crossland_zer,entra.crossland_zer,puntos.crossland_zer] =
calcula_crossland_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.crossland_zer,beta.crossla
nd_zer);
            if length(s_a.crossland_zer)<=puntos_discretizacion
                entra.crossland_zer=0;
            end
            %para los ejes
                if max(s_m.crossland_zer(:,1))*1.2>xmax
                    xmax=max(s_m.crossland_zer(:,1))*1.2;
                end

                if max(s_a.crossland_zer(:,1))*1.2>ymax
                    ymax=max(s_a.crossland_zer(:,1))*1.2;
                end
        %Ensayo s_-1--s_ut
            [s_m.crossland_ut,s_a.crossland_ut,entra.crossland_ut,puntos.crossland_ut] =
calcula_crossland_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.crossland_ut,beta.crosslan
d_ut);
            if length(s_a.crossland_ut)<=puntos_discretizacion
                entra.crossland_ut=0;
            end
            %para los ejes
                if max(s_m.crossland_ut(:,1))*1.2>xmax
                    xmax=max(s_m.crossland_ut(:,1))*1.2;
                end

                if max(s_a.crossland_ut(:,1))*1.2>ymax
                    ymax=max(s_a.crossland_ut(:,1))*1.2;
                end


    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   MATAKE   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Parametros del metodo
    %Ensayo s_-1--tau_-1
    alfa.matake_tau=2*(1/k)-1;
    beta.matake_tau=lf_tors_alt;
    %Ensayo s_-1--s_0
    alfa.matake_zer=(lf_tr_pul-lf_ax_alt)/(lf_ax_alt-2*lf_tr_pul);
    beta.matake_zer=(lf_ax_alt/2)*((-lf_tr_pul)/(lf_ax_alt-2*lf_tr_pul));
    %Ensayo s_-1--s_ut
    alfa.matake_ut=-(lf_ax_alt)/(lf_ax_alt-s_ut);%(s_ut-lf_ax_alt)/(lf_ax_alt);
    beta.matake_ut=(lf_ax_alt/2)*(-s_ut/(lf_ax_alt-s_ut));%s_ut/2;


        %Ensayo s_-1--tau_-1
            [s_m.matake_tau,s_a.matake_tau,entra.matake_tau,puntos.matake_tau] =
calcula_matake_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.matake_tau,beta.matake_tau);
            if length(s_a.matake_tau)<=puntos_discretizacion
                entra.matake_tau=0;
            end
            %para los ejes
                if max(s_m.matake_tau(:,1))*1.2>xmax
                    xmax=max(s_m.matake_tau(:,1))*1.2;
                end

                if max(s_a.matake_tau(:,1))*1.2>ymax
                    ymax=max(s_a.matake_tau(:,1))*1.2;
                end
        %Ensayo s_-1--s_0
            [s_m.matake_zer,s_a.matake_zer,entra.matake_zer,puntos.matake_zer] =
calcula_matake_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.matake_zer,beta.matake_zer);
            if length(s_a.matake_zer)<=puntos_discretizacion
                entra.matake_zer=0;
            end
            %para los ejes
                if max(s_m.matake_zer(:,1))*1.2>xmax
                    xmax=max(s_m.matake_zer(:,1))*1.2;
                end

                if max(s_a.matake_zer(:,1))*1.2>ymax
                    ymax=max(s_a.matake_zer(:,1))*1.2;
```

```
                end

        %Ensayo s_-1--s_ut
            [s_m.matake_ut,s_a.matake_ut,entra.matake_ut,puntos.matake_ut] =
calcula_matake_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.matake_ut,beta.matake_ut);
            if length(s_a.matake_ut)<=puntos_discretizacion
                entra.matake_ut=0;
            end
            %para los ejes
                if max(s_m.matake_ut(:,1))*1.2>xmax
                    xmax=max(s_m.matake_ut(:,1))*1.2;
                end

                if max(s_a.matake_ut(:,1))*1.2>ymax
                    ymax=max(s_a.matake_ut(:,1))*1.2;
                end


    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    FINDLEY    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Parametros del metodo
    %Ensayo s_-1--tau_-1
    alfa.findley_tau=(2-k)/(2*sqrt(k-1));
    beta.findley_tau=lf_ax_alt/(2*sqrt(k-1));
    %Ensayo s_-1--s_0
    alfa.findley_zer=calcula_alfa_findley_zer(lf_ax_alt,lf_tr_pul, tolerancia);
    beta.findley_zer=(lf_tr_pul/2)*(sqrt(4*alfa.findley_zer^2+1)+2*alfa.findley_zer);
    %Ensayo s_-1--s_ut
    alfa.findley_ut=(lf_ax_alt/2)*(sqrt(s_ut*(s_ut-lf_ax_alt)))/(s_ut*(s_ut-lf_ax_alt));
    beta.findley_ut=(lf_ax_alt/2)*(sqrt(s_ut*(s_ut-lf_ax_alt)))/(s_ut-lf_ax_alt);


        %Ensayo s_-1--tau_-1
            [s_m.findley_tau,s_a.findley_tau,entra.findley_tau,puntos.findley_tau] =
calcula_findley_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.findley_tau,beta.findley_tau
);
            if length(s_a.findley_tau)<=puntos_discretizacion
                entra.findley_tau=0;
            end
            %para los ejes
                if max(s_m.findley_tau(:,1))*1.2>xmax
                    xmax=max(s_m.findley_tau(:,1))*1.2;
                end

                if max(s_a.findley_tau(:,1))*1.2>ymax
                    ymax=max(s_a.findley_tau(:,1))*1.2;
                end
        %Ensayo s_-1--s_0
            [s_m.findley_zer,s_a.findley_zer,entra.findley_zer,puntos.findley_zer] =
calcula_findley_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.findley_zer,beta.findley_zer
);
            if length(s_a.findley_zer)<=puntos_discretizacion
                entra.findley_zer=0;
            end
            %para los ejes
                if max(s_m.findley_zer(:,1))*1.2>xmax
                    xmax=max(s_m.findley_zer(:,1))*1.2;
                end

                if max(s_a.findley_zer(:,1))*1.2>ymax
                    ymax=max(s_a.findley_zer(:,1))*1.2;
                end
        %Ensayo s_-1--s_ut
            [s_m.findley_ut,s_a.findley_ut,entra.findley_ut,puntos.findley_ut] =
calcula_findley_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.findley_ut,beta.findley_ut);
            if length(s_a.findley_ut)<=puntos_discretizacion
                entra.findley_ut=0;
            end
            %para los ejes
                if max(s_m.findley_ut(:,1))*1.2>xmax
                    xmax=max(s_m.findley_ut(:,1))*1.2;
                end

                if max(s_a.findley_ut(:,1))*1.2>ymax
```

```
                ymax=max(s_a.findley_ut(:,1))*1.2;
            end

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   DANG VAN   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Parametros del metodo
    %Ensayo s_-1--tau_-1
    alfa.dangvan_tau=3*((1/k)-0.5);
    beta.dangvan_tau=lf_tors_alt;
    %Ensayo s_-1--s_0
    alfa.dangvan_zer=(3/2)*(lf_tr_pul-lf_ax_alt)/(lf_ax_alt-2*lf_tr_pul);
    beta.dangvan_zer=(lf_ax_alt/2)*(-lf_tr_pul)/(lf_ax_alt-2*lf_tr_pul);
    %Ensayo s_-1--s_ut
    alfa.dangvan_ut=(3/2)*(-lf_ax_alt)/(lf_ax_alt-s_ut);
    beta.dangvan_ut=0.5*((-lf_ax_alt*s_ut)/(lf_ax_alt-s_ut));


        %Ensayo s_-1--tau_-1
            [s_m.dangvan_tau,s_a.dangvan_tau,entra.dangvan_tau,puntos.dangvan_tau] =
calcula_dangvan_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.dangvan_tau,beta.dangvan_tau
);
            if length(s_a.dangvan_tau)<=puntos_discretizacion
                entra.dangvan_tau=0;
            end
            %para los ejes
                if max(s_m.dangvan_tau(:,1))*1.2>xmax
                    xmax=max(s_m.dangvan_tau(:,1))*1.2;
                end

                if max(s_a.dangvan_tau(:,1))*1.2>ymax
                    ymax=max(s_a.dangvan_tau(:,1))*1.2;
                end
        %Ensayo s_-1--s_0
            [s_m.dangvan_zer,s_a.dangvan_zer,entra.dangvan_zer,puntos.dangvan_zer] =
calcula_dangvan_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.dangvan_zer,beta.dangvan_zer
);
            if length(s_a.dangvan_zer)<=puntos_discretizacion
                entra.dangvan_zer=0;
            end
            %para los ejes
                if max(s_m.dangvan_zer(:,1))*1.2>xmax
                    xmax=max(s_m.dangvan_zer(:,1))*1.2;
                end

                if max(s_a.dangvan_zer(:,1))*1.2>ymax
                    ymax=max(s_a.dangvan_zer(:,1))*1.2;
                end
        %Ensayo s_-1--s_ut
            [s_m.dangvan_ut,s_a.dangvan_ut,entra.dangvan_ut,puntos.dangvan_ut] =
calcula_dangvan_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.dangvan_ut,beta.dangvan_ut);
            if length(s_a.dangvan_ut)<=puntos_discretizacion
                entra.dangvan_ut=0;
            end
            %para los ejes
                if max(s_m.dangvan_ut(:,1))*1.2>xmax
                    xmax=max(s_m.dangvan_ut(:,1))*1.2;
                end

                if max(s_a.dangvan_ut(:,1))*1.2>ymax
                    ymax=max(s_a.dangvan_ut(:,1))*1.2;
                end


    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   PAPUGA   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Parametros del metodo
    papuga.a_p=((4*k^2)/(4+k^2))^2;
    papuga.b_p=(8*lf_ax_alt*k^2*(4-k^2))/((4+k^2)^2);

    % Datos Papuga
    papuga.delta_angulo=20;

    %Se calcula Papuga
```

```
    [s_m.papuga,s_a.papuga,entra.papuga] =
calcula_papuga_uniaxial(puntos_discretizacion,s_ut,avance,lf_ax_alt,lf_tors_alt,lf_tr_pul,tolerancia,papuga.a_
p,papuga.b_p,papuga.delta_angulo);
    if length(s_a.papuga)<=puntos_discretizacion
        entra.papuga=0;
    end

    %para los ejes
    if max(s_m.papuga(:,1))*1.2>xmax
        xmax=max(s_m.papuga(:,1))*1.2;
    end

    if max(s_a.papuga(:,1))*1.2>ymax
        ymax=max(s_a.papuga(:,1))*1.2;
    end

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    TORSION PURO    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    avance_t=round((2*t_ut)/puntos_discretizacion);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    SINES    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        %Ensayo s_-1--tau_-1
            t_m.sines_tau=zeros(puntos_discretizacion,1);
            t_a.sines_tau=zeros(puntos_discretizacion,1);

            for ii=1:puntos_discretizacion
                if ii==1
                    t_m.sines_tau(ii)=-t_ut;
                elseif ii==puntos_discretizacion
                    t_m.sines_tau(ii)=t_ut;
                else
                    t_m.sines_tau(ii)=t_m.sines_tau(ii-1)+avance_t;
                end

                if ii~=1 && t_m.sines_tau(ii)~=0
                    if t_m.sines_tau(ii-1)<0 && t_m.sines_tau(ii)>0
                        t_m.sines_tau(ii)=0;
                    end
                end

                t_a.sines_tau(ii)=(3/sqrt(6))*beta.sines_tau;

            end
            %Se normalizan los datos
                t_m.sines_tau=t_m.sines_tau/t_ut;
                t_a.sines_tau=t_a.sines_tau/lf_tors_alt;
            %para los ejes
            if max(t_m.sines_tau(:,1))*1.2>xmax_t
                xmax_t=max(t_m.sines_tau(:,1))*1.2;
            end

            if max(t_a.sines_tau(:,1))*1.2>ymax_t
                ymax_t=max(t_a.sines_tau(:,1))*1.2;
            end

        %Ensayo s_-1--s_0
            t_m.sines_zer=t_m.sines_tau*t_ut;
            t_a.sines_zer=zeros(puntos_discretizacion,1);

            for ii=1:puntos_discretizacion
                t_a.sines_zer(ii)=(3/sqrt(6))*beta.sines_zer;
            end
            %Se normalizan los datos
                t_m.sines_zer=t_m.sines_zer/t_ut;
                t_a.sines_zer=t_a.sines_zer/lf_tors_alt;
            %para los ejes
            if max(t_m.sines_zer(:,1))*1.2>xmax_t
                xmax_t=max(t_m.sines_zer(:,1))*1.2;
            end

            if max(t_a.sines_zer(:,1))*1.2>ymax_t
```

```
            ymax_t=max(t_a.sines_zer(:,1))*1.2;
        end

    %Ensayo s_-1--s_ut
        t_m.sines_ut=t_m.sines_tau*t_ut;
        t_a.sines_ut=zeros(puntos_discretizacion,1);

        for ii=1:puntos_discretizacion
            t_a.sines_ut(ii)=(3/sqrt(6))*beta.sines_ut;
        end
        %Se normalizan los datos
            t_m.sines_ut=t_m.sines_ut/t_ut;
            t_a.sines_ut=t_a.sines_ut/lf_tors_alt;
        %para los ejes
        if max(t_m.sines_ut(:,1))*1.2>xmax_t
            xmax_t=max(t_m.sines_ut(:,1))*1.2;
        end

        if max(t_a.sines_ut(:,1))*1.2>ymax_t
            ymax_t=max(t_a.sines_ut(:,1))*1.2;
        end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   CROSSLAND   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Ensayo s_-1--tau_-1
        t_m.crossland_tau=t_m.sines_tau*t_ut;
        t_a.crossland_tau=zeros(puntos_discretizacion,1);

        for ii=1:puntos_discretizacion
            t_a.crossland_tau(ii)=(3/sqrt(6))*beta.crossland_tau;
        end
        %Se normalizan los datos
            t_m.crossland_tau=t_m.crossland_tau/t_ut;
            t_a.crossland_tau=t_a.crossland_tau/lf_tors_alt;
        %para los ejes
        if max(t_m.crossland_tau(:,1))*1.2>xmax_t
            xmax_t=max(t_m.crossland_tau(:,1))*1.2;
        end

        if max(t_a.crossland_tau(:,1))*1.2>ymax_t
            ymax_t=max(t_a.crossland_tau(:,1))*1.2;
        end

    %Ensayo s_-1--s_0
        t_m.crossland_zer=t_m.crossland_tau*t_ut;
        t_a.crossland_zer=zeros(puntos_discretizacion,1);

        for ii=1:puntos_discretizacion
            t_a.crossland_zer(ii)=(3/sqrt(6))*beta.crossland_zer;
        end
        %Se normalizan los datos
            t_m.crossland_zer=t_m.crossland_zer/t_ut;
            t_a.crossland_zer=t_a.crossland_zer/lf_tors_alt;
        %para los ejes
        if max(t_m.crossland_zer(:,1))*1.2>xmax_t
            xmax_t=max(t_m.crossland_zer(:,1))*1.2;
        end

        if max(t_a.crossland_zer(:,1))*1.2>ymax_t
            ymax_t=max(t_a.crossland_zer(:,1))*1.2;
        end

    %Ensayo s_-1--s_ut
        t_m.crossland_ut=t_m.crossland_tau*t_ut;
        t_a.crossland_ut=zeros(puntos_discretizacion,1);

        for ii=1:puntos_discretizacion
            t_a.crossland_ut(ii)=(3/sqrt(6))*beta.crossland_ut;
        end
        %Se normalizan los datos
            t_m.crossland_ut=t_m.crossland_ut/t_ut;
            t_a.crossland_ut=t_a.crossland_ut/lf_tors_alt;
        %para los ejes
        if max(t_m.crossland_ut(:,1))*1.2>xmax_t
```

```
            xmax_t=max(t_m.crossland_ut(:,1))*1.2;
        end

        if max(t_a.crossland_ut(:,1))*1.2>ymax_t
            ymax_t=max(t_a.crossland_ut(:,1))*1.2;
        end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   MATAKE   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Ensayo s_-1--tau_-1
        t_m.matake_tau=t_m.sines_tau*t_ut;
        t_a.matake_tau=zeros(puntos_discretizacion,1);

        for ii=1:puntos_discretizacion
            t_a.matake_tau(ii)=beta.matake_tau;
        end
        %Se normalizan los datos
            t_m.matake_tau=t_m.matake_tau/t_ut;
            t_a.matake_tau=t_a.matake_tau/lf_tors_alt;
        %para los ejes
        if max(t_m.matake_tau(:,1))*1.2>xmax_t
            xmax_t=max(t_m.matake_tau(:,1))*1.2;
        end

        if max(t_a.matake_tau(:,1))*1.2>ymax_t
            ymax_t=max(t_a.matake_tau(:,1))*1.2;
        end

    %Ensayo s_-1--s_0
        t_m.matake_zer=t_m.matake_tau*t_ut;
        t_a.matake_zer=zeros(puntos_discretizacion,1);

        for ii=1:puntos_discretizacion
            t_a.matake_zer(ii)=beta.matake_zer;
        end
        %Se normalizan los datos
            t_m.matake_zer=t_m.matake_zer/t_ut;
            t_a.matake_zer=t_a.matake_zer/lf_tors_alt;
        %para los ejes
        if max(t_m.matake_zer(:,1))*1.2>xmax_t
            xmax_t=max(t_m.matake_zer(:,1))*1.2;
        end

        if max(t_a.matake_zer(:,1))*1.2>ymax_t
            ymax_t=max(t_a.matake_zer(:,1))*1.2;
        end

    %Ensayo s_-1--s_ut
        t_m.matake_ut=t_m.matake_tau*t_ut;
        t_a.matake_ut=zeros(puntos_discretizacion,1);

        for ii=1:puntos_discretizacion
            t_a.matake_ut(ii)=beta.matake_ut;
        end
        %Se normalizan los datos
            t_m.matake_ut=t_m.matake_ut/t_ut;
            t_a.matake_ut=t_a.matake_ut/lf_tors_alt;
        %para los ejes
        if max(t_m.matake_ut(:,1))*1.2>xmax_t
            xmax_t=max(t_m.matake_ut(:,1))*1.2;
        end

        if max(t_a.matake_ut(:,1))*1.2>ymax_t
            ymax_t=max(t_a.matake_ut(:,1))*1.2;
        end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   FINDLEY   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Importante tener en cuenta que aqui el avance es avance_t porque t_ut<s_ut
    %Ensayo s_-1--tau_-1
        [t_m.findley_tau,t_a.findley_tau,entra_t.findley_tau,puntos.findley_tau] = 
calcula_findley_torsion(puntos_discretizacion,t_ut,avance_t,lf_tors_alt,tolerancia,alfa.findley_tau,beta.findl
ey_tau);
        if length(t_a.findley_tau)<=puntos_discretizacion
            entra_t.findley_tau=0;
        end
        %para los ejes
```

```
            if max(t_m.findley_tau(:,1))*1.2>xmax_t
                xmax_t=max(t_m.findley_tau(:,1))*1.2;
            end

            if max(t_a.findley_tau(:,1))*1.2>ymax_t
                ymax_t=max(t_a.findley_tau(:,1))*1.2;
            end
        %Ensayo s_-1--s_0
            [t_m.findley_zer,t_a.findley_zer,entra_t.findley_zer,puntos.findley_zer] =
calcula_findley_torsion(puntos_discretizacion,t_ut,avance_t,lf_tors_alt,tolerancia,alfa.findley_zer,beta.findl
ey_zer);
            if length(t_a.findley_zer)<=puntos_discretizacion
                entra_t.findley_zer=0;
            end
            %para los ejes
                if max(t_m.findley_zer(:,1))*1.2>xmax_t
                    xmax_t=max(t_m.findley_zer(:,1))*1.2;
                end

                if max(t_a.findley_zer(:,1))*1.2>ymax_t
                    ymax_t=max(t_a.findley_zer(:,1))*1.2;
                end
        %Ensayo s_-1--t_ut
            [t_m.findley_ut,t_a.findley_ut,entra_t.findley_ut,puntos.findley_ut] =
calcula_findley_torsion(puntos_discretizacion,t_ut,avance_t,lf_tors_alt,tolerancia,alfa.findley_ut,beta.findle
y_ut);
            if length(t_a.findley_ut)<=puntos_discretizacion
                entra_t.findley_ut=0;
            end
            %para los ejes
                if max(t_m.findley_ut(:,1))*1.2>xmax_t
                    xmax_t=max(t_m.findley_ut(:,1))*1.2;
                end

                if max(t_a.findley_ut(:,1))*1.2>ymax_t
                    ymax_t=max(t_a.findley_ut(:,1))*1.2;
                end

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   DANG VAN   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %Ensayo s_-1--tau_-1
            t_m.dangvan_tau=t_m.sines_tau*t_ut;
            t_a.dangvan_tau=zeros(puntos_discretizacion,1);

            for ii=1:puntos_discretizacion
                t_a.dangvan_tau(ii)=beta.dangvan_tau;
            end
            %Se normalizan los datos
                t_m.dangvan_tau=t_m.dangvan_tau/t_ut;
                t_a.dangvan_tau=t_a.dangvan_tau/lf_tors_alt;
            %para los ejes
            if max(t_m.dangvan_tau(:,1))*1.2>xmax_t
                xmax_t=max(t_m.dangvan_tau(:,1))*1.2;
            end

            if max(t_a.dangvan_tau(:,1))*1.2>ymax_t
                ymax_t=max(t_a.dangvan_tau(:,1))*1.2;
            end

        %Ensayo s_-1--s_0
            t_m.dangvan_zer=t_m.dangvan_tau*t_ut;
            t_a.dangvan_zer=zeros(puntos_discretizacion,1);

            for ii=1:puntos_discretizacion
                t_a.dangvan_zer(ii)=beta.dangvan_zer;
            end
            %Se normalizan los datos
                t_m.dangvan_zer=t_m.dangvan_zer/t_ut;
                t_a.dangvan_zer=t_a.dangvan_zer/lf_tors_alt;
            %para los ejes
            if max(t_m.dangvan_zer(:,1))*1.2>xmax_t
                xmax_t=max(t_m.dangvan_zer(:,1))*1.2;
            end

            if max(t_a.dangvan_zer(:,1))*1.2>ymax_t
```

```
                ymax_t=max(t_a.dangvan_zer(:,1))*1.2;
            end

        %Ensayo s_-1--s_ut
            t_m.dangvan_ut=t_m.dangvan_tau*t_ut;
            t_a.dangvan_ut=zeros(puntos_discretizacion,1);

            for ii=1:puntos_discretizacion
                t_a.dangvan_ut(ii)=beta.dangvan_ut;
            end
            %Se normalizan los datos
                t_m.dangvan_ut=t_m.dangvan_ut/t_ut;
                t_a.dangvan_ut=t_a.dangvan_ut/lf_tors_alt;
            %para los ejes
            if max(t_m.dangvan_ut(:,1))*1.2>xmax_t
                xmax_t=max(t_m.dangvan_ut(:,1))*1.2;
            end

            if max(t_a.dangvan_ut(:,1))*1.2>ymax_t
                ymax_t=max(t_a.dangvan_ut(:,1))*1.2;
            end



    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    PAPUGA    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Se calcula Papuga
    [t_m.papuga,t_a.papuga,entra_t.papuga] =
calcula_papuga_torsion(puntos_discretizacion,t_ut,avance_t,lf_ax_alt,lf_tors_alt,lf_tr_pul,tolerancia,papuga.a
_p,papuga.b_p,papuga.delta_angulo);
    if length(t_a.papuga)<=puntos_discretizacion
        entra_t.papuga=0;
    end

    %para los ejes
        if max(t_m.papuga(:,1))*1.2>xmax_t
            xmax_t=max(t_m.papuga(:,1))*1.2;
        end

        if max(t_a.papuga(:,1))*1.2>ymax_t
            ymax_t=max(t_a.papuga(:,1))*1.2;
        end


    %% Ahora se calculan los criterios de ajuste de fatiga uniaxial

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    DIETMANN    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %se calcula el avance para dietmann
    intervalo=s_ut-(-s_ut);
    avance_dietmann=round(intervalo/puntos_discretizacion);
    if mod(s_ut,avance_dietmann)==0
        fin_dietmann=puntos_discretizacion;
    else
        fin_dietmann=puntos_discretizacion+2;
    end

        s_m.dietmann=zeros(fin_dietmann,1);
        s_a.dietmann=zeros(fin_dietmann,1);

    n_e=0;
    for ii=1:fin_dietmann
        if ii==1
            s_m.dietmann(ii)=-s_ut;
        else
            s_m.dietmann(ii)=s_m.dietmann(ii-1)+avance_dietmann;
            if s_m.dietmann(ii)<s_ut && ii==fin_dietmann
                fin_dietmann=fin_dietmann+1;
                s_m.dietmann=[s_m.dietmann;0];
                s_a.dietmann=[s_a.dietmann;0];
            end
            if s_m.dietmann(ii)>s_ut
                s_m.dietmann(ii)=s_ut;%Porque es donde corta segun la formula
                n_e=1;
            end
```

```
    end

    if ii~=1 && s_m.dietmann(ii)~=0
        if s_m.dietmann(ii-1)<0 && s_m.dietmann(ii)>0
            s_m.dietmann(ii)=0;
        end
    end

    s_a.dietmann(ii)=lf_ax_alt*sqrt(1-(s_m.dietmann(ii)/s_ut));

end

if n_e==0
    s_m.dietmann(ii+1)=s_ut;
    s_a.dietmann(ii+1)=lf_ax_alt*sqrt(1-(s_m.dietmann(ii+1)/s_ut));
end
%Se normalizan los datos
    s_m.dietmann=s_m.dietmann/s_ut;
    s_a.dietmann=s_a.dietmann/lf_ax_alt;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   GERBER   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%se calcula el avance para gerber
intervalo=s_ut-(-s_ut);
avance_gerber=round(intervalo/puntos_discretizacion);
if mod(s_ut,avance_gerber)==0
    fin_gerber=puntos_discretizacion;
else
    fin_gerber=puntos_discretizacion+2;
end

    s_m.gerber=zeros(fin_gerber,1);
    s_a.gerber=zeros(fin_gerber,1);

n_e=0;
for ii=1:fin_gerber
    if ii==1
        s_m.gerber(ii)=-s_ut;
    else
        s_m.gerber(ii)=s_m.gerber(ii-1)+avance_gerber;
        if s_m.gerber(ii)<s_ut && ii==fin_gerber
            fin_gerber=fin_gerber+1;
            s_m.gerber=[s_m.gerber;0];
            s_a.gerber=[s_a.gerber;0];
        end
        if s_m.gerber(ii)>s_ut
            s_m.gerber(ii)=s_ut;%Porque es donde corta segun la formula
            n_e=1;
        end
    end

    if ii~=1 && s_m.gerber(ii)~=0
        if s_m.gerber(ii-1)<0 && s_m.gerber(ii)>0
            s_m.gerber(ii)=0;
        end
    end

    s_a.gerber(ii)=lf_ax_alt*(1-(s_m.gerber(ii)/s_ut)^2);

end

if n_e==0
    s_m.gerber(ii+1)=s_ut;
    s_a.gerber(ii+1)=lf_ax_alt*(1-(s_m.gerber(ii+1)/s_ut)^2);
end

%Se normalizan los datos
    s_m.gerber=s_m.gerber/s_ut;
    s_a.gerber=s_a.gerber/lf_ax_alt;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   MARIN   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%se calcula el avance para marin
intervalo=s_ut-(-s_ut);
avance_marin=round(intervalo/puntos_discretizacion);
if mod(s_ut,avance_marin)==0
```

```
    fin_marin=puntos_discretizacion;
else
    fin_marin=puntos_discretizacion+2;
end

    s_m.marin=zeros(fin_marin,1);
    s_a.marin=zeros(fin_marin,1);

n_e=0;
for ii=1:fin_marin
    if ii==1
        s_m.marin(ii)=-s_ut;
    else
        s_m.marin(ii)=s_m.marin(ii-1)+avance_marin;
        if s_m.marin(ii)<s_ut && ii==fin_marin
            fin_marin=fin_marin+1;
            s_m.marin=[s_m.marin;0];
            s_a.marin=[s_a.marin;0];
        end
        if s_m.marin(ii)>s_ut
            s_m.marin(ii)=s_ut;%Porque es donde corta segun la formula
            n_e=1;
        end
    end

    if ii~=1 && s_m.marin(ii)~=0
        if s_m.marin(ii-1)<0 && s_m.marin(ii)>0
            s_m.marin(ii)=0;
        end
    end

    s_a.marin(ii)=lf_ax_alt*sqrt(1-(s_m.marin(ii)/s_ut)^2);

end

if n_e==0
    s_m.marin(ii+1)=s_ut;
    s_a.marin(ii+1)=lf_ax_alt*sqrt(1-(s_m.marin(ii+1)/s_ut)^2);
end

%Se normalizan los datos
    s_m.marin=s_m.marin/s_ut;
    s_a.marin=s_a.marin/lf_ax_alt;

%Se hace el paralelismo con torsión

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   DIETMANN   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%se calcula el avance para dietmann
intervalo=t_ut-(-t_ut);
avance_dietmann=round(intervalo/puntos_discretizacion);
if mod(t_ut,avance_dietmann)==0
    fin_dietmann=puntos_discretizacion;
else
    fin_dietmann=puntos_discretizacion+2;
end

    t_m.dietmann=zeros(fin_dietmann,1);
    t_a.dietmann=zeros(fin_dietmann,1);

n_e=0;
for ii=1:fin_dietmann
    if ii==1
        t_m.dietmann(ii)=-t_ut;
    else
        t_m.dietmann(ii)=t_m.dietmann(ii-1)+avance_dietmann;
        if t_m.dietmann(ii)<t_ut && ii==fin_dietmann
            fin_dietmann=fin_dietmann+1;
            t_m.dietmann=[t_m.dietmann;0];
            t_a.dietmann=[t_a.dietmann;0];
        end
        if t_m.dietmann(ii)>t_ut
            t_m.dietmann(ii)=t_ut;%Porque es donde corta segun la formula
            n_e=1;
        end
```

```
    end

    if ii~=1 && t_m.dietmann(ii)~=0
        if t_m.dietmann(ii-1)<0 && t_m.dietmann(ii)>0
            t_m.dietmann(ii)=0;
        end
    end

    t_a.dietmann(ii)=lf_tors_alt*sqrt(1-(t_m.dietmann(ii)/t_ut));

end
if n_e==0
    t_m.dietmann(ii+1)=t_ut;
    t_a.dietmann(ii+1)=lf_tors_alt*sqrt(1-(t_m.dietmann(ii+1)/t_ut));
end
%Se normalizan los datos
    t_m.dietmann=t_m.dietmann/t_ut;
    t_a.dietmann=t_a.dietmann/lf_tors_alt;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   GERBER   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%se calcula el avance para gerber
intervalo=t_ut-(-t_ut);
avance_gerber=round(intervalo/puntos_discretizacion);
if mod(t_ut,avance_gerber)==0
    fin_gerber=puntos_discretizacion;
else
    fin_gerber=puntos_discretizacion+2;
end

    t_m.gerber=zeros(fin_gerber,1);
    t_a.gerber=zeros(fin_gerber,1);

n_e=0;
for ii=1:fin_gerber
    if ii==1
        t_m.gerber(ii)=-t_ut;
    else
        t_m.gerber(ii)=t_m.gerber(ii-1)+avance_gerber;
        if t_m.gerber(ii)<t_ut && ii==fin_gerber
            fin_gerber=fin_gerber+1;
            t_m.gerber=[t_m.gerber;0];
            t_a.gerber=[t_a.gerber;0];
        end
        if t_m.gerber(ii)>t_ut
            t_m.gerber(ii)=t_ut;%Porque es donde corta segun la formula
            n_e=1;
        end
    end

    if ii~=1 && t_m.gerber(ii)~=0
        if t_m.gerber(ii-1)<0 && t_m.gerber(ii)>0
            t_m.gerber(ii)=0;
        end
    end

    t_a.gerber(ii)=lf_tors_alt*(1-(t_m.gerber(ii)/t_ut)^2);

end
if n_e==0
    t_m.gerber(ii+1)=t_ut;
    t_a.gerber(ii+1)=lf_tors_alt*(1-(t_m.gerber(ii+1)/t_ut)^2);
end
%Se normalizan los datos
    t_m.gerber=t_m.gerber/t_ut;
    t_a.gerber=t_a.gerber/lf_tors_alt;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   MARIN   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%se calcula el avance para marin
intervalo=t_ut-(-t_ut);
avance_marin=round(intervalo/puntos_discretizacion);
if mod(t_ut,avance_marin)==0
    fin_marin=puntos_discretizacion;
else
    fin_marin=puntos_discretizacion+2;
```

```
    end

        t_m.marin=zeros(fin_marin,1);
        t_a.marin=zeros(fin_marin,1);

    n_e=0;
    for ii=1:fin_marin
        if ii==1
            t_m.marin(ii)=-t_ut;
        else
            t_m.marin(ii)=t_m.marin(ii-1)+avance_marin;
            if t_m.marin(ii)<t_ut && ii==fin_marin
                fin_marin=fin_marin+1;
                t_m.marin=[t_m.marin;0];
                t_a.marin=[t_a.marin;0];
            end
            if t_m.marin(ii)>t_ut
                t_m.marin(ii)=t_ut;%Porque es donde corta segun la formula
                n_e=1;
            end
        end

        if ii~=1 && t_m.marin(ii)~=0
            if t_m.marin(ii-1)<0 && t_m.marin(ii)>0
                t_m.marin(ii)=0;
            end
        end

        t_a.marin(ii)=lf_tors_alt*sqrt(1-(t_m.marin(ii)/t_ut)^2);

    end
    if n_e==0
        t_m.marin(ii+1)=t_ut;
        t_a.marin(ii+1)=lf_tors_alt*sqrt(1-(t_m.marin(ii+1)/t_ut)^2);
    end
    %Se normalizan los datos
        t_m.marin=t_m.marin/t_ut;
        t_a.marin=t_a.marin/lf_tors_alt;


    %%%%%%%%%%%%%%%%% REPRESENTACIÓN %%%%%%%%%%%%%%%%%

    % en la startupFcn se van a dibujar Crossland, Matake, Findley y Papuga

    %en axial puro

    %Crossland
    hold(app.Representacion, 'on');

    if entra.crossland_tau==1

dib.crossland_tau_1=plot(app.Representacion,s_m.crossland_tau(1:(puntos_discretizacion+1),1),s_a.crossland_tau
(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.5569,0.6588,0.4667]); %Color: Verdoso

dib.crossland_tau_2=plot(app.Representacion,s_m.crossland_tau((puntos_discretizacion+1):end,1),s_a.crossland_t
au((puntos_discretizacion+1):end,1),'--','Color',[0.5569,0.6588,0.4667],'LineWidth',2); %Color: Verdoso
    else

dib.crossland_tau=plot(app.Representacion,s_m.crossland_tau,s_a.crossland_tau,'LineWidth',2,'Color',[0.5569,0.
6588,0.4667]); %Color: Verdoso
    end

    %Matake
    if entra.matake_tau==1

dib.matake_tau_1=plot(app.Representacion,s_m.matake_tau(1:(puntos_discretizacion+1),1),s_a.matake_tau(1:(punto
s_discretizacion+1),1),'LineWidth',2,'Color',[0.4510,0.5490,0.7294]); %Color: Azul

dib.matake_tau_2=plot(app.Representacion,s_m.matake_tau((puntos_discretizacion+1):end,1),s_a.matake_tau((punto
s_discretizacion+1):end,1),'--','Color',[0.4510,0.5490,0.7294],'LineWidth',2); %Color: Azul
    else

dib.matake_tau=plot(app.Representacion,s_m.matake_tau,s_a.matake_tau,'LineWidth',2,'Color',[0.4510,0.5490,0.72
94]); %Color: Azul
```

```
    end

    %Findley
    if entra.findley_tau==1

dib.findley_tau_1=plot(app.Representacion,s_m.findley_tau(1:(puntos_discretizacion+1),1),s_a.findley_tau(1:(pu
ntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib.findley_tau_2=plot(app.Representacion,s_m.findley_tau((puntos_discretizacion+1):end,1),s_a.findley_tau((pu
ntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
    else

dib.findley_tau=plot(app.Representacion,s_m.findley_tau,s_a.findley_tau,'LineWidth',2,'Color',[0.8510,0.3255,0
.0980]); %Color: Marron
    end

    %Papuga
    if entra.papuga==1

dib.papuga_1=plot(app.Representacion,s_m.papuga(1:(puntos_discretizacion+1),1),s_a.papuga(1:(puntos_discretiza
cion+1),1),'LineWidth',2,'Color',[0.50,0.50,0.50]); %Color: Gris oscuro

dib.papuga_2=plot(app.Representacion,s_m.papuga((puntos_discretizacion+1):end,1),s_a.papuga((puntos_discretiza
cion+1):end,1),'--','Color',[0.50,0.50,0.50],'LineWidth',2); %Color: Gris oscuro
    else
        dib.papuga=plot(app.Representacion,s_m.papuga,s_a.papuga,'LineWidth',2,'Color',[0.50,0.50,0.50]);
%Color: Gris oscuro
    end

    %%
    % en torsion puro
    %%

    %Crossland
    hold(app.Representacion_torsion, 'on');


dib_t.crossland_tau=plot(app.Representacion_torsion,t_m.crossland_tau,t_a.crossland_tau,'LineWidth',2,'Color',
[0.5569,0.6588,0.4667]); %Color: Verdoso


    %Matake


dib_t.matake_tau=plot(app.Representacion_torsion,t_m.matake_tau,t_a.matake_tau,'LineWidth',2,'Color',[0.4510,0
.5490,0.7294]); %Color: Azul


    %Findley
    if entra_t.findley_tau==1

dib_t.findley_tau_1=plot(app.Representacion_torsion,t_m.findley_tau(1:(puntos_discretizacion+1),1),t_a.findley
_tau(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib_t.findley_tau_2=plot(app.Representacion_torsion,t_m.findley_tau((puntos_discretizacion+1):end,1),t_a.findl
ey_tau((puntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
    else

dib_t.findley_tau=plot(app.Representacion_torsion,t_m.findley_tau,t_a.findley_tau,'LineWidth',2,'Color',[0.851
0,0.3255,0.0980]); %Color: Marron
    end

    %Papuga
    if entra_t.papuga==1

dib_t.papuga_1=plot(app.Representacion_torsion,t_m.papuga(1:(puntos_discretizacion+1),1),t_a.papuga(1:(puntos_
discretizacion+1),1),'LineWidth',2,'Color',[0.50,0.50,0.50]); %Color: Gris oscuro

dib_t.papuga_2=plot(app.Representacion_torsion,t_m.papuga((puntos_discretizacion+1):end,1),t_a.papuga((puntos_
discretizacion+1):end,1),'--','Color',[0.50,0.50,0.50],'LineWidth',2); %Color: Gris oscuro
    else

dib_t.papuga=plot(app.Representacion_torsion,t_m.papuga,t_a.papuga,'LineWidth',2,'Color',[0.50,0.50,0.50]);
%Color: Gris oscuro
```

```
    end

    %Para que los ejes sean cuadrados
    daspect(app.Representacion,[1 1 1]);
    daspect(app.Representacion_torsion,[1 1 1]);

    app.Representacion.XLim=[xmin_eje xmax_eje];
    app.Representacion.YLim=[ymin_eje ymax_eje];

    app.Representacion_torsion.XLim=[xmin_eje xmax_eje];
    app.Representacion_torsion.YLim=[ymin_eje ymax_eje];

    %Se añaden los ejes en líneas más estrechas y color negro
    %Representacion Axial Puro
    %eje x
    dib.axial_x=plot(app.Representacion,[-1.1 xmax*1.1/1.2],[0 0],'-','LineWidth',0.4,'Color','k');

dib.axial_x_flecha=plot(app.Representacion,xmax*1.1/1.2,0,'>','LineWidth',2,'MarkerSize',2,'MarkerEdgeColor','
k','MarkerFaceColor','k');
    %eje y
    dib.axial_y=plot(app.Representacion,[0 0],[0 ymax*1.1/1.2],'-','LineWidth',0.4,'Color','k');

dib.axial_y_flecha=plot(app.Representacion,0,ymax*1.1/1.2,'^','LineWidth',2,'MarkerSize',2,'MarkerEdgeColor','
k','MarkerFaceColor','k');

    %Representacion Torsion Puro
    %eje x
    dib_t.torsion_x=plot(app.Representacion_torsion,[-1.1 xmax_t*1.1/1.2],[0 0],'-
','LineWidth',0.4,'Color','k');

dib_t.torsion_x_flecha=plot(app.Representacion_torsion,xmax_t*1.1/1.2,0,'>','LineWidth',2,'MarkerSize',2,'Mark
erEdgeColor','k','MarkerFaceColor','k');
    %eje y
    dib_t.torsion_y=plot(app.Representacion_torsion,[0 0],[0 ymax_t*1.1/1.2],'-','LineWidth',0.4,'Color','k');

dib_t.torsion_y_flecha=plot(app.Representacion_torsion,0,ymax_t*1.1/1.2,'^','LineWidth',2,'MarkerSize',2,'Mark
erEdgeColor','k','MarkerFaceColor','k');

    %Grid on
    grid(app.Representacion,'on');
    xticks(app.Representacion,(-2:0.25:round(xmax*1.2)));
    yticks(app.Representacion,(-2:0.25:round(ymax*1.2)));
    grid(app.Representacion_torsion,'on');
    xticks(app.Representacion_torsion,(-2:0.25:round(xmax_t*1.2)));
    yticks(app.Representacion_torsion,(-2:0.25:round(ymax_t*1.2)));

    %Se muestran los valores de alfa y beta (en este caso, para el ensayo s_-1--tau_-1)
    app.AlfaSines.Value=0;
    app.BetaSines.Value=beta.sines_tau;
    app.AlfaCrossland.Value=alfa.crossland_tau;
    app.BetaCrossland.Value=beta.crossland_tau;
    app.AlfaMatake.Value=alfa.matake_tau;
    app.BetaMatake.Value=beta.matake_tau;
    app.AlfaFindley.Value=alfa.findley_tau;
    app.BetaFindley.Value=beta.findley_tau;
    app.AlfaDangVan.Value=alfa.dangvan_tau;
    app.BetaDangVan.Value=beta.dangvan_tau;
    app.a_p_Papuga.Value=papuga.a_p;
    app.b_p_Papuga.Value=papuga.b_p;


        end

        % Callback function
        function ypEditFieldValueChanged(app, event)

        end

        % Callback function
        function EditFieldValueChanged(app, event)

        end

        % Callback function
```

```
function KEditFieldValueChanged(app, event)

end

% Callback function
function EditField_2ValueChanged(app, event)

end

% Callback function
function u_tEditFieldValueChanged(app, event)

end

% Callback function
function u_rEditFieldValueChanged(app, event)

end

% Callback function
function PuntosacalcularSpinnerValueChanged(app, event)

end

% Callback function
function sigma_ypEditFieldValueChanged(app, event)

end

% Callback function
function EditField_3ValueChanged(app, event)

end

% Callback function
function KEditField_2ValueChanged(app, event)

end

% Callback function
function EditField_4ValueChanged(app, event)

end

% Callback function
function u_tEditField_2ValueChanged(app, event)

end

% Callback function
function u_rEditField_2ValueChanged(app, event)

end

% Callback function
function CalcularButtonPushed(app, event)

end

% Callback function
function ToleranciaSpinnerValueChanged(app, event)

end

% Callback function
function MtodoDropDownValueChanged(app, event)

end

% Callback function
function alfaEditFieldValueChanged(app, event)

end
```

```
        % Callback function
        function betaEditFieldValueChanged(app, event)

        end

        % Callback function
        function AlfaEditFieldValueChanged(app, event)

        end

        % Callback function
        function BetaEditFieldValueChanged(app, event)

        end

        % Value changed function: CuadrculaSwitch
        function CuadrculaSwitchValueChanged(app, event)

            global mallado
            mallado = app.CuadrculaSwitch.Value;
            if mallado=='Si'
                grid(app.Representacion,'on')
                grid(app.Representacion_torsion,'on')
            else
                grid(app.Representacion,'off')
                grid(app.Representacion_torsion,'off')
            end
        end

        % Value changed function: CrosslandCheckBox
        function CrosslandCheckBoxValueChanged(app, event)

            global situacion
            global entra
            global s_m
            global s_a
            global t_m
            global t_a
            global puntos_discretizacion
            global dib
            global dib_t

            situacion.crossland = app.CrosslandCheckBox.Value;

            if situacion.crossland==1

                hold(app.Representacion,'on');
                hold(app.Representacion_torsion,'on');
                %Se dibuja crossland
                if situacion.tau==1
                    if entra.crossland_tau==1

dib.crossland_tau_1=plot(app.Representacion,s_m.crossland_tau(1:(puntos_discretizacion+1),1),s_a.crossland_tau
(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.5569,0.6588,0.4667]); %Color: Verdoso

dib.crossland_tau_2=plot(app.Representacion,s_m.crossland_tau((puntos_discretizacion+1):end,1),s_a.crossland_t
au((puntos_discretizacion+1):end,1),'--','Color',[0.5569,0.6588,0.4667],'LineWidth',2); %Color: Verdoso
                    else

dib.crossland_tau=plot(app.Representacion,s_m.crossland_tau,s_a.crossland_tau,'LineWidth',2,'Color',[0.5569,0.
6588,0.4667]); %Color: Verdoso
                    end

dib_t.crossland_tau=plot(app.Representacion_torsion,t_m.crossland_tau,t_a.crossland_tau,'LineWidth',2,'Color',
[0.5569,0.6588,0.4667]); %Color: Verdoso
                elseif situacion.zer==1
                    if entra.crossland_zer==1

dib.crossland_zer_1=plot(app.Representacion,s_m.crossland_zer(1:(puntos_discretizacion+1),1),s_a.crossland_zer
(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.5569,0.6588,0.4667]); %Color: Verdoso

dib.crossland_zer_2=plot(app.Representacion,s_m.crossland_zer((puntos_discretizacion+1):end,1),s_a.crossland_z
er((puntos_discretizacion+1):end,1),'--','Color',[0.5569,0.6588,0.4667],'LineWidth',2); %Color: Verdoso
                    else
```

```
dib.crossland_zer=plot(app.Representacion,s_m.crossland_zer,s_a.crossland_zer,'LineWidth',2,'Color',[0.5569,0.
6588,0.4667]); %Color: Verdoso
                    end

dib_t.crossland_zer=plot(app.Representacion_torsion,t_m.crossland_zer,t_a.crossland_zer,'LineWidth',2,'Color',
[0.5569,0.6588,0.4667]); %Color: Verdoso
                elseif situacion.ut==1
                    if entra.crossland_ut==1

dib.crossland_ut_1=plot(app.Representacion,s_m.crossland_ut(1:(puntos_discretizacion+1),1),s_a.crossland_ut(1:
(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.5569,0.6588,0.4667]); %Color: Verdoso

dib.crossland_ut_2=plot(app.Representacion,s_m.crossland_ut((puntos_discretizacion+1):end,1),s_a.crossland_ut(
(puntos_discretizacion+1):end,1),'--','Color',[0.5569,0.6588,0.4667],'LineWidth',2); %Color: Verdoso
                    else

dib.crossland_ut=plot(app.Representacion,s_m.crossland_ut,s_a.crossland_ut,'LineWidth',2,'Color',[0.5569,0.658
8,0.4667]); %Color: Verdoso
                    end

dib_t.crossland_ut=plot(app.Representacion_torsion,t_m.crossland_ut,t_a.crossland_ut,'LineWidth',2,'Color',[0.
5569,0.6588,0.4667]); %Color: Verdoso
                end

            else

                %Se borra
                if situacion.tau==1
                    if entra.crossland_tau==1
                        delete(dib.crossland_tau_1)
                        delete(dib.crossland_tau_2)
                    else
                        delete(dib.crossland_tau)
                    end
                    delete(dib_t.crossland_tau)
                elseif situacion.zer==1
                    if entra.crossland_zer==1
                        delete(dib.crossland_zer_1)
                        delete(dib.crossland_zer_2)
                    else
                        delete(dib.crossland_zer)
                    end
                    delete(dib_t.crossland_zer)
                elseif situacion.ut==1
                    if entra.crossland_ut==1
                        delete(dib.crossland_ut_1)
                        delete(dib.crossland_ut_2)
                    else
                        delete(dib.crossland_ut)
                    end
                    delete(dib_t.crossland_ut)
                end

            end


        end

        % Value changed function: MatakeCheckBox
        function MatakeCheckBoxValueChanged(app, event)

            global situacion
            global entra
            global s_m
            global s_a
            global t_m
            global t_a
            global puntos_discretizacion
            global dib
            global dib_t

            situacion.matake = app.MatakeCheckBox.Value;
```

```
            if situacion.matake==1

                hold(app.Representacion,'on');
                hold(app.Representacion_torsion,'on');
                %Se dibuja matake
                if situacion.tau==1
                    if entra.matake_tau==1

dib.matake_tau_1=plot(app.Representacion,s_m.matake_tau(1:(puntos_discretizacion+1),1),s_a.matake_tau(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.4510,0.5490,0.7294]); %Color: Azul

dib.matake_tau_2=plot(app.Representacion,s_m.matake_tau((puntos_discretizacion+1):end,1),s_a.matake_tau((puntos_discretizacion+1):end,1),'--','Color',[0.4510,0.5490,0.7294],'LineWidth',2); %Color: Azul
                    else

dib.matake_tau=plot(app.Representacion,s_m.matake_tau,s_a.matake_tau,'LineWidth',2,'Color',[0.4510,0.5490,0.7294]); %Color: Azul
                    end

dib_t.matake_tau=plot(app.Representacion_torsion,t_m.matake_tau,t_a.matake_tau,'LineWidth',2,'Color',[0.4510,0.5490,0.7294]); %Color: Azul
                elseif situacion.zer==1
                    if entra.matake_zer==1

dib.matake_zer_1=plot(app.Representacion,s_m.matake_zer(1:(puntos_discretizacion+1),1),s_a.matake_zer(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.4510,0.5490,0.7294]); %Color: Azul

dib.matake_zer_2=plot(app.Representacion,s_m.matake_zer((puntos_discretizacion+1):end,1),s_a.matake_zer((puntos_discretizacion+1):end,1),'--','Color',[0.4510,0.5490,0.7294],'LineWidth',2); %Color: Azul
                    else

dib.matake_zer=plot(app.Representacion,s_m.matake_zer,s_a.matake_zer,'LineWidth',2,'Color',[0.4510,0.5490,0.7294]); %Color: Azul
                    end

dib_t.matake_zer=plot(app.Representacion_torsion,t_m.matake_zer,t_a.matake_zer,'LineWidth',2,'Color',[0.4510,0.5490,0.7294]); %Color: Azul
                elseif situacion.ut==1

dib.matake_ut=plot(app.Representacion,s_m.matake_ut,s_a.matake_ut,'LineWidth',2,'Color',[0.4510,0.5490,0.7294]); %Color: Azul

dib_t.matake_ut=plot(app.Representacion_torsion,t_m.matake_ut,t_a.matake_ut,'LineWidth',2,'Color',[0.4510,0.5490,0.7294]); %Color: Azul
                end

            else

                %Se borra
                if situacion.tau==1
                    if entra.matake_tau==1
                        delete(dib.matake_tau_1)
                        delete(dib.matake_tau_2)
                    else
                        delete(dib.matake_tau)
                    end
                    delete(dib_t.matake_tau)
                elseif situacion.zer==1
                    if entra.matake_zer==1
                        delete(dib.matake_zer_1)
                        delete(dib.matake_zer_2)
                    else
                        delete(dib.matake_zer)
                    end
                    delete(dib_t.matake_zer)
                elseif situacion.ut==1
                    delete(dib.matake_ut)
                    delete(dib_t.matake_ut)
                end

            end


        end
```

```
        % Value changed function: FindleyCheckBox
        function FindleyCheckBoxValueChanged(app, event)

            global situacion
            global entra
            global entra_t
            global s_m
            global s_a
            global t_m
            global t_a
            global puntos_discretizacion
            global dib
            global dib_t

            situacion.findley = app.FindleyCheckBox.Value;


            if situacion.findley==1

                hold(app.Representacion,'on');
                hold(app.Representacion_torsion,'on');
                %Se dibuja findley
                if situacion.tau==1
                    if entra.findley_tau==1

dib.findley_tau_1=plot(app.Representacion,s_m.findley_tau(1:(puntos_discretizacion+1),1),s_a.findley_tau(1:(pu
ntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib.findley_tau_2=plot(app.Representacion,s_m.findley_tau((puntos_discretizacion+1):end,1),s_a.findley_tau((pu
ntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
                    else

dib.findley_tau=plot(app.Representacion,s_m.findley_tau,s_a.findley_tau,'LineWidth',2,'Color',[0.8510,0.3255,0
.0980]); %Color: Marron
                    end

                    if entra_t.findley_tau==1

dib_t.findley_tau_1=plot(app.Representacion_torsion,t_m.findley_tau(1:(puntos_discretizacion+1),1),t_a.findley
_tau(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib_t.findley_tau_2=plot(app.Representacion_torsion,t_m.findley_tau((puntos_discretizacion+1):end,1),t_a.findl
ey_tau((puntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
                    else

dib_t.findley_tau=plot(app.Representacion_torsion,t_m.findley_tau,t_a.findley_tau,'LineWidth',2,'Color',[0.851
0,0.3255,0.0980]); %Color: Marron
                    end

                elseif situacion.zer==1
                    if entra.findley_zer==1

dib.findley_zer_1=plot(app.Representacion,s_m.findley_zer(1:(puntos_discretizacion+1),1),s_a.findley_zer(1:(pu
ntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib.findley_zer_2=plot(app.Representacion,s_m.findley_zer((puntos_discretizacion+1):end,1),s_a.findley_zer((pu
ntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
                    else

dib.findley_zer=plot(app.Representacion,s_m.findley_zer,s_a.findley_zer,'LineWidth',2,'Color',[0.8510,0.3255,0
.0980]); %Color: Marron
                    end

                    if entra_t.findley_zer==1

dib_t.findley_zer_1=plot(app.Representacion_torsion,t_m.findley_zer(1:(puntos_discretizacion+1),1),t_a.findley
_zer(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib_t.findley_zer_2=plot(app.Representacion_torsion,t_m.findley_zer((puntos_discretizacion+1):end,1),t_a.findl
ey_zer((puntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
                    else
```

```
dib_t.findley_zer=plot(app.Representacion_torsion,t_m.findley_zer,t_a.findley_zer,'LineWidth',2,'Color',[0.851
0,0.3255,0.0980]); %Color: Marron
                    end

                elseif situacion.ut==1
                    if entra.findley_ut==1

dib.findley_ut_1=plot(app.Representacion,s_m.findley_ut(1:(puntos_discretizacion+1),1),s_a.findley_ut(1:(punto
s_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib.findley_ut_2=plot(app.Representacion,s_m.findley_ut((puntos_discretizacion+1):end,1),s_a.findley_ut((punto
s_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
                    else

dib.findley_ut=plot(app.Representacion,s_m.findley_ut,s_a.findley_ut,'LineWidth',2,'Color',[0.8510,0.3255,0.09
80]); %Color: Marron
                    end

                    if entra_t.findley_ut==1

dib_t.findley_ut_1=plot(app.Representacion_torsion,t_m.findley_ut(1:(puntos_discretizacion+1),1),t_a.findley_u
t(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib_t.findley_ut_2=plot(app.Representacion_torsion,t_m.findley_ut((puntos_discretizacion+1):end,1),t_a.findley
_ut((puntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
                    else

dib_t.findley_ut=plot(app.Representacion_torsion,t_m.findley_ut,t_a.findley_ut,'LineWidth',2,'Color',[0.8510,0
.3255,0.0980]); %Color: Marron
                    end

                end

            else

                %Se borra
                if situacion.tau==1
                    if entra.findley_tau==1
                        delete(dib.findley_tau_1)
                        delete(dib.findley_tau_2)
                    else
                        delete(dib.findley_tau)
                    end

                    if entra_t.findley_tau==1
                        delete(dib_t.findley_tau_1)
                        delete(dib_t.findley_tau_2)
                    else
                        delete(dib_t.findley_tau)
                    end

                elseif situacion.zer==1
                    if entra.findley_zer==1
                        delete(dib.findley_zer_1)
                        delete(dib.findley_zer_2)
                    else
                        delete(dib.findley_zer)
                    end

                    if entra_t.findley_zer==1
                        delete(dib_t.findley_zer_1)
                        delete(dib_t.findley_zer_2)
                    else
                        delete(dib_t.findley_zer)
                    end

                elseif situacion.ut==1
                    if entra.findley_ut==1
                        delete(dib.findley_ut_1)
                        delete(dib.findley_ut_2)
                    else
                        delete(dib.findley_ut)
                    end
```

```
                if entra_t.findley_ut==1
                    delete(dib_t.findley_ut_1)
                    delete(dib_t.findley_ut_2)
                else
                    delete(dib_t.findley_ut)
                end

            end

        end


    end

    % Value changed function: DangVanCheckBox
    function DangVanCheckBoxValueChanged(app, event)
        global situacion
        global entra
        global s_m
        global s_a
        global t_m
        global t_a
        global puntos_discretizacion
        global dib
        global dib_t

        situacion.dangvan = app.DangVanCheckBox.Value;

        if situacion.dangvan==1

            hold(app.Representacion,'on');
            hold(app.Representacion_torsion,'on');
            %Se dibuja dangvan
            if situacion.tau==1
                if entra.dangvan_tau==1

dib.dangvan_tau_1=plot(app.Representacion,s_m.dangvan_tau(1:(puntos_discretizacion+1),1),s_a.dangvan_tau(1:(pu
ntos_discretizacion+1),1),'LineWidth',2,'Color',[0.9294,0.6941,0.1255]); %Color: Naranja

dib.dangvan_tau_2=plot(app.Representacion,s_m.dangvan_tau((puntos_discretizacion+1):end,1),s_a.dangvan_tau((pu
ntos_discretizacion+1):end,1),'--','Color',[0.9294,0.6941,0.1255],'LineWidth',2); %Color: Naranja
                else

dib.dangvan_tau=plot(app.Representacion,s_m.dangvan_tau,s_a.dangvan_tau,'LineWidth',2,'Color',[0.9294,0.6941,0
.1255]); %Color: Naranja
                end

dib_t.dangvan_tau=plot(app.Representacion_torsion,t_m.dangvan_tau,t_a.dangvan_tau,'LineWidth',2,'Color',[0.929
4,0.6941,0.1255]); %Color: Naranja
            elseif situacion.zer==1
                if entra.dangvan_zer==1

dib.dangvan_zer_1=plot(app.Representacion,s_m.dangvan_zer(1:(puntos_discretizacion+1),1),s_a.dangvan_zer(1:(pu
ntos_discretizacion+1),1),'LineWidth',2,'Color',[0.9294,0.6941,0.1255]); %Color: Naranja

dib.dangvan_zer_2=plot(app.Representacion,s_m.dangvan_zer((puntos_discretizacion+1):end,1),s_a.dangvan_zer((pu
ntos_discretizacion+1):end,1),'--','Color',[0.9294,0.6941,0.1255],'LineWidth',2); %Color: Naranja
                else

dib.dangvan_zer=plot(app.Representacion,s_m.dangvan_zer,s_a.dangvan_zer,'LineWidth',2,'Color',[0.9294,0.6941,0
.1255]); %Color: Naranja
                end

dib_t.dangvan_zer=plot(app.Representacion_torsion,t_m.dangvan_zer,t_a.dangvan_zer,'LineWidth',2,'Color',[0.929
4,0.6941,0.1255]); %Color: Naranja
            elseif situacion.ut==1

dib.dangvan_ut=plot(app.Representacion,s_m.dangvan_ut,s_a.dangvan_ut,'LineWidth',2,'Color',[0.9294,0.6941,0.12
55]); %Color: Naranja

dib_t.dangvan_ut=plot(app.Representacion_torsion,t_m.dangvan_ut,t_a.dangvan_ut,'LineWidth',2,'Color',[0.9294,0
.6941,0.1255]); %Color: Naranja
            end
```

```
            else

                %Se borra
                if situacion.tau==1
                    if entra.dangvan_tau==1
                        delete(dib.dangvan_tau_1)
                        delete(dib.dangvan_tau_2)
                    else
                        delete(dib.dangvan_tau)
                    end
                    delete(dib_t.dangvan_tau)
                elseif situacion.zer==1
                    if entra.dangvan_zer==1
                        delete(dib.dangvan_zer_1)
                        delete(dib.dangvan_zer_2)
                    else
                        delete(dib.dangvan_zer)
                    end
                    delete(dib_t.dangvan_zer)
                elseif situacion.ut==1
                    delete(dib.dangvan_ut)
                    delete(dib_t.dangvan_ut)
                end

            end


        end

        % Button pushed function: SalirButton
        function SalirButtonPushed(app, event)
            respuesta=questdlg('¿Desea salir del programa?','Salir','Si','No','No');
            if strcmp(respuesta,'Si')
                close(app.pag_principal);
            end
        end

        % Value changed function: GoodmanCheckBox
        function GoodmanCheckBoxValueChanged(app, event)
            global situacion
            global dib
            global dib_t

            situacion.goodman = app.GoodmanCheckBox.Value;

            if situacion.goodman==1
                hold(app.Representacion,'on');
                hold(app.Representacion_torsion,'on');
                dib.goodman=plot(app.Representacion,[-1 1],[2
0],':','LineWidth',1.5,'Color',[0.8000,0.8000,0.8000]); %Color: Gris
                dib_t.goodman=plot(app.Representacion_torsion,[-1 1],[2
0],':','LineWidth',1.5,'Color',[0.8000,0.8000,0.8000]); %Color: Gris

            else
                %Se borra
                delete(dib.goodman)
                delete(dib_t.goodman)
            end
        end

        % Callback function
        function SoderbergCheckBoxValueChanged(app, event)

        end

        % Callback function
        function MorrowCheckBoxValueChanged(app, event)

        end

        % Value changed function: DietmannCheckBox
        function DietmannCheckBoxValueChanged(app, event)
```

```
            global situacion
            global dib
            global s_m
            global s_a
            global t_m
            global t_a
            global dib_t

            situacion.dietmann = app.DietmannCheckBox.Value;

            if situacion.dietmann==1
                hold(app.Representacion,'on');
                hold(app.Representacion_torsion,'on');
                dib.dietmann=plot(app.Representacion,s_m.dietmann,s_a.dietmann,'--
','LineWidth',1.5,'Color',[0.8000,0.8000,0.8000]); %Color: Gris
                dib_t.dietmann=plot(app.Representacion_torsion,t_m.dietmann,t_a.dietmann,'--
','LineWidth',1.5,'Color',[0.8000,0.8000,0.8000]); %Color: Gris

            else
                %Se borra
                delete(dib.dietmann)
                delete(dib_t.dietmann)
            end

        end

        % Callback function
        function ButtonPushed(app, event)

        end

        % Callback function
        function Button_2Pushed(app, event)

        end

        % Callback function
        function EditField_5ValueChanged(app, event)

        end

        % Value changed function: GerberCheckBox_2
        function GerberCheckBox_2ValueChanged(app, event)

            global situacion
            global dib
            global s_m
            global s_a
            global t_m
            global t_a
            global dib_t

            situacion.gerber = app.GerberCheckBox_2.Value;

            if situacion.gerber==1
                hold(app.Representacion,'on');
                hold(app.Representacion_torsion,'on');
                dib.gerber=plot(app.Representacion,s_m.gerber,s_a.gerber,'-
.','LineWidth',1.5,'Color',[0.8000,0.8000,0.8000]); %Color: Gris
                dib_t.gerber=plot(app.Representacion_torsion,t_m.gerber,t_a.gerber,'-
.','LineWidth',1.5,'Color',[0.8000,0.8000,0.8000]); %Color: Gris
            else
                %Se borra
                delete(dib.gerber)
                delete(dib_t.gerber)
            end

        end

        % Value changed function: MarinElipseCheckBox
        function MarinElipseCheckBoxValueChanged(app, event)

            global situacion
            global dib
```

```
            global s_m
            global s_a
            global t_m
            global t_a
            global dib_t

            situacion.marin = app.MarinElipseCheckBox.Value;

            if situacion.marin==1
                hold(app.Representacion,'on');
                hold(app.Representacion_torsion,'on');
                dib.marin=plot(app.Representacion,s_m.marin,s_a.marin,'-
','LineWidth',1.5,'Color',[0.8000,0.8000,0.8000]); %Color: Gris
                dib_t.marin=plot(app.Representacion_torsion,t_m.marin,t_a.marin,'-
','LineWidth',1.5,'Color',[0.8000,0.8000,0.8000]); %Color: Gris
            else
                %Se borra
                delete(dib.marin)
                delete(dib_t.marin)
            end

        end

        % Callback function
        function Button_6Pushed(app, event)

        end

        % Value changed function: CheckBox
        function CheckBoxValueChanged(app, event)

            global situacion
            global entra
            global entra_t
            global dib
            global s_m
            global s_a
            global dib_t
            global t_m
            global t_a
            global puntos_discretizacion
            global alfa
            global beta

            situacion.tau = app.CheckBox.Value;

            if situacion.tau==1

                %Se borran los graficos anteriores
                %sines
                if situacion.sines==1
                    if situacion.zer==1
                        if entra.sines_zer==1
                            delete(dib.sines_zer_1)
                            delete(dib.sines_zer_2)
                        else
                            delete(dib.sines_zer)
                        end
                        delete(dib_t.sines_zer)
                    elseif situacion.ut==1
                        if entra.sines_ut==1
                            delete(dib.sines_ut_1)
                            delete(dib.sines_ut_2)
                        else
                            delete(dib.sines_ut)
                        end
                        delete(dib_t.sines_ut)
                    end
                end
                %Crossland
                if situacion.crossland==1
                    if situacion.zer==1
                        if entra.crossland_zer==1
                            delete(dib.crossland_zer_1)
```

```
                delete(dib.crossland_zer_2)
            else
                delete(dib.crossland_zer)
            end
            delete(dib_t.crossland_zer)
        elseif situacion.ut==1
            if entra.crossland_ut==1
                delete(dib.crossland_ut_1)
                delete(dib.crossland_ut_2)
            else
                delete(dib.crossland_ut)
            end
            delete(dib_t.crossland_ut)
        end
    end
    %matake
    if situacion.matake==1
        if situacion.zer==1
            if entra.matake_zer==1
                delete(dib.matake_zer_1)
                delete(dib.matake_zer_2)
            else
                delete(dib.matake_zer)
            end
            delete(dib_t.matake_zer)
        elseif situacion.ut==1
            delete(dib.matake_ut)
            delete(dib_t.matake_ut)
        end

    end
    %findley
    if situacion.findley==1
        if situacion.zer==1
            if entra.findley_zer==1
                delete(dib.findley_zer_1)
                delete(dib.findley_zer_2)
            else
                delete(dib.findley_zer)
            end

            if entra_t.findley_zer==1
                delete(dib_t.findley_zer_1)
                delete(dib_t.findley_zer_2)
            else
                delete(dib_t.findley_zer)
            end

        elseif situacion.ut==1
            if entra.findley_ut==1
                delete(dib.findley_ut_1)
                delete(dib.findley_ut_2)
            else
                delete(dib.findley_ut)
            end

            if entra_t.findley_ut==1
                delete(dib_t.findley_ut_1)
                delete(dib_t.findley_ut_2)
            else
                delete(dib_t.findley_ut)
            end

        end

    end
    %dangvan
    if situacion.dangvan==1
        if situacion.zer==1
            if entra.dangvan_zer==1
                delete(dib.dangvan_zer_1)
                delete(dib.dangvan_zer_2)
            else
                delete(dib.dangvan_zer)
```

```
                    end
                    delete(dib_t.dangvan_zer)
                elseif situacion.ut==1
                    delete(dib.dangvan_ut)
                    delete(dib_t.dangvan_ut)
                end
            end

            situacion.zer=0;
            app.CheckBox_2.Value=0;
            situacion.ut=0;
            app.CheckBox_3.Value=0;
            nuevo=1;

        elseif situacion.tau==0 && situacion.zer==0 && situacion.ut==0
            situacion.tau=1;
            app.CheckBox.Value=situacion.tau;
            nuevo=0;
        end


        if situacion.tau==1 && nuevo==1
            hold (app.Representacion,'off');
            hold (app.Representacion_torsion,'off');

            hold (app.Representacion,'on');
            hold (app.Representacion_torsion,'on');
            %Para que los ejes sean cuadrados
            daspect(app.Representacion,[1 1 1]);
            daspect(app.Representacion_torsion,[1 1 1]);
            %Sines
            if situacion.sines==1
%                 situacion.sines=0;
%                 app.SinesCheckBox.Value=situacion.sines;

dib_t.sines_tau=plot(app.Representacion_torsion,t_m.sines_tau,t_a.sines_tau,'LineWidth',2,'Color',[0.4941,0.18
43,0.5569]); %Color: Morado
            end

            %Crossland
            if situacion.crossland==1
                if entra.crossland_tau==1

dib.crossland_tau_1=plot(app.Representacion,s_m.crossland_tau(1:(puntos_discretizacion+1),1),s_a.crossland_tau
(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.5569,0.6588,0.4667]); %Color: Verdoso

dib.crossland_tau_2=plot(app.Representacion,s_m.crossland_tau((puntos_discretizacion+1):end,1),s_a.crossland_t
au((puntos_discretizacion+1):end,1),'--','Color',[0.5569,0.6588,0.4667],'LineWidth',2); %Color: Verdoso
                else

dib.crossland_tau=plot(app.Representacion,s_m.crossland_tau,s_a.crossland_tau,'LineWidth',2,'Color',[0.5569,0.
6588,0.4667]); %Color: Verdoso
                end

dib_t.crossland_tau=plot(app.Representacion_torsion,t_m.crossland_tau,t_a.crossland_tau,'LineWidth',2,'Color',
[0.5569,0.6588,0.4667]); %Color: Verdoso
            end

            %Matake
            if situacion.matake==1
                if entra.matake_tau==1

dib.matake_tau_1=plot(app.Representacion,s_m.matake_tau(1:(puntos_discretizacion+1),1),s_a.matake_tau(1:(punto
s_discretizacion+1),1),'LineWidth',2,'Color',[0.4510,0.5490,0.7294]); %Color: Azul

dib.matake_tau_2=plot(app.Representacion,s_m.matake_tau((puntos_discretizacion+1):end,1),s_a.matake_tau((punto
s_discretizacion+1):end,1),'--','Color',[0.4510,0.5490,0.7294],'LineWidth',2); %Color: Azul
                else

dib.matake_tau=plot(app.Representacion,s_m.matake_tau,s_a.matake_tau,'LineWidth',2,'Color',[0.4510,0.5490,0.72
94]); %Color: Azul
                end
```

```
dib_t.matake_tau=plot(app.Representacion_torsion,t_m.matake_tau,t_a.matake_tau,'LineWidth',2,'Color',[0.4510,0
.5490,0.7294]); %Color: Azul
                end

                %Findley
                if situacion.findley==1
                    if entra.findley_tau==1

dib.findley_tau_1=plot(app.Representacion,s_m.findley_tau(1:(puntos_discretizacion+1),1),s_a.findley_tau(1:(pu
ntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib.findley_tau_2=plot(app.Representacion,s_m.findley_tau((puntos_discretizacion+1):end,1),s_a.findley_tau((pu
ntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
                    else

dib.findley_tau=plot(app.Representacion,s_m.findley_tau,s_a.findley_tau,'LineWidth',2,'Color',[0.8510,0.3255,0
.0980]); %Color: Marron
                    end

                    if entra_t.findley_tau==1

dib_t.findley_tau_1=plot(app.Representacion_torsion,t_m.findley_tau(1:(puntos_discretizacion+1),1),t_a.findley
_tau(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib_t.findley_tau_2=plot(app.Representacion_torsion,t_m.findley_tau((puntos_discretizacion+1):end,1),t_a.findl
ey_tau((puntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
                    else

dib_t.findley_tau=plot(app.Representacion_torsion,t_m.findley_tau,t_a.findley_tau,'LineWidth',2,'Color',[0.851
0,0.3255,0.0980]); %Color: Marron
                    end

                end

                %Dang Van
                if situacion.dangvan==1
                    if entra.dangvan_tau==1

dib.dangvan_tau_1=plot(app.Representacion,s_m.dangvan_tau(1:(puntos_discretizacion+1),1),s_a.dangvan_tau(1:(pu
ntos_discretizacion+1),1),'LineWidth',2,'Color',[0.9294,0.6941,0.1255]); %Color: Naranja

dib.dangvan_tau_2=plot(app.Representacion,s_m.dangvan_tau((puntos_discretizacion+1):end,1),s_a.dangvan_tau((pu
ntos_discretizacion+1):end,1),'--','Color',[0.9294,0.6941,0.1255],'LineWidth',2); %Color: Naranja
                    else

dib.dangvan_tau=plot(app.Representacion,s_m.dangvan_tau,s_a.dangvan_tau,'LineWidth',2,'Color',[0.9294,0.6941,0
.1255]); %Color: Naranja
                    end

dib_t.dangvan_tau=plot(app.Representacion_torsion,t_m.dangvan_tau,t_a.dangvan_tau,'LineWidth',2,'Color',[0.929
4,0.6941,0.1255]); %Color: Naranja
                end

            end

%Se muestran los valores de alfa y beta (en este caso, para el ensayo s_-1--tau_-1)
app.AlfaSines.Value=0;
app.BetaSines.Value=beta.sines_tau;
app.AlfaCrossland.Value=alfa.crossland_tau;
app.BetaCrossland.Value=beta.crossland_tau;
app.AlfaMatake.Value=alfa.matake_tau;
app.BetaMatake.Value=beta.matake_tau;
app.AlfaFindley.Value=alfa.findley_tau;
app.BetaFindley.Value=beta.findley_tau;
app.AlfaDangVan.Value=alfa.dangvan_tau;
app.BetaDangVan.Value=beta.dangvan_tau;

        end

        % Value changed function: SinesCheckBox
        function SinesCheckBoxValueChanged(app, event)

            global situacion
```

```
            global entra
            global s_m
            global s_a
            global t_m
            global t_a
            global puntos_discretizacion
            global dib
            global dib_t

            situacion.sines = app.SinesCheckBox.Value;

            if situacion.sines==1

                hold(app.Representacion,'on');
                hold(app.Representacion_torsion,'on');

                if situacion.tau==1
%                     situacion.sines=0;
%                     app.SinesCheckBox.Value=0;

dib_t.sines_tau=plot(app.Representacion_torsion,t_m.sines_tau,t_a.sines_tau,'LineWidth',2,'Color',[0.4941,0.18
43,0.5569]); %Color: Morado
                    mensaje=sprintf('El método de Sines para el caso Uniaxial Puro no se puede calcular con la
pareja de ensayos seleccionada');
                    uialert(app.pag_principal,mensaje,'Datos no válidos','Icon','warning');
                else

                    %Se dibuja sines
                    if situacion.zer==1
                        if entra.sines_zer==1

dib.sines_zer_1=plot(app.Representacion,s_m.sines_zer(1:(puntos_discretizacion+1),1),s_a.sines_zer(1:(puntos_d
iscretizacion+1),1),'LineWidth',2,'Color',[0.4941,0.1843,0.5569]); %Color: Morado

dib.sines_zer_2=plot(app.Representacion,s_m.sines_zer((puntos_discretizacion+1):end,1),s_a.sines_zer((puntos_d
iscretizacion+1):end,1),'--','Color',[0.4941,0.1843,0.5569],'LineWidth',2); %Color: Morado
                        else

dib.sines_zer=plot(app.Representacion,s_m.sines_zer,s_a.sines_zer,'LineWidth',2,'Color',[0.4941,0.1843,0.5569]
); %Color: Morado
                        end

dib_t.sines_zer=plot(app.Representacion_torsion,t_m.sines_zer,t_a.sines_zer,'LineWidth',2,'Color',[0.4941,0.18
43,0.5569]); %Color: Morado
                    elseif situacion.ut==1
                        if entra.sines_ut==1

dib.sines_ut_1=plot(app.Representacion,s_m.sines_ut(1:(puntos_discretizacion+1),1),s_a.sines_ut(1:(puntos_disc
retizacion+1),1),'LineWidth',2,'Color',[0.4941,0.1843,0.5569]); %Color: Morado

dib.sines_ut_2=plot(app.Representacion,s_m.sines_ut((puntos_discretizacion+1):end,1),s_a.sines_ut((puntos_disc
retizacion+1):end,1),'--','Color',[0.4941,0.1843,0.5569],'LineWidth',2); %Color: Morado
                        else

dib.sines_ut=plot(app.Representacion,s_m.sines_ut,s_a.sines_ut,'LineWidth',2,'Color',[0.4941,0.1843,0.5569]);
%Color: Morado
                        end

dib_t.sines_ut=plot(app.Representacion_torsion,t_m.sines_ut,t_a.sines_ut,'LineWidth',2,'Color',[0.4941,0.1843,
0.5569]); %Color: Morado
                    end
                end

            else

                %Se borra
                if situacion.tau==1
                    delete(dib_t.sines_tau)
                elseif situacion.zer==1
                    if entra.sines_zer==1
                        delete(dib.sines_zer_1)
                        delete(dib.sines_zer_2)
                    else
                        delete(dib.sines_zer)
```

```
            end
            delete(dib_t.sines_zer)
        elseif situacion.ut==1
            if entra.sines_ut==1
                delete(dib.sines_ut_1)
                delete(dib.sines_ut_2)
            else
                delete(dib.sines_ut)
            end
            delete(dib_t.sines_ut)
        end

    end
end

% Value changed function: CheckBox_2
function CheckBox_2ValueChanged(app, event)

    global situacion
    global entra
    global dib
    global s_m
    global s_a
    global entra_t
    global dib_t
    global t_m
    global t_a
    global puntos_discretizacion
    global alfa
    global beta

    situacion.zer = app.CheckBox_2.Value;

    if situacion.zer==1

        %Se borran los graficos anteriores
        %sines
        if situacion.sines==1
            if situacion.tau==1
                delete(dib_t.sines_tau)
            elseif situacion.ut==1
                if entra.sines_ut==1
                    delete(dib.sines_ut_1)
                    delete(dib.sines_ut_2)
                else
                    delete(dib.sines_ut)
                end
                delete(dib_t.sines_ut)
            end
        end
        %Crossland
        if situacion.crossland==1
            if situacion.tau==1
                if entra.crossland_tau==1
                    delete(dib.crossland_tau_1)
                    delete(dib.crossland_tau_2)
                else
                    delete(dib.crossland_tau)
                end
                delete(dib_t.crossland_tau)
            elseif situacion.ut==1
                if entra.crossland_ut==1
                    delete(dib.crossland_ut_1)
                    delete(dib.crossland_ut_2)
                else
                    delete(dib.crossland_ut)
                end
                delete(dib_t.crossland_ut)
            end
        end
        %matake
        if situacion.matake==1
            if situacion.tau==1
                if entra.matake_tau==1
```

```
                    delete(dib.matake_tau_1)
                    delete(dib.matake_tau_2)
                else
                    delete(dib.matake_tau)
                end
                delete(dib_t.matake_tau)
            elseif situacion.ut==1
                delete(dib.matake_ut)
                delete(dib_t.matake_ut)
            end

        end
        %findley
        if situacion.findley==1
            if situacion.tau==1
                if entra.findley_tau==1
                    delete(dib.findley_tau_1)
                    delete(dib.findley_tau_2)
                else
                    delete(dib.findley_tau)
                end

                if entra_t.findley_tau==1
                    delete(dib_t.findley_tau_1)
                    delete(dib_t.findley_tau_2)
                else
                    delete(dib_t.findley_tau)
                end

            elseif situacion.ut==1
                if entra.findley_ut==1
                    delete(dib.findley_ut_1)
                    delete(dib.findley_ut_2)
                else
                    delete(dib.findley_ut)
                end

                if entra_t.findley_ut==1
                    delete(dib_t.findley_ut_1)
                    delete(dib_t.findley_ut_2)
                else
                    delete(dib_t.findley_ut)
                end

            end
        end
        %dangvan
        if situacion.dangvan==1
            if situacion.tau==1
                if entra.dangvan_tau==1
                    delete(dib.dangvan_tau_1)
                    delete(dib.dangvan_tau_2)
                else
                    delete(dib.dangvan_tau)
                end
                delete(dib_t.dangvan_tau)
            elseif situacion.ut==1
                delete(dib.dangvan_ut)
                delete(dib_t.dangvan_ut)
            end
        end

        situacion.tau=0;
        app.CheckBox.Value=0;
        situacion.ut=0;
        app.CheckBox_3.Value=0;
        nuevo=1;


    elseif situacion.zer==0 && situacion.tau==0 && situacion.ut==0
        situacion.zer=1;
        app.CheckBox_2.Value=situacion.zer;
        nuevo=0;
    end
```

```
            if situacion.zer==1 && nuevo==1
                hold (app.Representacion,'off');
                hold (app.Representacion_torsion,'off');

                hold (app.Representacion,'on');
                hold (app.Representacion_torsion,'on');
                %Para que los ejes sean cuadrados
                daspect(app.Representacion,[1 1 1]);
                daspect(app.Representacion_torsion,[1 1 1]);
                %Sines
                if situacion.sines==1
                    if entra.sines_zer==1

dib.sines_zer_1=plot(app.Representacion,s_m.sines_zer(1:(puntos_discretizacion+1),1),s_a.sines_zer(1:(puntos_d
iscretizacion+1),1),'LineWidth',2,'Color',[0.4941,0.1843,0.5569]); %Color: Morado

dib.sines_zer_2=plot(app.Representacion,s_m.sines_zer((puntos_discretizacion+1):end,1),s_a.sines_zer((puntos_d
iscretizacion+1):end,1),'--','Color',[0.4941,0.1843,0.5569],'LineWidth',2); %Color: Morado
                    else

dib.sines_zer=plot(app.Representacion,s_m.sines_zer,s_a.sines_zer,'LineWidth',2,'Color',[0.4941,0.1843,0.5569]
); %Color: Morado
                    end

dib_t.sines_zer=plot(app.Representacion_torsion,t_m.sines_zer,t_a.sines_zer,'LineWidth',2,'Color',[0.4941,0.18
43,0.5569]); %Color: Morado
                end

                %Crossland
                if situacion.crossland==1
                    if entra.crossland_zer==1

dib.crossland_zer_1=plot(app.Representacion,s_m.crossland_zer(1:(puntos_discretizacion+1),1),s_a.crossland_zer
(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.5569,0.6588,0.4667]); %Color: Verdoso

dib.crossland_zer_2=plot(app.Representacion,s_m.crossland_zer((puntos_discretizacion+1):end,1),s_a.crossland_z
er((puntos_discretizacion+1):end,1),'--','Color',[0.5569,0.6588,0.4667],'LineWidth',2); %Color: Verdoso
                    else

dib.crossland_zer=plot(app.Representacion,s_m.crossland_zer,s_a.crossland_zer,'LineWidth',2,'Color',[0.5569,0.
6588,0.4667]); %Color: Verdoso
                    end

dib_t.crossland_zer=plot(app.Representacion_torsion,t_m.crossland_zer,t_a.crossland_zer,'LineWidth',2,'Color',
[0.5569,0.6588,0.4667]); %Color: Verdoso
                end

                %Matake
                if situacion.matake==1
                    if entra.matake_zer==1

dib.matake_zer_1=plot(app.Representacion,s_m.matake_zer(1:(puntos_discretizacion+1),1),s_a.matake_zer(1:(punto
s_discretizacion+1),1),'LineWidth',2,'Color',[0.4510,0.5490,0.7294]); %Color: Azul

dib.matake_zer_2=plot(app.Representacion,s_m.matake_zer((puntos_discretizacion+1):end,1),s_a.matake_zer((punto
s_discretizacion+1):end,1),'--','Color',[0.4510,0.5490,0.7294],'LineWidth',2); %Color: Azul
                    else

dib.matake_zer=plot(app.Representacion,s_m.matake_zer,s_a.matake_zer,'LineWidth',2,'Color',[0.4510,0.5490,0.72
94]); %Color: Azul
                    end

dib_t.matake_zer=plot(app.Representacion_torsion,t_m.matake_zer,t_a.matake_zer,'LineWidth',2,'Color',[0.4510,0
.5490,0.7294]); %Color: Azul
                end

                %Findley
                if situacion.findley==1
                    if entra.findley_zer==1

dib.findley_zer_1=plot(app.Representacion,s_m.findley_zer(1:(puntos_discretizacion+1),1),s_a.findley_zer(1:(pu
ntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron
```

```
dib.findley_zer_2=plot(app.Representacion,s_m.findley_zer((puntos_discretizacion+1):end,1),s_a.findley_zer((pu
ntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
                    else

dib.findley_zer=plot(app.Representacion,s_m.findley_zer,s_a.findley_zer,'LineWidth',2,'Color',[0.8510,0.3255,0
.0980]); %Color: Marron
                    end

                    if entra_t.findley_zer==1

dib_t.findley_zer_1=plot(app.Representacion_torsion,t_m.findley_zer(1:(puntos_discretizacion+1),1),t_a.findley
_zer(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib_t.findley_zer_2=plot(app.Representacion_torsion,t_m.findley_zer((puntos_discretizacion+1):end,1),t_a.findl
ey_zer((puntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
                    else

dib_t.findley_zer=plot(app.Representacion_torsion,t_m.findley_zer,t_a.findley_zer,'LineWidth',2,'Color',[0.851
0,0.3255,0.0980]); %Color: Marron
                    end

                end

                %Dang Van
                if situacion.dangvan==1
                    if entra.dangvan_zer==1

dib.dangvan_zer_1=plot(app.Representacion,s_m.dangvan_zer(1:(puntos_discretizacion+1),1),s_a.dangvan_zer(1:(pu
ntos_discretizacion+1),1),'LineWidth',2,'Color',[0.9294,0.6941,0.1255]); %Color: Naranja

dib.dangvan_zer_2=plot(app.Representacion,s_m.dangvan_zer((puntos_discretizacion+1):end,1),s_a.dangvan_zer((pu
ntos_discretizacion+1):end,1),'--','Color',[0.9294,0.6941,0.1255],'LineWidth',2); %Color: Naranja
                    else

dib.dangvan_zer=plot(app.Representacion,s_m.dangvan_zer,s_a.dangvan_zer,'LineWidth',2,'Color',[0.9294,0.6941,0
.1255]); %Color: Naranja
                    end

dib_t.dangvan_zer=plot(app.Representacion_torsion,t_m.dangvan_zer,t_a.dangvan_zer,'LineWidth',2,'Color',[0.929
4,0.6941,0.1255]); %Color: Naranja
                end

            end

%Se muestran los valores de alfa y beta (en este caso, para el ensayo s_-1--s_-0)
app.AlfaSines.Value=alfa.sines_zer;
app.BetaSines.Value=beta.sines_zer;
app.AlfaCrossland.Value=alfa.crossland_zer;
app.BetaCrossland.Value=beta.crossland_zer;
app.AlfaMatake.Value=alfa.matake_zer;
app.BetaMatake.Value=beta.matake_zer;
app.AlfaFindley.Value=alfa.findley_zer;
app.BetaFindley.Value=beta.findley_zer;
app.AlfaDangVan.Value=alfa.dangvan_zer;
app.BetaDangVan.Value=beta.dangvan_zer;

        end

        % Value changed function: CheckBox_3
        function CheckBox_3ValueChanged(app, event)

            global situacion
            global entra
            global dib
            global s_m
            global s_a
            global entra_t
            global dib_t
            global t_m
            global t_a
            global puntos_discretizacion
            global alfa
            global beta
```

```
situacion.ut = app.CheckBox_3.Value;

if situacion.ut==1

    %Se borran los graficos anteriores
    %sines
    if situacion.sines==1
        if situacion.tau==1
            delete(dib_t.sines_tau)
        elseif situacion.zer==1
            if entra.sines_zer==1
                delete(dib.sines_zer_1)
                delete(dib.sines_zer_2)
            else
                delete(dib.sines_zer)
            end
            delete(dib_t.sines_zer)
        end
    end
    %Crossland
    if situacion.crossland==1
        if situacion.tau==1
            if entra.crossland_tau==1
                delete(dib.crossland_tau_1)
                delete(dib.crossland_tau_2)
            else
                delete(dib.crossland_tau)
            end
            delete(dib_t.crossland_tau)
        elseif situacion.zer==1
            if entra.crossland_zer==1
                delete(dib.crossland_zer_1)
                delete(dib.crossland_zer_2)
            else
                delete(dib.crossland_zer)
            end
            delete(dib_t.crossland_zer)
        end
    end
    %matake
    if situacion.matake==1
        if situacion.tau==1
            if entra.matake_tau==1
                delete(dib.matake_tau_1)
                delete(dib.matake_tau_2)
            else
                delete(dib.matake_tau)
            end
            delete(dib_t.matake_tau)
        elseif situacion.zer==1
            if entra.matake_zer==1
                delete(dib.matake_zer_1)
                delete(dib.matake_zer_2)
            else
                delete(dib.matake_zer)
            end
            delete(dib_t.matake_zer)
        end

    end
    %findley
    if situacion.findley==1
        if situacion.tau==1
            if entra.findley_tau==1
                delete(dib.findley_tau_1)
                delete(dib.findley_tau_2)
            else
                delete(dib.findley_tau)
            end

            if entra_t.findley_tau==1
                delete(dib_t.findley_tau_1)
                delete(dib_t.findley_tau_2)
```

```
                    else
                        delete(dib_t.findley_tau)
                    end

                elseif situacion.zer==1
                    if entra.findley_zer==1
                        delete(dib.findley_zer_1)
                        delete(dib.findley_zer_2)
                    else
                        delete(dib.findley_zer)
                    end

                    if entra_t.findley_zer==1
                        delete(dib_t.findley_zer_1)
                        delete(dib_t.findley_zer_2)
                    else
                        delete(dib_t.findley_zer)
                    end

                end
            end
            %dangvan
            if situacion.dangvan==1
                if situacion.tau==1
                    if entra.dangvan_tau==1
                        delete(dib.dangvan_tau_1)
                        delete(dib.dangvan_tau_2)
                    else
                        delete(dib.dangvan_tau)
                    end
                    delete(dib_t.dangvan_tau)
                elseif situacion.zer==1
                    if entra.dangvan_zer==1
                        delete(dib.dangvan_zer_1)
                        delete(dib.dangvan_zer_2)
                    else
                        delete(dib.dangvan_zer)
                    end
                    delete(dib_t.dangvan_zer)
                end
            end

            situacion.tau=0;
            app.CheckBox.Value=0;
            situacion.zer=0;
            app.CheckBox_2.Value=0;
            nuevo=1;


        elseif situacion.ut==0 && situacion.tau==0 && situacion.zer==0
            situacion.ut=1;
            app.CheckBox_3.Value=situacion.ut;
            nuevo=0;
        end

        if situacion.ut==1 && nuevo==1
            hold (app.Representacion,'off');
            hold (app.Representacion_torsion,'off');

            hold (app.Representacion,'on');
            hold (app.Representacion_torsion,'on');
            %Para que los ejes sean cuadrados
            daspect(app.Representacion,[1 1 1]);
            daspect(app.Representacion_torsion,[1 1 1]);
            %Sines
            if situacion.sines==1
                if entra.sines_ut==1

dib.sines_ut_1=plot(app.Representacion,s_m.sines_ut(1:(puntos_discretizacion+1),1),s_a.sines_ut(1:(puntos_disc
retizacion+1),1),'LineWidth',2,'Color',[0.4941,0.1843,0.5569]); %Color: Morado

dib.sines_ut_2=plot(app.Representacion,s_m.sines_ut((puntos_discretizacion+1):end,1),s_a.sines_ut((puntos_disc
retizacion+1):end,1),'--','Color',[0.4941,0.1843,0.5569],'LineWidth',2); %Color: Morado
                else
```

```
dib.sines_ut=plot(app.Representacion,s_m.sines_ut,s_a.sines_ut,'LineWidth',2,'Color',[0.4941,0.1843,0.5569]);
%Color: Morado
                    end

dib_t.sines_ut=plot(app.Representacion_torsion,t_m.sines_ut,t_a.sines_ut,'LineWidth',2,'Color',[0.4941,0.1843,
0.5569]); %Color: Morado
                end

                %Crossland
                if situacion.crossland==1
                    if entra.crossland_ut==1

dib.crossland_ut_1=plot(app.Representacion,s_m.crossland_ut(1:(puntos_discretizacion+1),1),s_a.crossland_ut(1:
(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.5569,0.6588,0.4667]); %Color: Verdoso

dib.crossland_ut_2=plot(app.Representacion,s_m.crossland_ut((puntos_discretizacion+1):end,1),s_a.crossland_ut(
(puntos_discretizacion+1):end,1),'--','Color',[0.5569,0.6588,0.4667],'LineWidth',2); %Color: Verdoso
                    else

dib.crossland_ut=plot(app.Representacion,s_m.crossland_ut,s_a.crossland_ut,'LineWidth',2,'Color',[0.5569,0.658
8,0.4667]); %Color: Verdoso
                    end

dib_t.crossland_ut=plot(app.Representacion_torsion,t_m.crossland_ut,t_a.crossland_ut,'LineWidth',2,'Color',[0.
5569,0.6588,0.4667]); %Color: Verdoso
                end

                %Matake
                if situacion.matake==1

dib.matake_ut=plot(app.Representacion,s_m.matake_ut,s_a.matake_ut,'LineWidth',2,'Color',[0.4510,0.5490,0.7294]
); %Color: Azul

dib_t.matake_ut=plot(app.Representacion_torsion,t_m.matake_ut,t_a.matake_ut,'LineWidth',2,'Color',[0.4510,0.54
90,0.7294]); %Color: Azul
                end

                %Findley
                if situacion.findley==1
                    if entra.findley_ut==1

dib.findley_ut_1=plot(app.Representacion,s_m.findley_ut(1:(puntos_discretizacion+1),1),s_a.findley_ut(1:(punto
s_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib.findley_ut_2=plot(app.Representacion,s_m.findley_ut((puntos_discretizacion+1):end,1),s_a.findley_ut((punto
s_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
                    else

dib.findley_ut=plot(app.Representacion,s_m.findley_ut,s_a.findley_ut,'LineWidth',2,'Color',[0.8510,0.3255,0.09
80]); %Color: Marron
                    end

                    if entra_t.findley_ut==1

dib_t.findley_ut_1=plot(app.Representacion_torsion,t_m.findley_ut(1:(puntos_discretizacion+1),1),t_a.findley_u
t(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib_t.findley_ut_2=plot(app.Representacion_torsion,t_m.findley_ut((puntos_discretizacion+1):end,1),t_a.findley
_ut((puntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
                    else

dib_t.findley_ut=plot(app.Representacion_torsion,t_m.findley_ut,t_a.findley_ut,'LineWidth',2,'Color',[0.8510,0
.3255,0.0980]); %Color: Marron
                    end

                end

                %Dang Van
                if situacion.dangvan==1

dib.dangvan_ut=plot(app.Representacion,s_m.dangvan_ut,s_a.dangvan_ut,'LineWidth',2,'Color',[0.9294,0.6941,0.12
55]); %Color: Naranja
```

```
dib_t.dangvan_ut=plot(app.Representacion_torsion,t_m.dangvan_ut,t_a.dangvan_ut,'LineWidth',2,'Color',[0.9294,0
.6941,0.1255]); %Color: Naranja
                end

            end
%Se muestran los valores de alfa y beta (en este caso, para el ensayo s_-1--s_-0)
app.AlfaSines.Value=alfa.sines_ut;
app.BetaSines.Value=beta.sines_ut;
app.AlfaCrossland.Value=alfa.crossland_ut;
app.BetaCrossland.Value=beta.crossland_ut;
app.AlfaMatake.Value=alfa.matake_ut;
app.BetaMatake.Value=beta.matake_ut;
app.AlfaFindley.Value=alfa.findley_ut;
app.BetaFindley.Value=beta.findley_ut;
app.AlfaDangVan.Value=alfa.dangvan_ut;
app.BetaDangVan.Value=beta.dangvan_ut;

        end

        % Value changed function: AlfaSines
        function AlfaSinesValueChanged(app, event)
            %value = app.AlfaSines.Value;

        end

        % Value changed function: BetaSines
        function BetaSinesValueChanged(app, event)
            %value = app.BetaSines.Value;

        end

        % Value changed function: AlfaCrossland
        function AlfaCrosslandValueChanged(app, event)
            %value = app.AlfaCrossland.Value;

        end

        % Value changed function: BetaCrossland
        function BetaCrosslandValueChanged(app, event)
            %value = app.BetaCrossland.Value;

        end

        % Value changed function: AlfaMatake
        function AlfaMatakeValueChanged(app, event)
            %value = app.AlfaMatake.Value;

        end

        % Value changed function: BetaMatake
        function BetaMatakeValueChanged(app, event)
            %value = app.BetaMatake.Value;

        end

        % Value changed function: AlfaFindley
        function AlfaFindleyValueChanged(app, event)
           %value = app.AlfaFindley.Value;

        end

        % Value changed function: BetaFindley
        function BetaFindleyValueChanged(app, event)
            %value = app.BetaFindley.Value;

        end

        % Value changed function: AlfaDangVan
        function AlfaDangVanValueChanged(app, event)
            %value = app.AlfaDangVan.Value;

        end
```

```
        % Value changed function: BetaDangVan
        function BetaDangVanValueChanged(app, event)
            %value = app.BetaDangVan.Value;

        end

        % Callback function
        function Button_3Pushed(app, event)

        end

        % Value changed function: Csteel2CheckBox
        function Csteel2CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.c_steel = app.Csteel2CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;

            if situacion.c_steel==1

dib.c_steel=plot(app.Representacion,x.c_steel,y.c_steel,'o','MarkerSize',7,'MarkerEdgeColor',[1,0.6,0.6],'Mark
erFaceColor',[1,0.6,0.6]); %Color: Salmon
            else
                delete(dib.c_steel)
            end

        end

        % Value changed function: Carbonsteel3CheckBox
        function Carbonsteel3CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.carb_steel_013 = app.Carbonsteel3CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;

            if situacion.carb_steel_013==1

dib.carb_steel_013=plot(app.Representacion,x.carb_steel_013,y.carb_steel_013,'d','MarkerSize',7,'MarkerEdgeCol
or',[0,0.4471,0.7412],'MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
            else
                delete(dib.carb_steel_013)
            end
        end

        % Value changed function: Carbonsteel4CheckBox
        function Carbonsteel4CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.carb_steel_029 = app.Carbonsteel4CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;

            if situacion.carb_steel_029==1
```

```
dib.carb_steel_029=plot(app.Representacion,x.carb_steel_029,y.carb_steel_029,'s','MarkerSize',7,'MarkerEdgeCol
or',[0.6353,0.0784,0.1843],'MarkerFaceColor',[0.6353,0.0784,0.1843]); %Color: Granate
            else
                delete(dib.carb_steel_029)
            end
        end

        % Value changed function: AISI4130steel5CheckBox
        function AISI4130steel5CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.AISI_4130 = app.AISI4130steel5CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.AISI_4130==1

dib.AISI_4130=plot(app.Representacion,x.AISI_4130,y.AISI_4130,'^','MarkerSize',7,'MarkerEdgeColor',[0.4667,0.6
745,0.1882],'MarkerFaceColor',[0.4667,0.6745,0.1882]); %Color: Verde
            else
                delete(dib.AISI_4130)
            end
        end

        % Value changed function: AISI4340steel6CheckBox
        function AISI4340steel6CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.AISI_4340 = app.AISI4340steel6CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.AISI_4340==1

dib.AISI_4340=plot(app.Representacion,x.AISI_4340,y.AISI_4340,'x','MarkerSize',7,'MarkerEdgeColor',[0.1490,0.1
490,0.1490],'MarkerFaceColor',[0.1490,0.1490,0.1490]); %Color: Negro
            else
                delete(dib.AISI_4340)
            end
        end

        % Value changed function: En25steel7CheckBox
        function En25steel7CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.En_25 = app.En25steel7CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.En_25==1
```

```
dib.En_25=plot(app.Representacion,x.En_25,y.En_25,'d','MarkerSize',7,'MarkerEdgeColor',[0.3020,0.7451,0.9333],
'MarkerFaceColor',[0.3020,0.7451,0.9333]); %Color: Azul claro
            else
                delete(dib.En_25)
            end
        end

        % Value changed function: NiCrMo8steel8CheckBox
        function NiCrMo8steel8CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.NiCrMo8 = app.NiCrMo8steel8CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.NiCrMo8==1

dib.NiCrMo8=plot(app.Representacion,x.NiCrMo8,y.NiCrMo8,'p','MarkerSize',7,'MarkerEdgeColor',[1.0000,0.0745,0.
6510],'MarkerFaceColor',[1.0000,0.0745,0.6510]); %Color: Rosa
            else
                delete(dib.NiCrMo8)
            end
        end

        % Value changed function: CrMo4steel9CheckBox
        function CrMo4steel9CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.CrMo4 = app.CrMo4steel9CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.CrMo4==1

dib.CrMo4=plot(app.Representacion,x.CrMo4,y.CrMo4,'o','MarkerSize',7,'MarkerEdgeColor',[0.6510,0.6510,0.6510],
'MarkerFaceColor',[0.6510,0.6510,0.6510]); %Color: Amarillo
            else
                delete(dib.CrMo4)
            end
        end

        % Value changed function: SAE52100steel10CheckBox
        function SAE52100steel10CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.SAE_52100 = app.SAE52100steel10CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.SAE_52100==1
```

```
dib.SAE_52100=plot(app.Representacion,x.SAE_52100,y.SAE_52100,'*','MarkerSize',7,'MarkerEdgeColor',[0,0.4471,0
.7412],'MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
            else
                delete(dib.SAE_52100)
            end
        end

        % Value changed function: Mildsteel11CheckBox
        function Mildsteel11CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.Mild_Steel = app.Mildsteel11CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.Mild_Steel==1

dib.Mild_Steel=plot(app.Representacion,x.Mild_Steel,y.Mild_Steel,'d','MarkerSize',7,'MarkerEdgeColor',[0.6510,
0.6510,0.6510],'MarkerFaceColor',[0.6510,0.6510,0.6510]); %color:gris
            else
                delete(dib.Mild_Steel)
            end
        end

        % Value changed function: NiCralloysteel11CheckBox
        function NiCralloysteel11CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.NiCr_alloy = app.NiCralloysteel11CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.NiCr_alloy==1

dib.NiCr_alloy=plot(app.Representacion,x.NiCr_alloy,y.NiCr_alloy,'s','MarkerSize',7,'MarkerEdgeColor',[1.0000,
0.4118,0.1608],'MarkerFaceColor',[1.0000,0.4118,0.1608]); %color:naranja
            else
                delete(dib.NiCr_alloy)
            end
        end

        % Value changed function: SAE413011CheckBox
        function SAE413011CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.SAE_4130 = app.SAE413011CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.SAE_4130==1
```

```
dib.SAE_4130=plot(app.Representacion,x.SAE_4130,y.SAE_4130,'>','MarkerSize',7,'MarkerEdgeColor',[0.4941,0.1843
,0.5569],'MarkerFaceColor',[0.4941,0.1843,0.5569]); %color:morado
            else
                delete(dib.SAE_4130)
            end
        end

        % Value changed function: ST311CheckBox
        function ST311CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.S24_T3 = app.ST311CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.S24_T3==1

dib.S24_T3=plot(app.Representacion,x.S24_T3,y.S24_T3,'p','MarkerSize',7,'MarkerEdgeColor',[0.3255,0.7608,0.682
4],'MarkerFaceColor',[0.3255,0.7608,0.6824]); %color:Azul turquesa
            else
                delete(dib.S24_T3)
            end
        end

        % Value changed function: ST611CheckBox_2
        function ST611CheckBox_2ValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.S75_T6 = app.ST611CheckBox_2.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.S75_T6==1

dib.S75_T6=plot(app.Representacion,x.S75_T6,y.S75_T6,'*','MarkerSize',7,'MarkerEdgeColor',[0.8314,0.3725,0.486
3],'MarkerFaceColor',[0.8314,0.3725,0.4863]); %color:rojo
            else
                delete(dib.S75_T6)
            end
        end

        % Value changed function: T611CheckBox_2
        function T611CheckBox_2ValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.a_2014_T6 = app.T611CheckBox_2.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.a_2014_T6==1
```

```
dib.a_2014_T6=plot(app.Representacion,x.a_2014_T6,y.a_2014_T6,'o','MarkerSize',7,'MarkerEdgeColor',[0.2667,0.6
784,0.1294],'MarkerFaceColor',[0.2667,0.6784,0.1294]); %color:Verde
            else
                delete(dib.a_2014_T6)
            end
        end

        % Value changed function: T411CheckBox
        function T411CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.a_2024_T4 = app.T411CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.a_2024_T4==1

dib.a_2024_T4=plot(app.Representacion,x.a_2024_T4,y.a_2024_T4,'h','MarkerSize',7,'MarkerEdgeColor',[0.6902,0.5
412,0.3569],'MarkerFaceColor',[0.6902,0.5412,0.3569]); %color:Marron
            else
                delete(dib.a_2024_T4)
            end
        end

        % Value changed function: T611CheckBox_3
        function T611CheckBox_3ValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.a_6061_T6 = app.T611CheckBox_3.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.a_6061_T6==1

dib.a_6061_T6=plot(app.Representacion,x.a_6061_T6,y.a_6061_T6,'o','MarkerSize',7,'MarkerEdgeColor',[0.9490,0.0
392,0.0392],'MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
            else
                delete(dib.a_6061_T6)
            end

        end

        % Value changed function: T611CheckBox
        function T611CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.a_7075_T6 = app.T611CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.a_7075_T6==1
```

```
dib.a_7075_T6=plot(app.Representacion,x.a_7075_T6,y.a_7075_T6,'v','MarkerSize',7,'MarkerEdgeColor',[0.8667,0.5
882,0.8784],'MarkerFaceColor',[0.8667,0.5882,0.8784]); %color:Morado
            else
                delete(dib.a_7075_T6)
            end
        end

        % Value changed function: ST411CheckBox
        function ST411CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.s_24_T4 = app.ST411CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.s_24_T4==1

dib.s_24_T4=plot(app.Representacion,x.s_24_T4,y.s_24_T4,'h','MarkerSize',7,'MarkerEdgeColor',[0.9804,0.4902,0.
1373],'MarkerFaceColor',[0.9804,0.4902,0.1373]); %color:Naranja
            else
                delete(dib.s_24_T4)
            end
        end

        % Value changed function: ST611CheckBox
        function ST611CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.s_14_T6 = app.ST611CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;

            if situacion.s_14_T6==1

dib.s_14_T6=plot(app.Representacion,x.s_14_T6,y.s_14_T6,'<','MarkerSize',7,'MarkerEdgeColor',[0.8000,0.5294,0.
0941],'MarkerFaceColor',[0.8000,0.5294,0.0941]); %color:Marron
            else
                delete(dib.s_14_T6)
            end
        end

        % Value changed function: Zn065Zr11CheckBox
        function Zn065Zr11CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.zn_25_zr65 = app.Zn065Zr11CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.zn_25_zr65==1

dib.zn_25_zr65=plot(app.Representacion,x.zn_25_zr65,y.zn_25_zr65,'d','MarkerSize',7,'MarkerEdgeColor',[0.9490,
0.0392,0.0392],'MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
```

```
            else
                delete(dib.zn_25_zr65)
            end
        end

        % Value changed function: Zn066Zr11CheckBox
        function Zn066Zr11CheckBoxValueChanged(app, event)
            global situacion
            global dib
            global x
            global y

            situacion.zn_56_zr66 = app.Zn066Zr11CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;


            if situacion.zn_56_zr66==1

dib.zn_56_zr66=plot(app.Representacion,x.zn_56_zr66,y.zn_56_zr66,'s','MarkerSize',7,'MarkerEdgeColor',[0.1765,
0.7098,0.4000],'MarkerFaceColor',[0.1765,0.7098,0.4000]); %color:Verde
            else
                delete(dib.zn_56_zr66)
            end
        end

        % Value changed function: Cnormalizedsteel13CheckBox_2
        function Cnormalizedsteel13CheckBox_2ValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.C_12norm = app.Cnormalizedsteel13CheckBox_2.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.C_12norm==1

dib_t.C_12norm=plot(app.Representacion_torsion,x.C_12norm,y.C_12norm,'d','MarkerSize',7,'MarkerEdgeColor',[0,0
.4471,0.7412],'MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
            else
                delete(dib_t.C_12norm)
            end
        end

        % Value changed function: Cnormalizedsteel13CheckBox
        function Cnormalizedsteel13CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.C_49norm = app.Cnormalizedsteel13CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.C_49norm==1

dib_t.C_49norm=plot(app.Representacion_torsion,x.C_49norm,y.C_49norm,'*','MarkerSize',7,'MarkerEdgeColor','k',
'MarkerFaceColor','k'); %color:Negro
            else
                delete(dib_t.C_49norm)
            end
        end
```

```
        % Value changed function: En25TNICrMosteel14CheckBox
        function En25TNICrMosteel14CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.En_25T = app.En25TNICrMosteel14CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.En_25T==1

dib_t.En_25T=plot(app.Representacion_torsion,x.En_25T,y.En_25T,'*','MarkerSize',7,'MarkerEdgeColor',[0.6353,0.
0784,0.1843],'MarkerFaceColor',[0.6353,0.0784,0.1843]); %Color: Granate
            else
                delete(dib_t.En_25T)
            end
        end

        % Value changed function: VDSiCrspringsteel15CheckBox
        function VDSiCrspringsteel15CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.VDSiCr = app.VDSiCrspringsteel15CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.VDSiCr==1

dib_t.VDSiCr=plot(app.Representacion_torsion,x.VDSiCr,y.VDSiCr,'d','MarkerSize',7,'MarkerEdgeColor',[0.6353,0.
0784,0.1843],'MarkerFaceColor',[0.6353,0.0784,0.1843]); %Color: Granate
            else
                delete(dib_t.VDSiCr)
            end
        end

        % Value changed function: SAE3140HRsteel16CheckBox
        function SAE3140HRsteel16CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.sae_3140_HR = app.SAE3140HRsteel16CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.sae_3140_HR==1

dib_t.sae_3140_HR=plot(app.Representacion_torsion,x.sae_3140_HR,y.sae_3140_HR,'>','MarkerSize',7,'MarkerEdgeCo
lor',[1.0000,0.0745,0.6510],'MarkerFaceColor',[1.0000,0.0745,0.6510]); %Color: Rosa
            else
                delete(dib_t.sae_3140_HR)
            end
        end

        % Value changed function: SAE3140QTsteel16CheckBox
        function SAE3140QTsteel16CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
```

```
            global x
            global y

            situacion.sae_3140_QT = app.SAE3140QTsteel16CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.sae_3140_QT==1

dib_t.sae_3140_QT=plot(app.Representacion_torsion,x.sae_3140_QT,y.sae_3140_QT,'^','MarkerSize',7,'MarkerEdgeCo
lor',[0.4667,0.6745,0.1882],'MarkerFaceColor',[0.4667,0.6745,0.1882]); %Color: Verde
            else
                delete(dib_t.sae_3140_QT)
            end
        end

        % Value changed function: Csorbiticsteel13CheckBox
        function Csorbiticsteel13CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.C_49sorb = app.Csorbiticsteel13CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.C_49sorb==1

dib_t.C_49sorb=plot(app.Representacion_torsion,x.C_49sorb,y.C_49sorb,'o','MarkerSize',7,'MarkerEdgeColor',[1,0
.6,0.6],'MarkerFaceColor',[1,0.6,0.6]); %Color: Salmon
            else
                delete(dib_t.C_49sorb)
            end
        end

        % Value changed function: NiCrMo3steel19CheckBox
        function NiCrMo3steel19CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.NiCrMo3 = app.NiCrMo3steel19CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.NiCrMo3==1

dib_t.NiCrMo3=plot(app.Representacion_torsion,x.NiCrMo3,y.NiCrMo3,'o','MarkerSize',7,'MarkerEdgeColor',[0.4941
,0.1843,0.5569],'MarkerFaceColor',[0.4941,0.1843,0.5569]); %color:morado
            else
                delete(dib_t.NiCrMo3)
            end
        end

        % Value changed function: STAluminium20CheckBox
        function STAluminium20CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.st_14_Al = app.STAluminium20CheckBox.Value;
```

```
            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.st_14_Al==1

dib_t.st_14_Al=plot(app.Representacion_torsion,x.st_14_Al,y.st_14_Al,'s','MarkerSize',7,'MarkerEdgeColor',[0,0
.4471,0.7412],'MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
            else
                delete(dib_t.st_14_Al)
            end
        end

        % Value changed function: Aluminium76ST6121CheckBox
        function Aluminium76ST6121CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.al_76 = app.Aluminium76ST6121CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.al_76==1

dib_t.al_76=plot(app.Representacion_torsion,x.al_76,y.al_76,'h','MarkerSize',7,'MarkerEdgeColor',[1.0000,0.074
5,0.6510],'MarkerFaceColor',[1.0000,0.0745,0.6510]); %Color: Rosa
            else
                delete(dib_t.al_76)
            end
        end

        % Value changed function: Csorbiticsteel17CheckBox
        function Csorbiticsteel17CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.C_12sorb = app.Csorbiticsteel17CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.C_12sorb==1

dib_t.C_12sorb=plot(app.Representacion_torsion,x.C_12sorb,y.C_12sorb,'<','MarkerSize',7,'MarkerEdgeColor',[0.8
000,0.5294,0.0941],'MarkerFaceColor',[0.8000,0.5294,0.0941]); %color:Marron
            else
                delete(dib_t.C_12sorb)
            end
        end

        % Value changed function: CrMo4Vsteel22CheckBox
        function CrMo4Vsteel22CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.CrMo4V = app.CrMo4Vsteel22CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;
```

```
            if situacion.CrMo4V==1

dib_t.CrMo4V=plot(app.Representacion_torsion,x.CrMo4V,y.CrMo4V,'s','MarkerSize',7,'MarkerEdgeColor',[0.3020,0.
7451,0.9333],'MarkerFaceColor',[0.3020,0.7451,0.9333]); %Color: Azul claro
            else
                delete(dib_t.CrMo4V)
            end
        end

        % Value changed function: CrNiMo6steel12CheckBox
        function CrNiMo6steel12CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.CrNiMo6_steel = app.CrNiMo6steel12CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.CrNiMo6_steel==1

dib_t.CrNiMo6_steel=plot(app.Representacion_torsion,x.CrNiMo6_steel,y.CrNiMo6_steel,'o','MarkerSize',7,'Marker
EdgeColor',[0.8392,0.8510,0.2980],'MarkerFaceColor',[0.8392,0.8510,0.2980]); %Color: Amarillo
            else
                delete(dib_t.CrNiMo6_steel)
            end
        end

        % Value changed function: JISSCM440steel24CheckBox
        function JISSCM440steel24CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.JIS = app.JISSCM440steel24CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.JIS==1

dib_t.JIS=plot(app.Representacion_torsion,x.JIS,y.JIS,'v','MarkerSize',7,'MarkerEdgeColor',[0.4941,0.1843,0.55
69],'MarkerFaceColor',[0.4941,0.1843,0.5569]); %color:morado
            else
                delete(dib_t.JIS)
            end
        end

        % Value changed function: MnCr5steel25CheckBox
        function MnCr5steel25CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.MnCr5 = app.MnCr5steel25CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.MnCr5==1

dib_t.MnCr5=plot(app.Representacion_torsion,x.MnCr5,y.MnCr5,'*','MarkerSize',7,'MarkerEdgeColor',[0,0.4471,0.7
412],'MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
            else
```

```
                delete(dib_t.MnCr5)
            end
        end

        % Value changed function: Csteel26CheckBox
        function Csteel26CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.c_01 = app.Csteel26CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.c_01==1

dib_t.c_01=plot(app.Representacion_torsion,x.c_01,y.c_01,'o','MarkerSize',7,'MarkerEdgeColor',[0.1765,0.7098,0
.4000],'MarkerFaceColor',[0.1765,0.7098,0.4000]); %color:Verde
            else
                delete(dib_t.c_01)
            end
        end

        % Value changed function: St35steel27CheckBox
        function St35steel27CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.ST_35 = app.St35steel27CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.ST_35==1

dib_t.ST_35=plot(app.Representacion_torsion,x.ST_35,y.ST_35,'h','MarkerSize',7,'MarkerEdgeColor',[0.9490,0.039
2,0.0392],'MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
            else
                delete(dib_t.ST_35)
            end
        end

        % Value changed function: CrMo4steel23CheckBox
        function CrMo4steel23CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.CrMo4_steel = app.CrMo4steel23CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.CrMo4_steel==1

dib_t.CrMo4_steel=plot(app.Representacion_torsion,x.CrMo4_steel,y.CrMo4_steel,'x','MarkerSize',7,'MarkerEdgeCo
lor',[0.1490,0.1490,0.1490],'MarkerFaceColor',[0.1490,0.1490,0.1490]); %Color: Negro
            else
                delete(dib_t.CrMo4_steel)
            end
        end

        % Value changed function: BSSS65Asteel28CheckBox
```

```
        function BSSS65Asteel28CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.bss_s65 = app.BSSS65Asteel28CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.bss_s65==1

dib_t.bss_s65=plot(app.Representacion_torsion,x.bss_s65,y.bss_s65,'^','MarkerSize',7,'MarkerEdgeColor',[0.9490
,0.0392,0.0392],'MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
            else
                delete(dib_t.bss_s65)
            end
        end

        % Value changed function: Cr4steel29CheckBox
        function Cr4steel29CheckBoxValueChanged(app, event)
            global situacion
            global dib_t
            global x
            global y

            situacion.Cr4_steel = app.Cr4steel29CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.Cr4_steel==1

dib_t.Cr4_steel=plot(app.Representacion_torsion,x.Cr4_steel,y.Cr4_steel,'p','MarkerSize',7,'MarkerEdgeColor',[
0.9804,0.4902,0.1373],'MarkerFaceColor',[0.9804,0.4902,0.1373]); %color:Naranja
            else
                delete(dib_t.Cr4_steel)
            end
        end

        % Value changed function: CNisteeltreatA13CheckBox
        function CNisteeltreatA13CheckBoxValueChanged(app, event)

            global situacion
            global dib_t
            global x
            global y

            situacion.C_35Ni_A = app.CNisteeltreatA13CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.C_35Ni_A ==1
                dib_t.C_35Ni_A =plot(app.Representacion_torsion,x.C_35Ni_A ,y.C_35Ni_A
,'s','MarkerSize',7,'MarkerEdgeColor',[0.6353,0.0784,0.1843],'MarkerFaceColor',[0.6353,0.0784,0.1843]);
%Color: Granate
            else
                delete(dib_t.C_35Ni_A)
            end
        end

        % Value changed function: CNisteeltreatB13CheckBox
        function CNisteeltreatB13CheckBoxValueChanged(app, event)

            global situacion
            global dib_t
```

```
            global x
            global y

            situacion.C_35Ni_B = app.CNisteeltreatB13CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.C_35Ni_B ==1
                dib_t.C_35Ni_B =plot(app.Representacion_torsion,x.C_35Ni_B ,y.C_35Ni_B
,'s','MarkerSize',7,'MarkerEdgeColor',[0.4667,0.6745,0.1882],'MarkerFaceColor',[0.4667,0.6745,0.1882]);
%Color: Verde
            else
                delete(dib_t.C_35Ni_B)
            end
        end

        % Value changed function: CNisteeltreatD13CheckBox
        function CNisteeltreatD13CheckBoxValueChanged(app, event)

            global situacion
            global dib_t
            global x
            global y

            situacion.C_35Ni_D = app.CNisteeltreatD13CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.C_35Ni_D ==1
                dib_t.C_35Ni_D =plot(app.Representacion_torsion,x.C_35Ni_D ,y.C_35Ni_D
,'p','MarkerSize',7,'MarkerEdgeColor',[0.3255,0.7608,0.6824],'MarkerFaceColor',[0.3255,0.7608,0.6824]);
%color:Azul turquesa
            else
                delete(dib_t.C_35Ni_D)
            end
        end

        % Value changed function: CrNisteeltreatD17CheckBox
        function CrNisteeltreatD17CheckBoxValueChanged(app, event)

            global situacion
            global dib_t
            global x
            global y

            situacion.CrNi_D = app.CrNisteeltreatD17CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.CrNi_D ==1
                dib_t.CrNi_D =plot(app.Representacion_torsion,x.CrNi_D ,y.CrNi_D
,'<','MarkerSize',7,'MarkerEdgeColor',[0.6510,0.6510,0.6510],'MarkerFaceColor',[0.6510,0.6510,0.6510]);
%color:gris
            else
                delete(dib_t.CrNi_D)
            end
        end

        % Value changed function: CK35steel23CheckBox
        function CK35steel23CheckBoxValueChanged(app, event)

            global situacion
            global dib_t
            global x
            global y
```

```
            situacion.CK_35 = app.CK35steel23CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.CK_35 ==1
                dib_t.CK_35
=plot(app.Representacion_torsion,x.CK_35,y.CK_35,'h','MarkerSize',7,'MarkerEdgeColor',[0.8667,0.5882,0.8784],'
MarkerFaceColor',[0.8667,0.5882,0.8784]); %color:Morado
            else
                delete(dib_t.CK_35)
            end
        end

        % Value changed function: CrMo4steel18CheckBox
        function CrMo4steel18CheckBoxValueChanged(app, event)

            global situacion
            global dib_t
            global x
            global y

            situacion.Cr_Mo_42 = app.CrMo4steel18CheckBox.Value;

            situacion.select_all_torsion =0;
            app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;
            situacion.unselect_all_torsion =0;
            app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion;

            if situacion.Cr_Mo_42 ==1
                dib_t.Cr_Mo_42
=plot(app.Representacion_torsion,x.Cr_Mo_42,y.Cr_Mo_42,'v','MarkerSize',7,'MarkerEdgeColor',[0.6902,0.5412,0.3
569],'MarkerFaceColor',[0.6902,0.5412,0.3569]); %color:Marron
            else
                delete(dib_t.Cr_Mo_42)
            end
        end

        % Callback function
        function SelectallCheckBoxValueChanged(app, event)

        end

        % Callback function
        function UnselectallCheckBoxValueChanged(app, event)

        end

        % Value changed function: Selectall_uniaxialCheckBox
        function Selectall_uniaxialCheckBoxValueChanged2(app, event)
%             value = app.Selectall_uniaxialCheckBox.Value;
            global situacion
            global dib
            global x
            global y

            situacion.select_all_uniaxial = app.Selectall_uniaxialCheckBox.Value;

            if situacion.select_all_uniaxial ==1
                situacion.unselect_all_uniaxial =0;
                app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;
                if situacion.c_steel ==0
                    situacion.c_steel=1;
                    app.Csteel2CheckBox.Value=1;

dib.c_steel=plot(app.Representacion,x.c_steel,y.c_steel,'o','MarkerSize',7,'MarkerEdgeColor',[1,0.6,0.6],'Mark
erFaceColor',[1,0.6,0.6]); %Color: Salmon
                end
                if situacion.carb_steel_013 ==0
                    situacion.carb_steel_013 =1;
                    app.Carbonsteel3CheckBox.Value=1;
```

```
dib.carb_steel_013=plot(app.Representacion,x.carb_steel_013,y.carb_steel_013,'d','MarkerSize',7,'MarkerEdgeCol
or',[0,0.4471,0.7412],'MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
                end
                if situacion.carb_steel_029 ==0
                    situacion.carb_steel_029 =1;
                    app.Carbonsteel4CheckBox.Value=1;

dib.carb_steel_029=plot(app.Representacion,x.carb_steel_029,y.carb_steel_029,'s','MarkerSize',7,'MarkerEdgeCol
or',[0.6353,0.0784,0.1843],'MarkerFaceColor',[0.6353,0.0784,0.1843]); %Color: Granate
                end
                if situacion.AISI_4130 ==0
                    situacion.AISI_4130 =1;
                    app.AISI4130steel5CheckBox.Value=1;

dib.AISI_4130=plot(app.Representacion,x.AISI_4130,y.AISI_4130,'^','MarkerSize',7,'MarkerEdgeColor',[0.4667,0.6
745,0.1882],'MarkerFaceColor',[0.4667,0.6745,0.1882]); %Color: Verde
                end
                if situacion.AISI_4340 ==0
                    situacion.AISI_4340 =1;
                    app.AISI4340steel6CheckBox.Value=1;

dib.AISI_4340=plot(app.Representacion,x.AISI_4340,y.AISI_4340,'x','MarkerSize',7,'MarkerEdgeColor',[0.1490,0.1
490,0.1490],'MarkerFaceColor',[0.1490,0.1490,0.1490]); %Color: Negro
                end
                if situacion.En_25 ==0
                    situacion.En_25 =1;
                    app.En25steel7CheckBox.Value=1;

dib.En_25=plot(app.Representacion,x.En_25,y.En_25,'d','MarkerSize',7,'MarkerEdgeColor',[0.3020,0.7451,0.9333],
'MarkerFaceColor',[0.3020,0.7451,0.9333]); %Color: Azul claro
                end
                if situacion.NiCrMo8 ==0
                    situacion.NiCrMo8 =1;
                    app.NiCrMo8steel8CheckBox.Value=1;

dib.NiCrMo8=plot(app.Representacion,x.NiCrMo8,y.NiCrMo8,'p','MarkerSize',7,'MarkerEdgeColor',[1.0000,0.0745,0.
6510],'MarkerFaceColor',[1.0000,0.0745,0.6510]); %Color: Rosa
                end
                if situacion.CrMo4 ==0
                    situacion.CrMo4 =1;
                    app.CrMo4steel9CheckBox.Value=1;

dib.CrMo4=plot(app.Representacion,x.CrMo4,y.CrMo4,'o','MarkerSize',7,'MarkerEdgeColor',[0.6510,0.6510,0.6510],
'MarkerFaceColor',[0.6510,0.6510,0.6510]); %Color: Gris
                end
                if situacion.SAE_52100 ==0
                    situacion.SAE_52100 =1;
                    app.SAE52100steel10CheckBox.Value=1;

dib.SAE_52100=plot(app.Representacion,x.SAE_52100,y.SAE_52100,'*','MarkerSize',7,'MarkerEdgeColor',[0,0.4471,0
.7412],'MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
                end
                if situacion.zn_25_zr65 ==0
                    situacion.zn_25_zr65 =1;
                    app.Zn065Zr11CheckBox.Value=1;

dib.zn_25_zr65=plot(app.Representacion,x.zn_25_zr65,y.zn_25_zr65,'d','MarkerSize',7,'MarkerEdgeColor',[0.9490,
0.0392,0.0392],'MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
                end
                if situacion.zn_56_zr66 ==0
                    situacion.zn_56_zr66 =1;
                    app.Zn066Zr11CheckBox.Value=1;

dib.zn_56_zr66=plot(app.Representacion,x.zn_56_zr66,y.zn_56_zr66,'s','MarkerSize',7,'MarkerEdgeColor',[0.1765,
0.7098,0.4000],'MarkerFaceColor',[0.1765,0.7098,0.4000]); %color:Verde
                end
                if situacion.Mild_Steel ==0
                    situacion.Mild_Steel =1;
                    app.Mildsteel11CheckBox.Value=1;

dib.Mild_Steel=plot(app.Representacion,x.Mild_Steel,y.Mild_Steel,'d','MarkerSize',7,'MarkerEdgeColor',[0.6510,
0.6510,0.6510],'MarkerFaceColor',[0.6510,0.6510,0.6510]); %color:gris
                end
```

```
                if situacion.a_6061_T6==0
                    situacion.a_6061_T6=1;
                    app.T611CheckBox_3.Value=1;

dib.a_6061_T6=plot(app.Representacion,x.a_6061_T6,y.a_6061_T6,'o','MarkerSize',7,'MarkerEdgeColor',[0.9490,0.0
392,0.0392],'MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
                end
                if situacion.NiCr_alloy ==0
                    situacion.NiCr_alloy =1;
                    app.NiCralloysteel11CheckBox.Value=1;

dib.NiCr_alloy=plot(app.Representacion,x.NiCr_alloy,y.NiCr_alloy,'s','MarkerSize',7,'MarkerEdgeColor',[1.0000,
0.4118,0.1608],'MarkerFaceColor',[1.0000,0.4118,0.1608]); %color:naranja
                end
                if situacion.SAE_4130 ==0
                    situacion.SAE_4130 =1;
                    app.SAE413011CheckBox.Value=1;

dib.SAE_4130=plot(app.Representacion,x.SAE_4130,y.SAE_4130,'>','MarkerSize',7,'MarkerEdgeColor',[0.4941,0.1843
,0.5569],'MarkerFaceColor',[0.4941,0.1843,0.5569]); %color:morado
                end
                if situacion.S24_T3 ==0
                    situacion.S24_T3 =1;
                    app.ST311CheckBox.Value=1;

dib.S24_T3=plot(app.Representacion,x.S24_T3,y.S24_T3,'p','MarkerSize',7,'MarkerEdgeColor',[0.3255,0.7608,0.682
4],'MarkerFaceColor',[0.3255,0.7608,0.6824]); %color:Azul turquesa
                end
                if situacion.S75_T6 ==0
                    situacion.S75_T6 =1;
                    app.ST611CheckBox_2.Value=1;

dib.S75_T6=plot(app.Representacion,x.S75_T6,y.S75_T6,'*','MarkerSize',7,'MarkerEdgeColor',[0.8314,0.3725,0.486
3],'MarkerFaceColor',[0.8314,0.3725,0.4863]); %color:rojo
                end
                if situacion.a_2014_T6 ==0
                    situacion.a_2014_T6 =1;
                    app.T611CheckBox_2.Value=1;

dib.a_2014_T6=plot(app.Representacion,x.a_2014_T6,y.a_2014_T6,'o','MarkerSize',7,'MarkerEdgeColor',[0.2667,0.6
784,0.1294],'MarkerFaceColor',[0.2667,0.6784,0.1294]); %color:Verde
                end
                if situacion.a_2024_T4 ==0
                    situacion.a_2024_T4 =1;
                    app.T411CheckBox.Value=1;

dib.a_2024_T4=plot(app.Representacion,x.a_2024_T4,y.a_2024_T4,'h','MarkerSize',7,'MarkerEdgeColor',[0.6902,0.5
412,0.3569],'MarkerFaceColor',[0.6902,0.5412,0.3569]); %color:Marron
                end
                if situacion.a_7075_T6 ==0
                    situacion.a_7075_T6 =1;
                    app.T611CheckBox.Value=1;

dib.a_7075_T6=plot(app.Representacion,x.a_7075_T6,y.a_7075_T6,'v','MarkerSize',7,'MarkerEdgeColor',[0.8667,0.5
882,0.8784],'MarkerFaceColor',[0.8667,0.5882,0.8784]); %color:Morado
                end
                if situacion.s_24_T4 ==0
                    situacion.s_24_T4 =1;
                    app.ST411CheckBox.Value=1;

dib.s_24_T4=plot(app.Representacion,x.s_24_T4,y.s_24_T4,'h','MarkerSize',7,'MarkerEdgeColor',[0.9804,0.4902,0.
1373],'MarkerFaceColor',[0.9804,0.4902,0.1373]); %color:Naranja
                end
                if situacion.s_14_T6 ==0
                    situacion.s_14_T6 =1;
                    app.ST611CheckBox.Value=1;

dib.s_14_T6=plot(app.Representacion,x.s_14_T6,y.s_14_T6,'<','MarkerSize',7,'MarkerEdgeColor',[0.8000,0.5294,0.
0941],'MarkerFaceColor',[0.8000,0.5294,0.0941]); %color:Marron
                end
                if situacion.CrNiMo6_uniaxial==0
                    situacion.CrNiMo6_uniaxial=1;
                    app.CrNiMo6steel1CheckBox.Value=1;
```

```
dib.CrNiMo6_uniaxial=plot(app.Representacion,x.CrNiMo6_uniaxial,y.CrNiMo6_uniaxial,'o','MarkerSize',7,'MarkerE
dgeColor',[0.8392,0.8510,0.2980],'MarkerFaceColor',[0.8392,0.8510,0.2980]); %Color: Amarillo
                end

            end
        end

        % Value changed function: Unselectall_uniaxialCheckBox
        function Unselectall_uniaxialCheckBoxValueChanged2(app, event)
%             value = app.Unselectall_uniaxialCheckBox.Value;

            global situacion
            global dib

            situacion.unselect_all_uniaxial = app.Unselectall_uniaxialCheckBox.Value;

            if situacion.unselect_all_uniaxial==1
                situacion.select_all_uniaxial =0;
                app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;

                if situacion.CrNiMo6_uniaxial ==1
                    situacion.CrNiMo6_uniaxial=0;
                    app.CrNiMo6steel1CheckBox.Value=0;
                    delete(dib.CrNiMo6_uniaxial)
                end
                if situacion.c_steel ==1
                    situacion.c_steel=0;
                    app.Csteel2CheckBox.Value=0;
                    delete(dib.c_steel)
                end
                if situacion.carb_steel_013 ==1
                    situacion.carb_steel_013 =0;
                    app.Carbonsteel3CheckBox.Value=0;
                    delete(dib.carb_steel_013)
                end
                if situacion.carb_steel_029 ==1
                    situacion.carb_steel_029 =0;
                    app.Carbonsteel4CheckBox.Value=0;
                    delete(dib.carb_steel_029)
                end
                if situacion.AISI_4130 ==1
                    situacion.AISI_4130 =0;
                    app.AISI4130steel5CheckBox.Value=0;
                    delete(dib.AISI_4130)
                end
                if situacion.AISI_4340 ==1
                    situacion.AISI_4340 =0;
                    app.AISI4340steel6CheckBox.Value=0;
                    delete(dib.AISI_4340)
                end
                if situacion.En_25 ==1
                    situacion.En_25 =0;
                    app.En25steel7CheckBox.Value=0;
                    delete(dib.En_25)
                end
                if situacion.NiCrMo8 ==1
                    situacion.NiCrMo8 =0;
                    app.NiCrMo8steel8CheckBox.Value=0;
                    delete(dib.NiCrMo8)
                end
                if situacion.CrMo4 ==1
                    situacion.CrMo4 =0;
                    app.CrMo4steel9CheckBox.Value=0;
                    delete(dib.CrMo4)
                end
                if situacion.SAE_52100 ==1
                    situacion.SAE_52100 =0;
                    app.SAE52100steel10CheckBox.Value=0;
                    delete(dib.SAE_52100)
                end
                if situacion.zn_25_zr65 ==1
                    situacion.zn_25_zr65 =0;
                    app.Zn065Zr11CheckBox.Value=0;
```

```
                delete(dib.zn_25_zr65)
            end
            if situacion.zn_56_zr66 ==1
                situacion.zn_56_zr66 =0;
                app.Zn066Zr11CheckBox.Value=0;
                delete(dib.zn_56_zr66)
            end
            if situacion.Mild_Steel ==1
                situacion.Mild_Steel =0;
                app.Mildsteel11CheckBox.Value=0;
                delete(dib.Mild_Steel)
            end
            if situacion.a_6061_T6==1
                situacion.a_6061_T6=0;
                app.T611CheckBox_3.Value=0;
                delete(dib.a_6061_T6)
            end
            if situacion.NiCr_alloy ==1
                situacion.NiCr_alloy =0;
                app.NiCralloysteel11CheckBox.Value=0;
                delete(dib.NiCr_alloy)
            end
            if situacion.SAE_4130 ==1
                situacion.SAE_4130 =0;
                app.SAE413011CheckBox.Value=0;
                delete(dib.SAE_4130)
            end
            if situacion.S24_T3 ==1
                situacion.S24_T3 =0;
                app.ST311CheckBox.Value=0;
                delete(dib.S24_T3)
            end
            if situacion.S75_T6 ==1
                situacion.S75_T6 =0;
                app.ST611CheckBox_2.Value=0;
                delete(dib.S75_T6)
            end
            if situacion.a_2014_T6 ==1
                situacion.a_2014_T6 =0;
                app.T611CheckBox_2.Value=0;
                delete(dib.a_2014_T6)
            end
            if situacion.a_2024_T4 ==1
                situacion.a_2024_T4 =0;
                app.T411CheckBox.Value=0;
                delete(dib.a_2024_T4)
            end
            if situacion.a_7075_T6 ==1
                situacion.a_7075_T6 =0;
                app.T611CheckBox.Value=0;
                delete(dib.a_7075_T6)
            end
            if situacion.s_24_T4 ==1
                situacion.s_24_T4 =0;
                app.ST411CheckBox.Value=0;
                delete(dib.s_24_T4)
            end
            if situacion.s_14_T6 ==1
                situacion.s_14_T6 =0;
                app.ST611CheckBox.Value=0;
                delete(dib.s_14_T6)
            end

        end
    end

    % Value changed function: Selectall_torsionCheckBox
    function Selectall_torsionCheckBoxValueChanged(app, event)
%         value = app.Selectall_torsionCheckBox.Value;
        global situacion
        global dib_t
        global x
        global y
```

```
            situacion.select_all_torsion = app.Selectall_torsionCheckBox.Value;

            if situacion.select_all_torsion ==1
                situacion.unselect_all_torsion = 0;
                app.Unselectall_torsionCheckBox.Value=situacion.unselect_all_torsion ;
                if situacion.C_12norm ==0
                situacion.C_12norm =1;
                app.Cnormalizedsteel13CheckBox_2.Value=1;

dib_t.C_12norm=plot(app.Representacion_torsion,x.C_12norm,y.C_12norm,'d','MarkerSize',7,'MarkerEdgeColor',[0,0
.4471,0.7412],'MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
                end
                if situacion.C_49norm ==0
                situacion.C_49norm =1;
                app.Cnormalizedsteel13CheckBox.Value=1;

dib_t.C_49norm=plot(app.Representacion_torsion,x.C_49norm,y.C_49norm,'*','MarkerSize',7,'MarkerEdgeColor','k',
'MarkerFaceColor','k'); %color:Negro
                end
                if situacion.En_25T ==0
                situacion.En_25T =1;
                app.En25TNICrMosteel14CheckBox.Value=1;

dib_t.En_25T=plot(app.Representacion_torsion,x.En_25T,y.En_25T,'*','MarkerSize',7,'MarkerEdgeColor',[0.6353,0.
0784,0.1843],'MarkerFaceColor',[0.6353,0.0784,0.1843]); %Color: Granate
                end
                if situacion.VDSiCr ==0
                situacion.VDSiCr =1;
                app.VDSiCrspringsteel15CheckBox.Value=1;

dib_t.VDSiCr=plot(app.Representacion_torsion,x.VDSiCr,y.VDSiCr,'d','MarkerSize',7,'MarkerEdgeColor',[0.6353,0.
0784,0.1843],'MarkerFaceColor',[0.6353,0.0784,0.1843]); %Color: Granate
                end
                if situacion.sae_3140_HR ==0
                situacion.sae_3140_HR =1;
                app.SAE3140HRsteel16CheckBox.Value=1;

dib_t.sae_3140_HR=plot(app.Representacion_torsion,x.sae_3140_HR,y.sae_3140_HR,'>','MarkerSize',7,'MarkerEdgeCo
lor',[1.0000,0.0745,0.6510],'MarkerFaceColor',[1.0000,0.0745,0.6510]); %Color: Rosa
                end
                if situacion.sae_3140_QT ==0
                situacion.sae_3140_QT =1;
                app.SAE3140QTsteel16CheckBox.Value=1;

dib_t.sae_3140_QT=plot(app.Representacion_torsion,x.sae_3140_QT,y.sae_3140_QT,'^','MarkerSize',7,'MarkerEdgeCo
lor',[0.4667,0.6745,0.1882],'MarkerFaceColor',[0.4667,0.6745,0.1882]); %Color: Verde
                end
                if situacion.C_49sorb ==0
                situacion.C_49sorb =1;
                app.Csorbiticsteel13CheckBox.Value=1;

dib_t.C_49sorb=plot(app.Representacion_torsion,x.C_49sorb,y.C_49sorb,'o','MarkerSize',7,'MarkerEdgeColor',[1,0
.6,0.6],'MarkerFaceColor',[1,0.6,0.6]); %Color: Salmon
                end
                if situacion.C_35Ni_A ==0
                situacion.C_35Ni_A =1;
                app.CNisteeltreatA13CheckBox.Value=1;
                dib_t.C_35Ni_A =plot(app.Representacion_torsion,x.C_35Ni_A ,y.C_35Ni_A
,'s','MarkerSize',7,'MarkerEdgeColor',[0.6353,0.0784,0.1843],'MarkerFaceColor',[0.6353,0.0784,0.1843]);
%Color: Granate
                end
                if situacion.C_35Ni_B ==0
                situacion.C_35Ni_B =1;
                app.CNisteeltreatB13CheckBox.Value=1;
                dib_t.C_35Ni_B =plot(app.Representacion_torsion,x.C_35Ni_B ,y.C_35Ni_B
,'s','MarkerSize',7,'MarkerEdgeColor',[0.4667,0.6745,0.1882],'MarkerFaceColor',[0.4667,0.6745,0.1882]);
%Color: Verde
                end
                if situacion.C_35Ni_D ==0
                situacion.C_35Ni_D =1;
                app.CNisteeltreatD13CheckBox.Value=1;
                dib_t.C_35Ni_D =plot(app.Representacion_torsion,x.C_35Ni_D ,y.C_35Ni_D
,'p','MarkerSize',7,'MarkerEdgeColor',[0.3255,0.7608,0.6824],'MarkerFaceColor',[0.3255,0.7608,0.6824]);
%color:Azul turquesa
```

```
                end
                if situacion.CrNi_D ==0
                situacion.CrNi_D =1;
                app.CrNisteeltreatD17CheckBox.Value=1;
                dib_t.CrNi_D =plot(app.Representacion_torsion,x.CrNi_D ,y.CrNi_D
,'<','MarkerSize',7,'MarkerEdgeColor',[0.6510,0.6510,0.6510],'MarkerFaceColor',[0.6510,0.6510,0.6510]);
%color:gris
                end
                if situacion.CK_35 ==0
                situacion.CK_35 =1;
                app.CK35steel23CheckBox.Value=1;
                dib_t.CK_35
=plot(app.Representacion_torsion,x.CK_35,y.CK_35,'h','MarkerSize',7,'MarkerEdgeColor',[0.8667,0.5882,0.8784],'
MarkerFaceColor',[0.8667,0.5882,0.8784]); %color:Morado
                end
                if situacion.Cr_Mo_42 ==0
                situacion.Cr_Mo_42 =1;
                app.CrMo4steel18CheckBox.Value=1;
                dib_t.Cr_Mo_42
=plot(app.Representacion_torsion,x.Cr_Mo_42,y.Cr_Mo_42,'v','MarkerSize',7,'MarkerEdgeColor',[0.6902,0.5412,0.3
569],'MarkerFaceColor',[0.6902,0.5412,0.3569]); %color:Marron
                end
                if situacion.NiCrMo3 ==0
                situacion.NiCrMo3 =1;
                app.NiCrMo3steel19CheckBox.Value=1;

dib_t.NiCrMo3=plot(app.Representacion_torsion,x.NiCrMo3,y.NiCrMo3,'o','MarkerSize',7,'MarkerEdgeColor',[0.4941
,0.1843,0.5569],'MarkerFaceColor',[0.4941,0.1843,0.5569]); %color:morado
                end
                if situacion.st_14_Al==0
                situacion.st_14_Al=1;
                app.STAluminium20CheckBox.Value=1;

dib_t.st_14_Al=plot(app.Representacion_torsion,x.st_14_Al,y.st_14_Al,'s','MarkerSize',7,'MarkerEdgeColor',[0,0
.4471,0.7412],'MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
                end
                if situacion.al_76 ==0
                situacion.al_76 =1;
                app.Aluminium76ST6121CheckBox.Value=1;

dib_t.al_76=plot(app.Representacion_torsion,x.al_76,y.al_76,'h','MarkerSize',7,'MarkerEdgeColor',[1.0000,0.074
5,0.6510],'MarkerFaceColor',[1.0000,0.0745,0.6510]); %Color: Rosa
                end
                if situacion.C_12sorb ==0
                situacion.C_12sorb =1;
                app.Csorbiticsteel17CheckBox.Value=1;

dib_t.C_12sorb=plot(app.Representacion_torsion,x.C_12sorb,y.C_12sorb,'<','MarkerSize',7,'MarkerEdgeColor',[0.8
000,0.5294,0.0941],'MarkerFaceColor',[0.8000,0.5294,0.0941]); %color:Marron
                end
                if situacion.CrMo4V ==0
                situacion.CrMo4V =1;
                app.CrMo4Vsteel22CheckBox.Value=1;

dib_t.CrMo4V=plot(app.Representacion_torsion,x.CrMo4V,y.CrMo4V,'s','MarkerSize',7,'MarkerEdgeColor',[0.3020,0.
7451,0.9333],'MarkerFaceColor',[0.3020,0.7451,0.9333]); %Color: Azul claro
                end
                if situacion.CrNiMo6_steel ==0
                situacion.CrNiMo6_steel =1;
                app.CrNiMo6steel12CheckBox.Value=1;

dib_t.CrNiMo6_steel=plot(app.Representacion_torsion,x.CrNiMo6_steel,y.CrNiMo6_steel,'o','MarkerSize',7,'Marker
EdgeColor',[0.8392,0.8510,0.2980],'MarkerFaceColor',[0.8392,0.8510,0.2980]); %Color: Amarillo
                end
                if situacion.JIS==0
                situacion.JIS=1;
                app.JISSCM440steel24CheckBox.Value=1;

dib_t.JIS=plot(app.Representacion_torsion,x.JIS,y.JIS,'v','MarkerSize',7,'MarkerEdgeColor',[0.4941,0.1843,0.55
69],'MarkerFaceColor',[0.4941,0.1843,0.5569]); %color:morado
                end
                if situacion.MnCr5 ==0
                situacion.MnCr5 =1;
                app.MnCr5steel25CheckBox.Value=1;
```

```
dib_t.MnCr5=plot(app.Representacion_torsion,x.MnCr5,y.MnCr5,'*','MarkerSize',7,'MarkerEdgeColor',[0,0.4471,0.7
412],'MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
                end
                if situacion.c_01 ==0
                situacion.c_01 =1;
                app.Csteel26CheckBox.Value=1;

dib_t.c_01=plot(app.Representacion_torsion,x.c_01,y.c_01,'o','MarkerSize',7,'MarkerEdgeColor',[0.1765,0.7098,0
.4000],'MarkerFaceColor',[0.1765,0.7098,0.4000]); %color:Verde
                end
                if situacion.ST_35 ==0
                situacion.ST_35 =1;
                app.St35steel27CheckBox.Value=1;

dib_t.ST_35=plot(app.Representacion_torsion,x.ST_35,y.ST_35,'h','MarkerSize',7,'MarkerEdgeColor',[0.9490,0.039
2,0.0392],'MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
                end
                if situacion.CrMo4_steel==0
                situacion.CrMo4_steel=1;
                app.CrMo4steel23CheckBox.Value=1;

dib_t.CrMo4_steel=plot(app.Representacion_torsion,x.CrMo4_steel,y.CrMo4_steel,'x','MarkerSize',7,'MarkerEdgeCo
lor',[0.1490,0.1490,0.1490],'MarkerFaceColor',[0.1490,0.1490,0.1490]); %Color: Negro
                end
                if situacion.bss_s65 ==0
                situacion.bss_s65 =1;
                app.BSSS65Asteel28CheckBox.Value=1;

dib_t.bss_s65=plot(app.Representacion_torsion,x.bss_s65,y.bss_s65,'^','MarkerSize',7,'MarkerEdgeColor',[0.9490
,0.0392,0.0392],'MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
                end
                if situacion.Cr4_steel==0
                situacion.Cr4_steel=1;
                app.Cr4steel29CheckBox.Value=1;

dib_t.Cr4_steel=plot(app.Representacion_torsion,x.Cr4_steel,y.Cr4_steel,'p','MarkerSize',7,'MarkerEdgeColor',[
0.9804,0.4902,0.1373],'MarkerFaceColor',[0.9804,0.4902,0.1373]); %color:Naranja
                end


            end
        end

        % Value changed function: Unselectall_torsionCheckBox
        function Unselectall_torsionCheckBoxValueChanged(app, event)
%             value = app.Unselectall_torsionCheckBox.Value;
            global situacion
            global dib_t

            situacion.unselect_all_torsion = app.Unselectall_torsionCheckBox.Value;

            if situacion.unselect_all_torsion ==1
                situacion.select_all_torsion = 0;
                app.Selectall_torsionCheckBox.Value=situacion.select_all_torsion;

                if situacion.C_12norm ==1
                situacion.C_12norm =0;
                app.Cnormalizedsteel13CheckBox_2.Value=0;
                delete(dib_t.C_12norm)
                end
                if situacion.C_49norm ==1
                situacion.C_49norm =0;
                app.Cnormalizedsteel13CheckBox.Value=0;
                delete(dib_t.C_49norm)
                end
                if situacion.En_25T ==1
                situacion.En_25T =0;
                app.En25TNICrMosteel14CheckBox.Value=0;
                delete(dib_t.En_25T)
                end
                if situacion.VDSiCr ==1
                situacion.VDSiCr =0;
                app.VDSiCrspringsteel15CheckBox.Value=0;
```

```
delete(dib_t.VDSiCr)
end
if situacion.sae_3140_HR ==1
situacion.sae_3140_HR =0;
app.SAE3140HRsteel16CheckBox.Value=0;
delete(dib_t.sae_3140_HR)
end
if situacion.sae_3140_QT ==1
situacion.sae_3140_QT =0;
app.SAE3140QTsteel16CheckBox.Value=0;
delete(dib_t.sae_3140_QT)
end
if situacion.C_49sorb ==1
situacion.C_49sorb =0;
app.Csorbiticsteel13CheckBox.Value=0;
delete(dib_t.C_49sorb)
end
if situacion.C_35Ni_A ==1
situacion.C_35Ni_A =0;
app.CNisteeltreatA13CheckBox.Value=0;
delete(dib_t.C_35Ni_A)
end
if situacion.C_35Ni_B ==1
situacion.C_35Ni_B =0;
app.CNisteeltreatB13CheckBox.Value=0;
delete(dib_t.C_35Ni_B)
end
if situacion.C_35Ni_D ==1
situacion.C_35Ni_D =0;
app.CNisteeltreatD13CheckBox.Value=0;
delete(dib_t.C_35Ni_D)
end
if situacion.CrNi_D ==1
situacion.CrNi_D =0;
app.CrNisteeltreatD17CheckBox.Value=0;
delete(dib_t.CrNi_D)
end
if situacion.CK_35 ==1
situacion.CK_35 =0;
app.CK35steel23CheckBox.Value=0;
delete(dib_t.CK_35)
end
if situacion.Cr_Mo_42 ==1
situacion.Cr_Mo_42 =0;
app.CrMo4steel18CheckBox.Value=0;
delete(dib_t.Cr_Mo_42)
end
if situacion.NiCrMo3 ==1
situacion.NiCrMo3 =0;
app.NiCrMo3steel19CheckBox.Value=0;
delete(dib_t.NiCrMo3)
end
if situacion.st_14_Al==1
situacion.st_14_Al=0;
app.STAluminium20CheckBox.Value=0;
delete(dib_t.st_14_Al)
end
if situacion.al_76 ==1
situacion.al_76 =0;
app.Aluminium76ST6121CheckBox.Value=0;
delete(dib_t.al_76)
end
if situacion.C_12sorb ==1
situacion.C_12sorb =0;
app.Csorbiticsteel17CheckBox.Value=0;
delete(dib_t.C_12sorb)
end
if situacion.CrMo4V ==1
situacion.CrMo4V =0;
app.CrMo4Vsteel22CheckBox.Value=0;
delete(dib_t.CrMo4V)
end
if situacion.CrNiMo6_steel ==1
situacion.CrNiMo6_steel =0;
```

```
                app.CrNiMo6steel12CheckBox.Value=0;
                delete(dib_t.CrNiMo6_steel)
                end
                if situacion.JIS==1
                situacion.JIS=0;
                app.JISSCM440steel24CheckBox.Value=0;
                delete(dib_t.JIS)
                end
                if situacion.MnCr5 ==1
                situacion.MnCr5 =0;
                app.MnCr5steel25CheckBox.Value=0;
                delete(dib_t.MnCr5)
                end
                if situacion.c_01 ==1
                situacion.c_01 =0;
                app.Csteel26CheckBox.Value=0;
                delete(dib_t.c_01)
                end
                if situacion.ST_35 ==1
                situacion.ST_35 =0;
                app.St35steel27CheckBox.Value=0;
                delete(dib_t.ST_35)
                end
                if situacion.CrMo4_steel==1
                situacion.CrMo4_steel=0;
                app.CrMo4steel23CheckBox.Value=0;
                delete(dib_t.CrMo4_steel)
                end
                if situacion.bss_s65 ==1
                situacion.bss_s65 =0;
                app.BSSS65Asteel28CheckBox.Value=0;
                delete(dib_t.bss_s65)
                end
                if situacion.Cr4_steel==1
                situacion.Cr4_steel=0;
                app.Cr4steel29CheckBox.Value=0;
                delete(dib_t.Cr4_steel)
                end

            end
        end

        % Button pushed function: Button
        function ButtonPushed2(app, event)


            mensaje = {'[1] Pallarés Santasmartas, L., Albizuri Irigoyen, J., Avilés Ajuria, A., & Avilés
González, R. (2018). Mean Stress Effect on the Axial Fatigue Strength of DIN 34CrNiMo6 Quenched and Tempered
Steel. Metals, 4(213), 8.';...
                        ' ';...
                        '[2] Ukrainetz, P. (1960). The Effect of the Mean Stress on the Endurance Limit. The
University of Columbia.';...
                        ' ';...
                        '[3] Gough, H. (1924). The Fatigue of Metals.';...
                        ' ';...
                        '[4] Forrest, P. (1962). Fatigue of Metals. Pergamon Press Inc.';...
                        ' ';...
                        '[5] Grover, H., Bishop, S., & Jackson, L. (1951). Axial Load Fatigue Tests of
Unnotched Sheet Specimens of 24S-T3 and 75S-T6 Aluminum Alloys and SAE 4130 Steels. National Advisory
Committee for Aeronautics.';...
                        ' ';...
                        '[6] Trapp, W., & Schwartz, R. (1953). Elevated Temperature Fatigue Properties of SAE
4340 Steel.';...
                        ' ';...
                        '[7] O'Connor, H., & Morrison, J. (1956). The Effect of Mean Stress on the Push-Pull
Fatigue Properties of an Alloy Steel. In Proceedings of the International Conference on Fatigue of Metals,
102-109.';...
                        ' ';...
                        '[8] Grün, P., Troost, A., Akin, O., & Klubberg. (1991). Langzeitund
Dauerschwingfestigkeit des Vergütungsstahls 25CrMo4 bei mehrachsiger Beanspruchung durch dreischwingende
Lastspannungen. Materialwissenschaft Werkstofftechnik.';...
                        ' ';...
                        '[9] Schäfer, H., Hempen, M., Klubberg, F., & Beiss, P. (2001).
Mittelspannungsempfindlichkeit metallischer Werkstoffe bei schwingender Beanspruchung. Aachen.';...
```

```
                        ' ';...
                        '[10] Bomas, H., Bacher-Hoechst, M., Kienzler, R., Kunow, S., Loewisch, G., Muehleder,
& Schroeder, R. (2010). Crack initiation and endurance limit of a hard steel under multiaxial cyclic loads.
Fatigue Fract. Eng. Mater. Struct. ';...
                        ' ';...
                        '[11] Frost NE, M. K. (1974). Metal fatigue. Oxford: Oxford University Press.';...
                        ' ';...
                        '[12] Pallarés-Santasmartas, L., Albizuri, J., Avilés, A., Saintier, N., & Merzeau, J.
(2018). Influence of mean shear stress on the torsional fatigue behaviour of 34CrNiMo6 steel. International
Journal of Fatigue.';...
                        ' ';...
                        '[13] Jasper, T., & Moore, H. (1924). An investigation of the fatigue of metals.
University of Illinois Engineering Experiment Station.';...
                        ' ';...
                        '[14] WT, C. (1956). Fatigue strength in shear of an alloy steel, with particular
reference to the effect of mean stress and directional properties.';...
                        ' ';...
                        '[15] Mayer, H. (2015). Cyclic torsion very high cycle fatigue of VDSiCr spring steel
at different load ratios,» International Journal of Fatigue. ';...
                        ' ';...
                        '[16] JO, S. (1939). The effect of range of stress on the torsional fatigue strength of
steel. University of Illinois Engineering Experiment Station.';...
                        ' ';...
                        '[17] Moore, H., & Jasper, T. (1925). An investigation of the fatigue of metals. Series
of 1925. A report of the investigation. University of Illinois Engineering Experiment Station.';...
                        ' ';...
                        '[18] T., B. (1986). Festigkeitsverhalten von Stählen unter mehrachsiger
phasenverschobener Schwingbeanspruchung mit unterschiedlichen Schwingungsformen und Frequenzen.
Stuttgart.';...
                        ' ';...
                        '[19] Davoli, P. (2003). Independence of the torsional fatigue limit upon a mean shear
stress. International Journal of Fatigue.';...
                        ' ';...
                        '[20] Sauer, J. (1948). A study of fatigue phenomena under combined stress. Proceedings
of the seventh international congress for applied mechanics.';...
                        ' ';...
                        '[21] Findley, W. (1953). Combined-stress fatigue strength of 76S-T61 Aluminum alloy
with superimposed mean stresses and corrections for yielding.';...
                        ' ';...
                        '[22]  Lempp, W. (1977). Festigkeitsverhalten von Stählen bei mehrachsiger
Dauerschwingbeanspruchung durch Normalspannungen mit überlagerten phasengleichen und phasenverschobenen
Schubspannungen. University of Stuttgart.';...
                        ' ';...
                        '[23] Baier, F. (1970). Zeit- und Dauerfestigkeit bei überlagerter statischer und
schwingender Zug-Druck- und Torsionbeanspruchung. Stuttgart.';...
                        ' ';...
                        '[24] The Society of Materials Science. (1996). JSMS Databook: Databook on fatigue
strength of metallic materials.';...
                        ' ';...
                        '[25] Lüpfert, H.-P., & Spies, H.-J. (2001). EinfluB von Druckvorspannungen auf die
Dauerfestigkeit metallischer Werkstoffe bei ein-und mehrachsiger Beanspruchung. Mat-wiss u Werkstofftech.';...
                        ' ';...
                        '[26] Issler, L. (1973). Festigkeitsverhalten metallischer Werkstoffe bei mehrachsiger
phasenverschobener Schwingbeanspruchung. Stuttgart.';...
                        ' ';...
                        '[27] Gough, H., Pollard, H., & Clenshaw, W. (1951). Some experiments on the resistance
of metals to fatigue under combined stresses. Aeronautical Research Council reports and memoranda.';...
                        ' ';...
                        '[28] Richter, H. (1984). Schubspannungsintensitätshypothese –weitere experimentelle
und theoretische Untersuchungen. Konstruktion.'};

            uialert(app.pag_principal,mensaje,'Referencias de los ensayos experimentales de cada
material','Icon','info');


        end

        % Value changed function: EditField_3
        function EditField_3ValueChanged2(app, event)

            global lf_ax_alt
            global k
            global lf_tors_alt
            global s_ut
```

```
        global s_0_m
        global s_0_g
        global lf_tr_pul

        if app.EditField_3.Value>0
            lf_ax_alt =app.EditField_3.Value;
            lf_tors_alt=lf_ax_alt/k;
            app.EditField_4.Value=lf_tors_alt;

            s_0_m=sqrt((lf_ax_alt^2*s_ut^2)/(lf_ax_alt^2+s_ut^2));
            app.s_0_Marin.Value=s_0_m;
            s_0_g=(lf_ax_alt*s_ut)/(lf_ax_alt+s_ut);
            app.s_0_Goodman.Value=s_0_g;

            if lf_tr_pul<s_0_g || lf_tr_pul>s_0_m
                lf_tr_pul=s_0_g;
                app.EditField_5.Value=lf_tr_pul;
            end

            %Se enciende la lampara en naranja para indicar que los calculos no están actualizados
            app.Lamp_calcular.Color='[0.93,0.69,0.13]'; %Naranja


        else
            app.EditField_3.Value=lf_ax_alt;
            uialert(app.pag_principal,'Se deben introducir datos numéricos positivos','Datos no válidos');
        end
    end

    % Value changed function: kEditField
    function kEditFieldValueChanged2(app, event)

        global k
        global lf_ax_alt
        global lf_tors_alt
        global s_0_m
        global s_0_g
        global s_ut

        if app.kEditField.Value>0
            k =app.kEditField.Value;
            lf_tors_alt=lf_ax_alt/k;
            app.EditField_4.Value=lf_tors_alt;

            s_0_m=sqrt((lf_ax_alt^2*s_ut^2)/(lf_ax_alt^2+s_ut^2));
            app.s_0_Marin.Value=s_0_m;
            s_0_g=(lf_ax_alt*s_ut)/(lf_ax_alt+s_ut);
            app.s_0_Goodman.Value=s_0_g;

            %Se enciende la lampara en naranja para indicar que los calculos no están actualizados
            app.Lamp_calcular.Color='[0.93,0.69,0.13]'; %Naranja

        else
            app.kEditField.Value=k;
            uialert(app.pag_principal,'Se deben introducir datos numéricos positivos','Datos no válidos');
        end
    end

    % Value changed function: EditField_4
    function EditField_4ValueChanged2(app, event)

        global k
        global lf_ax_alt
        global lf_tors_alt
        global s_0_m
        global s_0_g
        global s_ut

        if app.EditField_4.Value>0
            lf_tors_alt=app.EditField_4.Value;
            k=lf_ax_alt/lf_tors_alt;
            app.kEditField.Value=k;

            s_0_m=sqrt((lf_ax_alt^2*s_ut^2)/(lf_ax_alt^2+s_ut^2));
```

```
                app.s_0_Marin.Value=s_0_m;
                s_0_g=(lf_ax_alt*s_ut)/(lf_ax_alt+s_ut);
                app.s_0_Goodman.Value=s_0_g;

                %Se enciende la lampara en naranja para indicar que los calculos no están actualizados
                app.Lamp_calcular.Color='[0.93,0.69,0.13]'; %Naranja

            else
                app.EditField_4.Value=lf_tors_alt;
                uialert(app.pag_principal,'Se deben introducir datos numéricos positivos','Datos no válidos');
            end
        end

        % Value changed function: EditField_5
        function EditField_5ValueChanged2(app, event)

            global lf_tr_pul
            global s_0_g
            global s_0_m

            if app.EditField_5.Value>0
                lf_tr_pul =app.EditField_5.Value;

                if lf_tr_pul<s_0_g || lf_tr_pul>s_0_m
                    mensaje=sprintf('El valor del límite de fatiga de tracción pulsante debería estar
comprendido entre los límites definidos por los puntos de corte con las curvas de Goodman y Marin');
                    uialert(app.pag_principal,mensaje,'Warning','Icon','warning');
                    lf_tr_pul=s_0_g;
                    app.EditField_5.Value=lf_tr_pul;
                end

                %Se enciende la lampara en naranja para indicar que los calculos no están actualizados
                app.Lamp_calcular.Color='[0.93,0.69,0.13]'; %Naranja

            else
                app.EditField_5.Value=lf_tr_pul;
                uialert(app.pag_principal,'Se deben introducir datos numéricos positivos','Datos no válidos');
            end
        end

        % Value changed function: EditField_6
        function EditField_6ValueChanged2(app, event)

            global s_ut
            global s_0_m
            global s_0_g
            global lf_ax_alt
            global lf_tr_pul

            if app.EditField_6.Value>0
                s_ut =app.EditField_6.Value;

                s_0_m=sqrt((lf_ax_alt^2*s_ut^2)/(lf_ax_alt^2+s_ut^2));
                app.s_0_Marin.Value=s_0_m;
                s_0_g=(lf_ax_alt*s_ut)/(lf_ax_alt+s_ut);
                app.s_0_Goodman.Value=s_0_g;
                if lf_tr_pul<s_0_g || lf_tr_pul>s_0_m
                    lf_tr_pul=s_0_g;
                    app.EditField_5.Value=lf_tr_pul;
                end

                %Se enciende la lampara en naranja para indicar que los calculos no están actualizados
                app.Lamp_calcular.Color='[0.93,0.69,0.13]'; %Naranja

            else
                app.EditField_6.Value=s_ut;
                uialert(app.pag_principal,'Se deben introducir datos numéricos positivos','Datos no válidos');
            end
        end

        % Value changed function: s_0_Goodman
        function s_0_GoodmanValueChanged(app, event)

            global lf_ax_alt
```

```
            global s_ut
            global s_0_g

            s_0_g=(lf_ax_alt*s_ut)/(lf_ax_alt+s_ut);
            app.s_0_Goodman.Value=s_0_g;

        end

        % Value changed function: s_0_Marin
        function s_0_MarinValueChanged(app, event)

            global lf_ax_alt
            global s_ut
            global s_0_m

            s_0_m=sqrt((lf_ax_alt^2*s_ut^2)/(lf_ax_alt^2+s_ut^2));
            app.s_0_Marin.Value=s_0_m;

        end

        % Value changed function: PuntosacalcularSpinner
        function PuntosacalcularSpinnerValueChanged2(app, event)

%             global puntos_discretizacion
%
%             puntos_discretizacion = app.PuntosacalcularSpinner.Value;

            %Se enciende la lampara en naranja para indicar que los calculos no están actualizados
            app.Lamp_calcular.Color='[0.93,0.69,0.13]'; %Naranja
        end

        % Value changed function: ToleranciaSpinner
        function ToleranciaSpinnerValueChanged2(app, event)

            global tolerancia

            tolerancia = app.ToleranciaSpinner.Value;

            %Se enciende la lampara en naranja para indicar que los calculos no están actualizados
            app.Lamp_calcular.Color='[0.93,0.69,0.13]'; %Naranja
        end

        % Button pushed function: Button_2
        function Button_2Pushed2(app, event)

            mensaje=sprintf(' Número de puntos en los que se discretiza el \n intérvalo -1< ?m/?ut <1');
            uialert(app.pag_principal,mensaje,'Información','Icon','info');

        end

        % Button pushed function: Button_3
        function Button_3Pushed2(app, event)

            mensaje=sprintf(' Precisión con la que se calcula el \n resultado numéricamente (Newton-
Raphson)');
            uialert(app.pag_principal,mensaje,'Información','Icon','info');

        end

        % Button pushed function: Button_4
        function Button_4Pushed(app, event)

            mensaje=sprintf('El valor del límite de fatiga de tracción pulsante debería estar comprendido
entre los límites definidos por los puntos de corte con las curvas de Goodman y Marin');
            uialert(app.pag_principal,mensaje,'Información','Icon','info');

        end

        % Value changed function: PapugaCheckBox
        function PapugaCheckBoxValueChanged(app, event)
%             value = app.PapugaCheckBox.Value;
            global situacion
            global entra
            global entra_t
```

```
            global s_m
            global s_a
            global t_m
            global t_a
            global puntos_discretizacion
            global dib
            global dib_t

            situacion.papuga = app.PapugaCheckBox.Value;

            if situacion.papuga==1

                hold(app.Representacion,'on');
                hold(app.Representacion_torsion,'on');
                %Se dibuja papuga
                    if entra.papuga==1

dib.papuga_1=plot(app.Representacion,s_m.papuga(1:(puntos_discretizacion+1),1),s_a.papuga(1:(puntos_discretiza
cion+1),1),'LineWidth',2,'Color',[0.50,0.50,0.50]); %Color: Gris oscuro

dib.papuga_2=plot(app.Representacion,s_m.papuga((puntos_discretizacion+1):end,1),s_a.papuga((puntos_discretiza
cion+1):end,1),'--','Color',[0.50,0.50,0.50],'LineWidth',2); %Color: Gris oscuro
                    else

dib.papuga=plot(app.Representacion,s_m.papuga,s_a.papuga,'LineWidth',2,'Color',[0.50,0.50,0.50]); %Color: Gris
oscuro
                    end
                    if entra_t.papuga==1

dib_t.papuga_1=plot(app.Representacion_torsion,t_m.papuga(1:(puntos_discretizacion+1),1),t_a.papuga(1:(puntos_
discretizacion+1),1),'LineWidth',2,'Color',[0.50,0.50,0.50]); %Color: Gris oscuro

dib_t.papuga_2=plot(app.Representacion_torsion,t_m.papuga((puntos_discretizacion+1):end,1),t_a.papuga((puntos_
discretizacion+1):end,1),'--','Color',[0.50,0.50,0.50],'LineWidth',2); %Color: Gris oscuro
                    else

dib_t.papuga=plot(app.Representacion_torsion,t_m.papuga,t_a.papuga,'LineWidth',2,'Color',[0.50,0.50,0.50]);
%Color: Gris oscuro
                    end

            else

                %Se borra
                if entra.papuga==1
                    delete(dib.papuga_1)
                    delete(dib.papuga_2)
                else
                    delete(dib.papuga)
                end
                if entra_t.papuga==1
                    delete(dib_t.papuga_1)
                    delete(dib_t.papuga_2)
                else
                    delete(dib_t.papuga)
                end
            end



        end

        % Value changed function: a_p_Papuga
        function a_p_PapugaValueChanged(app, event)
%             value = app.a_p_Papuga.Value;

        end

        % Value changed function: b_p_Papuga
        function b_p_PapugaValueChanged(app, event)
%             value = app.b_p_Papuga.Value;

        end

        % Button pushed function: CalcularButton
```

```
        function CalcularButtonPushed2(app, event)
    global lf_ax_alt
    global k
    global lf_tors_alt
    global s_ut
    global t_ut
    global lf_tr_pul
    global avance
    global avance_t
    global tolerancia
%     global mallado

    global alfa
    global beta
    global s_m
    global s_a
    global t_m
    global t_a
    global puntos
    global entra
    global entra_t
    global xmin_eje
    global xmax
    global xmax_t
    global xmax_eje
    global ymin_eje
    global ymax
    global ymax_t
    global ymax_eje
    global dib
    global dib_t
    global situacion
    global puntos_discretizacion
    global papuga

    %Se comienza actualizando el valor de los puntos discretizados
    puntos_discretizacion = app.PuntosacalcularSpinner.Value;

    %Ahora se borra el anterior
    %ejes
    delete(dib.axial_x)
    delete(dib.axial_x_flecha)
    delete(dib.axial_y)
    delete(dib.axial_y_flecha)
    delete(dib_t.torsion_x)
    delete(dib_t.torsion_x_flecha)
    delete(dib_t.torsion_y)
    delete(dib_t.torsion_y_flecha)
    %curvas
    if situacion.sines==1

        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');

        if situacion.tau==1
            delete(dib_t.sines_tau)
        else
            if situacion.zer==1
                if entra.sines_zer==1
                    delete(dib.sines_zer_1)
                    delete(dib.sines_zer_2)
                else
                    delete(dib.sines_zer)
                end
                delete(dib_t.sines)
            elseif situacion.ut==1
                if entra.sines_ut==1
                    delete(dib.sines_ut_1)
                    delete(dib.sines_ut_2)
                else
                    delete(dib.sines_ut)
                end
                delete(dib_t.sines_ut)
            end
```

```
    end
end

if situacion.crossland==1

    hold(app.Representacion,'on');
    hold(app.Representacion_torsion,'on');
    if situacion.tau==1
        if entra.crossland_tau==1
            delete(dib.crossland_tau_1)
            delete(dib.crossland_tau_2)
        else
            delete(dib.crossland_tau)
        end
        delete(dib_t.crossland_tau)
    elseif situacion.zer==1
        if entra.crossland_zer==1
            delete(dib.crossland_zer_1)
            delete(dib.crossland_zer_2)
        else
            delete(dib.crossland_zer)
        end
        delete(dib_t.crossland_zer)
    elseif situacion.ut==1
        if entra.crossland_ut==1
            delete(dib.crossland_ut_1)
            delete(dib.crossland_ut_2)
        else
            delete(dib.crossland_ut)
        end
        delete(dib_t.crossland_ut)
    end
end

if situacion.matake==1

    hold(app.Representacion,'on');
    hold(app.Representacion_torsion,'on');
    if situacion.tau==1
        if entra.matake_tau==1
            delete(dib.matake_tau_1)
            delete(dib.matake_tau_2)
        else
            delete(dib.matake_tau)
        end
            delete(dib_t.matake_tau)
    elseif situacion.zer==1
        if entra.matake_zer==1
            delete(dib.matake_zer_1)
            delete(dib.matake_zer_2)
        else
            delete(dib.matake_zer)
        end
        delete(dib_t.matake_zer)
    elseif situacion.ut==1
        delete(dib.matake_ut)
        delete(dib_t.matake_ut)
    end
end

if situacion.findley==1

    hold(app.Representacion,'on');
    hold(app.Representacion_torsion,'on');
    if situacion.tau==1
        if entra.findley_tau==1
            delete(dib.findley_tau_1)
            delete(dib.findley_tau_2)
        else
            delete(dib.findley_tau)
        end
        if entra_t.findley_tau==1
            delete(dib_t.findley_tau_1)
            delete(dib_t.findley_tau_2)
```

```
            else
                delete(dib_t.findley_tau)
            end
        elseif situacion.zer==1
            if entra.findley_zer==1
                delete(dib.findley_zer_1)
                delete(dib.findley_zer_2)
            else
                delete(dib.findley_zer)
            end
            if entra_t.findley_zer==1
                delete(dib_t.findley_zer_1)
                delete(dib_t.findley_zer_2)
            else
                delete(dib_t.findley_zer)
            end
        elseif situacion.ut==1
            if entra.findley_ut==1
                delete(dib.findley_ut_1)
                delete(dib.findley_ut_2)
            else
                delete(dib.findley_ut)
            end
            if entra_t.findley_ut==1
                delete(dib_t.findley_ut_1)
                delete(dib_t.findley_ut_2)
            else
                delete(dib_t.findley_ut)
            end
        end
    end

    if situacion.dangvan==1
        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');
        if situacion.tau==1
            if entra.dangvan_tau==1
                delete(dib.dangvan_tau_1)
                delete(dib.dangvan_tau_2)
            else
                delete(dib.dangvan_tau)
            end
            delete(dib_t.dangvan_tau)
        elseif situacion.zer==1
            if entra.dangvan_zer==1
                delete(dib.dangvan_zer_1)
                delete(dib.dangvan_zer_2)
            else
                delete(dib.dangvan_zer)
            end
            delete(dib_t.dangvan_zer)
        elseif situacion.ut==1
            delete(dib.dangvan_ut)
            delete(dib_t.dangvan_ut)
        end
    end

    if situacion.papuga==1

        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');
        if entra.papuga==1
            delete(dib.papuga_1)
            delete(dib.papuga_2)
        else
            delete(dib.papuga)
        end
        if entra_t.papuga==1
            delete(dib_t.papuga_1)
            delete(dib_t.papuga_2)
        else
            delete(dib_t.papuga)
        end
    end
```

```
    if situacion.goodman==1
        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');
        delete(dib.goodman)
        delete(dib_t.goodman)
    end

    if situacion.dietmann==1
        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');
        delete(dib.dietmann)
        delete(dib_t.dietmann)
    end

    if situacion.gerber==1
        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');
        delete(dib.gerber)
        delete(dib_t.gerber)
    end

    if situacion.marin==1
        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');
        delete(dib.marin)
        delete(dib_t.marin)
    end



    %se calcula el avance
    avance=round((2*s_ut)/puntos_discretizacion);


    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    UNIAXIAL PURO    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    SINES    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Parametros del metodo
    %Ensayo s_-1--tau_-1
        %No se puede sacar alfa
    beta.sines_tau=(sqrt(2)/3)*lf_ax_alt;
    %Ensayo s_-1--s_0
    alfa.sines_zer=sqrt(2)*((lf_ax_alt/lf_tr_pul)-1);
    beta.sines_zer=sqrt(2)/3*lf_ax_alt;
    %Ensayo s_-1--s_ut
    alfa.sines_ut=sqrt(2)*lf_ax_alt/s_ut;
    beta.sines_ut=sqrt(2)/3*lf_ax_alt;


        %Ensayo s_-1--tau_-1
            % No se puede sacar alfa
        %Ensayo s_-1--s_0
            [s_m.sines_zer,s_a.sines_zer,entra.sines_zer,puntos.sines_zer] =
calcula_sines_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.sines_zer,beta.sines_zer);
            if length(s_a.sines_zer)<=puntos_discretizacion
                entra.sines_zer=0;
            end
            %para los ejes
                if max(s_m.sines_zer(:,1))*1.2>xmax
                    xmax=max(s_m.sines_zer(:,1))*1.2;
                end

                if max(s_a.sines_zer(:,1))*1.2>ymax
                    ymax=max(s_a.sines_zer(:,1))*1.2;
                end
        %Ensayo s_-1--s_ut
            [s_m.sines_ut,s_a.sines_ut,entra.sines_ut,puntos.sines_ut] =
calcula_sines_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.sines_ut,beta.sines_ut);
            if length(s_a.sines_ut)<=puntos_discretizacion
                entra.sines_ut=0;
```

```
            end
            %para los ejes
                if max(s_m.sines_ut(:,1))*1.2>xmax
                    xmax=max(s_m.sines_ut(:,1))*1.2;
                end

                if max(s_a.sines_ut(:,1))*1.2>ymax
                    ymax=max(s_a.sines_ut(:,1))*1.2;
                end


    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    CROSSLAND    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Parametros del metodo
    %Ensayo s_-1--tau_-1
    alfa.crossland_tau=sqrt(2)*(sqrt(3)*(1/k)-1);
    beta.crossland_tau=sqrt(2/3)*lf_tors_alt;
    %Ensayo s_-1--s_0
    alfa.crossland_zer=sqrt(2)*(lf_tr_pul-lf_ax_alt)/(lf_ax_alt-2*lf_tr_pul);
    beta.crossland_zer=sqrt(2)/3*lf_ax_alt*(-lf_tr_pul)/(lf_ax_alt-2*lf_tr_pul);
    %Ensayo s_-1--s_ut
    alfa.crossland_ut=sqrt(2)*lf_ax_alt/(s_ut-lf_ax_alt);
    beta.crossland_ut=sqrt(2)/3*(s_ut*lf_ax_alt)/(s_ut-lf_ax_alt);


        %Ensayo s_-1--tau_-1
            [s_m.crossland_tau,s_a.crossland_tau,entra.crossland_tau,puntos.crossland_tau] =
calcula_crossland_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.crossland_tau,beta.crossla
nd_tau);
            if length(s_a.crossland_tau)<=puntos_discretizacion
                entra.crossland_tau=0;
            end
            %para los ejes
                if max(s_m.crossland_tau(:,1))*1.2>xmax
                    xmax=max(s_m.crossland_tau(:,1))*1.2;
                end

                if max(s_a.crossland_tau(:,1))*1.2>ymax
                    ymax=max(s_a.crossland_tau(:,1))*1.2;
                end
        %Ensayo s_-1--s_0
            [s_m.crossland_zer,s_a.crossland_zer,entra.crossland_zer,puntos.crossland_zer] =
calcula_crossland_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.crossland_zer,beta.crossla
nd_zer);
            if length(s_a.crossland_zer)<=puntos_discretizacion
                entra.crossland_zer=0;
            end
            %para los ejes
                if max(s_m.crossland_zer(:,1))*1.2>xmax
                    xmax=max(s_m.crossland_zer(:,1))*1.2;
                end

                if max(s_a.crossland_zer(:,1))*1.2>ymax
                    ymax=max(s_a.crossland_zer(:,1))*1.2;
                end
        %Ensayo s_-1--s_ut
            [s_m.crossland_ut,s_a.crossland_ut,entra.crossland_ut,puntos.crossland_ut] =
calcula_crossland_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.crossland_ut,beta.crosslan
d_ut);
            if length(s_a.crossland_ut)<=puntos_discretizacion
                entra.crossland_ut=0;
            end
            %para los ejes
                if max(s_m.crossland_ut(:,1))*1.2>xmax
                    xmax=max(s_m.crossland_ut(:,1))*1.2;
                end

                if max(s_a.crossland_ut(:,1))*1.2>ymax
                    ymax=max(s_a.crossland_ut(:,1))*1.2;
                end


    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    MATAKE    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    %Parametros del metodo
    %Ensayo s_-1--tau_-1
    alfa.matake_tau=2*(1/k)-1;
    beta.matake_tau=lf_tors_alt;
    %Ensayo s_-1--s_0
    alfa.matake_zer=(lf_tr_pul-lf_ax_alt)/(lf_ax_alt-2*lf_tr_pul);
    beta.matake_zer=(lf_ax_alt/2)*((-lf_tr_pul)/(lf_ax_alt-2*lf_tr_pul));
    %Ensayo s_-1--s_ut
    alfa.matake_ut=-(lf_ax_alt)/(lf_ax_alt-s_ut);%(s_ut-lf_ax_alt)/(lf_ax_alt);
    beta.matake_ut=(lf_ax_alt/2)*(-s_ut/(lf_ax_alt-s_ut));%s_ut/2;


        %Ensayo s_-1--tau_-1
            [s_m.matake_tau,s_a.matake_tau,entra.matake_tau,puntos.matake_tau] =
calcula_matake_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.matake_tau,beta.matake_tau);
            if length(s_a.matake_tau)<=puntos_discretizacion
                entra.matake_tau=0;
            end
            %para los ejes
                if max(s_m.matake_tau(:,1))*1.2>xmax
                    xmax=max(s_m.matake_tau(:,1))*1.2;
                end

                if max(s_a.matake_tau(:,1))*1.2>ymax
                    ymax=max(s_a.matake_tau(:,1))*1.2;
                end
        %Ensayo s_-1--s_0
            [s_m.matake_zer,s_a.matake_zer,entra.matake_zer,puntos.matake_zer] =
calcula_matake_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.matake_zer,beta.matake_zer);
            if length(s_a.matake_zer)<=puntos_discretizacion
                entra.matake_zer=0;
            end
            %para los ejes
                if max(s_m.matake_zer(:,1))*1.2>xmax
                    xmax=max(s_m.matake_zer(:,1))*1.2;
                end

                if max(s_a.matake_zer(:,1))*1.2>ymax
                    ymax=max(s_a.matake_zer(:,1))*1.2;
                end

        %Ensayo s_-1--s_ut
            [s_m.matake_ut,s_a.matake_ut,entra.matake_ut,puntos.matake_ut] =
calcula_matake_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.matake_ut,beta.matake_ut);
            if length(s_a.matake_ut)<=puntos_discretizacion
                entra.matake_ut=0;
            end
            %para los ejes
                if max(s_m.matake_ut(:,1))*1.2>xmax
                    xmax=max(s_m.matake_ut(:,1))*1.2;
                end

                if max(s_a.matake_ut(:,1))*1.2>ymax
                    ymax=max(s_a.matake_ut(:,1))*1.2;
                end



    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    FINDLEY    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Parametros del metodo
    %Ensayo s_-1--tau_-1
    alfa.findley_tau=(2-k)/(2*sqrt(k-1));
    beta.findley_tau=lf_ax_alt/(2*sqrt(k-1));
    %Ensayo s_-1--s_0
    alfa.findley_zer=calcula_alfa_findley_zer(lf_ax_alt,lf_tr_pul, tolerancia);
    beta.findley_zer=(lf_tr_pul/2)*(sqrt(4*alfa.findley_zer^2+1)+2*alfa.findley_zer);
    %Ensayo s_-1--s_ut
    alfa.findley_ut=(lf_ax_alt/2)*(sqrt(s_ut*(s_ut-lf_ax_alt)))/(s_ut*(s_ut-lf_ax_alt));
    beta.findley_ut=(lf_ax_alt/2)*(sqrt(s_ut*(s_ut-lf_ax_alt)))/(s_ut-lf_ax_alt);


        %Ensayo s_-1--tau_-1
```

```
            [s_m.findley_tau,s_a.findley_tau,entra.findley_tau,puntos.findley_tau] =
calcula_findley_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.findley_tau,beta.findley_tau
);
            if length(s_a.findley_tau)<=puntos_discretizacion
                entra.findley_tau=0;
            end
            %para los ejes
                if max(s_m.findley_tau(:,1))*1.2>xmax
                    xmax=max(s_m.findley_tau(:,1))*1.2;
                end

                if max(s_a.findley_tau(:,1))*1.2>ymax
                    ymax=max(s_a.findley_tau(:,1))*1.2;
                end
        %Ensayo s_-1--s_0
            [s_m.findley_zer,s_a.findley_zer,entra.findley_zer,puntos.findley_zer] =
calcula_findley_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.findley_zer,beta.findley_zer
);
            if length(s_a.findley_zer)<=puntos_discretizacion
                entra.findley_zer=0;
            end
            %para los ejes
                if max(s_m.findley_zer(:,1))*1.2>xmax
                    xmax=max(s_m.findley_zer(:,1))*1.2;
                end

                if max(s_a.findley_zer(:,1))*1.2>ymax
                    ymax=max(s_a.findley_zer(:,1))*1.2;
                end
        %Ensayo s_-1--s_ut
            [s_m.findley_ut,s_a.findley_ut,entra.findley_ut,puntos.findley_ut] =
calcula_findley_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.findley_ut,beta.findley_ut);
            if length(s_a.findley_ut)<=puntos_discretizacion
                entra.findley_ut=0;
            end
            %para los ejes
                if max(s_m.findley_ut(:,1))*1.2>xmax
                    xmax=max(s_m.findley_ut(:,1))*1.2;
                end

                if max(s_a.findley_ut(:,1))*1.2>ymax
                    ymax=max(s_a.findley_ut(:,1))*1.2;
                end


    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   DANG VAN   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Parametros del metodo
    %Ensayo s_-1--tau_-1
    alfa.dangvan_tau=3*((1/k)-0.5);
    beta.dangvan_tau=lf_tors_alt;
    %Ensayo s_-1--s_0
    alfa.dangvan_zer=(3/2)*(lf_tr_pul-lf_ax_alt)/(lf_ax_alt-2*lf_tr_pul);
    beta.dangvan_zer=(lf_ax_alt/2)*(-lf_tr_pul)/(lf_ax_alt-2*lf_tr_pul);
    %Ensayo s_-1--s_ut
    alfa.dangvan_ut=(3/2)*(-lf_ax_alt)/(lf_ax_alt-s_ut);
    beta.dangvan_ut=0.5*((-lf_ax_alt*s_ut)/(lf_ax_alt-s_ut));


        %Ensayo s_-1--tau_-1
            [s_m.dangvan_tau,s_a.dangvan_tau,entra.dangvan_tau,puntos.dangvan_tau] =
calcula_dangvan_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.dangvan_tau,beta.dangvan_tau
);
            if length(s_a.dangvan_tau)<=puntos_discretizacion
                entra.dangvan_tau=0;
            end
            %para los ejes
                if max(s_m.dangvan_tau(:,1))*1.2>xmax
                    xmax=max(s_m.dangvan_tau(:,1))*1.2;
                end

                if max(s_a.dangvan_tau(:,1))*1.2>ymax
                    ymax=max(s_a.dangvan_tau(:,1))*1.2;
                end
```

```
        %Ensayo s_-1--s_0
            [s_m.dangvan_zer,s_a.dangvan_zer,entra.dangvan_zer,puntos.dangvan_zer] =
calcula_dangvan_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.dangvan_zer,beta.dangvan_zer
);
            if length(s_a.dangvan_zer)<=puntos_discretizacion
                entra.dangvan_zer=0;
            end
            %para los ejes
                if max(s_m.dangvan_zer(:,1))*1.2>xmax
                    xmax=max(s_m.dangvan_zer(:,1))*1.2;
                end

                if max(s_a.dangvan_zer(:,1))*1.2>ymax
                    ymax=max(s_a.dangvan_zer(:,1))*1.2;
                end
        %Ensayo s_-1--s_ut
            [s_m.dangvan_ut,s_a.dangvan_ut,entra.dangvan_ut,puntos.dangvan_ut] =
calcula_dangvan_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa.dangvan_ut,beta.dangvan_ut);
            if length(s_a.dangvan_ut)<=puntos_discretizacion
                entra.dangvan_ut=0;
            end
            %para los ejes
                if max(s_m.dangvan_ut(:,1))*1.2>xmax
                    xmax=max(s_m.dangvan_ut(:,1))*1.2;
                end

                if max(s_a.dangvan_ut(:,1))*1.2>ymax
                    ymax=max(s_a.dangvan_ut(:,1))*1.2;
                end



    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    PAPUGA    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Parametros del metodo
    papuga.a_p=((4*k^2)/(4+k^2))^2;
    papuga.b_p=(8*lf_ax_alt*k^2*(4-k^2))/((4+k^2)^2);

    % Datos Papuga
    papuga.delta_angulo=20;

    %Se calcula Papuga
    [s_m.papuga,s_a.papuga,entra.papuga] =
calcula_papuga_uniaxial(puntos_discretizacion,s_ut,avance,lf_ax_alt,lf_tors_alt,lf_tr_pul,tolerancia,papuga.a_
p,papuga.b_p,papuga.delta_angulo);
    if length(s_a.papuga)<=puntos_discretizacion
        entra.papuga=0;
    end
    %para los ejes
    if max(s_m.papuga(:,1))*1.2>xmax
        xmax=max(s_m.papuga(:,1))*1.2;
    end

    if max(s_a.papuga(:,1))*1.2>ymax
        ymax=max(s_a.papuga(:,1))*1.2;
    end



    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    TORSION PURO    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    avance_t=round((2*t_ut)/puntos_discretizacion);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%    SINES    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        %Ensayo s_-1--tau_-1
            t_m.sines_tau=zeros(puntos_discretizacion,1);
            t_a.sines_tau=zeros(puntos_discretizacion,1);

            for ii=1:puntos_discretizacion
                if ii==1
                    t_m.sines_tau(ii)=-t_ut;
```

```
            elseif ii==puntos_discretizacion
                t_m.sines_tau(ii)=t_ut;
            else
                t_m.sines_tau(ii)=t_m.sines_tau(ii-1)+avance_t;
            end

            if ii~=1 && t_m.sines_tau(ii)~=0
                if t_m.sines_tau(ii-1)<0 && t_m.sines_tau(ii)>0
                    t_m.sines_tau(ii)=0;
                end
            end

            t_a.sines_tau(ii)=(3/sqrt(6))*beta.sines_tau;

        end
        %Se normalizan los datos
            t_m.sines_tau=t_m.sines_tau/t_ut;
            t_a.sines_tau=t_a.sines_tau/lf_tors_alt;
        %para los ejes
        if max(t_m.sines_tau(:,1))*1.2>xmax_t
            xmax_t=max(t_m.sines_tau(:,1))*1.2;
        end

        if max(t_a.sines_tau(:,1))*1.2>ymax_t
            ymax_t=max(t_a.sines_tau(:,1))*1.2;
        end

    %Ensayo s_-1--s_0
        t_m.sines_zer=t_m.sines_tau*t_ut;
        t_a.sines_zer=zeros(puntos_discretizacion,1);

        for ii=1:puntos_discretizacion
            t_a.sines_zer(ii)=(3/sqrt(6))*beta.sines_zer;
        end
        %Se normalizan los datos
            t_m.sines_zer=t_m.sines_zer/t_ut;
            t_a.sines_zer=t_a.sines_zer/lf_tors_alt;
        %para los ejes
        if max(t_m.sines_zer(:,1))*1.2>xmax_t
            xmax_t=max(t_m.sines_zer(:,1))*1.2;
        end

        if max(t_a.sines_zer(:,1))*1.2>ymax_t
            ymax_t=max(t_a.sines_zer(:,1))*1.2;
        end

    %Ensayo s_-1--s_ut
        t_m.sines_ut=t_m.sines_tau*t_ut;
        t_a.sines_ut=zeros(puntos_discretizacion,1);

        for ii=1:puntos_discretizacion
            t_a.sines_ut(ii)=(3/sqrt(6))*beta.sines_ut;
        end
        %Se normalizan los datos
            t_m.sines_ut=t_m.sines_ut/t_ut;
            t_a.sines_ut=t_a.sines_ut/lf_tors_alt;
        %para los ejes
        if max(t_m.sines_ut(:,1))*1.2>xmax_t
            xmax_t=max(t_m.sines_ut(:,1))*1.2;
        end

        if max(t_a.sines_ut(:,1))*1.2>ymax_t
            ymax_t=max(t_a.sines_ut(:,1))*1.2;
        end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   CROSSLAND   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Ensayo s_-1--tau_-1
        t_m.crossland_tau=t_m.sines_tau*t_ut;
        t_a.crossland_tau=zeros(puntos_discretizacion,1);

        for ii=1:puntos_discretizacion
            t_a.crossland_tau(ii)=(3/sqrt(6))*beta.crossland_tau;
        end
```

```
        %Se normalizan los datos
            t_m.crossland_tau=t_m.crossland_tau/t_ut;
            t_a.crossland_tau=t_a.crossland_tau/lf_tors_alt;
        %para los ejes
        if max(t_m.crossland_tau(:,1))*1.2>xmax_t
            xmax_t=max(t_m.crossland_tau(:,1))*1.2;
        end

        if max(t_a.crossland_tau(:,1))*1.2>ymax_t
            ymax_t=max(t_a.crossland_tau(:,1))*1.2;
        end

    %Ensayo s_-1--s_0
        t_m.crossland_zer=t_m.crossland_tau*t_ut;
        t_a.crossland_zer=zeros(puntos_discretizacion,1);

        for ii=1:puntos_discretizacion
            t_a.crossland_zer(ii)=(3/sqrt(6))*beta.crossland_zer;
        end
        %Se normalizan los datos
            t_m.crossland_zer=t_m.crossland_zer/t_ut;
            t_a.crossland_zer=t_a.crossland_zer/lf_tors_alt;
        %para los ejes
        if max(t_m.crossland_zer(:,1))*1.2>xmax_t
            xmax_t=max(t_m.crossland_zer(:,1))*1.2;
        end

        if max(t_a.crossland_zer(:,1))*1.2>ymax_t
            ymax_t=max(t_a.crossland_zer(:,1))*1.2;
        end

    %Ensayo s_-1--s_ut
        t_m.crossland_ut=t_m.crossland_tau*t_ut;
        t_a.crossland_ut=zeros(puntos_discretizacion,1);

        for ii=1:puntos_discretizacion
            t_a.crossland_ut(ii)=(3/sqrt(6))*beta.crossland_ut;
        end
        %Se normalizan los datos
            t_m.crossland_ut=t_m.crossland_ut/t_ut;
            t_a.crossland_ut=t_a.crossland_ut/lf_tors_alt;
        %para los ejes
        if max(t_m.crossland_ut(:,1))*1.2>xmax_t
            xmax_t=max(t_m.crossland_ut(:,1))*1.2;
        end

        if max(t_a.crossland_ut(:,1))*1.2>ymax_t
            ymax_t=max(t_a.crossland_ut(:,1))*1.2;
        end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   MATAKE   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Ensayo s_-1--tau_-1
        t_m.matake_tau=t_m.sines_tau*t_ut;
        t_a.matake_tau=zeros(puntos_discretizacion,1);

        for ii=1:puntos_discretizacion
            t_a.matake_tau(ii)=beta.matake_tau;
        end
        %Se normalizan los datos
            t_m.matake_tau=t_m.matake_tau/t_ut;
            t_a.matake_tau=t_a.matake_tau/lf_tors_alt;
        %para los ejes
        if max(t_m.matake_tau(:,1))*1.2>xmax_t
            xmax_t=max(t_m.matake_tau(:,1))*1.2;
        end

        if max(t_a.matake_tau(:,1))*1.2>ymax_t
            ymax_t=max(t_a.matake_tau(:,1))*1.2;
        end

    %Ensayo s_-1--s_0
        t_m.matake_zer=t_m.matake_tau*t_ut;
        t_a.matake_zer=zeros(puntos_discretizacion,1);
```

```
            for ii=1:puntos_discretizacion
                t_a.matake_zer(ii)=beta.matake_zer;
            end
            %Se normalizan los datos
                t_m.matake_zer=t_m.matake_zer/t_ut;
                t_a.matake_zer=t_a.matake_zer/lf_tors_alt;
            %para los ejes
            if max(t_m.matake_zer(:,1))*1.2>xmax_t
                xmax_t=max(t_m.matake_zer(:,1))*1.2;
            end

            if max(t_a.matake_zer(:,1))*1.2>ymax_t
                ymax_t=max(t_a.matake_zer(:,1))*1.2;
            end

        %Ensayo s_-1--s_ut
            t_m.matake_ut=t_m.matake_tau*t_ut;
            t_a.matake_ut=zeros(puntos_discretizacion,1);

            for ii=1:puntos_discretizacion
                t_a.matake_ut(ii)=beta.matake_ut;
            end
            %Se normalizan los datos
                t_m.matake_ut=t_m.matake_ut/t_ut;
                t_a.matake_ut=t_a.matake_ut/lf_tors_alt;
            %para los ejes
            if max(t_m.matake_ut(:,1))*1.2>xmax_t
                xmax_t=max(t_m.matake_ut(:,1))*1.2;
            end

            if max(t_a.matake_ut(:,1))*1.2>ymax_t
                ymax_t=max(t_a.matake_ut(:,1))*1.2;
            end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   FINDLEY   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Importante tener en cuenta que aqui el avance es avance_t porque t_ut<s_ut
        %Ensayo s_-1--tau_-1
            [t_m.findley_tau,t_a.findley_tau,entra_t.findley_tau,puntos.findley_tau] =
calcula_findley_torsion(puntos_discretizacion,t_ut,avance_t,lf_tors_alt,tolerancia,alfa.findley_tau,beta.findl
ey_tau);
            if length(t_a.findley_tau)<=puntos_discretizacion
                entra_t.findley_tau=0;
            end
            %para los ejes
                if max(t_m.findley_tau(:,1))*1.2>xmax_t
                    xmax_t=max(t_m.findley_tau(:,1))*1.2;
                end

                if max(t_a.findley_tau(:,1))*1.2>ymax_t
                    ymax_t=max(t_a.findley_tau(:,1))*1.2;
                end
        %Ensayo s_-1--s_0
            [t_m.findley_zer,t_a.findley_zer,entra_t.findley_zer,puntos.findley_zer] =
calcula_findley_torsion(puntos_discretizacion,t_ut,avance_t,lf_tors_alt,tolerancia,alfa.findley_zer,beta.findl
ey_zer);
            if length(t_a.findley_zer)<=puntos_discretizacion
                entra_t.findley_zer=0;
            end
            %para los ejes
                if max(t_m.findley_zer(:,1))*1.2>xmax_t
                    xmax_t=max(t_m.findley_zer(:,1))*1.2;
                end

                if max(t_a.findley_zer(:,1))*1.2>ymax_t
                    ymax_t=max(t_a.findley_zer(:,1))*1.2;
                end
        %Ensayo s_-1--t_ut
            [t_m.findley_ut,t_a.findley_ut,entra_t.findley_ut,puntos.findley_ut] =
calcula_findley_torsion(puntos_discretizacion,t_ut,avance_t,lf_tors_alt,tolerancia,alfa.findley_ut,beta.findle
y_ut);
            if length(t_a.findley_ut)<=puntos_discretizacion
                entra_t.findley_ut=0;
            end
            %para los ejes
                if max(t_m.findley_ut(:,1))*1.2>xmax_t
```

```
                xmax_t=max(t_m.findley_ut(:,1))*1.2;
            end

            if max(t_a.findley_ut(:,1))*1.2>ymax_t
                ymax_t=max(t_a.findley_ut(:,1))*1.2;
            end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   DANG VAN   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Ensayo s_-1--tau_-1
        t_m.dangvan_tau=t_m.sines_tau*t_ut;
        t_a.dangvan_tau=zeros(puntos_discretizacion,1);

        for ii=1:puntos_discretizacion
            t_a.dangvan_tau(ii)=beta.dangvan_tau;
        end
        %Se normalizan los datos
            t_m.dangvan_tau=t_m.dangvan_tau/t_ut;
            t_a.dangvan_tau=t_a.dangvan_tau/lf_tors_alt;
        %para los ejes
        if max(t_m.dangvan_tau(:,1))*1.2>xmax_t
            xmax_t=max(t_m.dangvan_tau(:,1))*1.2;
        end

        if max(t_a.dangvan_tau(:,1))*1.2>ymax_t
            ymax_t=max(t_a.dangvan_tau(:,1))*1.2;
        end

    %Ensayo s_-1--s_0
        t_m.dangvan_zer=t_m.dangvan_tau*t_ut;
        t_a.dangvan_zer=zeros(puntos_discretizacion,1);

        for ii=1:puntos_discretizacion
            t_a.dangvan_zer(ii)=beta.dangvan_zer;
        end
        %Se normalizan los datos
            t_m.dangvan_zer=t_m.dangvan_zer/t_ut;
            t_a.dangvan_zer=t_a.dangvan_zer/lf_tors_alt;
        %para los ejes
        if max(t_m.dangvan_zer(:,1))*1.2>xmax_t
            xmax_t=max(t_m.dangvan_zer(:,1))*1.2;
        end

        if max(t_a.dangvan_zer(:,1))*1.2>ymax_t
            ymax_t=max(t_a.dangvan_zer(:,1))*1.2;
        end

    %Ensayo s_-1--s_ut
        t_m.dangvan_ut=t_m.dangvan_tau*t_ut;
        t_a.dangvan_ut=zeros(puntos_discretizacion,1);

        for ii=1:puntos_discretizacion
            t_a.dangvan_ut(ii)=beta.dangvan_ut;
        end
        %Se normalizan los datos
            t_m.dangvan_ut=t_m.dangvan_ut/t_ut;
            t_a.dangvan_ut=t_a.dangvan_ut/lf_tors_alt;
        %para los ejes
        if max(t_m.dangvan_ut(:,1))*1.2>xmax_t
            xmax_t=max(t_m.dangvan_ut(:,1))*1.2;
        end

        if max(t_a.dangvan_ut(:,1))*1.2>ymax_t
            ymax_t=max(t_a.dangvan_ut(:,1))*1.2;
        end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   PAPUGA   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    %Se calcula Papuga
    [t_m.papuga,t_a.papuga,entra_t.papuga] =
calcula_papuga_torsion(puntos_discretizacion,t_ut,avance_t,lf_ax_alt,lf_tors_alt,lf_tr_pul,tolerancia,papuga.a
_p,papuga.b_p,papuga.delta_angulo);
    if length(t_a.papuga)<=puntos_discretizacion
        entra_t.papuga=0;
```

```
end

%para los ejes
    if max(t_m.papuga(:,1))*1.2>xmax_t
        xmax_t=max(t_m.papuga(:,1))*1.2;
    end

    if max(t_a.papuga(:,1))*1.2>ymax_t
        ymax_t=max(t_a.papuga(:,1))*1.2;
    end


%% Ahora se calculan los criterios de ajuste de fatiga uniaxial


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   DIETMANN   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%se calcula el avance para dietmann
intervalo=s_ut-(-s_ut);
avance_dietmann=round(intervalo/puntos_discretizacion);
if mod(s_ut,avance_dietmann)==0
    fin_dietmann=puntos_discretizacion;
else
    fin_dietmann=puntos_discretizacion+2;
end

    s_m.dietmann=zeros(fin_dietmann,1);
    s_a.dietmann=zeros(fin_dietmann,1);

n_e=0;
for ii=1:fin_dietmann
    if ii==1
        s_m.dietmann(ii)=-s_ut;
    else
        s_m.dietmann(ii)=s_m.dietmann(ii-1)+avance_dietmann;
        if s_m.dietmann(ii)<s_ut && ii==fin_dietmann
            fin_dietmann=fin_dietmann+1;
            s_m.dietmann=[s_m.dietmann;0];
            s_a.dietmann=[s_a.dietmann;0];
        end
        if s_m.dietmann(ii)>s_ut
            s_m.dietmann(ii)=s_ut;%Porque es donde corta segun la formula
            n_e=1;
        end
    end

    if ii~=1 && s_m.dietmann(ii)~=0
        if s_m.dietmann(ii-1)<0 && s_m.dietmann(ii)>0
            s_m.dietmann(ii)=0;
        end
    end

    s_a.dietmann(ii)=lf_ax_alt*sqrt(1-(s_m.dietmann(ii)/s_ut));

end

if n_e==0
    s_m.dietmann(ii+1)=s_ut;
    s_a.dietmann(ii+1)=lf_ax_alt*sqrt(1-(s_m.dietmann(ii+1)/s_ut));
end
%Se normalizan los datos
    s_m.dietmann=s_m.dietmann/s_ut;
    s_a.dietmann=s_a.dietmann/lf_ax_alt;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   GERBER   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%se calcula el avance para gerber
intervalo=s_ut-(-s_ut);
avance_gerber=round(intervalo/puntos_discretizacion);
if mod(s_ut,avance_gerber)==0
    fin_gerber=puntos_discretizacion;
else
    fin_gerber=puntos_discretizacion+2;
end
```

```
        s_m.gerber=zeros(fin_gerber,1);
        s_a.gerber=zeros(fin_gerber,1);

    n_e=0;
    for ii=1:fin_gerber
        if ii==1
            s_m.gerber(ii)=-s_ut;
        else
            s_m.gerber(ii)=s_m.gerber(ii-1)+avance_gerber;
            if s_m.gerber(ii)<s_ut && ii==fin_gerber
                fin_gerber=fin_gerber+1;
                s_m.gerber=[s_m.gerber;0];
                s_a.gerber=[s_a.gerber;0];
            end
            if s_m.gerber(ii)>s_ut
                s_m.gerber(ii)=s_ut;%Porque es donde corta segun la formula
                n_e=1;
            end
        end

        if ii~=1 && s_m.gerber(ii)~=0
            if s_m.gerber(ii-1)<0 && s_m.gerber(ii)>0
                s_m.gerber(ii)=0;
            end
        end

        s_a.gerber(ii)=lf_ax_alt*(1-(s_m.gerber(ii)/s_ut)^2);

    end

    if n_e==0
        s_m.gerber(ii+1)=s_ut;
        s_a.gerber(ii+1)=lf_ax_alt*(1-(s_m.gerber(ii+1)/s_ut)^2);
    end

    %Se normalizan los datos
        s_m.gerber=s_m.gerber/s_ut;
        s_a.gerber=s_a.gerber/lf_ax_alt;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   MARIN   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %se calcula el avance para marin
    intervalo=s_ut-(-s_ut);
    avance_marin=round(intervalo/puntos_discretizacion);
    if mod(s_ut,avance_marin)==0
        fin_marin=puntos_discretizacion;
    else
        fin_marin=puntos_discretizacion+2;
    end

        s_m.marin=zeros(fin_marin,1);
        s_a.marin=zeros(fin_marin,1);

    n_e=0;
    for ii=1:fin_marin
        if ii==1
            s_m.marin(ii)=-s_ut;
        else
            s_m.marin(ii)=s_m.marin(ii-1)+avance_marin;
            if s_m.marin(ii)<s_ut && ii==fin_marin
                fin_marin=fin_marin+1;
                s_m.marin=[s_m.marin;0];
                s_a.marin=[s_a.marin;0];
            end
            if s_m.marin(ii)>s_ut
                s_m.marin(ii)=s_ut;%Porque es donde corta segun la formula
                n_e=1;
            end
        end

        if ii~=1 && s_m.marin(ii)~=0
            if s_m.marin(ii-1)<0 && s_m.marin(ii)>0
                s_m.marin(ii)=0;
            end
        end
```

```
    s_a.marin(ii)=lf_ax_alt*sqrt(1-(s_m.marin(ii)/s_ut)^2);

end

if n_e==0
    s_m.marin(ii+1)=s_ut;
    s_a.marin(ii+1)=lf_ax_alt*sqrt(1-(s_m.marin(ii+1)/s_ut)^2);
end

%Se normalizan los datos
    s_m.marin=s_m.marin/s_ut;
    s_a.marin=s_a.marin/lf_ax_alt;

%Se hace el paralelismo con torsión

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   DIETMANN   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%se calcula el avance para dietmann
intervalo=t_ut-(-t_ut);
avance_dietmann=round(intervalo/puntos_discretizacion);
if mod(t_ut,avance_dietmann)==0
    fin_dietmann=puntos_discretizacion;
else
    fin_dietmann=puntos_discretizacion+2;
end

    t_m.dietmann=zeros(fin_dietmann,1);
    t_a.dietmann=zeros(fin_dietmann,1);

n_e=0;
for ii=1:fin_dietmann
    if ii==1
        t_m.dietmann(ii)=-t_ut;
    else
        t_m.dietmann(ii)=t_m.dietmann(ii-1)+avance_dietmann;
        if t_m.dietmann(ii)<t_ut && ii==fin_dietmann
            fin_dietmann=fin_dietmann+1;
            t_m.dietmann=[t_m.dietmann;0];
            t_a.dietmann=[t_a.dietmann;0];
        end
        if t_m.dietmann(ii)>t_ut
            t_m.dietmann(ii)=t_ut;%Porque es donde corta segun la formula
            n_e=1;
        end
    end

    if ii~=1 && t_m.dietmann(ii)~=0
        if t_m.dietmann(ii-1)<0 && t_m.dietmann(ii)>0
            t_m.dietmann(ii)=0;
        end
    end

    t_a.dietmann(ii)=lf_tors_alt*sqrt(1-(t_m.dietmann(ii)/t_ut));

end
if n_e==0
    t_m.dietmann(ii+1)=t_ut;
    t_a.dietmann(ii+1)=lf_tors_alt*sqrt(1-(t_m.dietmann(ii+1)/t_ut));
end
%Se normalizan los datos
    t_m.dietmann=t_m.dietmann/t_ut;
    t_a.dietmann=t_a.dietmann/lf_tors_alt;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   GERBER   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%se calcula el avance para gerber
intervalo=t_ut-(-t_ut);
avance_gerber=round(intervalo/puntos_discretizacion);
if mod(t_ut,avance_gerber)==0
    fin_gerber=puntos_discretizacion;
else
    fin_gerber=puntos_discretizacion+2;
end

    t_m.gerber=zeros(fin_gerber,1);
```

```
    t_a.gerber=zeros(fin_gerber,1);

n_e=0;
for ii=1:fin_gerber
    if ii==1
        t_m.gerber(ii)=-t_ut;
    else
        t_m.gerber(ii)=t_m.gerber(ii-1)+avance_gerber;
        if t_m.gerber(ii)<t_ut && ii==fin_gerber
            fin_gerber=fin_gerber+1;
            t_m.gerber=[t_m.gerber;0];
            t_a.gerber=[t_a.gerber;0];
        end
        if t_m.gerber(ii)>t_ut
            t_m.gerber(ii)=t_ut;%Porque es donde corta segun la formula
            n_e=1;
        end
    end

    if ii~=1 && t_m.gerber(ii)~=0
        if t_m.gerber(ii-1)<0 && t_m.gerber(ii)>0
            t_m.gerber(ii)=0;
        end
    end

    t_a.gerber(ii)=lf_tors_alt*(1-(t_m.gerber(ii)/t_ut)^2);

end
if n_e==0
    t_m.gerber(ii+1)=t_ut;
    t_a.gerber(ii+1)=lf_tors_alt*(1-(t_m.gerber(ii+1)/t_ut)^2);
end
%Se normalizan los datos
    t_m.gerber=t_m.gerber/t_ut;
    t_a.gerber=t_a.gerber/lf_tors_alt;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   MARIN   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%se calcula el avance para marin
intervalo=t_ut-(-t_ut);
avance_marin=round(intervalo/puntos_discretizacion);
if mod(t_ut,avance_marin)==0
    fin_marin=puntos_discretizacion;
else
    fin_marin=puntos_discretizacion+2;
end

    t_m.marin=zeros(fin_marin,1);
    t_a.marin=zeros(fin_marin,1);

n_e=0;
for ii=1:fin_marin
    if ii==1
        t_m.marin(ii)=-t_ut;
    else
        t_m.marin(ii)=t_m.marin(ii-1)+avance_marin;
        if t_m.marin(ii)<t_ut && ii==fin_marin
            fin_marin=fin_marin+1;
            t_m.marin=[t_m.marin;0];
            t_a.marin=[t_a.marin;0];
        end
        if t_m.marin(ii)>t_ut
            t_m.marin(ii)=t_ut;%Porque es donde corta segun la formula
            n_e=1;
        end
    end

    if ii~=1 && t_m.marin(ii)~=0
        if t_m.marin(ii-1)<0 && t_m.marin(ii)>0
            t_m.marin(ii)=0;
        end
    end

    t_a.marin(ii)=lf_tors_alt*sqrt(1-(t_m.marin(ii)/t_ut)^2);
```

```
    end
    if n_e==0
        t_m.marin(ii+1)=t_ut;
        t_a.marin(ii+1)=lf_tors_alt*sqrt(1-(t_m.marin(ii+1)/t_ut)^2);
    end
    %Se normalizan los datos
        t_m.marin=t_m.marin/t_ut;
        t_a.marin=t_a.marin/lf_tors_alt;


    %%%%%%%%%%%%%%%%% REPRESENTACIÓN %%%%%%%%%%%%%%%%%


    if situacion.sines==1

        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');

        if situacion.tau==1
            %Se muestra el nuevo

dib_t.sines_tau=plot(app.Representacion_torsion,t_m.sines_tau,t_a.sines_tau,'LineWidth',2,'Color',[0.4941,0.18
43,0.5569]); %Color: Morado
            mensaje=sprintf('El método de Sines para el caso Uniaxial Puro no se puede calcular con la pareja
de ensayos seleccionada');
            uialert(app.pag_principal,mensaje,'Datos no válidos','Icon','warning');
        else

            %Se dibuja sines
            if situacion.zer==1
                if entra.sines_zer==1

dib.sines_zer_1=plot(app.Representacion,s_m.sines_zer(1:(puntos_discretizacion+1),1),s_a.sines_zer(1:(puntos_d
iscretizacion+1),1),'LineWidth',2,'Color',[0.4941,0.1843,0.5569]); %Color: Morado

dib.sines_zer_2=plot(app.Representacion,s_m.sines_zer((puntos_discretizacion+1):end,1),s_a.sines_zer((puntos_d
iscretizacion+1):end,1),'--','Color',[0.4941,0.1843,0.5569],'LineWidth',2); %Color: Morado
                else

dib.sines_zer=plot(app.Representacion,s_m.sines_zer,s_a.sines_zer,'LineWidth',2,'Color',[0.4941,0.1843,0.5569]
); %Color: Morado
                end

dib_t.sines_zer=plot(app.Representacion_torsion,t_m.sines_zer,t_a.sines_zer,'LineWidth',2,'Color',[0.4941,0.18
43,0.5569]); %Color: Morado
            elseif situacion.ut==1
                if entra.sines_ut==1

dib.sines_ut_1=plot(app.Representacion,s_m.sines_ut(1:(puntos_discretizacion+1),1),s_a.sines_ut(1:(puntos_disc
retizacion+1),1),'LineWidth',2,'Color',[0.4941,0.1843,0.5569]); %Color: Morado

dib.sines_ut_2=plot(app.Representacion,s_m.sines_ut((puntos_discretizacion+1):end,1),s_a.sines_ut((puntos_disc
retizacion+1):end,1),'--','Color',[0.4941,0.1843,0.5569],'LineWidth',2); %Color: Morado
                else

dib.sines_ut=plot(app.Representacion,s_m.sines_ut,s_a.sines_ut,'LineWidth',2,'Color',[0.4941,0.1843,0.5569]);
%Color: Morado
                end

dib_t.sines_ut=plot(app.Representacion_torsion,t_m.sines_ut,t_a.sines_ut,'LineWidth',2,'Color',[0.4941,0.1843,
0.5569]); %Color: Morado
            end
        end
    end

    if situacion.crossland==1

        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');
        %Se dibuja crossland
        if situacion.tau==1
            if entra.crossland_tau==1
```

```
dib.crossland_tau_1=plot(app.Representacion,s_m.crossland_tau(1:(puntos_discretizacion+1),1),s_a.crossland_tau
(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.5569,0.6588,0.4667]); %Color: Verdoso

dib.crossland_tau_2=plot(app.Representacion,s_m.crossland_tau((puntos_discretizacion+1):end,1),s_a.crossland_t
au((puntos_discretizacion+1):end,1),'--','Color',[0.5569,0.6588,0.4667],'LineWidth',2); %Color: Verdoso
            else

dib.crossland_tau=plot(app.Representacion,s_m.crossland_tau,s_a.crossland_tau,'LineWidth',2,'Color',[0.5569,0.
6588,0.4667]); %Color: Verdoso
            end

dib_t.crossland_tau=plot(app.Representacion_torsion,t_m.crossland_tau,t_a.crossland_tau,'LineWidth',2,'Color',
[0.5569,0.6588,0.4667]); %Color: Verdoso
        elseif situacion.zer==1
            if entra.crossland_zer==1

dib.crossland_zer_1=plot(app.Representacion,s_m.crossland_zer(1:(puntos_discretizacion+1),1),s_a.crossland_zer
(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.5569,0.6588,0.4667]); %Color: Verdoso

dib.crossland_zer_2=plot(app.Representacion,s_m.crossland_zer((puntos_discretizacion+1):end,1),s_a.crossland_z
er((puntos_discretizacion+1):end,1),'--','Color',[0.5569,0.6588,0.4667],'LineWidth',2); %Color: Verdoso
            else

dib.crossland_zer=plot(app.Representacion,s_m.crossland_zer,s_a.crossland_zer,'LineWidth',2,'Color',[0.5569,0.
6588,0.4667]); %Color: Verdoso
            end

dib_t.crossland_zer=plot(app.Representacion_torsion,t_m.crossland_zer,t_a.crossland_zer,'LineWidth',2,'Color',
[0.5569,0.6588,0.4667]); %Color: Verdoso
        elseif situacion.ut==1
            if entra.crossland_ut==1

dib.crossland_ut_1=plot(app.Representacion,s_m.crossland_ut(1:(puntos_discretizacion+1),1),s_a.crossland_ut(1:
(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.5569,0.6588,0.4667]); %Color: Verdoso

dib.crossland_ut_2=plot(app.Representacion,s_m.crossland_ut((puntos_discretizacion+1):end,1),s_a.crossland_ut(
(puntos_discretizacion+1):end,1),'--','Color',[0.5569,0.6588,0.4667],'LineWidth',2); %Color: Verdoso
            else

dib.crossland_ut=plot(app.Representacion,s_m.crossland_ut,s_a.crossland_ut,'LineWidth',2,'Color',[0.5569,0.658
8,0.4667]); %Color: Verdoso
            end

dib_t.crossland_ut=plot(app.Representacion_torsion,t_m.crossland_ut,t_a.crossland_ut,'LineWidth',2,'Color',[0.
5569,0.6588,0.4667]); %Color: Verdoso
        end
    end

    if situacion.matake==1

        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');
        %Se dibuja matake
        if situacion.tau==1
            if entra.matake_tau==1

dib.matake_tau_1=plot(app.Representacion,s_m.matake_tau(1:(puntos_discretizacion+1),1),s_a.matake_tau(1:(punto
s_discretizacion+1),1),'LineWidth',2,'Color',[0.4510,0.5490,0.7294]); %Color: Azul

dib.matake_tau_2=plot(app.Representacion,s_m.matake_tau((puntos_discretizacion+1):end,1),s_a.matake_tau((punto
s_discretizacion+1):end,1),'--','Color',[0.4510,0.5490,0.7294],'LineWidth',2); %Color: Azul
            else

dib.matake_tau=plot(app.Representacion,s_m.matake_tau,s_a.matake_tau,'LineWidth',2,'Color',[0.4510,0.5490,0.72
94]); %Color: Azul
            end

dib_t.matake_tau=plot(app.Representacion_torsion,t_m.matake_tau,t_a.matake_tau,'LineWidth',2,'Color',[0.4510,0
.5490,0.7294]); %Color: Azul
        elseif situacion.zer==1
            if entra.matake_zer==1
```

```
dib.matake_zer_1=plot(app.Representacion,s_m.matake_zer(1:(puntos_discretizacion+1),1),s_a.matake_zer(1:(punto
s_discretizacion+1),1),'LineWidth',2,'Color',[0.4510,0.5490,0.7294]); %Color: Azul

dib.matake_zer_2=plot(app.Representacion,s_m.matake_zer((puntos_discretizacion+1):end,1),s_a.matake_zer((punto
s_discretizacion+1):end,1),'--','Color',[0.4510,0.5490,0.7294],'LineWidth',2); %Color: Azul
            else

dib.matake_zer=plot(app.Representacion,s_m.matake_zer,s_a.matake_zer,'LineWidth',2,'Color',[0.4510,0.5490,0.72
94]); %Color: Azul
            end

dib_t.matake_zer=plot(app.Representacion_torsion,t_m.matake_zer,t_a.matake_zer,'LineWidth',2,'Color',[0.4510,0
.5490,0.7294]); %Color: Azul
        elseif situacion.ut==1

dib.matake_ut=plot(app.Representacion,s_m.matake_ut,s_a.matake_ut,'LineWidth',2,'Color',[0.4510,0.5490,0.7294]
); %Color: Azul

dib_t.matake_ut=plot(app.Representacion_torsion,t_m.matake_ut,t_a.matake_ut,'LineWidth',2,'Color',[0.4510,0.54
90,0.7294]); %Color: Azul
        end
    end

    if situacion.findley==1

        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');
        %Se dibuja findley
        if situacion.tau==1
            if entra.findley_tau==1

dib.findley_tau_1=plot(app.Representacion,s_m.findley_tau(1:(puntos_discretizacion+1),1),s_a.findley_tau(1:(pu
ntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib.findley_tau_2=plot(app.Representacion,s_m.findley_tau((puntos_discretizacion+1):end,1),s_a.findley_tau((pu
ntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
            else

dib.findley_tau=plot(app.Representacion,s_m.findley_tau,s_a.findley_tau,'LineWidth',2,'Color',[0.8510,0.3255,0
.0980]); %Color: Marron
            end
            if entra_t.findley_tau==1

dib_t.findley_tau_1=plot(app.Representacion_torsion,t_m.findley_tau(1:(puntos_discretizacion+1),1),t_a.findley
_tau(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib_t.findley_tau_2=plot(app.Representacion_torsion,t_m.findley_tau((puntos_discretizacion+1):end,1),t_a.findl
ey_tau((puntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
            else

dib_t.findley_tau=plot(app.Representacion_torsion,t_m.findley_tau,t_a.findley_tau,'LineWidth',2,'Color',[0.851
0,0.3255,0.0980]); %Color: Marron
            end
        elseif situacion.zer==1
            if entra.findley_zer==1

dib.findley_zer_1=plot(app.Representacion,s_m.findley_zer(1:(puntos_discretizacion+1),1),s_a.findley_zer(1:(pu
ntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib.findley_zer_2=plot(app.Representacion,s_m.findley_zer((puntos_discretizacion+1):end,1),s_a.findley_zer((pu
ntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
            else

dib.findley_zer=plot(app.Representacion,s_m.findley_zer,s_a.findley_zer,'LineWidth',2,'Color',[0.8510,0.3255,0
.0980]); %Color: Marron
            end
            if entra_t.findley_zer==1

dib_t.findley_zer_1=plot(app.Representacion_torsion,t_m.findley_zer(1:(puntos_discretizacion+1),1),t_a.findley
_zer(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib_t.findley_zer_2=plot(app.Representacion_torsion,t_m.findley_zer((puntos_discretizacion+1):end,1),t_a.findl
ey_zer((puntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
```

```
            else

dib_t.findley_zer=plot(app.Representacion_torsion,t_m.findley_zer,t_a.findley_zer,'LineWidth',2,'Color',[0.851
0,0.3255,0.0980]); %Color: Marron
            end
        elseif situacion.ut==1
            if entra.findley_ut==1

dib.findley_ut_1=plot(app.Representacion,s_m.findley_ut(1:(puntos_discretizacion+1),1),s_a.findley_ut(1:(punto
s_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib.findley_ut_2=plot(app.Representacion,s_m.findley_ut((puntos_discretizacion+1):end,1),s_a.findley_ut((punto
s_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
            else

dib.findley_ut=plot(app.Representacion,s_m.findley_ut,s_a.findley_ut,'LineWidth',2,'Color',[0.8510,0.3255,0.09
80]); %Color: Marron
            end
            if entra_t.findley_ut==1

dib_t.findley_ut_1=plot(app.Representacion_torsion,t_m.findley_ut(1:(puntos_discretizacion+1),1),t_a.findley_u
t(1:(puntos_discretizacion+1),1),'LineWidth',2,'Color',[0.8510,0.3255,0.0980]); %Color: Marron

dib_t.findley_ut_2=plot(app.Representacion_torsion,t_m.findley_ut((puntos_discretizacion+1):end,1),t_a.findley
_ut((puntos_discretizacion+1):end,1),'--','Color',[0.8510,0.3255,0.0980],'LineWidth',2); %Color: Marron
            else

dib_t.findley_ut=plot(app.Representacion_torsion,t_m.findley_ut,t_a.findley_ut,'LineWidth',2,'Color',[0.8510,0
.3255,0.0980]); %Color: Marron
            end
        end
    end

    if situacion.dangvan==1
        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');
        %Se dibuja dangvan
        if situacion.tau==1
            if entra.dangvan_tau==1

dib.dangvan_tau_1=plot(app.Representacion,s_m.dangvan_tau(1:(puntos_discretizacion+1),1),s_a.dangvan_tau(1:(pu
ntos_discretizacion+1),1),'LineWidth',2,'Color',[0.9294,0.6941,0.1255]); %Color: Naranja

dib.dangvan_tau_2=plot(app.Representacion,s_m.dangvan_tau((puntos_discretizacion+1):end,1),s_a.dangvan_tau((pu
ntos_discretizacion+1):end,1),'--','Color',[0.9294,0.6941,0.1255],'LineWidth',2); %Color: Naranja
            else

dib.dangvan_tau=plot(app.Representacion,s_m.dangvan_tau,s_a.dangvan_tau,'LineWidth',2,'Color',[0.9294,0.6941,0
.1255]); %Color: Naranja
            end

dib_t.dangvan_tau=plot(app.Representacion_torsion,t_m.dangvan_tau,t_a.dangvan_tau,'LineWidth',2,'Color',[0.929
4,0.6941,0.1255]); %Color: Naranja
        elseif situacion.zer==1
            if entra.dangvan_zer==1

dib.dangvan_zer_1=plot(app.Representacion,s_m.dangvan_zer(1:(puntos_discretizacion+1),1),s_a.dangvan_zer(1:(pu
ntos_discretizacion+1),1),'LineWidth',2,'Color',[0.9294,0.6941,0.1255]); %Color: Naranja

dib.dangvan_zer_2=plot(app.Representacion,s_m.dangvan_zer((puntos_discretizacion+1):end,1),s_a.dangvan_zer((pu
ntos_discretizacion+1):end,1),'--','Color',[0.9294,0.6941,0.1255],'LineWidth',2); %Color: Naranja
            else

dib.dangvan_zer=plot(app.Representacion,s_m.dangvan_zer,s_a.dangvan_zer,'LineWidth',2,'Color',[0.9294,0.6941,0
.1255]); %Color: Naranja
            end

dib_t.dangvan_zer=plot(app.Representacion_torsion,t_m.dangvan_zer,t_a.dangvan_zer,'LineWidth',2,'Color',[0.929
4,0.6941,0.1255]); %Color: Naranja
        elseif situacion.ut==1

dib.dangvan_ut=plot(app.Representacion,s_m.dangvan_ut,s_a.dangvan_ut,'LineWidth',2,'Color',[0.9294,0.6941,0.12
55]); %Color: Naranja
```

```
dib_t.dangvan_ut=plot(app.Representacion_torsion,t_m.dangvan_ut,t_a.dangvan_ut,'LineWidth',2,'Color',[0.9294,0
.6941,0.1255]); %Color: Naranja
        end
    end

    if situacion.papuga==1

        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');
        %Se dibuja papuga
        if entra.papuga==1

dib.papuga_1=plot(app.Representacion,s_m.papuga(1:(puntos_discretizacion+1),1),s_a.papuga(1:(puntos_discretiza
cion+1),1),'LineWidth',2,'Color',[0.50,0.50,0.50]); %Color: Gris oscuro

dib.papuga_2=plot(app.Representacion,s_m.papuga((puntos_discretizacion+1):end,1),s_a.papuga((puntos_discretiza
cion+1):end,1),'--','Color',[0.50,0.50,0.50],'LineWidth',2); %Color: Gris oscuro
        else
            dib.papuga=plot(app.Representacion,s_m.papuga,s_a.papuga,'LineWidth',2,'Color',[0.50,0.50,0.50]);
%Color: Gris oscuro
        end
        if entra_t.papuga==1

dib_t.papuga_1=plot(app.Representacion_torsion,t_m.papuga(1:(puntos_discretizacion+1),1),t_a.papuga(1:(puntos_
discretizacion+1),1),'LineWidth',2,'Color',[0.50,0.50,0.50]); %Color: Gris oscuro

dib_t.papuga_2=plot(app.Representacion_torsion,t_m.papuga((puntos_discretizacion+1):end,1),t_a.papuga((puntos_
discretizacion+1):end,1),'--','Color',[0.50,0.50,0.50],'LineWidth',2); %Color: Gris oscuro
        else

dib_t.papuga=plot(app.Representacion_torsion,t_m.papuga,t_a.papuga,'LineWidth',2,'Color',[0.50,0.50,0.50]);
%Color: Gris oscuro
        end
    end

    if situacion.goodman==1
        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');
        dib.goodman=plot(app.Representacion,[-1 1],[2 0],':','LineWidth',1.5,'Color',[0.8000,0.8000,0.8000]);
%Color: Gris
        dib_t.goodman=plot(app.Representacion_torsion,[-1 1],[2
0],':','LineWidth',1.5,'Color',[0.8000,0.8000,0.8000]); %Color: Gris
    end

    if situacion.dietmann==1
        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');
        dib.dietmann=plot(app.Representacion,s_m.dietmann,s_a.dietmann,'--
','LineWidth',1.5,'Color',[0.8000,0.8000,0.8000]); %Color: Gris
        dib_t.dietmann=plot(app.Representacion_torsion,t_m.dietmann,t_a.dietmann,'--
','LineWidth',1.5,'Color',[0.8000,0.8000,0.8000]); %Color: Gris
    end

    if situacion.gerber==1
        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');
        dib.gerber=plot(app.Representacion,s_m.gerber,s_a.gerber,'-
.','LineWidth',1.5,'Color',[0.8000,0.8000,0.8000]); %Color: Gris
        dib_t.gerber=plot(app.Representacion_torsion,t_m.gerber,t_a.gerber,'-
.','LineWidth',1.5,'Color',[0.8000,0.8000,0.8000]); %Color: Gris
    end

    if situacion.marin==1
        hold(app.Representacion,'on');
        hold(app.Representacion_torsion,'on');
        dib.marin=plot(app.Representacion,s_m.marin,s_a.marin,'-
','LineWidth',1.5,'Color',[0.8000,0.8000,0.8000]); %Color: Gris
        dib_t.marin=plot(app.Representacion_torsion,t_m.marin,t_a.marin,'-
','LineWidth',1.5,'Color',[0.8000,0.8000,0.8000]); %Color: Gris
    end

    %Para que los ejes sean cuadrados
    daspect(app.Representacion,[1 1 1]);
```

```
    daspect(app.Representacion_torsion,[1 1 1]);

    app.Representacion.XLim=[xmin_eje xmax_eje];
    app.Representacion.YLim=[ymin_eje ymax_eje];

    app.Representacion_torsion.XLim=[xmin_eje xmax_eje];
    app.Representacion_torsion.YLim=[ymin_eje ymax_eje];

    %Se añaden los ejes en líneas más estrechas y color negro
    %Representacion Axial Puro
    %eje x
    dib.axial_x=plot(app.Representacion,[-1.1 xmax*1.1/1.2],[0 0],'-','LineWidth',0.4,'Color','k');

dib.axial_x_flecha=plot(app.Representacion,xmax*1.1/1.2,0,'>','LineWidth',2,'MarkerSize',2,'MarkerEdgeColor','
k','MarkerFaceColor','k');
    %eje y
    dib.axial_y=plot(app.Representacion,[0 0],[0 ymax*1.1/1.2],'-','LineWidth',0.4,'Color','k');

dib.axial_y_flecha=plot(app.Representacion,0,ymax*1.1/1.2,'^','LineWidth',2,'MarkerSize',2,'MarkerEdgeColor','
k','MarkerFaceColor','k');

    %Representacion Torsion Puro
    %eje x
    dib_t.torsion_x=plot(app.Representacion_torsion,[-1.1 xmax_t*1.1/1.2],[0 0],'-
','LineWidth',0.4,'Color','k');

dib_t.torsion_x_flecha=plot(app.Representacion_torsion,xmax_t*1.1/1.2,0,'>','LineWidth',2,'MarkerSize',2,'Mark
erEdgeColor','k','MarkerFaceColor','k');
    %eje y
    dib_t.torsion_y=plot(app.Representacion_torsion,[0 0],[0 ymax_t*1.1/1.2],'-','LineWidth',0.4,'Color','k');

dib_t.torsion_y_flecha=plot(app.Representacion_torsion,0,ymax_t*1.1/1.2,'^','LineWidth',2,'MarkerSize',2,'Mark
erEdgeColor','k','MarkerFaceColor','k');

    %Grid on
    grid(app.Representacion,'on');
    xticks(app.Representacion,(-2:0.25:round(xmax*1.2)));
    yticks(app.Representacion,(-2:0.25:round(ymax*1.2)));
    grid(app.Representacion_torsion,'on');
    xticks(app.Representacion_torsion,(-2:0.25:round(xmax_t*1.2)));
    yticks(app.Representacion_torsion,(-2:0.25:round(ymax_t*1.2)));

    %Se muestran los valores de alfa y beta
    if situacion.tau==1
        app.AlfaSines.Value=0;
        app.BetaSines.Value=beta.sines_tau;
        app.AlfaCrossland.Value=alfa.crossland_tau;
        app.BetaCrossland.Value=beta.crossland_tau;
        app.AlfaMatake.Value=alfa.matake_tau;
        app.BetaMatake.Value=beta.matake_tau;
        app.AlfaFindley.Value=alfa.findley_tau;
        app.BetaFindley.Value=beta.findley_tau;
        app.AlfaDangVan.Value=alfa.dangvan_tau;
        app.BetaDangVan.Value=beta.dangvan_tau;
    elseif situacion.zer==1
        app.AlfaSines.Value=0;
        app.BetaSines.Value=beta.sines_zer;
        app.AlfaCrossland.Value=alfa.crossland_zer;
        app.BetaCrossland.Value=beta.crossland_zer;
        app.AlfaMatake.Value=alfa.matake_zer;
        app.BetaMatake.Value=beta.matake_zer;
        app.AlfaFindley.Value=alfa.findley_zer;
        app.BetaFindley.Value=beta.findley_zer;
        app.AlfaDangVan.Value=alfa.dangvan_zer;
        app.BetaDangVan.Value=beta.dangvan_zer;
    elseif situacion.ut==1
        app.AlfaSines.Value=0;
        app.BetaSines.Value=beta.sines_ut;
        app.AlfaCrossland.Value=alfa.crossland_ut;
        app.BetaCrossland.Value=beta.crossland_ut;
        app.AlfaMatake.Value=alfa.matake_ut;
        app.BetaMatake.Value=beta.matake_ut;
        app.AlfaFindley.Value=alfa.findley_ut;
        app.BetaFindley.Value=beta.findley_ut;
```

```
        app.AlfaDangVan.Value=alfa.dangvan_ut;
        app.BetaDangVan.Value=beta.dangvan_ut;
    end
    %Se muestran los valores de a_p y b_p
        app.a_p_Papuga.Value=papuga.a_p;
        app.b_p_Papuga.Value=papuga.b_p;


    %Se enciende la lampara en verde para indicar que los calculos se han actualizado
    app.Lamp_calcular.Color='[0.6039 0.8588 0.4196]'; %Verde


        end

        % Value changed function: CrNiMo6steel1CheckBox
        function CrNiMo6steel1CheckBoxValueChanged(app, event)
%             value = app.CrNiMo6steel1CheckBox.Value;
            global situacion
            global dib
            global x
            global y

            situacion.CrNiMo6_uniaxial = app.CrNiMo6steel1CheckBox.Value;

            situacion.select_all_uniaxial =0;
            app.Selectall_uniaxialCheckBox.Value=situacion.select_all_uniaxial;
            situacion.unselect_all_uniaxial =0;
            app.Unselectall_uniaxialCheckBox.Value=situacion.unselect_all_uniaxial;

            if situacion.CrNiMo6_uniaxial==1

dib.CrNiMo6_uniaxial=plot(app.Representacion,x.CrNiMo6_uniaxial,y.CrNiMo6_uniaxial,'o','MarkerSize',7,'MarkerE
dgeColor',[0.8392,0.8510,0.2980],'MarkerFaceColor',[0.8392,0.8510,0.2980]); %Color: Amarillo
            else
                delete(dib.CrNiMo6_uniaxial)
            end
        end

        % Button pushed function: ImportarDatosButton
        function ImportarDatosButtonPushed(app, event)
            global x_exc_uni
            global y_exc_uni
            global x_exc_tor
            global y_exc_tor
            global num_materiales
            global cont_uni
            global cont_tor
            global pos_uni
            global pos_tor
            global sit

            %Datos importados colocados a cero
            sit.mat_1_uni=0;
            app.Material1_uniCheckBox.Value=0;
            sit.mat_2_uni=0;
            app.Material2_uniCheckBox.Value=0;
            sit.mat_3_uni=0;
            app.Material3_uniCheckBox.Value=0;
            sit.mat_4_uni=0;
            app.Material4_uniCheckBox.Value
            sit.mat_5_uni=0;
            app.Material5_uniCheckBox.Value=0;
            sit.mat_6_uni=0;
            app.Material6_uniCheckBox.Value=0;
            sit.mat_7_uni=0;
            app.Material7_uniCheckBox.Value=0;
            sit.mat_8_uni=0;
            app.Material8_uniCheckBox.Value=0;
            sit.mat_9_uni=0;
            app.Material9_uniCheckBox.Value=0;
            sit.mat_10_uni=0;
            app.Material10_uniCheckBox.Value=0;
            sit.mat_11_uni=0;
            app.Material11_uniCheckBox.Value=0;
```

```
sit.mat_12_uni=0;
app.Material12_uniCheckBox.Value=0;
sit.mat_13_uni=0;
app.Material13_uniCheckBox.Value=0;
sit.mat_14_uni=0;
app.Material14_uniCheckBox.Value=0;
sit.mat_15_uni=0;
app.Material15_uniCheckBox.Value=0;
sit.mat_16_uni=0;
app.Material16_uniCheckBox.Value=0;
sit.mat_17_uni=0;
app.Material17_uniCheckBox.Value=0;
sit.mat_18_uni=0;
app.Material18_uniCheckBox.Value=0;
sit.mat_19_uni=0;
app.Material19_uniCheckBox.Value=0;
sit.mat_20_uni=0;
app.Material20_uniCheckBox.Value=0;
sit.mat_21_uni=0;
app.Material21_uniCheckBox.Value=0;
sit.mat_22_uni=0;
app.Material22_uniCheckBox.Value=0;
sit.mat_1_tor=0;
app.Material1_torCheckBox.Value=0;
sit.mat_2_tor=0;
app.Material2_torCheckBox.Value=0;
sit.mat_3_tor=0;
app.Material3_torCheckBox.Value=0;
sit.mat_4_tor=0;
app.Material4_torCheckBox.Value=0;
sit.mat_5_tor=0;
app.Material5_torCheckBox.Value=0;
sit.mat_6_tor=0;
app.Material6_torCheckBox.Value=0;
sit.mat_7_tor=0;
app.Material7_torCheckBox.Value=0;
sit.mat_8_tor=0;
app.Material8_torCheckBox.Value=0;
sit.mat_9_tor=0;
app.Material9_torCheckBox.Value=0;
sit.mat_10_tor=0;
app.Material10_torCheckBox.Value=0;
sit.mat_11_tor=0;
app.Material11_torCheckBox.Value=0;
sit.mat_12_tor=0;
app.Material12_torCheckBox.Value=0;
sit.mat_13_tor=0;
app.Material13_torCheckBox.Value=0;
sit.mat_14_tor=0;
app.Material14_torCheckBox.Value=0;
sit.mat_15_tor=0;
app.Material15_torCheckBox.Value=0;
sit.mat_16_tor=0;
app.Material16_torCheckBox.Value=0;
sit.mat_17_tor=0;
app.Material17_torCheckBox.Value=0;
sit.mat_18_tor=0;
app.Material18_torCheckBox.Value=0;
sit.mat_19_tor=0;
app.Material19_torCheckBox.Value=0;
sit.mat_20_tor=0;
app.Material20_torCheckBox.Value=0;
sit.mat_21_tor=0;
app.Material21_torCheckBox.Value=0;
sit.mat_22_tor=0;
app.Material22_torCheckBox.Value=0;

sit.select_all_exc_uni=0;
app.Sel_all_exc_uni.Value=0;
sit.unselect_all_exc_uni=0;
app.Unsel_all_exc_uni.Value=0;
sit.select_all_exc_tor=0;
app.Sel_all_exc_tor.Value=0;
sit.unselect_all_exc_tor=0;
```

```
            app.Unsel_all_exc_tor.Value=0;

            cont_uni=0;
            cont_tor=0;

            msg = ('El archivo Excel debe seguir el formato establecido por defecto');
            title = 'Información';
            seleccion = uiconfirm(app.pag_principal,msg,title,'Options',{'Seleccionar
archivo','Cancelar'},'DefaultOption',1,'CancelOption',2,'Icon', 'info');
            pathname=0;
            if strcmp(seleccion,'Seleccionar archivo')

                [filename, pathname, ~] = uigetfile({'*.xls;*.xlsx;*.xlsm;*.xltx;*.xltm','ARCHIVO EXCEL
(*.xls, *.xlsx, *.xlsm, *.xltx, *.xltm)'},'Selecciona el archivo que contiene los datos a importar');

                if pathname~=0
                    app.CargandoLabel.Visible="on";
                    app.DireccinLabel.Visible="on";
                    app.direccion_excelLabel.Visible="on";
                    app.direccion_excelLabel.Text=pathname;
                    app.NombreLabel.Visible="on";
                    app.nombre_excelLabel.Visible="on";
                    app.nombre_excelLabel.Text=filename;

                    [NUM,TXT,~] = xlsread([pathname filename]);%es como que lo almacena en un bloc
considerando solo los numeros (lo demas-->NaN)


                    num_materiales=NUM(1,2);
                    pos_uni=zeros(num_materiales,1);
                    pos_tor=pos_uni;


                    vect_horizontal=NUM(9,:);
                    ultima_pos_real=length(vect_horizontal);
                    ult_pos_escrita=num_materiales*4-2;



                    if ult_pos_escrita>ultima_pos_real
                        mensaje = (['Error en los datos introducidos. Compruebe la celda del número de
materiales en la hoja Excel.']);
                        uialert(app.pag_principal,mensaje,'Datos no válidos');
                    elseif num_materiales>=1
                        for ii=1:num_materiales
                            columna=4*ii-2;
                            [datos_uniaxial,continuar_uniaxial,datos_torsion,continuar_torsion] =
comprueba_datos(NUM,columna);

                            if continuar_uniaxial==0 && continuar_torsion==0
                                mensaje = (['Error en los datos introducidos. Compruebe los datos en la hoja
Excel.']);
                                uialert(app.pag_principal,mensaje,'Datos no válidos');
                                break

                            elseif ii==1

                                if continuar_uniaxial==1
                                    app.Sel_all_exc_uni.Visible="on";
                                    app.Unsel_all_exc_uni.Visible="on";
                                    pos_uni(ii)=1;
                                    cont_uni=cont_uni+1;
                                    nombre_material=TXT(4,columna);
                                    app.Material1_uniCheckBox.Text=nombre_material;
                                    app.Material1_uniCheckBox.Visible="on";
                                    app.im_mat_1_uni.Visible="on";
                                    x_exc_uni.mat_1=datos_uniaxial(:,1);
                                    y_exc_uni.mat_1=datos_uniaxial(:,2);
                                end
                                if continuar_torsion==1
                                    app.Sel_all_exc_tor.Visible="on";
                                    app.Unsel_all_exc_tor.Visible="on";
                                    pos_tor(ii)=1;
                                    cont_tor=cont_tor+1;
```

```
            nombre_material=TXT(4,columna);
            app.Material1_torCheckBox.Text=nombre_material;
            app.Material1_torCheckBox.Visible="on";
            app.im_mat_1_tor.Visible="on";
            x_exc_tor.mat_1=datos_torsion(:,1);
            y_exc_tor.mat_1=datos_torsion(:,2);
        end

    elseif ii==2
        if continuar_uniaxial==1
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
            nombre_material=TXT(4,columna);
            app.Material2_uniCheckBox.Text=nombre_material;
            app.Material2_uniCheckBox.Visible="on";
            app.im_mat_2_uni.Visible="on";
            x_exc_uni.mat_2=datos_uniaxial(:,1);
            y_exc_uni.mat_2=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
            pos_tor(ii)=1;
            cont_tor=cont_tor+1;
            nombre_material=TXT(4,columna);
            app.Material2_torCheckBox.Text=nombre_material;
            app.Material2_torCheckBox.Visible="on";
            app.im_mat_2_tor.Visible="on";
            x_exc_tor.mat_2=datos_torsion(:,1);
            y_exc_tor.mat_2=datos_torsion(:,2);
        end

    elseif ii==3
        if continuar_uniaxial==1
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
            nombre_material=TXT(4,columna);
            app.Material3_uniCheckBox.Text=nombre_material;
            app.Material3_uniCheckBox.Visible="on";
            app.im_mat_3_uni.Visible="on";
            x_exc_uni.mat_3=datos_uniaxial(:,1);
            y_exc_uni.mat_3=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
            pos_tor(ii)=1;
            cont_tor=cont_tor+1;
            nombre_material=TXT(4,columna);
            app.Material3_torCheckBox.Text=nombre_material;
            app.Material3_torCheckBox.Visible="on";
            app.im_mat_3_tor.Visible="on";
            x_exc_tor.mat_3=datos_torsion(:,1);
            y_exc_tor.mat_3=datos_torsion(:,2);
        end

    elseif ii==4
        if continuar_uniaxial==1
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
            nombre_material=TXT(4,columna);
            app.Material4_uniCheckBox.Text=nombre_material;
            app.Material4_uniCheckBox.Visible="on";
            app.im_mat_4_uni.Visible="on";
            x_exc_uni.mat_4=datos_uniaxial(:,1);
            y_exc_uni.mat_4=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
```

```
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
            pos_tor(ii)=1;
            cont_tor=cont_tor+1;
            nombre_material=TXT(4,columna);
            app.Material4_torCheckBox.Text=nombre_material;
            app.Material4_torCheckBox.Visible="on";
            app.im_mat_4_tor.Visible="on";
            x_exc_tor.mat_4=datos_torsion(:,1);
            y_exc_tor.mat_4=datos_torsion(:,2);
        end

    elseif ii==5
        if continuar_uniaxial==1
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
            nombre_material=TXT(4,columna);
            app.Material5_uniCheckBox.Text=nombre_material;
            app.Material5_uniCheckBox.Visible="on";
            app.im_mat_5_uni.Visible="on";
            x_exc_uni.mat_5=datos_uniaxial(:,1);
            y_exc_uni.mat_5=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
            pos_tor(ii)=1;
            cont_tor=cont_tor+1;
            nombre_material=TXT(4,columna);
            app.Material5_torCheckBox.Text=nombre_material;
            app.Material5_torCheckBox.Visible="on";
            app.im_mat_5_tor.Visible="on";
            x_exc_tor.mat_5=datos_torsion(:,1);
            y_exc_tor.mat_5=datos_torsion(:,2);
        end

    elseif ii==6
        if continuar_uniaxial==1
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
            nombre_material=TXT(4,columna);
            app.Material6_uniCheckBox.Text=nombre_material;
            app.Material6_uniCheckBox.Visible="on";
            app.im_mat_6_uni.Visible="on";
            x_exc_uni.mat_6=datos_uniaxial(:,1);
            y_exc_uni.mat_6=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
            pos_tor(ii)=1;
            cont_tor=cont_tor+1;
            nombre_material=TXT(4,columna);
            app.Material6_torCheckBox.Text=nombre_material;
            app.Material6_torCheckBox.Visible="on";
            app.im_mat_6_tor.Visible="on";
            x_exc_tor.mat_6=datos_torsion(:,1);
            y_exc_tor.mat_6=datos_torsion(:,2);
        end

    elseif ii==7
        if continuar_uniaxial==1
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
            nombre_material=TXT(4,columna);
            app.Material7_uniCheckBox.Text=nombre_material;
            app.Material7_uniCheckBox.Visible="on";
            app.im_mat_7_uni.Visible="on";
```

```
            x_exc_uni.mat_7=datos_uniaxial(:,1);
            y_exc_uni.mat_7=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
            pos_tor(ii)=1;
            cont_tor=cont_tor+1;
            nombre_material=TXT(4,columna);
            app.Material7_torCheckBox.Text=nombre_material;
            app.Material7_torCheckBox.Visible="on";
            app.im_mat_7_tor.Visible="on";
            x_exc_tor.mat_7=datos_torsion(:,1);
            y_exc_tor.mat_7=datos_torsion(:,2);
        end

    elseif ii==8
        if continuar_uniaxial==1
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
            nombre_material=TXT(4,columna);
            app.Material8_uniCheckBox.Text=nombre_material;
            app.Material8_uniCheckBox.Visible="on";
            app.im_mat_8_uni.Visible="on";
            x_exc_uni.mat_8=datos_uniaxial(:,1);
            y_exc_uni.mat_8=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
            pos_tor(ii)=1;
            cont_tor=cont_tor+1;
            nombre_material=TXT(4,columna);
            app.Material8_torCheckBox.Text=nombre_material;
            app.Material8_torCheckBox.Visible="on";
            app.im_mat_8_tor.Visible="on";
            x_exc_tor.mat_8=datos_torsion(:,1);
            y_exc_tor.mat_8=datos_torsion(:,2);
        end

    elseif ii==9
        if continuar_uniaxial==1
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
            nombre_material=TXT(4,columna);
            app.Material9_uniCheckBox.Text=nombre_material;
            app.Material9_uniCheckBox.Visible="on";
            app.im_mat_9_uni.Visible="on";
            x_exc_uni.mat_9=datos_uniaxial(:,1);
            y_exc_uni.mat_9=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
            pos_tor(ii)=1;
            cont_tor=cont_tor+1;
            nombre_material=TXT(4,columna);
            app.Material9_torCheckBox.Text=nombre_material;
            app.Material9_torCheckBox.Visible="on";
            app.im_mat_9_tor.Visible="on";
            x_exc_tor.mat_9=datos_torsion(:,1);
            y_exc_tor.mat_9=datos_torsion(:,2);
        end

    elseif ii==10
        if continuar_uniaxial==1
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
```

```
            nombre_material=TXT(4,columna);
            app.Material10_uniCheckBox.Text=nombre_material;
            app.Material10_uniCheckBox.Visible="on";
            app.im_mat_10_uni.Visible="on";
            x_exc_uni.mat_10=datos_uniaxial(:,1);
            y_exc_uni.mat_10=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
            pos_tor(ii)=1;
            cont_tor=cont_tor+1;
            nombre_material=TXT(4,columna);
            app.Material10_torCheckBox.Text=nombre_material;
            app.Material10_torCheckBox.Visible="on";
            app.im_mat_10_tor.Visible="on";
            x_exc_tor.mat_10=datos_torsion(:,1);
            y_exc_tor.mat_10=datos_torsion(:,2);
        end

    elseif ii==11
        if continuar_uniaxial==1
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
            nombre_material=TXT(4,columna);
            app.Material11_uniCheckBox.Text=nombre_material;
            app.Material11_uniCheckBox.Visible="on";
            app.im_mat_11_uni.Visible="on";
            x_exc_uni.mat_11=datos_uniaxial(:,1);
            y_exc_uni.mat_11=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
            pos_tor(ii)=1;
            cont_tor=cont_tor+1;
            nombre_material=TXT(4,columna);
            app.Material11_torCheckBox.Text=nombre_material;
            app.Material11_torCheckBox.Visible="on";
            app.im_mat_11_tor.Visible="on";
            x_exc_tor.mat_11=datos_torsion(:,1);
            y_exc_tor.mat_11=datos_torsion(:,2);
        end

    elseif ii==12
        if continuar_uniaxial==1
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
            nombre_material=TXT(4,columna);
            app.Material12_uniCheckBox.Text=nombre_material;
            app.Material12_uniCheckBox.Visible="on";
            app.im_mat_12_uni.Visible="on";
            x_exc_uni.mat_12=datos_uniaxial(:,1);
            y_exc_uni.mat_12=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
            pos_tor(ii)=1;
            cont_tor=cont_tor+1;
            nombre_material=TXT(4,columna);
            app.Material12_torCheckBox.Text=nombre_material;
            app.Material12_torCheckBox.Visible="on";
            app.im_mat_12_tor.Visible="on";
            x_exc_tor.mat_12=datos_torsion(:,1);
            y_exc_tor.mat_12=datos_torsion(:,2);
        end

    elseif ii==13
        if continuar_uniaxial==1
```

```
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
            nombre_material=TXT(4,columna);
            app.Material13_uniCheckBox.Text=nombre_material;
            app.Material13_uniCheckBox.Visible="on";
            app.im_mat_13_uni.Visible="on";
            x_exc_uni.mat_13=datos_uniaxial(:,1);
            y_exc_uni.mat_13=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
            pos_tor(ii)=1;
            cont_tor=cont_tor+1;
            nombre_material=TXT(4,columna);
            app.Material13_torCheckBox.Text=nombre_material;
            app.Material13_torCheckBox.Visible="on";
            app.im_mat_13_tor.Visible="on";
            x_exc_tor.mat_13=datos_torsion(:,1);
            y_exc_tor.mat_13=datos_torsion(:,2);
        end

    elseif ii==14
        if continuar_uniaxial==1
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
            nombre_material=TXT(4,columna);
            app.Material14_uniCheckBox.Text=nombre_material;
            app.Material14_uniCheckBox.Visible="on";
            app.im_mat_14_uni.Visible="on";
            x_exc_uni.mat_14=datos_uniaxial(:,1);
            y_exc_uni.mat_14=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
            pos_tor(ii)=1;
            cont_tor=cont_tor+1;
            nombre_material=TXT(4,columna);
            app.Material14_torCheckBox.Text=nombre_material;
            app.Material14_torCheckBox.Visible="on";
            app.im_mat_14_tor.Visible="on";
            x_exc_tor.mat_14=datos_torsion(:,1);
            y_exc_tor.mat_14=datos_torsion(:,2);
        end

    elseif ii==15
        if continuar_uniaxial==1
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
            nombre_material=TXT(4,columna);
            app.Material15_uniCheckBox.Text=nombre_material;
            app.Material15_uniCheckBox.Visible="on";
            app.im_mat_15_uni.Visible="on";
            x_exc_uni.mat_15=datos_uniaxial(:,1);
            y_exc_uni.mat_15=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
            pos_tor(ii)=1;
            cont_tor=cont_tor+1;
            nombre_material=TXT(4,columna);
            app.Material15_torCheckBox.Text=nombre_material;
            app.Material15_torCheckBox.Visible="on";
            app.im_mat_15_tor.Visible="on";
            x_exc_tor.mat_15=datos_torsion(:,1);
            y_exc_tor.mat_15=datos_torsion(:,2);
```

```
    end

elseif ii==16
    if continuar_uniaxial==1
        app.Sel_all_exc_uni.Visible="on";
        app.Unsel_all_exc_uni.Visible="on";
        pos_uni(ii)=1;
        cont_uni=cont_uni+1;
        nombre_material=TXT(4,columna);
        app.Material16_uniCheckBox.Text=nombre_material;
        app.Material16_uniCheckBox.Visible="on";
        app.im_mat_16_uni.Visible="on";
        x_exc_uni.mat_16=datos_uniaxial(:,1);
        y_exc_uni.mat_16=datos_uniaxial(:,2);
    end
    if continuar_torsion==1
        app.Sel_all_exc_tor.Visible="on";
        app.Unsel_all_exc_tor.Visible="on";
        pos_tor(ii)=1;
        cont_tor=cont_tor+1;
        nombre_material=TXT(4,columna);
        app.Material16_torCheckBox.Text=nombre_material;
        app.Material16_torCheckBox.Visible="on";
        app.im_mat_16_tor.Visible="on";
        x_exc_tor.mat_16=datos_torsion(:,1);
        y_exc_tor.mat_16=datos_torsion(:,2);
    end

elseif ii==17
    if continuar_uniaxial==1
        app.Sel_all_exc_uni.Visible="on";
        app.Unsel_all_exc_uni.Visible="on";
        pos_uni(ii)=1;
        cont_uni=cont_uni+1;
        nombre_material=TXT(4,columna);
        app.Material17_uniCheckBox.Text=nombre_material;
        app.Material17_uniCheckBox.Visible="on";
        app.im_mat_17_uni.Visible="on";
        x_exc_uni.mat_17=datos_uniaxial(:,1);
        y_exc_uni.mat_17=datos_uniaxial(:,2);
    end
    if continuar_torsion==1
        app.Sel_all_exc_tor.Visible="on";
        app.Unsel_all_exc_tor.Visible="on";
        pos_tor(ii)=1;
        cont_tor=cont_tor+1;
        nombre_material=TXT(4,columna);
        app.Material17_torCheckBox.Text=nombre_material;
        app.Material17_torCheckBox.Visible="on";
        app.im_mat_17_tor.Visible="on";
        x_exc_tor.mat_17=datos_torsion(:,1);
        y_exc_tor.mat_17=datos_torsion(:,2);
    end

elseif ii==18
    if continuar_uniaxial==1
        app.Sel_all_exc_uni.Visible="on";
        app.Unsel_all_exc_uni.Visible="on";
        pos_uni(ii)=1;
        cont_uni=cont_uni+1;
        nombre_material=TXT(4,columna);
        app.Material18_uniCheckBox.Text=nombre_material;
        app.Material18_uniCheckBox.Visible="on";
        app.im_mat_18_uni.Visible="on";
        x_exc_uni.mat_18=datos_uniaxial(:,1);
        y_exc_uni.mat_18=datos_uniaxial(:,2);
    end
    if continuar_torsion==1
        app.Sel_all_exc_tor.Visible="on";
        app.Unsel_all_exc_tor.Visible="on";
        pos_tor(ii)=1;
        cont_tor=cont_tor+1;
        nombre_material=TXT(4,columna);
        app.Material18_torCheckBox.Text=nombre_material;
```

```
            app.Material18_torCheckBox.Visible="on";
            app.im_mat_18_tor.Visible="on";
            x_exc_tor.mat_18=datos_torsion(:,1);
            y_exc_tor.mat_18=datos_torsion(:,2);
        end

    elseif ii==19
        if continuar_uniaxial==1
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
            nombre_material=TXT(4,columna);
            app.Material19_uniCheckBox.Text=nombre_material;
            app.Material19_uniCheckBox.Visible="on";
            app.im_mat_19_uni.Visible="on";
            x_exc_uni.mat_19=datos_uniaxial(:,1);
            y_exc_uni.mat_19=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
            pos_tor(ii)=1;
            cont_tor=cont_tor+1;
            nombre_material=TXT(4,columna);
            app.Material19_torCheckBox.Text=nombre_material;
            app.Material19_torCheckBox.Visible="on";
            app.im_mat_19_tor.Visible="on";
            x_exc_tor.mat_19=datos_torsion(:,1);
            y_exc_tor.mat_19=datos_torsion(:,2);
        end

    elseif ii==20
        if continuar_uniaxial==1
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
            nombre_material=TXT(4,columna);
            app.Material20_uniCheckBox.Text=nombre_material;
            app.Material20_uniCheckBox.Visible="on";
            app.im_mat_20_uni.Visible="on";
            x_exc_uni.mat_20=datos_uniaxial(:,1);
            y_exc_uni.mat_20=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
            pos_tor(ii)=1;
            cont_tor=cont_tor+1;
            nombre_material=TXT(4,columna);
            app.Material20_torCheckBox.Text=nombre_material;
            app.Material20_torCheckBox.Visible="on";
            app.im_mat_20_tor.Visible="on";
            x_exc_tor.mat_20=datos_torsion(:,1);
            y_exc_tor.mat_20=datos_torsion(:,2);
        end

    elseif ii==21
        if continuar_uniaxial==1
            app.Sel_all_exc_uni.Visible="on";
            app.Unsel_all_exc_uni.Visible="on";
            pos_uni(ii)=1;
            cont_uni=cont_uni+1;
            nombre_material=TXT(4,columna);
            app.Material21_uniCheckBox.Text=nombre_material;
            app.Material21_uniCheckBox.Visible="on";
            app.im_mat_21_uni.Visible="on";
            x_exc_uni.mat_21=datos_uniaxial(:,1);
            y_exc_uni.mat_21=datos_uniaxial(:,2);
        end
        if continuar_torsion==1
            app.Sel_all_exc_tor.Visible="on";
            app.Unsel_all_exc_tor.Visible="on";
```

```
                                pos_tor(ii)=1;
                                cont_tor=cont_tor+1;
                                nombre_material=TXT(4,columna);
                                app.Material21_torCheckBox.Text=nombre_material;
                                app.Material21_torCheckBox.Visible="on";
                                app.im_mat_21_tor.Visible="on";
                                x_exc_tor.mat_21=datos_torsion(:,1);
                                y_exc_tor.mat_21=datos_torsion(:,2);
                            end

                        elseif ii==22
                            if continuar_uniaxial==1
                                app.Sel_all_exc_uni.Visible="on";
                                app.Unsel_all_exc_uni.Visible="on";
                                pos_uni(ii)=1;
                                cont_uni=cont_uni+1;
                                nombre_material=TXT(4,columna);
                                app.Material22_uniCheckBox.Text=nombre_material;
                                app.Material22_uniCheckBox.Visible="on";
                                app.im_mat_22_uni.Visible="on";
                                x_exc_uni.mat_22=datos_uniaxial(:,1);
                                y_exc_uni.mat_22=datos_uniaxial(:,2);
                            end
                            if continuar_torsion==1
                                app.Sel_all_exc_tor.Visible="on";
                                app.Unsel_all_exc_tor.Visible="on";
                                pos_tor(ii)=1;
                                cont_tor=cont_tor+1;
                                nombre_material=TXT(4,columna);
                                app.Material22_torCheckBox.Text=nombre_material;
                                app.Material22_torCheckBox.Visible="on";
                                app.im_mat_22_tor.Visible="on";
                                x_exc_tor.mat_22=datos_torsion(:,1);
                                y_exc_tor.mat_22=datos_torsion(:,2);
                            end

                        end


                    end
                end



    % nombre=TXT
    % numero=NUM
                app.CargandoLabel.Visible="off";
            end

        end

    end

    % Value changed function: Sel_all_exc_uni
    function Sel_all_exc_uniValueChanged(app, event)
%           value = app.Sel_all_exc_uni.Value;
        global sit
        global dib
        global x_exc_uni
        global y_exc_uni
        global pos_uni
        global num_materiales

        sit.select_all_exc_uni=app.Sel_all_exc_uni.Value;

        if sit.select_all_exc_uni==1
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;

            ii=1;
            if ii<=num_materiales
                if  sit.mat_1_uni==0 && pos_uni(ii)==1
                    sit.mat_1_uni=1;
                    app.Material1_uniCheckBox.Value=1;
```

```
dib.mat_1_uni=plot(app.Representacion,x_exc_uni.mat_1,y_exc_uni.mat_1,'o','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_2_uni==0 && pos_uni(ii)==1
                        sit.mat_2_uni=1;
                        app.Material2_uniCheckBox.Value=1;

dib.mat_2_uni=plot(app.Representacion,x_exc_uni.mat_2,y_exc_uni.mat_2,'s','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_3_uni==0 && pos_uni(ii)==1
                        sit.mat_3_uni=1;
                        app.Material3_uniCheckBox.Value=1;

dib.mat_3_uni=plot(app.Representacion,x_exc_uni.mat_3,y_exc_uni.mat_3,'v','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[0.4667,0.6745,0.1882]); %Color: Verde
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_4_uni==0 && pos_uni(ii)==1
                        sit.mat_4_uni=1;
                        app.Material4_uniCheckBox.Value=1;

dib.mat_4_uni=plot(app.Representacion,x_exc_uni.mat_4,y_exc_uni.mat_4,'o','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[1.0000,0.4118,0.1608]); %color:naranja
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_5_uni==0 && pos_uni(ii)==1
                        sit.mat_5_uni=1;
                        app.Material5_uniCheckBox.Value=1;

dib.mat_5_uni=plot(app.Representacion,x_exc_uni.mat_5,y_exc_uni.mat_5,'s','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[0.4941,0.1843,0.5569]); %color:morado
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_6_uni==0 && pos_uni(ii)==1
                        sit.mat_6_uni=1;
                        app.Material6_uniCheckBox.Value=1;

dib.mat_6_uni=plot(app.Representacion,x_exc_uni.mat_6,y_exc_uni.mat_6,'v','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[1,0.6,0.6]); %Color: Salmon
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_7_uni==0 && pos_uni(ii)==1
                        sit.mat_7_uni=1;
                        app.Material7_uniCheckBox.Value=1;

dib.mat_7_uni=plot(app.Representacion,x_exc_uni.mat_7,y_exc_uni.mat_7,'o','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[0.8667,0.5882,0.8784]); %color:Morado
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_8_uni==0 && pos_uni(ii)==1
                        sit.mat_8_uni=1;
                        app.Material8_uniCheckBox.Value=1;

dib.mat_8_uni=plot(app.Representacion,x_exc_uni.mat_8,y_exc_uni.mat_8,'s','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[0.6510,0.6510,0.6510]); %color:gris
                    end
```

```
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_9_uni==0 && pos_uni(ii)==1
                        sit.mat_9_uni=1;
                        app.Material9_uniCheckBox.Value=1;

dib.mat_9_uni=plot(app.Representacion,x_exc_uni.mat_9,y_exc_uni.mat_9,'v','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[0.3020,0.7451,0.9333]); %Color: Azul claro
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_10_uni==0 && pos_uni(ii)==1
                        sit.mat_10_uni=1;
                        app.Material10_uniCheckBox.Value=1;

dib.mat_10_uni=plot(app.Representacion,x_exc_uni.mat_10,y_exc_uni.mat_10,'o','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.6353,0.0784,0.1843]); %Color: Granate
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_11_uni==0 && pos_uni(ii)==1
                        sit.mat_11_uni=1;
                        app.Material11_uniCheckBox.Value=1;

dib.mat_11_uni=plot(app.Representacion,x_exc_uni.mat_11,y_exc_uni.mat_11,'s','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.8000,0.5294,0.0941]); %color:Marron
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_12_uni==0 && pos_uni(ii)==1
                        sit.mat_12_uni=1;
                        app.Material12_uniCheckBox.Value=1;

dib.mat_12_uni=plot(app.Representacion,x_exc_uni.mat_12,y_exc_uni.mat_12,'d','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.9804,0.4902,0.1373]); %color:Naranja
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_13_uni==0 && pos_uni(ii)==1
                        sit.mat_13_uni=1;
                        app.Material13_uniCheckBox.Value=1;

dib.mat_13_uni=plot(app.Representacion,x_exc_uni.mat_13,y_exc_uni.mat_13,'p','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.1765,0.7098,0.4000]); %color:Verde
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_14_uni==0 && pos_uni(ii)==1
                        sit.mat_14_uni=1;
                        app.Material14_uniCheckBox.Value=1;

dib.mat_14_uni=plot(app.Representacion,x_exc_uni.mat_14,y_exc_uni.mat_14,'>','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_15_uni==0 && pos_uni(ii)==1
                        sit.mat_15_uni=1;
                        app.Material15_uniCheckBox.Value=1;

dib.mat_15_uni=plot(app.Representacion,x_exc_uni.mat_15,y_exc_uni.mat_15,'d','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.8667,0.5882,0.8784]); %color:Morado
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_16_uni==0 && pos_uni(ii)==1
```

```
                        sit.mat_16_uni=1;
                        app.Material16_uniCheckBox.Value=1;

dib.mat_16_uni=plot(app.Representacion,x_exc_uni.mat_16,y_exc_uni.mat_16,'p','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_17_uni==0 && pos_uni(ii)==1
                        sit.mat_17_uni=1;
                        app.Material17_uniCheckBox.Value=1;

dib.mat_17_uni=plot(app.Representacion,x_exc_uni.mat_17,y_exc_uni.mat_17,'>','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.6510,0.6510,0.6510]); %color:gris
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_18_uni==0 && pos_uni(ii)==1
                        sit.mat_18_uni=1;
                        app.Material18_uniCheckBox.Value=1;

dib.mat_18_uni=plot(app.Representacion,x_exc_uni.mat_18,y_exc_uni.mat_18,'d','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.3020,0.7451,0.9333]); %Color: Azul claro
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_19_uni==0 && pos_uni(ii)==1
                        sit.mat_19_uni=1;
                        app.Material19_uniCheckBox.Value=1;

dib.mat_19_uni=plot(app.Representacion,x_exc_uni.mat_19,y_exc_uni.mat_19,'p','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.4941,0.1843,0.5569]); %color:morado
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_20_uni==0 && pos_uni(ii)==1
                        sit.mat_20_uni=1;
                        app.Material20_uniCheckBox.Value=1;

dib.mat_20_uni=plot(app.Representacion,x_exc_uni.mat_20,y_exc_uni.mat_20,'>','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[1,0.6,0.6]); %Color: Salmon
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_21_uni==0 && pos_uni(ii)==1
                        sit.mat_21_uni=1;
                        app.Material21_uniCheckBox.Value=1;

dib.mat_21_uni=plot(app.Representacion,x_exc_uni.mat_21,y_exc_uni.mat_21,'d','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_22_uni==0 && pos_uni(ii)==1
                        sit.mat_22_uni=1;
                        app.Material22_uniCheckBox.Value=1;

dib.mat_22_uni=plot(app.Representacion,x_exc_uni.mat_22,y_exc_uni.mat_22,'p','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
                    end
                end

            end

        end

        % Value changed function: Unsel_all_exc_uni
        function Unsel_all_exc_uniValueChanged(app, event)
```

```
%             value = app.Unsel_all_exc_uni.Value;
            global sit
            global dib
            global pos_uni
            global num_materiales

            sit.unselect_all_exc_uni=app.Unsel_all_exc_uni.Value;

            if sit.unselect_all_exc_uni==1
                sit.select_all_exc_uni=0;
                app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;

                ii=1;
                if ii<=num_materiales
                    if  sit.mat_1_uni==1 && pos_uni(ii)==1
                        sit.mat_1_uni=0;
                        app.Material1_uniCheckBox.Value=0;
                        delete(dib.mat_1_uni)
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_2_uni==1 && pos_uni(ii)==1
                        sit.mat_2_uni=0;
                        app.Material2_uniCheckBox.Value=0;
                        delete(dib.mat_2_uni)
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_3_uni==1 && pos_uni(ii)==1
                        sit.mat_3_uni=0;
                        app.Material3_uniCheckBox.Value=0;
                        delete(dib.mat_3_uni)
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_4_uni==1 && pos_uni(ii)==1
                        sit.mat_4_uni=0;
                        app.Material4_uniCheckBox.Value=0;
                        delete(dib.mat_4_uni)
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_4_uni==1 && pos_uni(ii)==1
                        sit.mat_4_uni=0;
                        app.Material4_uniCheckBox.Value=0;
                        delete(dib.mat_4_uni)
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_5_uni==1 && pos_uni(ii)==1
                        sit.mat_5_uni=0;
                        app.Material5_uniCheckBox.Value=0;
                        delete(dib.mat_5_uni)
                    end
                end
                if ii<=num_materiales
                    if  sit.mat_6_uni==1 && pos_uni(ii)==1
                        sit.mat_6_uni=0;
                        app.Material6_uniCheckBox.Value=0;
                        delete(dib.mat_6_uni)
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_7_uni==1 && pos_uni(ii)==1
                        sit.mat_7_uni=0;
                        app.Material7_uniCheckBox.Value=0;
                        delete(dib.mat_7_uni)
                    end
```

```
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_8_uni==1 && pos_uni(ii)==1
        sit.mat_8_uni=0;
        app.Material8_uniCheckBox.Value=0;
        delete(dib.mat_8_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_9_uni==1 && pos_uni(ii)==1
        sit.mat_9_uni=0;
        app.Material9_uniCheckBox.Value=0;
        delete(dib.mat_9_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_10_uni==1 && pos_uni(ii)==1
        sit.mat_10_uni=0;
        app.Material10_uniCheckBox.Value=0;
        delete(dib.mat_10_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_11_uni==1 && pos_uni(ii)==1
        sit.mat_11_uni=0;
        app.Material11_uniCheckBox.Value=0;
        delete(dib.mat_11_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_12_uni==1 && pos_uni(ii)==1
        sit.mat_12_uni=0;
        app.Material12_uniCheckBox.Value=0;
        delete(dib.mat_12_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_13_uni==1 && pos_uni(ii)==1
        sit.mat_13_uni=0;
        app.Material13_uniCheckBox.Value=0;
        delete(dib.mat_13_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_14_uni==1 && pos_uni(ii)==1
        sit.mat_14_uni=0;
        app.Material14_uniCheckBox.Value=0;
        delete(dib.mat_14_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_15_uni==1 && pos_uni(ii)==1
        sit.mat_15_uni=0;
        app.Material15_uniCheckBox.Value=0;
        delete(dib.mat_15_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_16_uni==1 && pos_uni(ii)==1
        sit.mat_16_uni=0;
        app.Material16_uniCheckBox.Value=0;
        delete(dib.mat_16_uni)
    end
end
ii=ii+1;
```

```
                if ii<=num_materiales
                    if  sit.mat_17_uni==1 && pos_uni(ii)==1
                        sit.mat_17_uni=0;
                        app.Material17_uniCheckBox.Value=0;
                        delete(dib.mat_17_uni)
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_18_uni==1 && pos_uni(ii)==1
                        sit.mat_18_uni=0;
                        app.Material18_uniCheckBox.Value=0;
                        delete(dib.mat_18_uni)
                    end
                end
                ii=ii+1;
               if ii<=num_materiales
                    if  sit.mat_19_uni==1 && pos_uni(ii)==1
                        sit.mat_19_uni=0;
                        app.Material19_uniCheckBox.Value=0;
                        delete(dib.mat_19_uni)
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_20_uni==1 && pos_uni(ii)==1
                        sit.mat_20_uni=0;
                        app.Material20_uniCheckBox.Value=0;
                        delete(dib.mat_20_uni)
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_21_uni==1 && pos_uni(ii)==1
                        sit.mat_21_uni=0;
                        app.Material21_uniCheckBox.Value=0;
                        delete(dib.mat_21_uni)
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_22_uni==1 && pos_uni(ii)==1
                        sit.mat_22_uni=0;
                        app.Material22_uniCheckBox.Value=0;
                        delete(dib.mat_22_uni)
                    end
                end

            end
        end

        % Value changed function: Material1_uniCheckBox
        function Material1_uniCheckBoxValueChanged(app, event)
%             value = app.Material1_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_1_uni=app.Material1_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_1_uni==1

dib.mat_1_uni=plot(app.Representacion,x_exc_uni.mat_1,y_exc_uni.mat_1,'o','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
            else
                delete(dib.mat_1_uni)
            end
```

```
        end

        % Value changed function: Material2_uniCheckBox
        function Material2_uniCheckBoxValueChanged(app, event)
%             value = app.Material2_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_2_uni=app.Material2_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_2_uni==1

dib.mat_2_uni=plot(app.Representacion,x_exc_uni.mat_2,y_exc_uni.mat_2,'s','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
            else
                delete(dib.mat_2_uni)
            end
        end

        % Value changed function: Material3_uniCheckBox
        function Material3_uniCheckBoxValueChanged(app, event)
%             value = app.Material3_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_3_uni=app.Material3_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_3_uni==1

dib.mat_3_uni=plot(app.Representacion,x_exc_uni.mat_3,y_exc_uni.mat_3,'v','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[0.4667,0.6745,0.1882]); %Color: Verde
            else
                delete(dib.mat_3_uni)
            end
        end

        % Value changed function: Material4_uniCheckBox
        function Material4_uniCheckBoxValueChanged(app, event)
%             value = app.Material4_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_4_uni=app.Material4_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_4_uni==1

dib.mat_4_uni=plot(app.Representacion,x_exc_uni.mat_4,y_exc_uni.mat_4,'o','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[1.0000,0.4118,0.1608]); %color:naranja
            else
                delete(dib.mat_4_uni)
```

```
            end
        end

        % Value changed function: Material5_uniCheckBox
        function Material5_uniCheckBoxValueChanged(app, event)
%             value = app.Material5_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_5_uni=app.Material5_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_5_uni==1

dib.mat_5_uni=plot(app.Representacion,x_exc_uni.mat_5,y_exc_uni.mat_5,'s','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[0.4941,0.1843,0.5569]); %color:morado
            else
                delete(dib.mat_5_uni)
            end
        end

        % Value changed function: Material6_uniCheckBox
        function Material6_uniCheckBoxValueChanged(app, event)
%             value = app.Material6_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_6_uni=app.Material6_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_6_uni==1

dib.mat_6_uni=plot(app.Representacion,x_exc_uni.mat_6,y_exc_uni.mat_6,'v','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[1,0.6,0.6]); %Color: Salmon
            else
                delete(dib.mat_6_uni)
            end
        end

        % Value changed function: Material7_uniCheckBox
        function Material7_uniCheckBoxValueChanged(app, event)
%             value = app.Material7_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_7_uni=app.Material7_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_7_uni==1

dib.mat_7_uni=plot(app.Representacion,x_exc_uni.mat_7,y_exc_uni.mat_7,'o','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[0.8667,0.5882,0.8784]); %color:Morado
            else
```

```
                delete(dib.mat_7_uni)
            end
        end

        % Value changed function: Material8_uniCheckBox
        function Material8_uniCheckBoxValueChanged(app, event)
%             value = app.Material8_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_8_uni=app.Material8_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_8_uni==1

dib.mat_8_uni=plot(app.Representacion,x_exc_uni.mat_8,y_exc_uni.mat_8,'s','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[0.6510,0.6510,0.6510]); %color:gris
            else
                delete(dib.mat_8_uni)
            end
        end

        % Value changed function: Material9_uniCheckBox
        function Material9_uniCheckBoxValueChanged(app, event)
%             value = app.Material9_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_9_uni=app.Material9_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_9_uni==1

dib.mat_9_uni=plot(app.Representacion,x_exc_uni.mat_9,y_exc_uni.mat_9,'v','MarkerSize',7,'MarkerEdgeColor','k'
,'MarkerFaceColor',[0.3020,0.7451,0.9333]); %Color: Azul claro
            else
                delete(dib.mat_9_uni)
            end
        end

        % Value changed function: Material10_uniCheckBox
        function Material10_uniCheckBoxValueChanged(app, event)
%             value = app.Material10_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_10_uni=app.Material10_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_10_uni==1

dib.mat_10_uni=plot(app.Representacion,x_exc_uni.mat_10,y_exc_uni.mat_10,'o','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.6353,0.0784,0.1843]); %Color: Granate
```

```
            else
                delete(dib.mat_10_uni)
            end
        end

        % Value changed function: Material11_uniCheckBox
        function Material11_uniCheckBoxValueChanged(app, event)
%             value = app.Material11_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_11_uni=app.Material11_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_11_uni==1

dib.mat_11_uni=plot(app.Representacion,x_exc_uni.mat_11,y_exc_uni.mat_11,'s','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.8000,0.5294,0.0941]); %color:Marron
            else
                delete(dib.mat_11_uni)
            end
        end

        % Value changed function: Material12_uniCheckBox
        function Material12_uniCheckBoxValueChanged(app, event)
%             value = app.Material12_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_12_uni=app.Material12_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_12_uni==1

dib.mat_12_uni=plot(app.Representacion,x_exc_uni.mat_12,y_exc_uni.mat_12,'d','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.9804,0.4902,0.1373]); %color:Naranja
            else
                delete(dib.mat_12_uni)
            end
        end

        % Value changed function: Material13_uniCheckBox
        function Material13_uniCheckBoxValueChanged(app, event)
%             value = app.Material13_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_13_uni=app.Material13_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_13_uni==1
```

```
dib.mat_13_uni=plot(app.Representacion,x_exc_uni.mat_13,y_exc_uni.mat_13,'p','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.1765,0.7098,0.4000]); %color:Verde
            else
                delete(dib.mat_13_uni)
            end
        end

        % Value changed function: Material14_uniCheckBox
        function Material14_uniCheckBoxValueChanged(app, event)
%             value = app.Material14_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_14_uni=app.Material14_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_14_uni==1

dib.mat_14_uni=plot(app.Representacion,x_exc_uni.mat_14,y_exc_uni.mat_14,'>','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
            else
                delete(dib.mat_14_uni)
            end
        end

        % Value changed function: Material15_uniCheckBox
        function Material15_uniCheckBoxValueChanged(app, event)
%             value = app.Material15_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_15_uni=app.Material15_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_15_uni==1

dib.mat_15_uni=plot(app.Representacion,x_exc_uni.mat_15,y_exc_uni.mat_15,'d','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.8667,0.5882,0.8784]); %color:Morado
            else
                delete(dib.mat_15_uni)
            end
        end

        % Value changed function: Material16_uniCheckBox
        function Material16_uniCheckBoxValueChanged(app, event)
%             value = app.Material16_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_16_uni=app.Material16_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;
```

```
            if sit.mat_16_uni==1

dib.mat_16_uni=plot(app.Representacion,x_exc_uni.mat_16,y_exc_uni.mat_16,'p','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
            else
                delete(dib.mat_16_uni)
            end
        end

        % Value changed function: Material17_uniCheckBox
        function Material17_uniCheckBoxValueChanged(app, event)
%             value = app.Material17_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_17_uni=app.Material17_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_17_uni==1

dib.mat_17_uni=plot(app.Representacion,x_exc_uni.mat_17,y_exc_uni.mat_17,'>','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.6510,0.6510,0.6510]); %color:gris
            else
                delete(dib.mat_17_uni)
            end
        end

        % Value changed function: Material18_uniCheckBox
        function Material18_uniCheckBoxValueChanged(app, event)
%             value = app.Material18_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_18_uni=app.Material18_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_18_uni==1

dib.mat_18_uni=plot(app.Representacion,x_exc_uni.mat_18,y_exc_uni.mat_18,'d','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.3020,0.7451,0.9333]); %Color: Azul claro
            else
                delete(dib.mat_18_uni)
            end
        end

        % Value changed function: Material19_uniCheckBox
        function Material19_uniCheckBoxValueChanged(app, event)
%             value = app.Material19_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_19_uni=app.Material19_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;
```

```
            if sit.mat_19_uni==1

dib.mat_19_uni=plot(app.Representacion,x_exc_uni.mat_19,y_exc_uni.mat_19,'p','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.4941,0.1843,0.5569]); %color:morado
            else
                delete(dib.mat_19_uni)
            end
        end

        % Value changed function: Material20_uniCheckBox
        function Material20_uniCheckBoxValueChanged(app, event)
%             value = app.Material20_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_20_uni=app.Material20_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_20_uni==1

dib.mat_20_uni=plot(app.Representacion,x_exc_uni.mat_20,y_exc_uni.mat_20,'>','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[1,0.6,0.6]); %Color: Salmon
            else
                delete(dib.mat_20_uni)
            end
        end

        % Value changed function: Material21_uniCheckBox
        function Material21_uniCheckBoxValueChanged(app, event)
%             value = app.Material21_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_21_uni=app.Material21_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;


            if sit.mat_21_uni==1

dib.mat_21_uni=plot(app.Representacion,x_exc_uni.mat_21,y_exc_uni.mat_21,'d','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
            else
                delete(dib.mat_21_uni)
            end
        end

        % Value changed function: Material22_uniCheckBox
        function Material22_uniCheckBoxValueChanged(app, event)
%             value = app.Material22_uniCheckBox.Value;
            global sit
            global dib
            global x_exc_uni
            global y_exc_uni

            sit.mat_22_uni=app.Material22_uniCheckBox.Value;

            sit.select_all_exc_uni=0;
            app.Sel_all_exc_uni.Value=sit.select_all_exc_uni;
            sit.unselect_all_exc_uni=0;
            app.Unsel_all_exc_uni.Value=sit.unselect_all_exc_uni;
```

```
            if sit.mat_22_uni==1

dib.mat_22_uni=plot(app.Representacion,x_exc_uni.mat_22,y_exc_uni.mat_22,'p','MarkerSize',7,'MarkerEdgeColor',
'k','MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
            else
                delete(dib.mat_22_uni)
            end
        end

        % Value changed function: Unsel_all_exc_tor
        function Unsel_all_exc_torValueChanged(app, event)
%             value = app.Unsel_all_exc_tor.Value;
            global sit
            global dib_t
            global pos_tor
            global num_materiales

            sit.unselect_all_exc_tor=app.Unsel_all_exc_tor.Value;

            if sit.unselect_all_exc_tor==1
                sit.select_all_exc_tor=0;
                app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;

                ii=1;
                if ii<=num_materiales
                    if  sit.mat_1_tor==1 && pos_tor(ii)==1
                        sit.mat_1_tor=0;
                        app.Material1_torCheckBox.Value=0;
                        delete(dib_t.mat_1_tor)
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_2_tor==1 && pos_tor(ii)==1
                        sit.mat_2_tor=0;
                        app.Material2_torCheckBox.Value=0;
                        delete(dib_t.mat_2_tor)
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_3_tor==1 && pos_tor(ii)==1
                        sit.mat_3_tor=0;
                        app.Material3_torCheckBox.Value=0;
                        delete(dib_t.mat_3_tor)
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_4_tor==1 && pos_tor(ii)==1
                        sit.mat_4_tor=0;
                        app.Material4_torCheckBox.Value=0;
                        delete(dib_t.mat_4_tor)
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_4_tor==1 && pos_tor(ii)==1
                        sit.mat_4_tor=0;
                        app.Material4_torCheckBox.Value=0;
                        delete(dib_t.mat_4_tor)
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_5_tor==1 && pos_tor(ii)==1
                        sit.mat_5_tor=0;
                        app.Material5_torCheckBox.Value=0;
                        delete(dib_t.mat_5_tor)
                    end
                end
                if ii<=num_materiales
```

```
    if  sit.mat_6_tor==1 && pos_tor(ii)==1
        sit.mat_6_tor=0;
        app.Material6_torCheckBox.Value=0;
        delete(dib_t.mat_6_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_7_tor==1 && pos_tor(ii)==1
        sit.mat_7_tor=0;
        app.Material7_torCheckBox.Value=0;
        delete(dib_t.mat_7_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_8_tor==1 && pos_tor(ii)==1
        sit.mat_8_tor=0;
        app.Material8_torCheckBox.Value=0;
        delete(dib_t.mat_8_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_9_tor==1 && pos_tor(ii)==1
        sit.mat_9_tor=0;
        app.Material9_torCheckBox.Value=0;
        delete(dib_t.mat_9_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_10_tor==1 && pos_tor(ii)==1
        sit.mat_10_tor=0;
        app.Material10_torCheckBox.Value=0;
        delete(dib_t.mat_10_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_11_tor==1 && pos_tor(ii)==1
        sit.mat_11_tor=0;
        app.Material11_torCheckBox.Value=0;
        delete(dib_t.mat_11_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_12_tor==1 && pos_tor(ii)==1
        sit.mat_12_tor=0;
        app.Material12_torCheckBox.Value=0;
        delete(dib_t.mat_12_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_13_tor==1 && pos_tor(ii)==1
        sit.mat_13_tor=0;
        app.Material13_torCheckBox.Value=0;
        delete(dib_t.mat_13_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_14_tor==1 && pos_tor(ii)==1
        sit.mat_14_tor=0;
        app.Material14_torCheckBox.Value=0;
        delete(dib_t.mat_14_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_15_tor==1 && pos_tor(ii)==1
        sit.mat_15_tor=0;
```

```
                    app.Material15_torCheckBox.Value=0;
                    delete(dib_t.mat_15_tor)
                end
            end
            ii=ii+1;
            if ii<=num_materiales
                if  sit.mat_16_tor==1 && pos_tor(ii)==1
                    sit.mat_16_tor=0;
                    app.Material16_torCheckBox.Value=0;
                    delete(dib_t.mat_16_tor)
                end
            end
            ii=ii+1;
            if ii<=num_materiales
                if  sit.mat_17_tor==1 && pos_tor(ii)==1
                    sit.mat_17_tor=0;
                    app.Material17_torCheckBox.Value=0;
                    delete(dib_t.mat_17_tor)
                end
            end
            ii=ii+1;
            if ii<=num_materiales
                if  sit.mat_18_tor==1 && pos_tor(ii)==1
                    sit.mat_18_tor=0;
                    app.Material18_torCheckBox.Value=0;
                    delete(dib_t.mat_18_tor)
                end
            end
            ii=ii+1;
           if ii<=num_materiales
                if  sit.mat_19_tor==1 && pos_tor(ii)==1
                    sit.mat_19_tor=0;
                    app.Material19_torCheckBox.Value=0;
                    delete(dib_t.mat_19_tor)
                end
            end
            ii=ii+1;
            if ii<=num_materiales
                if  sit.mat_20_tor==1 && pos_tor(ii)==1
                    sit.mat_20_tor=0;
                    app.Material20_torCheckBox.Value=0;
                    delete(dib_t.mat_20_tor)
                end
            end
            ii=ii+1;
            if ii<=num_materiales
                if  sit.mat_21_tor==1 && pos_tor(ii)==1
                    sit.mat_21_tor=0;
                    app.Material21_torCheckBox.Value=0;
                    delete(dib_t.mat_21_tor)
                end
            end
            ii=ii+1;
            if ii<=num_materiales
                if  sit.mat_22_tor==1 && pos_tor(ii)==1
                    sit.mat_22_tor=0;
                    app.Material22_torCheckBox.Value=0;
                    delete(dib_t.mat_22_tor)
                end
            end

        end
    end

    % Value changed function: Sel_all_exc_tor
    function Sel_all_exc_torValueChanged(app, event)
%         value = app.Sel_all_exc_tor.Value;
        global sit
        global dib_t
        global x_exc_tor
        global y_exc_tor
        global pos_tor
        global num_materiales
```

```
            sit.select_all_exc_tor=app.Sel_all_exc_tor.Value;

            if sit.select_all_exc_tor==1
                sit.unselect_all_exc_tor=0;
                app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;

                ii=1;
                if ii<=num_materiales
                    if  sit.mat_1_tor==0 && pos_tor(ii)==1
                        sit.mat_1_tor=1;
                        app.Material1_torCheckBox.Value=1;

dib_t.mat_1_tor=plot(app.Representacion_torsion,x_exc_tor.mat_1,y_exc_tor.mat_1,'o','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_2_tor==0 && pos_tor(ii)==1
                        sit.mat_2_tor=1;
                        app.Material2_torCheckBox.Value=1;

dib_t.mat_2_tor=plot(app.Representacion_torsion,x_exc_tor.mat_2,y_exc_tor.mat_2,'s','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_3_tor==0 && pos_tor(ii)==1
                        sit.mat_3_tor=1;
                        app.Material3_torCheckBox.Value=1;

dib_t.mat_3_tor=plot(app.Representacion_torsion,x_exc_tor.mat_3,y_exc_tor.mat_3,'v','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[0.4667,0.6745,0.1882]); %Color: Verde
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_4_tor==0 && pos_tor(ii)==1
                        sit.mat_4_tor=1;
                        app.Material4_torCheckBox.Value=1;

dib_t.mat_4_tor=plot(app.Representacion_torsion,x_exc_tor.mat_4,y_exc_tor.mat_4,'o','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[1.0000,0.4118,0.1608]); %color:naranja
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_5_tor==0 && pos_tor(ii)==1
                        sit.mat_5_tor=1;
                        app.Material5_torCheckBox.Value=1;

dib_t.mat_5_tor=plot(app.Representacion_torsion,x_exc_tor.mat_5,y_exc_tor.mat_5,'s','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[0.4941,0.1843,0.5569]); %color:morado
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_6_tor==0 && pos_tor(ii)==1
                        sit.mat_6_tor=1;
                        app.Material6_torCheckBox.Value=1;

dib_t.mat_6_tor=plot(app.Representacion_torsion,x_exc_tor.mat_6,y_exc_tor.mat_6,'v','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[1,0.6,0.6]); %Color: Salmon
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_7_tor==0 && pos_tor(ii)==1
                        sit.mat_7_tor=1;
                        app.Material7_torCheckBox.Value=1;

dib_t.mat_7_tor=plot(app.Representacion_torsion,x_exc_tor.mat_7,y_exc_tor.mat_7,'o','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[0.8667,0.5882,0.8784]); %color:Morado
```

```
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_8_tor==0 && pos_tor(ii)==1
                        sit.mat_8_tor=1;
                        app.Material8_torCheckBox.Value=1;

dib_t.mat_8_tor=plot(app.Representacion_torsion,x_exc_tor.mat_8,y_exc_tor.mat_8,'s','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[0.6510,0.6510,0.6510]); %color:gris
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_9_tor==0 && pos_tor(ii)==1
                        sit.mat_9_tor=1;
                        app.Material9_torCheckBox.Value=1;

dib_t.mat_9_tor=plot(app.Representacion_torsion,x_exc_tor.mat_9,y_exc_tor.mat_9,'v','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[0.3020,0.7451,0.9333]); %Color: Azul claro
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_10_tor==0 && pos_tor(ii)==1
                        sit.mat_10_tor=1;
                        app.Material10_torCheckBox.Value=1;

dib_t.mat_10_tor=plot(app.Representacion_torsion,x_exc_tor.mat_10,y_exc_tor.mat_10,'o','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.6353,0.0784,0.1843]); %Color: Granate
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_11_tor==0 && pos_tor(ii)==1
                        sit.mat_11_tor=1;
                        app.Material11_torCheckBox.Value=1;

dib_t.mat_11_tor=plot(app.Representacion_torsion,x_exc_tor.mat_11,y_exc_tor.mat_11,'s','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.8000,0.5294,0.0941]); %color:Marron
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_12_tor==0 && pos_tor(ii)==1
                        sit.mat_12_tor=1;
                        app.Material12_torCheckBox.Value=1;

dib_t.mat_12_tor=plot(app.Representacion_torsion,x_exc_tor.mat_12,y_exc_tor.mat_12,'d','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.9804,0.4902,0.1373]); %color:Naranja
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_13_tor==0 && pos_tor(ii)==1
                        sit.mat_13_tor=1;
                        app.Material13_torCheckBox.Value=1;

dib_t.mat_13_tor=plot(app.Representacion_torsion,x_exc_tor.mat_13,y_exc_tor.mat_13,'p','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.1765,0.7098,0.4000]); %color:Verde
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_14_tor==0 && pos_tor(ii)==1
                        sit.mat_14_tor=1;
                        app.Material14_torCheckBox.Value=1;

dib_t.mat_14_tor=plot(app.Representacion_torsion,x_exc_tor.mat_14,y_exc_tor.mat_14,'>','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
```

```
                    if  sit.mat_15_tor==0 && pos_tor(ii)==1
                        sit.mat_15_tor=1;
                        app.Material15_torCheckBox.Value=1;

dib_t.mat_15_tor=plot(app.Representacion_torsion,x_exc_tor.mat_15,y_exc_tor.mat_15,'d','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.8667,0.5882,0.8784]); %color:Morado
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_16_tor==0 && pos_tor(ii)==1
                        sit.mat_16_tor=1;
                        app.Material16_torCheckBox.Value=1;

dib_t.mat_16_tor=plot(app.Representacion_torsion,x_exc_tor.mat_16,y_exc_tor.mat_16,'p','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_17_tor==0 && pos_tor(ii)==1
                        sit.mat_17_tor=1;
                        app.Material17_torCheckBox.Value=1;

dib_t.mat_17_tor=plot(app.Representacion_torsion,x_exc_tor.mat_17,y_exc_tor.mat_17,'>','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.6510,0.6510,0.6510]); %color:gris
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_18_tor==0 && pos_tor(ii)==1
                        sit.mat_18_tor=1;
                        app.Material18_torCheckBox.Value=1;

dib_t.mat_18_tor=plot(app.Representacion_torsion,x_exc_tor.mat_18,y_exc_tor.mat_18,'d','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.3020,0.7451,0.9333]); %Color: Azul claro
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_19_tor==0 && pos_tor(ii)==1
                        sit.mat_19_tor=1;
                        app.Material19_torCheckBox.Value=1;

dib_t.mat_19_tor=plot(app.Representacion_torsion,x_exc_tor.mat_19,y_exc_tor.mat_19,'p','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.4941,0.1843,0.5569]); %color:morado
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_20_tor==0 && pos_tor(ii)==1
                        sit.mat_20_tor=1;
                        app.Material20_torCheckBox.Value=1;

dib_t.mat_20_tor=plot(app.Representacion_torsion,x_exc_tor.mat_20,y_exc_tor.mat_20,'>','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[1,0.6,0.6]); %Color: Salmon
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_21_tor==0 && pos_tor(ii)==1
                        sit.mat_21_tor=1;
                        app.Material21_torCheckBox.Value=1;

dib_t.mat_21_tor=plot(app.Representacion_torsion,x_exc_tor.mat_21,y_exc_tor.mat_21,'d','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
                    end
                end
                ii=ii+1;
                if ii<=num_materiales
                    if  sit.mat_22_tor==0 && pos_tor(ii)==1
                        sit.mat_22_tor=1;
                        app.Material22_torCheckBox.Value=1;
```

```
dib_t.mat_22_tor=plot(app.Representacion_torsion,x_exc_tor.mat_22,y_exc_tor.mat_22,'p','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
                    end
                end

            end

        end

        % Value changed function: Material1_torCheckBox
        function Material1_torCheckBoxValueChanged(app, event)
%             value = app.Material1_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_1_tor=app.Material1_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_1_tor==1

dib_t.mat_1_tor=plot(app.Representacion_torsion,x_exc_tor.mat_1,y_exc_tor.mat_1,'o','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
            else
                delete(dib_t.mat_1_tor)
            end
        end

        % Value changed function: Material2_torCheckBox
        function Material2_torCheckBoxValueChanged(app, event)
%             value = app.Material2_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_2_tor=app.Material2_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_2_tor==1

dib_t.mat_2_tor=plot(app.Representacion_torsion,x_exc_tor.mat_2,y_exc_tor.mat_2,'s','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
            else
                delete(dib_t.mat_2_tor)
            end
        end

        % Value changed function: Material3_torCheckBox
        function Material3_torCheckBoxValueChanged(app, event)
%             value = app.Material3_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_3_tor=app.Material3_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;
```

```
            if sit.mat_3_tor==1

dib_t.mat_3_tor=plot(app.Representacion_torsion,x_exc_tor.mat_3,y_exc_tor.mat_3,'v','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[0.4667,0.6745,0.1882]); %Color: Verde
            else
                delete(dib_t.mat_3_tor)
            end
        end

        % Value changed function: Material4_torCheckBox
        function Material4_torCheckBoxValueChanged(app, event)
%             value = app.Material4_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_4_tor=app.Material4_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_4_tor==1

dib_t.mat_4_tor=plot(app.Representacion_torsion,x_exc_tor.mat_4,y_exc_tor.mat_4,'o','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[1.0000,0.4118,0.1608]); %color:naranja
            else
                delete(dib_t.mat_4_tor)
            end
        end

        % Value changed function: Material5_torCheckBox
        function Material5_torCheckBoxValueChanged(app, event)
%             value = app.Material5_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_5_tor=app.Material5_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_5_tor==1

dib_t.mat_5_tor=plot(app.Representacion_torsion,x_exc_tor.mat_5,y_exc_tor.mat_5,'s','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[0.4941,0.1843,0.5569]); %color:morado
            else
                delete(dib_t.mat_5_tor)
            end
        end

        % Value changed function: Material6_torCheckBox
        function Material6_torCheckBoxValueChanged(app, event)
%             value = app.Material6_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_6_tor=app.Material6_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
```

```
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_6_tor==1

dib_t.mat_6_tor=plot(app.Representacion_torsion,x_exc_tor.mat_6,y_exc_tor.mat_6,'v','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[1,0.6,0.6]); %Color: Salmon
            else
                delete(dib_t.mat_6_tor)
            end
        end

        % Value changed function: Material7_torCheckBox
        function Material7_torCheckBoxValueChanged(app, event)
%             value = app.Material7_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_7_tor=app.Material7_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_7_tor==1

dib_t.mat_7_tor=plot(app.Representacion_torsion,x_exc_tor.mat_7,y_exc_tor.mat_7,'o','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[0.8667,0.5882,0.8784]); %color:Morado
            else
                delete(dib_t.mat_7_tor)
            end
        end

        % Value changed function: Material8_torCheckBox
        function Material8_torCheckBoxValueChanged(app, event)
%             value = app.Material8_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_8_tor=app.Material8_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_8_tor==1

dib_t.mat_8_tor=plot(app.Representacion_torsion,x_exc_tor.mat_8,y_exc_tor.mat_8,'s','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[0.6510,0.6510,0.6510]); %color:gris
            else
                delete(dib_t.mat_8_tor)
            end
        end

        % Value changed function: Material9_torCheckBox
        function Material9_torCheckBoxValueChanged(app, event)
%             value = app.Material9_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_9_tor=app.Material9_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
```

```
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_9_tor==1

dib_t.mat_9_tor=plot(app.Representacion_torsion,x_exc_tor.mat_9,y_exc_tor.mat_9,'v','MarkerSize',7,'MarkerEdge
Color','k','MarkerFaceColor',[0.3020,0.7451,0.9333]); %Color: Azul claro
            else
                delete(dib_t.mat_9_tor)
            end
        end

        % Value changed function: Material10_torCheckBox
        function Material10_torCheckBoxValueChanged(app, event)
%             value = app.Material10_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_10_tor=app.Material10_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_10_tor==1

dib_t.mat_10_tor=plot(app.Representacion_torsion,x_exc_tor.mat_10,y_exc_tor.mat_10,'o','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.6353,0.0784,0.1843]); %Color: Granate
            else
                delete(dib_t.mat_10_tor)
            end
        end

        % Value changed function: Material11_torCheckBox
        function Material11_torCheckBoxValueChanged(app, event)
%             value = app.Material11_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_11_tor=app.Material11_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_11_tor==1

dib_t.mat_11_tor=plot(app.Representacion_torsion,x_exc_tor.mat_11,y_exc_tor.mat_11,'s','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.8000,0.5294,0.0941]); %color:Marron
            else
                delete(dib_t.mat_11_tor)
            end
        end

        % Value changed function: Material12_torCheckBox
        function Material12_torCheckBoxValueChanged(app, event)
%             value = app.Material12_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_12_tor=app.Material12_torCheckBox.Value;

            sit.select_all_exc_tor=0;
```

```
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_12_tor==1

dib_t.mat_12_tor=plot(app.Representacion_torsion,x_exc_tor.mat_12,y_exc_tor.mat_12,'d','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.9804,0.4902,0.1373]); %color:Naranja
            else
                delete(dib_t.mat_12_tor)
            end
        end

        % Value changed function: Material13_torCheckBox
        function Material13_torCheckBoxValueChanged(app, event)
%             value = app.Material13_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_13_tor=app.Material13_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_13_tor==1

dib_t.mat_13_tor=plot(app.Representacion_torsion,x_exc_tor.mat_13,y_exc_tor.mat_13,'p','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.1765,0.7098,0.4000]); %color:Verde
            else
                delete(dib_t.mat_13_tor)
            end
        end

        % Value changed function: Material14_torCheckBox
        function Material14_torCheckBoxValueChanged(app, event)
%             value = app.Material14_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_14_tor=app.Material14_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_14_tor==1

dib_t.mat_14_tor=plot(app.Representacion_torsion,x_exc_tor.mat_14,y_exc_tor.mat_14,'>','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
            else
                delete(dib_t.mat_14_tor)
            end
        end

        % Value changed function: Material15_torCheckBox
        function Material15_torCheckBoxValueChanged(app, event)
%             value = app.Material15_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_15_tor=app.Material15_torCheckBox.Value;
```

```
            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_15_tor==1

dib_t.mat_15_tor=plot(app.Representacion_torsion,x_exc_tor.mat_15,y_exc_tor.mat_15,'d','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.8667,0.5882,0.8784]); %color:Morado
            else
                delete(dib_t.mat_15_tor)
            end
        end

        % Value changed function: Material16_torCheckBox
        function Material16_torCheckBoxValueChanged(app, event)
%             value = app.Material16_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_16_tor=app.Material16_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_16_tor==1

dib_t.mat_16_tor=plot(app.Representacion_torsion,x_exc_tor.mat_16,y_exc_tor.mat_16,'p','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
            else
                delete(dib_t.mat_16_tor)
            end
        end

        % Value changed function: Material17_torCheckBox
        function Material17_torCheckBoxValueChanged(app, event)
%             value = app.Material17_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_17_tor=app.Material17_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_17_tor==1

dib_t.mat_17_tor=plot(app.Representacion_torsion,x_exc_tor.mat_17,y_exc_tor.mat_17,'>','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.6510,0.6510,0.6510]); %color:gris
            else
                delete(dib_t.mat_17_tor)
            end
        end

        % Value changed function: Material18_torCheckBox
        function Material18_torCheckBoxValueChanged(app, event)
%             value = app.Material18_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_18_tor=app.Material18_torCheckBox.Value;
```

```
            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_18_tor==1

dib_t.mat_18_tor=plot(app.Representacion_torsion,x_exc_tor.mat_18,y_exc_tor.mat_18,'d','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.3020,0.7451,0.9333]); %Color: Azul claro
            else
                delete(dib_t.mat_18_tor)
            end
        end

        % Value changed function: Material19_torCheckBox
        function Material19_torCheckBoxValueChanged(app, event)
%             value = app.Material19_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_19_tor=app.Material19_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_19_tor==1

dib_t.mat_19_tor=plot(app.Representacion_torsion,x_exc_tor.mat_19,y_exc_tor.mat_19,'p','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.4941,0.1843,0.5569]); %color:morado
            else
                delete(dib_t.mat_19_tor)
            end
        end

        % Value changed function: Material20_torCheckBox
        function Material20_torCheckBoxValueChanged(app, event)
%             value = app.Material20_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_20_tor=app.Material20_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_20_tor==1

dib_t.mat_20_tor=plot(app.Representacion_torsion,x_exc_tor.mat_20,y_exc_tor.mat_20,'>','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[1,0.6,0.6]); %Color: Salmon
            else
                delete(dib_t.mat_20_tor)
            end
        end

        % Value changed function: Material21_torCheckBox
        function Material21_torCheckBoxValueChanged(app, event)
%             value = app.Material21_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor
```

```
            sit.mat_21_tor=app.Material21_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_21_tor==1

dib_t.mat_21_tor=plot(app.Representacion_torsion,x_exc_tor.mat_21,y_exc_tor.mat_21,'d','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0.9490,0.0392,0.0392]); %color:Rojo
            else
                delete(dib_t.mat_21_tor)
            end
        end

        % Value changed function: Material22_torCheckBox
        function Material22_torCheckBoxValueChanged(app, event)
%             value = app.Material22_torCheckBox.Value;
            global sit
            global dib_t
            global x_exc_tor
            global y_exc_tor

            sit.mat_22_tor=app.Material22_torCheckBox.Value;

            sit.select_all_exc_tor=0;
            app.Sel_all_exc_tor.Value=sit.select_all_exc_tor;
            sit.unselect_all_exc_tor=0;
            app.Unsel_all_exc_tor.Value=sit.unselect_all_exc_tor;


            if sit.mat_22_tor==1

dib_t.mat_22_tor=plot(app.Representacion_torsion,x_exc_tor.mat_22,y_exc_tor.mat_22,'p','MarkerSize',7,'MarkerE
dgeColor','k','MarkerFaceColor',[0,0.4471,0.7412]); %color:Azul marino
            else
                delete(dib_t.mat_22_tor)
            end
        end

        % Button pushed function: BorrardatosButton
        function BorrardatosButtonPushed(app, event)
            global num_materiales
            global pos_uni
            global pos_tor
            global sit
            global dib
            global dib_t

            ii=1;
            if ii<=num_materiales
                if  sit.mat_1_uni==1 && pos_uni(ii)==1
                    sit.mat_1_uni=0;
                    app.Material1_uniCheckBox.Value=0;
                    delete(dib.mat_1_uni)
                end
            end
            ii=ii+1;
            if ii<=num_materiales
                if  sit.mat_2_uni==1 && pos_uni(ii)==1
                    sit.mat_2_uni=0;
                    app.Material2_uniCheckBox.Value=0;
                    delete(dib.mat_2_uni)
                end
            end
            ii=ii+1;
            if ii<=num_materiales
                if  sit.mat_3_uni==1 && pos_uni(ii)==1
                    sit.mat_3_uni=0;
                    app.Material3_uniCheckBox.Value=0;
                    delete(dib.mat_3_uni)
                end
```

```
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_4_uni==1 && pos_uni(ii)==1
        sit.mat_4_uni=0;
        app.Material4_uniCheckBox.Value=0;
        delete(dib.mat_4_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_4_uni==1 && pos_uni(ii)==1
        sit.mat_4_uni=0;
        app.Material4_uniCheckBox.Value=0;
        delete(dib.mat_4_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_5_uni==1 && pos_uni(ii)==1
        sit.mat_5_uni=0;
        app.Material5_uniCheckBox.Value=0;
        delete(dib.mat_5_uni)
    end
end
if ii<=num_materiales
    if  sit.mat_6_uni==1 && pos_uni(ii)==1
        sit.mat_6_uni=0;
        app.Material6_uniCheckBox.Value=0;
        delete(dib.mat_6_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_7_uni==1 && pos_uni(ii)==1
        sit.mat_7_uni=0;
        app.Material7_uniCheckBox.Value=0;
        delete(dib.mat_7_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_8_uni==1 && pos_uni(ii)==1
        sit.mat_8_uni=0;
        app.Material8_uniCheckBox.Value=0;
        delete(dib.mat_8_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_9_uni==1 && pos_uni(ii)==1
        sit.mat_9_uni=0;
        app.Material9_uniCheckBox.Value=0;
        delete(dib.mat_9_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_10_uni==1 && pos_uni(ii)==1
        sit.mat_10_uni=0;
        app.Material10_uniCheckBox.Value=0;
        delete(dib.mat_10_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_11_uni==1 && pos_uni(ii)==1
        sit.mat_11_uni=0;
        app.Material11_uniCheckBox.Value=0;
        delete(dib.mat_11_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
```

```
        if  sit.mat_12_uni==1 && pos_uni(ii)==1
            sit.mat_12_uni=0;
            app.Material12_uniCheckBox.Value=0;
            delete(dib.mat_12_uni)
        end
    end
    ii=ii+1;
    if ii<=num_materiales
        if  sit.mat_13_uni==1 && pos_uni(ii)==1
            sit.mat_13_uni=0;
            app.Material13_uniCheckBox.Value=0;
            delete(dib.mat_13_uni)
        end
    end
    ii=ii+1;
    if ii<=num_materiales
        if  sit.mat_14_uni==1 && pos_uni(ii)==1
            sit.mat_14_uni=0;
            app.Material14_uniCheckBox.Value=0;
            delete(dib.mat_14_uni)
        end
    end
    ii=ii+1;
    if ii<=num_materiales
        if  sit.mat_15_uni==1 && pos_uni(ii)==1
            sit.mat_15_uni=0;
            app.Material15_uniCheckBox.Value=0;
            delete(dib.mat_15_uni)
        end
    end
    ii=ii+1;
    if ii<=num_materiales
        if  sit.mat_16_uni==1 && pos_uni(ii)==1
            sit.mat_16_uni=0;
            app.Material16_uniCheckBox.Value=0;
            delete(dib.mat_16_uni)
        end
    end
    ii=ii+1;
    if ii<=num_materiales
        if  sit.mat_17_uni==1 && pos_uni(ii)==1
            sit.mat_17_uni=0;
            app.Material17_uniCheckBox.Value=0;
            delete(dib.mat_17_uni)
        end
    end
    ii=ii+1;
    if ii<=num_materiales
        if  sit.mat_18_uni==1 && pos_uni(ii)==1
            sit.mat_18_uni=0;
            app.Material18_uniCheckBox.Value=0;
            delete(dib.mat_18_uni)
        end
    end
    ii=ii+1;
   if ii<=num_materiales
        if  sit.mat_19_uni==1 && pos_uni(ii)==1
            sit.mat_19_uni=0;
            app.Material19_uniCheckBox.Value=0;
            delete(dib.mat_19_uni)
        end
    end
    ii=ii+1;
    if ii<=num_materiales
        if  sit.mat_20_uni==1 && pos_uni(ii)==1
            sit.mat_20_uni=0;
            app.Material20_uniCheckBox.Value=0;
            delete(dib.mat_20_uni)
        end
    end
    ii=ii+1;
    if ii<=num_materiales
        if  sit.mat_21_uni==1 && pos_uni(ii)==1
            sit.mat_21_uni=0;
```

```
        app.Material21_uniCheckBox.Value=0;
        delete(dib.mat_21_uni)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_22_uni==1 && pos_uni(ii)==1
        sit.mat_22_uni=0;
        app.Material22_uniCheckBox.Value=0;
        delete(dib.mat_22_uni)
    end
end

ii=1;
if ii<=num_materiales
    if  sit.mat_1_tor==1 && pos_tor(ii)==1
        sit.mat_1_tor=0;
        app.Material1_torCheckBox.Value=0;
        delete(dib_t.mat_1_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_2_tor==1 && pos_tor(ii)==1
        sit.mat_2_tor=0;
        app.Material2_torCheckBox.Value=0;
        delete(dib_t.mat_2_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_3_tor==1 && pos_tor(ii)==1
        sit.mat_3_tor=0;
        app.Material3_torCheckBox.Value=0;
        delete(dib_t.mat_3_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_4_tor==1 && pos_tor(ii)==1
        sit.mat_4_tor=0;
        app.Material4_torCheckBox.Value=0;
        delete(dib_t.mat_4_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_4_tor==1 && pos_tor(ii)==1
        sit.mat_4_tor=0;
        app.Material4_torCheckBox.Value=0;
        delete(dib_t.mat_4_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_5_tor==1 && pos_tor(ii)==1
        sit.mat_5_tor=0;
        app.Material5_torCheckBox.Value=0;
        delete(dib_t.mat_5_tor)
    end
end
if ii<=num_materiales
    if  sit.mat_6_tor==1 && pos_tor(ii)==1
        sit.mat_6_tor=0;
        app.Material6_torCheckBox.Value=0;
        delete(dib_t.mat_6_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_7_tor==1 && pos_tor(ii)==1
        sit.mat_7_tor=0;
        app.Material7_torCheckBox.Value=0;
        delete(dib_t.mat_7_tor)
```

```
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_8_tor==1 && pos_tor(ii)==1
        sit.mat_8_tor=0;
        app.Material8_torCheckBox.Value=0;
        delete(dib_t.mat_8_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_9_tor==1 && pos_tor(ii)==1
        sit.mat_9_tor=0;
        app.Material9_torCheckBox.Value=0;
        delete(dib_t.mat_9_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_10_tor==1 && pos_tor(ii)==1
        sit.mat_10_tor=0;
        app.Material10_torCheckBox.Value=0;
        delete(dib_t.mat_10_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_11_tor==1 && pos_tor(ii)==1
        sit.mat_11_tor=0;
        app.Material11_torCheckBox.Value=0;
        delete(dib_t.mat_11_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_12_tor==1 && pos_tor(ii)==1
        sit.mat_12_tor=0;
        app.Material12_torCheckBox.Value=0;
        delete(dib_t.mat_12_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_13_tor==1 && pos_tor(ii)==1
        sit.mat_13_tor=0;
        app.Material13_torCheckBox.Value=0;
        delete(dib_t.mat_13_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_14_tor==1 && pos_tor(ii)==1
        sit.mat_14_tor=0;
        app.Material14_torCheckBox.Value=0;
        delete(dib_t.mat_14_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_15_tor==1 && pos_tor(ii)==1
        sit.mat_15_tor=0;
        app.Material15_torCheckBox.Value=0;
        delete(dib_t.mat_15_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_16_tor==1 && pos_tor(ii)==1
        sit.mat_16_tor=0;
        app.Material16_torCheckBox.Value=0;
        delete(dib_t.mat_16_tor)
    end
end
```

```
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_17_tor==1 && pos_tor(ii)==1
        sit.mat_17_tor=0;
        app.Material17_torCheckBox.Value=0;
        delete(dib_t.mat_17_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_18_tor==1 && pos_tor(ii)==1
        sit.mat_18_tor=0;
        app.Material18_torCheckBox.Value=0;
        delete(dib_t.mat_18_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_19_tor==1 && pos_tor(ii)==1
        sit.mat_19_tor=0;
        app.Material19_torCheckBox.Value=0;
        delete(dib_t.mat_19_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_20_tor==1 && pos_tor(ii)==1
        sit.mat_20_tor=0;
        app.Material20_torCheckBox.Value=0;
        delete(dib_t.mat_20_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_21_tor==1 && pos_tor(ii)==1
        sit.mat_21_tor=0;
        app.Material21_torCheckBox.Value=0;
        delete(dib_t.mat_21_tor)
    end
end
ii=ii+1;
if ii<=num_materiales
    if  sit.mat_22_tor==1 && pos_tor(ii)==1
        sit.mat_22_tor=0;
        app.Material22_torCheckBox.Value=0;
        delete(dib_t.mat_22_tor)
    end
end


app.CargandoLabel.Visible="off";
app.DireccinLabel.Visible="off";
app.direccion_excelLabel.Visible="off";
app.NombreLabel.Visible="off";
app.nombre_excelLabel.Visible="off";

app.Sel_all_exc_uni.Visible="off";
app.Unsel_all_exc_uni.Visible="off";
app.Sel_all_exc_tor.Visible="off";
app.Unsel_all_exc_tor.Visible="off";

app.Material1_uniCheckBox.Visible="off";
app.im_mat_1_uni.Visible="off";
app.Material1_torCheckBox.Visible="off";
app.im_mat_1_tor.Visible="off";
app.Material2_uniCheckBox.Visible="off";
app.im_mat_2_uni.Visible="off";
app.Material2_torCheckBox.Visible="off";
app.im_mat_2_tor.Visible="off";
app.Material3_uniCheckBox.Visible="off";
app.im_mat_3_uni.Visible="off";
app.Material3_torCheckBox.Visible="off";
app.im_mat_3_tor.Visible="off";
app.Material4_uniCheckBox.Visible="off";
```

```
app.im_mat_4_uni.Visible="off";
app.Material4_torCheckBox.Visible="off";
app.im_mat_4_tor.Visible="off";
app.Material5_uniCheckBox.Visible="off";
app.im_mat_5_uni.Visible="off";
app.Material5_torCheckBox.Visible="off";
app.im_mat_5_tor.Visible="off";
app.Material6_uniCheckBox.Visible="off";
app.im_mat_6_uni.Visible="off";
app.Material6_torCheckBox.Visible="off";
app.im_mat_6_tor.Visible="off";
app.Material7_uniCheckBox.Visible="off";
app.im_mat_7_uni.Visible="off";
app.Material7_torCheckBox.Visible="off";
app.im_mat_7_tor.Visible="off";
app.Material8_uniCheckBox.Visible="off";
app.im_mat_8_uni.Visible="off";
app.Material8_torCheckBox.Visible="off";
app.im_mat_8_tor.Visible="off";
app.Material9_uniCheckBox.Visible="off";
app.im_mat_9_uni.Visible="off";
app.Material9_torCheckBox.Visible="off";
app.im_mat_9_tor.Visible="off";
app.Material10_uniCheckBox.Visible="off";
app.im_mat_10_uni.Visible="off";
app.Material10_torCheckBox.Visible="off";
app.im_mat_10_tor.Visible="off";
app.Material11_uniCheckBox.Visible="off";
app.im_mat_11_uni.Visible="off";
app.Material11_torCheckBox.Visible="off";
app.im_mat_11_tor.Visible="off";
app.Material12_uniCheckBox.Visible="off";
app.im_mat_12_uni.Visible="off";
app.Material12_torCheckBox.Visible="off";
app.im_mat_12_tor.Visible="off";
app.Material13_uniCheckBox.Visible="off";
app.im_mat_13_uni.Visible="off";
app.Material13_torCheckBox.Visible="off";
app.im_mat_13_tor.Visible="off";
app.Material14_uniCheckBox.Visible="off";
app.im_mat_14_uni.Visible="off";
app.Material14_torCheckBox.Visible="off";
app.im_mat_14_tor.Visible="off";
app.Material15_uniCheckBox.Visible="off";
app.im_mat_15_uni.Visible="off";
app.Material15_torCheckBox.Visible="off";
app.im_mat_15_tor.Visible="off";
app.Material16_uniCheckBox.Visible="off";
app.im_mat_16_uni.Visible="off";
app.Material16_torCheckBox.Visible="off";
app.im_mat_16_tor.Visible="off";
app.Material17_uniCheckBox.Visible="off";
app.im_mat_17_uni.Visible="off";
app.Material17_torCheckBox.Visible="off";
app.im_mat_17_tor.Visible="off";
app.Material18_uniCheckBox.Visible="off";
app.im_mat_18_uni.Visible="off";
app.Material18_torCheckBox.Visible="off";
app.im_mat_18_tor.Visible="off";
app.Material19_uniCheckBox.Visible="off";
app.im_mat_19_uni.Visible="off";
app.Material19_torCheckBox.Visible="off";
app.im_mat_19_tor.Visible="off";
app.Material20_uniCheckBox.Visible="off";
app.im_mat_20_uni.Visible="off";
app.Material20_torCheckBox.Visible="off";
app.im_mat_20_tor.Visible="off";
app.Material21_uniCheckBox.Visible="off";
app.im_mat_21_uni.Visible="off";
app.Material21_torCheckBox.Visible="off";
app.im_mat_21_tor.Visible="off";
app.Material22_uniCheckBox.Visible="off";
app.im_mat_22_uni.Visible="off";
app.Material22_torCheckBox.Visible="off";
```

```
            app.im_mat_22_tor.Visible="off";


        end

        % Changes arrangement of the app based on UIFigure width
        function updateAppLayout(app, event)
            currentFigureWidth = app.pag_principal.Position(3);
            if(currentFigureWidth <= app.onePanelWidth)
                % Change to a 3x1 grid
                app.GridLayout.RowHeight = {755, 755, 755};
                app.GridLayout.ColumnWidth = {'1x'};
                app.CenterPanel.Layout.Row = 1;
                app.CenterPanel.Layout.Column = 1;
                app.LeftPanel.Layout.Row = 2;
                app.LeftPanel.Layout.Column = 1;
                app.RightPanel.Layout.Row = 3;
                app.RightPanel.Layout.Column = 1;
            elseif (currentFigureWidth > app.onePanelWidth && currentFigureWidth <= app.twoPanelWidth)
                % Change to a 2x2 grid
                app.GridLayout.RowHeight = {755, 755};
                app.GridLayout.ColumnWidth = {'1x', '1x'};
                app.CenterPanel.Layout.Row = 1;
                app.CenterPanel.Layout.Column = [1,2];
                app.LeftPanel.Layout.Row = 2;
                app.LeftPanel.Layout.Column = 1;
                app.RightPanel.Layout.Row = 2;
                app.RightPanel.Layout.Column = 2;
            else
                % Change to a 1x3 grid
                app.GridLayout.RowHeight = {'1x'};
                app.GridLayout.ColumnWidth = {341, '1x', 261};
                app.LeftPanel.Layout.Row = 1;
                app.LeftPanel.Layout.Column = 1;
                app.CenterPanel.Layout.Row = 1;
                app.CenterPanel.Layout.Column = 2;
                app.RightPanel.Layout.Row = 1;
                app.RightPanel.Layout.Column = 3;
            end
        end
    end

    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create pag_principal and hide until all components are created
            app.pag_principal = uifigure('Visible', 'off');
            app.pag_principal.AutoResizeChildren = 'off';
            app.pag_principal.Position = [100 100 1268 755];
            app.pag_principal.Name = 'MATLAB App';
            app.pag_principal.SizeChangedFcn = createCallbackFcn(app, @updateAppLayout, true);

            % Create GridLayout
            app.GridLayout = uigridlayout(app.pag_principal);
            app.GridLayout.ColumnWidth = {341, '1x', 261};
            app.GridLayout.RowHeight = {'1x'};
            app.GridLayout.ColumnSpacing = 0;
            app.GridLayout.RowSpacing = 0;
            app.GridLayout.Padding = [0 0 0 0];
            app.GridLayout.Scrollable = 'on';

            % Create LeftPanel
            app.LeftPanel = uipanel(app.GridLayout);
            app.LeftPanel.Layout.Row = 1;
            app.LeftPanel.Layout.Column = 1;

            % Create CenterPanel
            app.CenterPanel = uipanel(app.GridLayout);
            app.CenterPanel.Layout.Row = 1;
            app.CenterPanel.Layout.Column = 2;
```

```
            % Create Representacion
            app.Representacion = uiaxes(app.CenterPanel);
            title(app.Representacion, 'Estado tensional uniaxial \sigma_{xx,m} - \sigma_{xx,a}')
            xlabel(app.Representacion, '\sigma_{xx,m}/\sigma_{ut}')
            ylabel(app.Representacion, '\sigma_{xx,a}/\sigma_{-1}')
            app.Representacion.Position = [30 388 607 350];

            % Create Representacion_torsion
            app.Representacion_torsion = uiaxes(app.CenterPanel);
            title(app.Representacion_torsion, 'Estado tensional torsional \tau_{xx,m} - \tau_{xx,a}')
            xlabel(app.Representacion_torsion, '         \tau_{xx,m}/\tau_{ut}
(\tau_{ut}=0.75\sigma_{ut})')
            ylabel(app.Representacion_torsion, '\tau_{xx,a}/\tau_{-1}')
            app.Representacion_torsion.Position = [34 16 609 357];

            % Create CuadrculaSwitchLabel
            app.CuadrculaSwitchLabel = uilabel(app.CenterPanel);
            app.CuadrculaSwitchLabel.HorizontalAlignment = 'center';
            app.CuadrculaSwitchLabel.Position = [13 385 63 22];
            app.CuadrculaSwitchLabel.Text = 'Cuadrícula';

            % Create CuadrculaSwitch
            app.CuadrculaSwitch = uiswitch(app.CenterPanel, 'slider');
            app.CuadrculaSwitch.Items = {'Si', 'No'};
            app.CuadrculaSwitch.ValueChangedFcn = createCallbackFcn(app, @CuadrculaSwitchValueChanged, true);
            app.CuadrculaSwitch.Position = [23 360 45 20];
            app.CuadrculaSwitch.Value = 'Si';

            % Create RightPanel
            app.RightPanel = uipanel(app.GridLayout);
            app.RightPanel.Layout.Row = 1;
            app.RightPanel.Layout.Column = 3;

            % Create SinesCheckBox
            app.SinesCheckBox = uicheckbox(app.RightPanel);
            app.SinesCheckBox.ValueChangedFcn = createCallbackFcn(app, @SinesCheckBoxValueChanged, true);
            app.SinesCheckBox.Text = 'Sines';
            app.SinesCheckBox.FontColor = [0.4941 0.1843 0.5569];
            app.SinesCheckBox.Position = [15 509 52 22];

            % Create CrosslandCheckBox
            app.CrosslandCheckBox = uicheckbox(app.RightPanel);
            app.CrosslandCheckBox.ValueChangedFcn = createCallbackFcn(app, @CrosslandCheckBoxValueChanged,
true);
            app.CrosslandCheckBox.Text = 'Crossland';
            app.CrosslandCheckBox.FontColor = [0.5569 0.6588 0.4667];
            app.CrosslandCheckBox.Position = [15 470 76 22];
            app.CrosslandCheckBox.Value = true;

            % Create MatakeCheckBox
            app.MatakeCheckBox = uicheckbox(app.RightPanel);
            app.MatakeCheckBox.ValueChangedFcn = createCallbackFcn(app, @MatakeCheckBoxValueChanged, true);
            app.MatakeCheckBox.Text = 'Matake';
            app.MatakeCheckBox.FontColor = [0.451 0.549 0.7294];
            app.MatakeCheckBox.Position = [15 431 61 22];
            app.MatakeCheckBox.Value = true;

            % Create FindleyCheckBox
            app.FindleyCheckBox = uicheckbox(app.RightPanel);
            app.FindleyCheckBox.ValueChangedFcn = createCallbackFcn(app, @FindleyCheckBoxValueChanged, true);
            app.FindleyCheckBox.Text = 'Findley';
            app.FindleyCheckBox.FontColor = [0.851 0.3255 0.098];
            app.FindleyCheckBox.Position = [15 393 61 22];
            app.FindleyCheckBox.Value = true;

            % Create DangVanCheckBox
            app.DangVanCheckBox = uicheckbox(app.RightPanel);
            app.DangVanCheckBox.ValueChangedFcn = createCallbackFcn(app, @DangVanCheckBoxValueChanged, true);
            app.DangVanCheckBox.Text = 'Dang Van';
            app.DangVanCheckBox.FontColor = [0.9294 0.6902 0.1294];
            app.DangVanCheckBox.Position = [15 355 75 22];

            % Create GoodmanCheckBox
```

```
            app.GoodmanCheckBox = uicheckbox(app.RightPanel);
            app.GoodmanCheckBox.ValueChangedFcn = createCallbackFcn(app, @GoodmanCheckBoxValueChanged, true);
            app.GoodmanCheckBox.Text = 'Goodman';
            app.GoodmanCheckBox.Position = [15 198 75 22];

            % Create DietmannCheckBox
            app.DietmannCheckBox = uicheckbox(app.RightPanel);
            app.DietmannCheckBox.ValueChangedFcn = createCallbackFcn(app, @DietmannCheckBoxValueChanged,
true);
            app.DietmannCheckBox.Text = 'Dietmann';
            app.DietmannCheckBox.Position = [15 166 73 22];

            % Create MarinElipseCheckBox
            app.MarinElipseCheckBox = uicheckbox(app.RightPanel);
            app.MarinElipseCheckBox.ValueChangedFcn = createCallbackFcn(app, @MarinElipseCheckBoxValueChanged,
true);
            app.MarinElipseCheckBox.Text = 'Marin (Elipse)';
            app.MarinElipseCheckBox.Position = [15 103 96 22];

            % Create MtododeFatigaMultiaxialLabel
            app.MtododeFatigaMultiaxialLabel = uilabel(app.RightPanel);
            app.MtododeFatigaMultiaxialLabel.Position = [7 581 152 22];
            app.MtododeFatigaMultiaxialLabel.Text = 'Método de Fatiga Multiaxial';

            % Create CriteriosdeajusteFatigaLabel
            app.CriteriosdeajusteFatigaLabel = uilabel(app.RightPanel);
            app.CriteriosdeajusteFatigaLabel.Position = [7 234 139 22];
            app.CriteriosdeajusteFatigaLabel.Text = 'Criterios de ajuste Fatiga';

            % Create SalirButton
            app.SalirButton = uibutton(app.RightPanel, 'push');
            app.SalirButton.ButtonPushedFcn = createCallbackFcn(app, @SalirButtonPushed, true);
            app.SalirButton.Position = [177 39 74 22];
            app.SalirButton.Text = 'Salir';

            % Create GerberCheckBox_2
            app.GerberCheckBox_2 = uicheckbox(app.RightPanel);
            app.GerberCheckBox_2.ValueChangedFcn = createCallbackFcn(app, @GerberCheckBox_2ValueChanged,
true);
            app.GerberCheckBox_2.Text = 'Gerber';
            app.GerberCheckBox_2.Position = [15 134 59 22];

            % Create AlfaSines
            app.AlfaSines = uieditfield(app.RightPanel, 'numeric');
            app.AlfaSines.ValueChangedFcn = createCallbackFcn(app, @AlfaSinesValueChanged, true);
            app.AlfaSines.Editable = 'off';
            app.AlfaSines.BackgroundColor = [0.9412 0.9412 0.9412];
            app.AlfaSines.Position = [101 509 50 22];

            % Create BetaSines
            app.BetaSines = uieditfield(app.RightPanel, 'numeric');
            app.BetaSines.ValueChangedFcn = createCallbackFcn(app, @BetaSinesValueChanged, true);
            app.BetaSines.Editable = 'off';
            app.BetaSines.BackgroundColor = [0.9412 0.9412 0.9412];
            app.BetaSines.Position = [188 509 50 22];

            % Create AlfaCrossland
            app.AlfaCrossland = uieditfield(app.RightPanel, 'numeric');
            app.AlfaCrossland.ValueChangedFcn = createCallbackFcn(app, @AlfaCrosslandValueChanged, true);
            app.AlfaCrossland.Editable = 'off';
            app.AlfaCrossland.BackgroundColor = [0.9412 0.9412 0.9412];
            app.AlfaCrossland.Position = [101 470 50 22];

            % Create BetaCrossland
            app.BetaCrossland = uieditfield(app.RightPanel, 'numeric');
            app.BetaCrossland.ValueChangedFcn = createCallbackFcn(app, @BetaCrosslandValueChanged, true);
            app.BetaCrossland.Editable = 'off';
            app.BetaCrossland.BackgroundColor = [0.9412 0.9412 0.9412];
            app.BetaCrossland.Position = [188 470 50 22];

            % Create AlfaMatake
            app.AlfaMatake = uieditfield(app.RightPanel, 'numeric');
            app.AlfaMatake.ValueChangedFcn = createCallbackFcn(app, @AlfaMatakeValueChanged, true);
            app.AlfaMatake.Editable = 'off';
```

```
app.AlfaMatake.BackgroundColor = [0.9412 0.9412 0.9412];
app.AlfaMatake.Position = [101 431 50 22];

% Create BetaMatake
app.BetaMatake = uieditfield(app.RightPanel, 'numeric');
app.BetaMatake.ValueChangedFcn = createCallbackFcn(app, @BetaMatakeValueChanged, true);
app.BetaMatake.Editable = 'off';
app.BetaMatake.BackgroundColor = [0.9412 0.9412 0.9412];
app.BetaMatake.Position = [188 431 50 22];

% Create AlfaFindley
app.AlfaFindley = uieditfield(app.RightPanel, 'numeric');
app.AlfaFindley.ValueChangedFcn = createCallbackFcn(app, @AlfaFindleyValueChanged, true);
app.AlfaFindley.Editable = 'off';
app.AlfaFindley.BackgroundColor = [0.9412 0.9412 0.9412];
app.AlfaFindley.Position = [101 393 50 22];

% Create BetaFindley
app.BetaFindley = uieditfield(app.RightPanel, 'numeric');
app.BetaFindley.ValueChangedFcn = createCallbackFcn(app, @BetaFindleyValueChanged, true);
app.BetaFindley.Editable = 'off';
app.BetaFindley.BackgroundColor = [0.9412 0.9412 0.9412];
app.BetaFindley.Position = [188 393 50 22];

% Create AlfaDangVan
app.AlfaDangVan = uieditfield(app.RightPanel, 'numeric');
app.AlfaDangVan.ValueChangedFcn = createCallbackFcn(app, @AlfaDangVanValueChanged, true);
app.AlfaDangVan.Editable = 'off';
app.AlfaDangVan.BackgroundColor = [0.9412 0.9412 0.9412];
app.AlfaDangVan.Position = [101 355 50 22];

% Create BetaDangVan
app.BetaDangVan = uieditfield(app.RightPanel, 'numeric');
app.BetaDangVan.ValueChangedFcn = createCallbackFcn(app, @BetaDangVanValueChanged, true);
app.BetaDangVan.Editable = 'off';
app.BetaDangVan.BackgroundColor = [0.9412 0.9412 0.9412];
app.BetaDangVan.Position = [188 355 50 22];

% Create MtodoLabel
app.MtodoLabel = uilabel(app.RightPanel);
app.MtodoLabel.Position = [25 544 41 22];
app.MtodoLabel.Text = 'Método';

% Create Label_3
app.Label_3 = uilabel(app.RightPanel);
app.Label_3.Position = [121 544 10 22];
app.Label_3.Text = '?';

% Create Label_4
app.Label_4 = uilabel(app.RightPanel);
app.Label_4.Position = [208 544 10 22];
app.Label_4.Text = '?';

% Create CheckBox
app.CheckBox = uicheckbox(app.RightPanel);
app.CheckBox.ValueChangedFcn = createCallbackFcn(app, @CheckBoxValueChanged, true);
app.CheckBox.Text = '?   - ?  ';
app.CheckBox.Position = [7 624 58 22];
app.CheckBox.Value = true;

% Create ParejadeensayosempleadosLabel
app.ParejadeensayosempleadosLabel = uilabel(app.RightPanel);
app.ParejadeensayosempleadosLabel.Position = [7 660 167 22];
app.ParejadeensayosempleadosLabel.Text = 'Pareja de ensayos empleados';

% Create CheckBox_2
app.CheckBox_2 = uicheckbox(app.RightPanel);
app.CheckBox_2.ValueChangedFcn = createCallbackFcn(app, @CheckBox_2ValueChanged, true);
app.CheckBox_2.Text = '?   - ?   ';
app.CheckBox_2.Position = [93 625 64 22];

% Create CheckBox_3
app.CheckBox_3 = uicheckbox(app.RightPanel);
app.CheckBox_3.ValueChangedFcn = createCallbackFcn(app, @CheckBox_3ValueChanged, true);
```

```
app.CheckBox_3.Text = '?   - ?  ';
app.CheckBox_3.Position = [180 625 61 22];

% Create ResultadosLabel
app.ResultadosLabel = uilabel(app.RightPanel);
app.ResultadosLabel.FontWeight = 'bold';
app.ResultadosLabel.Position = [7 705 70 22];
app.ResultadosLabel.Text = 'Resultados';

% Create Image2
app.Image2 = uiimage(app.RightPanel);
app.Image2.Position = [126 198 65 22];
app.Image2.ImageSource = 'raya_puntos.PNG';

% Create Image2_2
app.Image2_2 = uiimage(app.RightPanel);
app.Image2_2.Position = [126 166 65 22];
app.Image2_2.ImageSource = 'raya_rayas.PNG';

% Create Image2_3
app.Image2_3 = uiimage(app.RightPanel);
app.Image2_3.Position = [123 134 77 22];
app.Image2_3.ImageSource = 'raya_rayapunto.PNG';

% Create Image2_4
app.Image2_4 = uiimage(app.RightPanel);
app.Image2_4.Position = [126 103 64 22];
app.Image2_4.ImageSource = 'raya_continua.PNG';

% Create PapugaCheckBox
app.PapugaCheckBox = uicheckbox(app.RightPanel);
app.PapugaCheckBox.ValueChangedFcn = createCallbackFcn(app, @PapugaCheckBoxValueChanged, true);
app.PapugaCheckBox.Text = 'Papuga';
app.PapugaCheckBox.FontColor = [0.502 0.502 0.502];
app.PapugaCheckBox.Position = [15 291 63 22];
app.PapugaCheckBox.Value = true;

% Create a_p_Papuga
app.a_p_Papuga = uieditfield(app.RightPanel, 'numeric');
app.a_p_Papuga.ValueChangedFcn = createCallbackFcn(app, @a_p_PapugaValueChanged, true);
app.a_p_Papuga.Editable = 'off';
app.a_p_Papuga.BackgroundColor = [0.9412 0.9412 0.9412];
app.a_p_Papuga.Position = [101 291 50 22];

% Create b_p_Papuga
app.b_p_Papuga = uieditfield(app.RightPanel, 'numeric');
app.b_p_Papuga.ValueChangedFcn = createCallbackFcn(app, @b_p_PapugaValueChanged, true);
app.b_p_Papuga.Editable = 'off';
app.b_p_Papuga.BackgroundColor = [0.9412 0.9412 0.9412];
app.b_p_Papuga.Position = [188 291 50 22];

% Create aLabel
app.aLabel = uilabel(app.RightPanel);
app.aLabel.Position = [121 320 10 22];
app.aLabel.Text = 'a';

% Create pLabel
app.pLabel = uilabel(app.RightPanel);
app.pLabel.FontSize = 9;
app.pLabel.Position = [128 316 11 22];
app.pLabel.Text = 'p';

% Create bLabel
app.bLabel = uilabel(app.RightPanel);
app.bLabel.Position = [208 320 11 22];
app.bLabel.Text = 'b';

% Create pLabel_2
app.pLabel_2 = uilabel(app.RightPanel);
app.pLabel_2.FontSize = 9;
app.pLabel_2.Position = [215 316 10 22];
app.pLabel_2.Text = 'p';

% Create Label_15
```

```
app.Label_15 = uilabel(app.RightPanel);
app.Label_15.FontSize = 8;
app.Label_15.Position = [33 618 24 22];
app.Label_15.Text = '-1';

% Create Label_16
app.Label_16 = uilabel(app.RightPanel);
app.Label_16.FontSize = 8;
app.Label_16.Position = [55 618 25 22];
app.Label_16.Text = '-1';

% Create Label_17
app.Label_17 = uilabel(app.RightPanel);
app.Label_17.FontSize = 8;
app.Label_17.Position = [118 619 25 22];
app.Label_17.Text = '-1';

% Create Label_18
app.Label_18 = uilabel(app.RightPanel);
app.Label_18.FontSize = 8;
app.Label_18.Position = [143 619 21 22];
app.Label_18.Text = '0';

% Create Label_19
app.Label_19 = uilabel(app.RightPanel);
app.Label_19.FontSize = 8;
app.Label_19.Position = [206 619 25 22];
app.Label_19.Text = '-1';

% Create utLabel_2
app.utLabel_2 = uilabel(app.RightPanel);
app.utLabel_2.FontSize = 8;
app.utLabel_2.Position = [229 619 15 22];
app.utLabel_2.Text = 'ut';

% Create TabGroup
app.TabGroup = uitabgroup(app.pag_principal);
app.TabGroup.Position = [-12 1 354 755];

% Create DatosTab
app.DatosTab = uitab(app.TabGroup);
app.DatosTab.Title = '   Datos';

% Create CalcularButton
app.CalcularButton = uibutton(app.DatosTab, 'push');
app.CalcularButton.ButtonPushedFcn = createCallbackFcn(app, @CalcularButtonPushed2, true);
app.CalcularButton.Position = [27 40 100 22];
app.CalcularButton.Text = 'Calcular';

% Create Button_2
app.Button_2 = uibutton(app.DatosTab, 'push');
app.Button_2.ButtonPushedFcn = createCallbackFcn(app, @Button_2Pushed2, true);
app.Button_2.Icon = 'uialert_info.png';
app.Button_2.Position = [208 158 26 22];
app.Button_2.Text = '';

% Create Button_3
app.Button_3 = uibutton(app.DatosTab, 'push');
app.Button_3.ButtonPushedFcn = createCallbackFcn(app, @Button_3Pushed2, true);
app.Button_3.Icon = 'uialert_info.png';
app.Button_3.Position = [208 117 26 22];
app.Button_3.Text = '';

% Create Lamp_calcular
app.Lamp_calcular = uilamp(app.DatosTab);
app.Lamp_calcular.Position = [138 41 20 20];
app.Lamp_calcular.Color = [0.9412 0.9412 0.9412];

% Create PuntosacalcularSpinnerLabel
app.PuntosacalcularSpinnerLabel = uilabel(app.DatosTab);
app.PuntosacalcularSpinnerLabel.WordWrap = 'on';
app.PuntosacalcularSpinnerLabel.Position = [35 154 94 31];
app.PuntosacalcularSpinnerLabel.Text = 'Puntos a calcular';
```

```
            % Create PuntosacalcularSpinner
            app.PuntosacalcularSpinner = uispinner(app.DatosTab);
            app.PuntosacalcularSpinner.Step = 10;
            app.PuntosacalcularSpinner.Limits = [10 500];
            app.PuntosacalcularSpinner.RoundFractionalValues = 'on';
            app.PuntosacalcularSpinner.ValueChangedFcn = createCallbackFcn(app,
@PuntosacalcularSpinnerValueChanged2, true);
            app.PuntosacalcularSpinner.HorizontalAlignment = 'center';
            app.PuntosacalcularSpinner.Position = [138 158 64 22];
            app.PuntosacalcularSpinner.Value = 20;

            % Create Label_5
            app.Label_5 = uilabel(app.DatosTab);
            app.Label_5.Position = [35 445 25 22];
            app.Label_5.Text = '?';

            % Create EditField_3
            app.EditField_3 = uieditfield(app.DatosTab, 'numeric');
            app.EditField_3.ValueChangedFcn = createCallbackFcn(app, @EditField_3ValueChanged2, true);
            app.EditField_3.Position = [77 445 49 22];
            app.EditField_3.Value = 615;

            % Create kEditFieldLabel
            app.kEditFieldLabel = uilabel(app.DatosTab);
            app.kEditFieldLabel.Position = [35 407 25 22];
            app.kEditFieldLabel.Text = 'k';

            % Create kEditField
            app.kEditField = uieditfield(app.DatosTab, 'numeric');
            app.kEditField.ValueChangedFcn = createCallbackFcn(app, @kEditFieldValueChanged2, true);
            app.kEditField.Position = [77 407 49 22];
            app.kEditField.Value = 1.4203;

            % Create Label_7
            app.Label_7 = uilabel(app.DatosTab);
            app.Label_7.Position = [35 370 94 22];
            app.Label_7.Text = '?';

            % Create EditField_4
            app.EditField_4 = uieditfield(app.DatosTab, 'numeric');
            app.EditField_4.ValueChangedFcn = createCallbackFcn(app, @EditField_4ValueChanged2, true);
            app.EditField_4.Position = [77 370 49 22];
            app.EditField_4.Value = 433;

            % Create EditField_6Label
            app.EditField_6Label = uilabel(app.DatosTab);
            app.EditField_6Label.Position = [35 259 25 22];
            app.EditField_6Label.Text = '?';

            % Create EditField_6
            app.EditField_6 = uieditfield(app.DatosTab, 'numeric');
            app.EditField_6.ValueChangedFcn = createCallbackFcn(app, @EditField_6ValueChanged2, true);
            app.EditField_6.Position = [77 259 49 22];
            app.EditField_6.Value = 1200;

            % Create ToleranciaSpinnerLabel
            app.ToleranciaSpinnerLabel = uilabel(app.DatosTab);
            app.ToleranciaSpinnerLabel.Position = [35 117 56 22];
            app.ToleranciaSpinnerLabel.Text = 'Tolerancia';

            % Create ToleranciaSpinner
            app.ToleranciaSpinner = uispinner(app.DatosTab);
            app.ToleranciaSpinner.Step = 0.01;
            app.ToleranciaSpinner.Limits = [0.001 0.2];
            app.ToleranciaSpinner.ValueChangedFcn = createCallbackFcn(app, @ToleranciaSpinnerValueChanged2,
true);
            app.ToleranciaSpinner.HorizontalAlignment = 'center';
            app.ToleranciaSpinner.Position = [138 117 64 22];
            app.ToleranciaSpinner.Value = 0.01;

            % Create Label_9
            app.Label_9 = uilabel(app.DatosTab);
            app.Label_9.Position = [35 314 25 22];
            app.Label_9.Text = '?';
```

```
% Create EditField_5
app.EditField_5 = uieditfield(app.DatosTab, 'numeric');
app.EditField_5.ValueChangedFcn = createCallbackFcn(app, @EditField_5ValueChanged2, true);
app.EditField_5.Position = [77 314 49 22];
app.EditField_5.Value = 480;

% Create Label_11
app.Label_11 = uilabel(app.DatosTab);
app.Label_11.Position = [141 342 25 22];
app.Label_11.Text = '?';

% Create s_0_Goodman
app.s_0_Goodman = uieditfield(app.DatosTab, 'numeric');
app.s_0_Goodman.ValueChangedFcn = createCallbackFcn(app, @s_0_GoodmanValueChanged, true);
app.s_0_Goodman.Editable = 'off';
app.s_0_Goodman.BackgroundColor = [0.9412 0.9412 0.9412];
app.s_0_Goodman.Position = [185 342 49 22];

% Create Label_14
app.Label_14 = uilabel(app.DatosTab);
app.Label_14.Position = [141 292 25 22];
app.Label_14.Text = '?';

% Create s_0_Marin
app.s_0_Marin = uieditfield(app.DatosTab, 'numeric');
app.s_0_Marin.ValueChangedFcn = createCallbackFcn(app, @s_0_MarinValueChanged, true);
app.s_0_Marin.Editable = 'off';
app.s_0_Marin.BackgroundColor = [0.9412 0.9412 0.9412];
app.s_0_Marin.Position = [186 292 49 22];

% Create UNIAXIALPUROLabel
app.UNIAXIALPUROLabel = uilabel(app.DatosTab);
app.UNIAXIALPUROLabel.FontWeight = 'bold';
app.UNIAXIALPUROLabel.Position = [45 668 100 22];
app.UNIAXIALPUROLabel.Text = 'UNIAXIAL PURO';

% Create Image
app.Image = uiimage(app.DatosTab);
app.Image.Position = [21 531 147 131];
app.Image.ImageSource = 'esquema_uniaxial_inicio.PNG';

% Create TORSINPUROLabel
app.TORSINPUROLabel = uilabel(app.DatosTab);
app.TORSINPUROLabel.FontWeight = 'bold';
app.TORSINPUROLabel.Position = [223 668 98 22];
app.TORSINPUROLabel.Text = 'TORSIÓN PURO';

% Create Image_2
app.Image_2 = uiimage(app.DatosTab);
app.Image_2.Position = [198 531 147 131];
app.Image_2.ImageSource = 'esquema_torsion_inicio.PNG';

% Create EstadostensionalesLabel
app.EstadostensionalesLabel = uilabel(app.DatosTab);
app.EstadostensionalesLabel.FontSize = 13;
app.EstadostensionalesLabel.FontWeight = 'bold';
app.EstadostensionalesLabel.Position = [24 698 131 22];
app.EstadostensionalesLabel.Text = 'Estados tensionales';

% Create DatosdelmaterialLabel
app.DatosdelmaterialLabel = uilabel(app.DatosTab);
app.DatosdelmaterialLabel.FontSize = 13;
app.DatosdelmaterialLabel.FontWeight = 'bold';
app.DatosdelmaterialLabel.Position = [24 484 117 22];
app.DatosdelmaterialLabel.Text = 'Datos del material';

% Create DatosparalaresolucinLabel
app.DatosparalaresolucinLabel = uilabel(app.DatosTab);
app.DatosparalaresolucinLabel.FontSize = 13;
app.DatosparalaresolucinLabel.FontWeight = 'bold';
app.DatosparalaresolucinLabel.Position = [24 205 156 22];
app.DatosparalaresolucinLabel.Text = 'Datos para la resolución';
```

```
            % Create Button_4
            app.Button_4 = uibutton(app.DatosTab, 'push');
            app.Button_4.ButtonPushedFcn = createCallbackFcn(app, @Button_4Pushed, true);
            app.Button_4.Icon = 'uialert_info.png';
            app.Button_4.Position = [244 317 26 22];
            app.Button_4.Text = '';

            % Create Label_6
            app.Label_6 = uilabel(app.DatosTab);
            app.Label_6.FontSize = 9;
            app.Label_6.Position = [42 440 25 22];
            app.Label_6.Text = '-1';

            % Create Label_8
            app.Label_8 = uilabel(app.DatosTab);
            app.Label_8.FontSize = 9;
            app.Label_8.Position = [42 365 25 22];
            app.Label_8.Text = '-1';

            % Create Label_10
            app.Label_10 = uilabel(app.DatosTab);
            app.Label_10.FontSize = 9;
            app.Label_10.Position = [42 308 25 22];
            app.Label_10.Text = '0';

            % Create Label_12
            app.Label_12 = uilabel(app.DatosTab);
            app.Label_12.FontSize = 9;
            app.Label_12.Position = [148 336 25 22];
            app.Label_12.Text = '0';

            % Create GLabel
            app.GLabel = uilabel(app.DatosTab);
            app.GLabel.FontSize = 8;
            app.GLabel.Position = [153 332 25 22];
            app.GLabel.Text = 'G';

            % Create Label_13
            app.Label_13 = uilabel(app.DatosTab);
            app.Label_13.FontSize = 9;
            app.Label_13.Position = [148 285 25 22];
            app.Label_13.Text = '0';

            % Create MLabel_2
            app.MLabel_2 = uilabel(app.DatosTab);
            app.MLabel_2.FontSize = 8;
            app.MLabel_2.Position = [153 281 25 22];
            app.MLabel_2.Text = 'M';

            % Create utLabel
            app.utLabel = uilabel(app.DatosTab);
            app.utLabel.FontSize = 9;
            app.utLabel.Position = [42 254 18 22];
            app.utLabel.Text = 'ut';

            % Create LibreriadematerialesTab
            app.LibreriadematerialesTab = uitab(app.TabGroup);
            app.LibreriadematerialesTab.Title = 'Libreria de materiales';

            % Create SAE52100steel10CheckBox
            app.SAE52100steel10CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.SAE52100steel10CheckBox.ValueChangedFcn = createCallbackFcn(app,
@SAE52100steel10CheckBoxValueChanged, true);
            app.SAE52100steel10CheckBox.Text = 'SAE52100 steel  [10]';
            app.SAE52100steel10CheckBox.Position = [16 424 135 22];

            % Create Mildsteel11CheckBox
            app.Mildsteel11CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.Mildsteel11CheckBox.ValueChangedFcn = createCallbackFcn(app, @Mildsteel11CheckBoxValueChanged,
true);
            app.Mildsteel11CheckBox.Text = 'Mild steel  [11]';
            app.Mildsteel11CheckBox.Position = [193 617 99 22];

            % Create NiCralloysteel11CheckBox
```

```
            app.NiCralloysteel11CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.NiCralloysteel11CheckBox.ValueChangedFcn = createCallbackFcn(app,
@NiCralloysteel11CheckBoxValueChanged, true);
            app.NiCralloysteel11CheckBox.Text = 'NiCr alloy steel  [11]';
            app.NiCralloysteel11CheckBox.Position = [193 569 128 22];

            % Create SAE413011CheckBox
            app.SAE413011CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.SAE413011CheckBox.ValueChangedFcn = createCallbackFcn(app, @SAE413011CheckBoxValueChanged,
true);
            app.SAE413011CheckBox.Text = 'SAE 4130  [11]';
            app.SAE413011CheckBox.Position = [193 545 102 22];

            % Create T611CheckBox
            app.T611CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.T611CheckBox.ValueChangedFcn = createCallbackFcn(app, @T611CheckBoxValueChanged, true);
            app.T611CheckBox.Text = '7075-T6  [11]';
            app.T611CheckBox.Position = [193 427 92 22];

            % Create ST411CheckBox
            app.ST411CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.ST411CheckBox.ValueChangedFcn = createCallbackFcn(app, @ST411CheckBoxValueChanged, true);
            app.ST411CheckBox.Text = '24S-T4  [11]';
            app.ST411CheckBox.Position = [193 404 87 22];

            % Create ST611CheckBox
            app.ST611CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.ST611CheckBox.ValueChangedFcn = createCallbackFcn(app, @ST611CheckBoxValueChanged, true);
            app.ST611CheckBox.Text = '14S-T6  [11]';
            app.ST611CheckBox.Position = [193 381 87 22];

            % Create ST311CheckBox
            app.ST311CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.ST311CheckBox.ValueChangedFcn = createCallbackFcn(app, @ST311CheckBoxValueChanged, true);
            app.ST311CheckBox.Text = '24S-T3  [11]';
            app.ST311CheckBox.Position = [193 521 87 22];

            % Create ST611CheckBox_2
            app.ST611CheckBox_2 = uicheckbox(app.LibreriadematerialesTab);
            app.ST611CheckBox_2.ValueChangedFcn = createCallbackFcn(app, @ST611CheckBox_2ValueChanged, true);
            app.ST611CheckBox_2.Text = '75S-T6  [11]';
            app.ST611CheckBox_2.Position = [193 497 87 22];

            % Create T611CheckBox_2
            app.T611CheckBox_2 = uicheckbox(app.LibreriadematerialesTab);
            app.T611CheckBox_2.ValueChangedFcn = createCallbackFcn(app, @T611CheckBox_2ValueChanged, true);
            app.T611CheckBox_2.Text = '2014-T6  [11]';
            app.T611CheckBox_2.Position = [193 473 92 22];

            % Create T411CheckBox
            app.T411CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.T411CheckBox.ValueChangedFcn = createCallbackFcn(app, @T411CheckBoxValueChanged, true);
            app.T411CheckBox.Text = '2024-T4  [11]';
            app.T411CheckBox.Position = [193 450 92 22];

            % Create Zn065Zr11CheckBox
            app.Zn065Zr11CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.Zn065Zr11CheckBox.ValueChangedFcn = createCallbackFcn(app, @Zn065Zr11CheckBoxValueChanged,
true);
            app.Zn065Zr11CheckBox.Text = '2.5%Zn 0.65%Zr  [11]';
            app.Zn065Zr11CheckBox.Position = [16 400 138 22];

            % Create Zn066Zr11CheckBox
            app.Zn066Zr11CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.Zn066Zr11CheckBox.ValueChangedFcn = createCallbackFcn(app, @Zn066Zr11CheckBoxValueChanged,
true);
            app.Zn066Zr11CheckBox.Text = '5.6%Zn 0.66%Zr  [11]';
            app.Zn066Zr11CheckBox.Position = [16 376 138 22];

            % Create CasoUniaxialpuroLabel
            app.CasoUniaxialpuroLabel = uilabel(app.LibreriadematerialesTab);
            app.CasoUniaxialpuroLabel.FontWeight = 'bold';
            app.CasoUniaxialpuroLabel.Position = [13 677 114 22];
            app.CasoUniaxialpuroLabel.Text = 'Caso Uniaxial puro';
```

```
            % Create CasoTorsinpuroLabel
            app.CasoTorsinpuroLabel = uilabel(app.LibreriadematerialesTab);
            app.CasoTorsinpuroLabel.FontWeight = 'bold';
            app.CasoTorsinpuroLabel.Position = [13 342 111 22];
            app.CasoTorsinpuroLabel.Text = 'Caso Torsión puro';

            % Create Cnormalizedsteel13CheckBox
            app.Cnormalizedsteel13CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.Cnormalizedsteel13CheckBox.ValueChangedFcn = createCallbackFcn(app,
@Cnormalizedsteel13CheckBoxValueChanged, true);
            app.Cnormalizedsteel13CheckBox.Text = '0.49C normalized steel [13]';
            app.Cnormalizedsteel13CheckBox.Position = [16 280 168 22];

            % Create Csorbiticsteel13CheckBox
            app.Csorbiticsteel13CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.Csorbiticsteel13CheckBox.ValueChangedFcn = createCallbackFcn(app,
@Csorbiticsteel13CheckBoxValueChanged, true);
            app.Csorbiticsteel13CheckBox.Text = '0.49C sorbitic steel  [13]';
            app.Csorbiticsteel13CheckBox.Position = [16 152 151 22];

            % Create Csorbiticsteel17CheckBox
            app.Csorbiticsteel17CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.Csorbiticsteel17CheckBox.ValueChangedFcn = createCallbackFcn(app,
@Csorbiticsteel17CheckBoxValueChanged, true);
            app.Csorbiticsteel17CheckBox.Text = '1.2C sorbitic steel  [17]';
            app.Csorbiticsteel17CheckBox.Position = [193 228 144 22];

            % Create SAE3140HRsteel16CheckBox
            app.SAE3140HRsteel16CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.SAE3140HRsteel16CheckBox.ValueChangedFcn = createCallbackFcn(app,
@SAE3140HRsteel16CheckBoxValueChanged, true);
            app.SAE3140HRsteel16CheckBox.Text = 'SAE 3140 H.R. steel  [16]';
            app.SAE3140HRsteel16CheckBox.Position = [16 202 159 22];

            % Create SAE3140QTsteel16CheckBox
            app.SAE3140QTsteel16CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.SAE3140QTsteel16CheckBox.ValueChangedFcn = createCallbackFcn(app,
@SAE3140QTsteel16CheckBoxValueChanged, true);
            app.SAE3140QTsteel16CheckBox.Text = 'SAE 3140 Q&T steel  [16]';
            app.SAE3140QTsteel16CheckBox.Position = [16 177 159 22];

            % Create STAluminium20CheckBox
            app.STAluminium20CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.STAluminium20CheckBox.ValueChangedFcn = createCallbackFcn(app,
@STAluminium20CheckBoxValueChanged, true);
            app.STAluminium20CheckBox.Text = '14 S-T Aluminium  [20]';
            app.STAluminium20CheckBox.Position = [193 280 143 22];

            % Create BSSS65Asteel28CheckBox
            app.BSSS65Asteel28CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.BSSS65Asteel28CheckBox.ValueChangedFcn = createCallbackFcn(app,
@BSSS65Asteel28CheckBoxValueChanged, true);
            app.BSSS65Asteel28CheckBox.Text = 'BSS S65A steel  [28]';
            app.BSSS65Asteel28CheckBox.Position = [193 27 134 22];

            % Create Aluminium76ST6121CheckBox
            app.Aluminium76ST6121CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.Aluminium76ST6121CheckBox.ValueChangedFcn = createCallbackFcn(app,
@Aluminium76ST6121CheckBoxValueChanged, true);
            app.Aluminium76ST6121CheckBox.Text = 'Aluminium 76ST61 [21]';
            app.Aluminium76ST6121CheckBox.Position = [193 254 147 22];

            % Create En25TNICrMosteel14CheckBox
            app.En25TNICrMosteel14CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.En25TNICrMosteel14CheckBox.ValueChangedFcn = createCallbackFcn(app,
@En25TNICrMosteel14CheckBoxValueChanged, true);
            app.En25TNICrMosteel14CheckBox.Text = 'En 25T NICrMo steel  [14]';
            app.En25TNICrMosteel14CheckBox.Position = [16 254 160 22];

            % Create Csteel26CheckBox
            app.Csteel26CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.Csteel26CheckBox.ValueChangedFcn = createCallbackFcn(app, @Csteel26CheckBoxValueChanged,
true);
```

```
            app.Csteel26CheckBox.Text = '0.1C steel  [26]';
            app.Csteel26CheckBox.Position = [193 102 103 22];

            % Create CrMo4steel23CheckBox
            app.CrMo4steel23CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.CrMo4steel23CheckBox.ValueChangedFcn = createCallbackFcn(app,
@CrMo4steel23CheckBoxValueChanged, true);
            app.CrMo4steel23CheckBox.Text = '34CrMo4 steel  [23]';
            app.CrMo4steel23CheckBox.Position = [193 52 127 22];

            % Create St35steel27CheckBox
            app.St35steel27CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.St35steel27CheckBox.ValueChangedFcn = createCallbackFcn(app, @St35steel27CheckBoxValueChanged,
true);
            app.St35steel27CheckBox.Text = 'St 35 steel  [27]';
            app.St35steel27CheckBox.Position = [193 77 105 22];

            % Create CrMo4Vsteel22CheckBox
            app.CrMo4Vsteel22CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.CrMo4Vsteel22CheckBox.ValueChangedFcn = createCallbackFcn(app,
@CrMo4Vsteel22CheckBoxValueChanged, true);
            app.CrMo4Vsteel22CheckBox.Text = '42CrMo4V steel  [22]';
            app.CrMo4Vsteel22CheckBox.Position = [193 202 135 22];

            % Create Csteel2CheckBox
            app.Csteel2CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.Csteel2CheckBox.ValueChangedFcn = createCallbackFcn(app, @Csteel2CheckBoxValueChanged, true);
            app.Csteel2CheckBox.Text = '0.1%C steel  [2]';
            app.Csteel2CheckBox.Position = [16 617 107 22];

            % Create Carbonsteel3CheckBox
            app.Carbonsteel3CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.Carbonsteel3CheckBox.ValueChangedFcn = createCallbackFcn(app,
@Carbonsteel3CheckBoxValueChanged, true);
            app.Carbonsteel3CheckBox.Text = '0.13 Carbon steel  [3]';
            app.Carbonsteel3CheckBox.Position = [16 592 137 22];

            % Create Carbonsteel4CheckBox
            app.Carbonsteel4CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.Carbonsteel4CheckBox.ValueChangedFcn = createCallbackFcn(app,
@Carbonsteel4CheckBoxValueChanged, true);
            app.Carbonsteel4CheckBox.Text = '0.29 Carbon steel  [4]';
            app.Carbonsteel4CheckBox.Position = [16 568 137 22];

            % Create AISI4130steel5CheckBox
            app.AISI4130steel5CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.AISI4130steel5CheckBox.ValueChangedFcn = createCallbackFcn(app,
@AISI4130steel5CheckBoxValueChanged, true);
            app.AISI4130steel5CheckBox.Text = 'AISI 4130 steel  [5]';
            app.AISI4130steel5CheckBox.Position = [16 544 123 22];

            % Create AISI4340steel6CheckBox
            app.AISI4340steel6CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.AISI4340steel6CheckBox.ValueChangedFcn = createCallbackFcn(app,
@AISI4340steel6CheckBoxValueChanged, true);
            app.AISI4340steel6CheckBox.Text = 'AISI 4340 steel  [6]';
            app.AISI4340steel6CheckBox.Position = [16 520 123 22];

            % Create En25steel7CheckBox
            app.En25steel7CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.En25steel7CheckBox.ValueChangedFcn = createCallbackFcn(app, @En25steel7CheckBoxValueChanged,
true);
            app.En25steel7CheckBox.Text = 'En 25 steel  [7]';
            app.En25steel7CheckBox.Position = [16 496 102 22];

            % Create NiCrMo8steel8CheckBox
            app.NiCrMo8steel8CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.NiCrMo8steel8CheckBox.ValueChangedFcn = createCallbackFcn(app,
@NiCrMo8steel8CheckBoxValueChanged, true);
            app.NiCrMo8steel8CheckBox.Text = '30NiCrMo8 steel  [8]';
            app.NiCrMo8steel8CheckBox.Position = [16 472 131 22];

            % Create CrMo4steel9CheckBox
            app.CrMo4steel9CheckBox = uicheckbox(app.LibreriadematerialesTab);
```

```
            app.CrMo4steel9CheckBox.ValueChangedFcn = createCallbackFcn(app, @CrMo4steel9CheckBoxValueChanged,
true);
            app.CrMo4steel9CheckBox.Text = '25CrMo4 steel  [9]';
            app.CrMo4steel9CheckBox.Position = [16 448 120 22];

            % Create T611CheckBox_3
            app.T611CheckBox_3 = uicheckbox(app.LibreriadematerialesTab);
            app.T611CheckBox_3.ValueChangedFcn = createCallbackFcn(app, @T611CheckBox_3ValueChanged, true);
            app.T611CheckBox_3.Text = '6061-T6  [11]';
            app.T611CheckBox_3.Position = [193 593 92 22];

            % Create Cr4steel29CheckBox
            app.Cr4steel29CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.Cr4steel29CheckBox.ValueChangedFcn = createCallbackFcn(app, @Cr4steel29CheckBoxValueChanged,
true);
            app.Cr4steel29CheckBox.Text = '34Cr4 steel  [29]';
            app.Cr4steel29CheckBox.Position = [193 2 110 22];

            % Create JISSCM440steel24CheckBox
            app.JISSCM440steel24CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.JISSCM440steel24CheckBox.ValueChangedFcn = createCallbackFcn(app,
@JISSCM440steel24CheckBoxValueChanged, true);
            app.JISSCM440steel24CheckBox.Text = 'JIS SCM440 steel [24]';
            app.JISSCM440steel24CheckBox.Position = [193 152 142 22];

            % Create MnCr5steel25CheckBox
            app.MnCr5steel25CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.MnCr5steel25CheckBox.ValueChangedFcn = createCallbackFcn(app,
@MnCr5steel25CheckBoxValueChanged, true);
            app.MnCr5steel25CheckBox.Text = '20MnCr5 steel  [25]';
            app.MnCr5steel25CheckBox.Position = [193 127 127 22];

            % Create NiCrMo3steel19CheckBox
            app.NiCrMo3steel19CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.NiCrMo3steel19CheckBox.ValueChangedFcn = createCallbackFcn(app,
@NiCrMo3steel19CheckBoxValueChanged, true);
            app.NiCrMo3steel19CheckBox.Text = '39NiCrMo3 steel  [19]';
            app.NiCrMo3steel19CheckBox.Position = [193 306 138 22];

            % Create VDSiCrspringsteel15CheckBox
            app.VDSiCrspringsteel15CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.VDSiCrspringsteel15CheckBox.ValueChangedFcn = createCallbackFcn(app,
@VDSiCrspringsteel15CheckBoxValueChanged, true);
            app.VDSiCrspringsteel15CheckBox.Text = 'VDSiCr spring steel  [15]';
            app.VDSiCrspringsteel15CheckBox.Position = [16 228 153 22];

            % Create CrNiMo6steel12CheckBox
            app.CrNiMo6steel12CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.CrNiMo6steel12CheckBox.ValueChangedFcn = createCallbackFcn(app,
@CrNiMo6steel12CheckBoxValueChanged, true);
            app.CrNiMo6steel12CheckBox.Text = '34CrNiMo6 steel  [12]';
            app.CrNiMo6steel12CheckBox.Position = [16 308 138 22];

            % Create CNisteeltreatA13CheckBox
            app.CNisteeltreatA13CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.CNisteeltreatA13CheckBox.ValueChangedFcn = createCallbackFcn(app,
@CNisteeltreatA13CheckBoxValueChanged, true);
            app.CNisteeltreatA13CheckBox.Text = '3.5C Ni steel treat.A  [13]';
            app.CNisteeltreatA13CheckBox.Position = [16 127 155 22];

            % Create CNisteeltreatB13CheckBox
            app.CNisteeltreatB13CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.CNisteeltreatB13CheckBox.ValueChangedFcn = createCallbackFcn(app,
@CNisteeltreatB13CheckBoxValueChanged, true);
            app.CNisteeltreatB13CheckBox.Text = '3.5C Ni steel treat.B  [13]';
            app.CNisteeltreatB13CheckBox.Position = [16 102 156 22];

            % Create CNisteeltreatD13CheckBox
            app.CNisteeltreatD13CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.CNisteeltreatD13CheckBox.ValueChangedFcn = createCallbackFcn(app,
@CNisteeltreatD13CheckBoxValueChanged, true);
            app.CNisteeltreatD13CheckBox.Text = '3.5C Ni steel treat.D  [13]';
            app.CNisteeltreatD13CheckBox.Position = [16 77 157 22];
```

```
            % Create CrNisteeltreatD17CheckBox
            app.CrNisteeltreatD17CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.CrNisteeltreatD17CheckBox.ValueChangedFcn = createCallbackFcn(app,
@CrNisteeltreatD17CheckBoxValueChanged, true);
            app.CrNisteeltreatD17CheckBox.Text = 'CrNi steel treat.D  [17]';
            app.CrNisteeltreatD17CheckBox.Position = [16 52 141 22];

            % Create CK35steel23CheckBox
            app.CK35steel23CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.CK35steel23CheckBox.ValueChangedFcn = createCallbackFcn(app, @CK35steel23CheckBoxValueChanged,
true);
            app.CK35steel23CheckBox.Text = 'CK35 steel  [23]';
            app.CK35steel23CheckBox.Position = [193 177 107 22];

            % Create CrMo4steel18CheckBox
            app.CrMo4steel18CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.CrMo4steel18CheckBox.ValueChangedFcn = createCallbackFcn(app,
@CrMo4steel18CheckBoxValueChanged, true);
            app.CrMo4steel18CheckBox.Text = '42CrMo4 steel  [18]';
            app.CrMo4steel18CheckBox.Position = [16 2 127 22];

            % Create Selectall_uniaxialCheckBox
            app.Selectall_uniaxialCheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.Selectall_uniaxialCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Selectall_uniaxialCheckBoxValueChanged2, true);
            app.Selectall_uniaxialCheckBox.Text = 'Select all';
            app.Selectall_uniaxialCheckBox.Position = [129 685 71 22];

            % Create Unselectall_uniaxialCheckBox
            app.Unselectall_uniaxialCheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.Unselectall_uniaxialCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Unselectall_uniaxialCheckBoxValueChanged2, true);
            app.Unselectall_uniaxialCheckBox.Text = 'Unselect all';
            app.Unselectall_uniaxialCheckBox.Position = [129 667 84 22];

            % Create Selectall_torsionCheckBox
            app.Selectall_torsionCheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.Selectall_torsionCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Selectall_torsionCheckBoxValueChanged, true);
            app.Selectall_torsionCheckBox.Text = 'Select all';
            app.Selectall_torsionCheckBox.Position = [129 350 71 22];

            % Create Unselectall_torsionCheckBox
            app.Unselectall_torsionCheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.Unselectall_torsionCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Unselectall_torsionCheckBoxValueChanged, true);
            app.Unselectall_torsionCheckBox.Text = 'Unselect all';
            app.Unselectall_torsionCheckBox.Position = [129 332 84 22];

            % Create Image5
            app.Image5 = uiimage(app.LibreriadematerialesTab);
            app.Image5.Position = [119 616 25 23];
            app.Image5.ImageSource = 'c_st01.PNG';

            % Create Image5_2
            app.Image5_2 = uiimage(app.LibreriadematerialesTab);
            app.Image5_2.Position = [148 590 25 23];
            app.Image5_2.ImageSource = 'C_st013.PNG';

            % Create Image5_3
            app.Image5_3 = uiimage(app.LibreriadematerialesTab);
            app.Image5_3.Position = [148 565 25 23];
            app.Image5_3.ImageSource = 'carb_steel_029.PNG';

            % Create Image5_4
            app.Image5_4 = uiimage(app.LibreriadematerialesTab);
            app.Image5_4.Position = [135 544 25 23];
            app.Image5_4.ImageSource = 'AISI_4130.PNG';

            % Create Image5_5
            app.Image5_5 = uiimage(app.LibreriadematerialesTab);
            app.Image5_5.Position = [135 519 25 23];
            app.Image5_5.ImageSource = 'AISI_4340.PNG';
```

```
% Create Image5_6
app.Image5_6 = uiimage(app.LibreriadematerialesTab);
app.Image5_6.Position = [114 496 25 23];
app.Image5_6.ImageSource = 'En_25.PNG';

% Create Image5_7
app.Image5_7 = uiimage(app.LibreriadematerialesTab);
app.Image5_7.Position = [143 471 25 23];
app.Image5_7.ImageSource = 'NiCrMo8.PNG';

% Create Image5_8
app.Image5_8 = uiimage(app.LibreriadematerialesTab);
app.Image5_8.Position = [133 447 25 23];
app.Image5_8.ImageSource = 'CrMo4.PNG';

% Create Image5_9
app.Image5_9 = uiimage(app.LibreriadematerialesTab);
app.Image5_9.Position = [149 422 25 23];
app.Image5_9.ImageSource = 'SAE_52100.PNG';

% Create Image5_10
app.Image5_10 = uiimage(app.LibreriadematerialesTab);
app.Image5_10.Position = [148 400 25 23];
app.Image5_10.ImageSource = 'zn_25_zr65.PNG';

% Create Image5_11
app.Image5_11 = uiimage(app.LibreriadematerialesTab);
app.Image5_11.Position = [150 373 25 25];
app.Image5_11.ImageSource = 'zn_56_zr66.PNG';

% Create Image5_12
app.Image5_12 = uiimage(app.LibreriadematerialesTab);
app.Image5_12.Position = [288 616 25 23];
app.Image5_12.ImageSource = 'Mild_Steel.PNG';

% Create Image5_13
app.Image5_13 = uiimage(app.LibreriadematerialesTab);
app.Image5_13.Position = [286 596 21 14];
app.Image5_13.ImageSource = 'a_6061_T6.PNG';

% Create Image5_14
app.Image5_14 = uiimage(app.LibreriadematerialesTab);
app.Image5_14.Position = [277 381 25 19];
app.Image5_14.ImageSource = 's_14_T6.PNG';

% Create Image5_15
app.Image5_15 = uiimage(app.LibreriadematerialesTab);
app.Image5_15.Position = [278 403 25 23];
app.Image5_15.ImageSource = 's_24_T4.PNG';

% Create Image5_16
app.Image5_16 = uiimage(app.LibreriadematerialesTab);
app.Image5_16.Position = [281 429 25 19];
app.Image5_16.ImageSource = 'a_7075_T6.PNG';

% Create Image5_17
app.Image5_17 = uiimage(app.LibreriadematerialesTab);
app.Image5_17.Position = [284 448 25 23];
app.Image5_17.ImageSource = 'a_2024_T6.PNG';

% Create Image5_18
app.Image5_18 = uiimage(app.LibreriadematerialesTab);
app.Image5_18.Position = [283 471 24 19];
app.Image5_18.ImageSource = 'a_2014_T6.PNG';

% Create Image5_19
app.Image5_19 = uiimage(app.LibreriadematerialesTab);
app.Image5_19.Position = [279 496 25 20];
app.Image5_19.ImageSource = 'S75_T6.PNG';

% Create Image5_20
app.Image5_20 = uiimage(app.LibreriadematerialesTab);
app.Image5_20.Position = [278 517 25 25];
app.Image5_20.ImageSource = 'S24_T3.PNG';
```

```
% Create Image5_21
app.Image5_21 = uiimage(app.LibreriademterialesTab);
app.Image5_21.Position = [292 544 25 21];
app.Image5_21.ImageSource = 'SAE_4130.PNG';

% Create Image5_22
app.Image5_22 = uiimage(app.LibreriademterialesTab);
app.Image5_22.Position = [317 564 26 26];
app.Image5_22.ImageSource = 'NiCr_alloy.PNG';

% Create Image5_23
app.Image5_23 = uiimage(app.LibreriademterialesTab);
app.Image5_23.Position = [169 25 25 23];
app.Image5_23.ImageSource = 'C_st013.PNG';

% Create Image5_24
app.Image5_24 = uiimage(app.LibreriademterialesTab);
app.Image5_24.Position = [179 281 14 19];
app.Image5_24.ImageSource = 'C_49norm.PNG';

% Create Image5_25
app.Image5_25 = uiimage(app.LibreriademterialesTab);
app.Image5_25.Position = [171 252 20 22];
app.Image5_25.ImageSource = 'En_25T.PNG';

% Create Image5_26
app.Image5_26 = uiimage(app.LibreriademterialesTab);
app.Image5_26.Position = [164 228 21 19];
app.Image5_26.ImageSource = 'VDSiCr.PNG';

% Create Image5_27
app.Image5_27 = uiimage(app.LibreriademterialesTab);
app.Image5_27.Position = [169 202 17 22];
app.Image5_27.ImageSource = 'sae_3140_HR.PNG';

% Create Image5_28
app.Image5_28 = uiimage(app.LibreriademterialesTab);
app.Image5_28.Position = [170 178 19 19];
app.Image5_28.ImageSource = 'sae_3140_QT.PNG';

% Create Image5_29
app.Image5_29 = uiimage(app.LibreriademterialesTab);
app.Image5_29.Position = [161 151 25 23];
app.Image5_29.ImageSource = 'C_49sorb.PNG';

% Create Image5_30
app.Image5_30 = uiimage(app.LibreriademterialesTab);
app.Image5_30.Position = [167 125 25 23];
app.Image5_30.ImageSource = 'C_35Ni_A.PNG';

% Create Image5_31
app.Image5_31 = uiimage(app.LibreriademterialesTab);
app.Image5_31.Position = [166 101 25 23];
app.Image5_31.ImageSource = 'C_35Ni_B.PNG';

% Create Image5_32
app.Image5_32 = uiimage(app.LibreriademterialesTab);
app.Image5_32.Position = [165 76 25 23];
app.Image5_32.ImageSource = 'C_35Ni_D.PNG';

% Create Image5_33
app.Image5_33 = uiimage(app.LibreriademterialesTab);
app.Image5_33.Position = [150 54 20 18];
app.Image5_33.ImageSource = 'CrNi_D.PNG';

% Create Image5_34
app.Image5_34 = uiimage(app.LibreriademterialesTab);
app.Image5_34.Position = [296 177 20 20];
app.Image5_34.ImageSource = 'CK_35.PNG';

% Create Image5_35
app.Image5_35 = uiimage(app.LibreriademterialesTab);
app.Image5_35.Position = [136 4 21 15];
```

```
app.Image5_35.ImageSource = 'Cr_Mo_42.PNG';

% Create Image5_36
app.Image5_36 = uiimage(app.LibreriadematerialesTab);
app.Image5_36.Position = [326 305 20 22];
app.Image5_36.ImageSource = 'NiCrMo3.PNG';

% Create Image5_37
app.Image5_37 = uiimage(app.LibreriadematerialesTab);
app.Image5_37.Position = [331 281 17 20];
app.Image5_37.ImageSource = 'st_14_Al.PNG';

% Create Image5_38
app.Image5_38 = uiimage(app.LibreriadematerialesTab);
app.Image5_38.Position = [335 255 17 18];
app.Image5_38.ImageSource = 'al_76.PNG';

% Create Image5_39
app.Image5_39 = uiimage(app.LibreriadematerialesTab);
app.Image5_39.Position = [332 229 19 19];
app.Image5_39.ImageSource = 'C_12sorb.PNG';

% Create Image5_40
app.Image5_40 = uiimage(app.LibreriadematerialesTab);
app.Image5_40.Position = [323 201 22 22];
app.Image5_40.ImageSource = 'CrMo4V.PNG';

% Create Image5_41
app.Image5_41 = uiimage(app.LibreriadematerialesTab);
app.Image5_41.Position = [147 308 25 23];
app.Image5_41.ImageSource = 'CrNiMo6_steel.PNG';

% Create Image5_42
app.Image5_42 = uiimage(app.LibreriadematerialesTab);
app.Image5_42.Position = [329 154 22 16];
app.Image5_42.ImageSource = 'JIS.PNG';

% Create Image5_43
app.Image5_43 = uiimage(app.LibreriadematerialesTab);
app.Image5_43.Position = [315 126 19 21];
app.Image5_43.ImageSource = 'MnCr5.PNG';

% Create Image5_44
app.Image5_44 = uiimage(app.LibreriadematerialesTab);
app.Image5_44.Position = [288 101 25 23];
app.Image5_44.ImageSource = 'c_01.PNG';

% Create Image5_45
app.Image5_45 = uiimage(app.LibreriadematerialesTab);
app.Image5_45.Position = [293 78 21 20];
app.Image5_45.ImageSource = 'ST_35.PNG';

% Create Image5_46
app.Image5_46 = uiimage(app.LibreriadematerialesTab);
app.Image5_46.Position = [312 51 25 23];
app.Image5_46.ImageSource = 'CrMo4_steel.PNG';

% Create Image5_47
app.Image5_47 = uiimage(app.LibreriadematerialesTab);
app.Image5_47.Position = [320 30 21 14];
app.Image5_47.ImageSource = 'bss_s65.PNG';

% Create Image5_48
app.Image5_48 = uiimage(app.LibreriadematerialesTab);
app.Image5_48.Position = [299 2 22 22];
app.Image5_48.ImageSource = 'Cr4_steel.PNG';

% Create ReferenciasLabel
app.ReferenciasLabel = uilabel(app.LibreriadematerialesTab);
app.ReferenciasLabel.Position = [16 706 70 22];
app.ReferenciasLabel.Text = 'Referencias';

% Create Button
app.Button = uibutton(app.LibreriadematerialesTab, 'push');
```

```
            app.Button.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed2, true);
            app.Button.Icon = 'uialert_info.png';
            app.Button.Position = [84 707 20 21];
            app.Button.Text = '';

            % Create Cnormalizedsteel13CheckBox_2
            app.Cnormalizedsteel13CheckBox_2 = uicheckbox(app.LibreriadematerialesTab);
            app.Cnormalizedsteel13CheckBox_2.ValueChangedFcn = createCallbackFcn(app,
@Cnormalizedsteel13CheckBox_2ValueChanged, true);
            app.Cnormalizedsteel13CheckBox_2.Text = '1.2C normalized steel [13]';
            app.Cnormalizedsteel13CheckBox_2.Position = [16 27 162 22];

            % Create CrNiMo6steel1CheckBox
            app.CrNiMo6steel1CheckBox = uicheckbox(app.LibreriadematerialesTab);
            app.CrNiMo6steel1CheckBox.ValueChangedFcn = createCallbackFcn(app,
@CrNiMo6steel1CheckBoxValueChanged, true);
            app.CrNiMo6steel1CheckBox.Text = '34CrNiMo6 steel  [1]';
            app.CrNiMo6steel1CheckBox.Position = [16 642 131 22];

            % Create Image5_49
            app.Image5_49 = uiimage(app.LibreriadematerialesTab);
            app.Image5_49.Position = [145 639 25 25];
            app.Image5_49.ImageSource = 'CrNiMo6_uniaxial.PNG';

            % Create AadirmaterialesTab
            app.AadirmaterialesTab = uitab(app.TabGroup);
            app.AadirmaterialesTab.Title = 'Añadir materiales';

            % Create Material9_uniCheckBox
            app.Material9_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material9_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material9_uniCheckBoxValueChanged, true);
            app.Material9_uniCheckBox.Visible = 'off';
            app.Material9_uniCheckBox.Text = 'Material 9_uni';
            app.Material9_uniCheckBox.Position = [16 368 97 22];

            % Create Material12_uniCheckBox
            app.Material12_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material12_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material12_uniCheckBoxValueChanged, true);
            app.Material12_uniCheckBox.Visible = 'off';
            app.Material12_uniCheckBox.Text = 'Material 12_uni';
            app.Material12_uniCheckBox.Position = [181 561 104 22];

            % Create Material14_uniCheckBox
            app.Material14_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material14_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material14_uniCheckBoxValueChanged, true);
            app.Material14_uniCheckBox.Visible = 'off';
            app.Material14_uniCheckBox.Text = 'Material 14_uni';
            app.Material14_uniCheckBox.Position = [181 512 104 22];

            % Create Material15_uniCheckBox
            app.Material15_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material15_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material15_uniCheckBoxValueChanged, true);
            app.Material15_uniCheckBox.Visible = 'off';
            app.Material15_uniCheckBox.Text = 'Material 15_uni';
            app.Material15_uniCheckBox.Position = [181 488 104 22];

            % Create Material20_uniCheckBox
            app.Material20_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material20_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material20_uniCheckBoxValueChanged, true);
            app.Material20_uniCheckBox.Visible = 'off';
            app.Material20_uniCheckBox.Text = 'Material 20_uni';
            app.Material20_uniCheckBox.Position = [181 368 104 22];

            % Create Material21_uniCheckBox
            app.Material21_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material21_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material21_uniCheckBoxValueChanged, true);
            app.Material21_uniCheckBox.Visible = 'off';
            app.Material21_uniCheckBox.Text = 'Material 21_uni';
```

```
            app.Material21_uniCheckBox.Position = [181 344 104 22];

            % Create Material22_uniCheckBox
            app.Material22_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material22_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material22_uniCheckBoxValueChanged, true);
            app.Material22_uniCheckBox.Visible = 'off';
            app.Material22_uniCheckBox.Text = 'Material 22_uni';
            app.Material22_uniCheckBox.Position = [181 320 104 22];

            % Create Material16_uniCheckBox
            app.Material16_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material16_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material16_uniCheckBoxValueChanged, true);
            app.Material16_uniCheckBox.Visible = 'off';
            app.Material16_uniCheckBox.Text = 'Material 16_uni';
            app.Material16_uniCheckBox.Position = [181 464 104 22];

            % Create Material17_uniCheckBox
            app.Material17_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material17_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material17_uniCheckBoxValueChanged, true);
            app.Material17_uniCheckBox.Visible = 'off';
            app.Material17_uniCheckBox.Text = 'Material 17_uni';
            app.Material17_uniCheckBox.Position = [181 440 104 22];

            % Create Material18_uniCheckBox
            app.Material18_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material18_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material18_uniCheckBoxValueChanged, true);
            app.Material18_uniCheckBox.Visible = 'off';
            app.Material18_uniCheckBox.Text = 'Material 18_uni';
            app.Material18_uniCheckBox.Position = [181 416 104 22];

            % Create Material19_uniCheckBox
            app.Material19_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material19_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material19_uniCheckBoxValueChanged, true);
            app.Material19_uniCheckBox.Visible = 'off';
            app.Material19_uniCheckBox.Text = 'Material 19_uni';
            app.Material19_uniCheckBox.Position = [181 392 104 22];

            % Create Material10_uniCheckBox
            app.Material10_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material10_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material10_uniCheckBoxValueChanged, true);
            app.Material10_uniCheckBox.Visible = 'off';
            app.Material10_uniCheckBox.Text = 'Material 10_uni';
            app.Material10_uniCheckBox.Position = [16 344 104 22];

            % Create Material11_uniCheckBox
            app.Material11_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material11_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material11_uniCheckBoxValueChanged, true);
            app.Material11_uniCheckBox.Visible = 'off';
            app.Material11_uniCheckBox.Text = 'Material 11_uni';
            app.Material11_uniCheckBox.Position = [16 320 103 22];

            % Create CasoUniaxialpuroLabel_2
            app.CasoUniaxialpuroLabel_2 = uilabel(app.AadirmaterialesTab);
            app.CasoUniaxialpuroLabel_2.FontWeight = 'bold';
            app.CasoUniaxialpuroLabel_2.Position = [14 600 114 22];
            app.CasoUniaxialpuroLabel_2.Text = 'Caso Uniaxial puro';

            % Create CasoTorsinpuroLabel_2
            app.CasoTorsinpuroLabel_2 = uilabel(app.AadirmaterialesTab);
            app.CasoTorsinpuroLabel_2.FontWeight = 'bold';
            app.CasoTorsinpuroLabel_2.Position = [14 286 111 22];
            app.CasoTorsinpuroLabel_2.Text = 'Caso Torsión puro';

            % Create Material1_uniCheckBox
            app.Material1_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material1_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material1_uniCheckBoxValueChanged, true);
```

```
            app.Material1_uniCheckBox.Visible = 'off';
            app.Material1_uniCheckBox.Text = 'Material 1_uni';
            app.Material1_uniCheckBox.Position = [16 561 97 22];

            % Create Material2_uniCheckBox
            app.Material2_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material2_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material2_uniCheckBoxValueChanged, true);
            app.Material2_uniCheckBox.Visible = 'off';
            app.Material2_uniCheckBox.Text = 'Material 2_uni';
            app.Material2_uniCheckBox.Position = [16 536 97 22];

            % Create Material3_uniCheckBox
            app.Material3_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material3_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material3_uniCheckBoxValueChanged, true);
            app.Material3_uniCheckBox.Visible = 'off';
            app.Material3_uniCheckBox.Text = 'Material 3_uni';
            app.Material3_uniCheckBox.Position = [16 512 97 22];

            % Create Material4_uniCheckBox
            app.Material4_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material4_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material4_uniCheckBoxValueChanged, true);
            app.Material4_uniCheckBox.Visible = 'off';
            app.Material4_uniCheckBox.Text = 'Material 4_uni';
            app.Material4_uniCheckBox.Position = [16 488 97 22];

            % Create Material5_uniCheckBox
            app.Material5_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material5_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material5_uniCheckBoxValueChanged, true);
            app.Material5_uniCheckBox.Visible = 'off';
            app.Material5_uniCheckBox.Text = 'Material 5_uni';
            app.Material5_uniCheckBox.Position = [16 464 97 22];

            % Create Material6_uniCheckBox
            app.Material6_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material6_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material6_uniCheckBoxValueChanged, true);
            app.Material6_uniCheckBox.Visible = 'off';
            app.Material6_uniCheckBox.Text = 'Material 6_uni';
            app.Material6_uniCheckBox.Position = [16 440 97 22];

            % Create Material7_uniCheckBox
            app.Material7_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material7_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material7_uniCheckBoxValueChanged, true);
            app.Material7_uniCheckBox.Visible = 'off';
            app.Material7_uniCheckBox.Text = 'Material 7_uni';
            app.Material7_uniCheckBox.Position = [16 416 97 22];

            % Create Material8_uniCheckBox
            app.Material8_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material8_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material8_uniCheckBoxValueChanged, true);
            app.Material8_uniCheckBox.Visible = 'off';
            app.Material8_uniCheckBox.Text = 'Material 8_uni';
            app.Material8_uniCheckBox.Position = [16 392 97 22];

            % Create Material13_uniCheckBox
            app.Material13_uniCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material13_uniCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material13_uniCheckBoxValueChanged, true);
            app.Material13_uniCheckBox.Visible = 'off';
            app.Material13_uniCheckBox.Text = 'Material 13_uni';
            app.Material13_uniCheckBox.Position = [181 536 104 22];

            % Create Sel_all_exc_uni
            app.Sel_all_exc_uni = uicheckbox(app.AadirmaterialesTab);
            app.Sel_all_exc_uni.ValueChangedFcn = createCallbackFcn(app, @Sel_all_exc_uniValueChanged, true);
            app.Sel_all_exc_uni.Visible = 'off';
            app.Sel_all_exc_uni.Text = 'Select all';
            app.Sel_all_exc_uni.Position = [131 610 71 22];
```

```
            % Create Unsel_all_exc_uni
            app.Unsel_all_exc_uni = uicheckbox(app.AadirmaterialesTab);
            app.Unsel_all_exc_uni.ValueChangedFcn = createCallbackFcn(app, @Unsel_all_exc_uniValueChanged,
true);
            app.Unsel_all_exc_uni.Visible = 'off';
            app.Unsel_all_exc_uni.Text = 'Unselect all';
            app.Unsel_all_exc_uni.Position = [131 588 84 22];

            % Create Sel_all_exc_tor
            app.Sel_all_exc_tor = uicheckbox(app.AadirmaterialesTab);
            app.Sel_all_exc_tor.ValueChangedFcn = createCallbackFcn(app, @Sel_all_exc_torValueChanged, true);
            app.Sel_all_exc_tor.Visible = 'off';
            app.Sel_all_exc_tor.Text = 'Select all';
            app.Sel_all_exc_tor.Position = [131 296 71 22];

            % Create Unsel_all_exc_tor
            app.Unsel_all_exc_tor = uicheckbox(app.AadirmaterialesTab);
            app.Unsel_all_exc_tor.ValueChangedFcn = createCallbackFcn(app, @Unsel_all_exc_torValueChanged,
true);
            app.Unsel_all_exc_tor.Visible = 'off';
            app.Unsel_all_exc_tor.Text = 'Unselect all';
            app.Unsel_all_exc_tor.Position = [131 274 84 22];

            % Create im_mat_1_uni
            app.im_mat_1_uni = uiimage(app.AadirmaterialesTab);
            app.im_mat_1_uni.Visible = 'off';
            app.im_mat_1_uni.Position = [125 560 25 23];
            app.im_mat_1_uni.ImageSource = 'Mat_1.PNG';

            % Create im_mat_2_uni
            app.im_mat_2_uni = uiimage(app.AadirmaterialesTab);
            app.im_mat_2_uni.Visible = 'off';
            app.im_mat_2_uni.Position = [125 534 25 23];
            app.im_mat_2_uni.ImageSource = 'Mat_2.PNG';

            % Create im_mat_3_uni
            app.im_mat_3_uni = uiimage(app.AadirmaterialesTab);
            app.im_mat_3_uni.Visible = 'off';
            app.im_mat_3_uni.Position = [125 509 25 23];
            app.im_mat_3_uni.ImageSource = 'Mat_3.PNG';

            % Create im_mat_4_uni
            app.im_mat_4_uni = uiimage(app.AadirmaterialesTab);
            app.im_mat_4_uni.Visible = 'off';
            app.im_mat_4_uni.Position = [125 488 25 23];
            app.im_mat_4_uni.ImageSource = 'Mat_4.PNG';

            % Create im_mat_5_uni
            app.im_mat_5_uni = uiimage(app.AadirmaterialesTab);
            app.im_mat_5_uni.Visible = 'off';
            app.im_mat_5_uni.Position = [125 463 25 23];
            app.im_mat_5_uni.ImageSource = 'Mat_5.PNG';

            % Create im_mat_6_uni
            app.im_mat_6_uni = uiimage(app.AadirmaterialesTab);
            app.im_mat_6_uni.Visible = 'off';
            app.im_mat_6_uni.Position = [125 440 25 23];
            app.im_mat_6_uni.ImageSource = 'Mat_6.PNG';

            % Create im_mat_7_uni
            app.im_mat_7_uni = uiimage(app.AadirmaterialesTab);
            app.im_mat_7_uni.Visible = 'off';
            app.im_mat_7_uni.Position = [125 415 25 23];
            app.im_mat_7_uni.ImageSource = 'Mat_7.PNG';

            % Create im_mat_8_uni
            app.im_mat_8_uni = uiimage(app.AadirmaterialesTab);
            app.im_mat_8_uni.Visible = 'off';
            app.im_mat_8_uni.Position = [125 391 25 23];
            app.im_mat_8_uni.ImageSource = 'Mat_8.PNG';

            % Create im_mat_9_uni
            app.im_mat_9_uni = uiimage(app.AadirmaterialesTab);
```

```
app.im_mat_9_uni.Visible = 'off';
app.im_mat_9_uni.Position = [125 366 25 23];
app.im_mat_9_uni.ImageSource = 'Mat_9.PNG';

% Create im_mat_10_uni
app.im_mat_10_uni = uiimage(app.AadirmaterialesTab);
app.im_mat_10_uni.Visible = 'off';
app.im_mat_10_uni.Position = [125 344 25 23];
app.im_mat_10_uni.ImageSource = 'Mat_10.PNG';

% Create im_mat_11_uni
app.im_mat_11_uni = uiimage(app.AadirmaterialesTab);
app.im_mat_11_uni.Visible = 'off';
app.im_mat_11_uni.Position = [125 319 25 25];
app.im_mat_11_uni.ImageSource = 'Mat_11.PNG';

% Create im_mat_12_uni
app.im_mat_12_uni = uiimage(app.AadirmaterialesTab);
app.im_mat_12_uni.Visible = 'off';
app.im_mat_12_uni.Position = [291 561 25 23];
app.im_mat_12_uni.ImageSource = 'Mat_12.PNG';

% Create im_mat_13_uni
app.im_mat_13_uni = uiimage(app.AadirmaterialesTab);
app.im_mat_13_uni.Visible = 'off';
app.im_mat_13_uni.Position = [291 535 25 24];
app.im_mat_13_uni.ImageSource = 'Mat_13.PNG';

% Create im_mat_22_uni
app.im_mat_22_uni = uiimage(app.AadirmaterialesTab);
app.im_mat_22_uni.Visible = 'off';
app.im_mat_22_uni.Position = [291 322 26 22];
app.im_mat_22_uni.ImageSource = 'Mat_22.PNG';

% Create im_mat_21_uni
app.im_mat_21_uni = uiimage(app.AadirmaterialesTab);
app.im_mat_21_uni.Visible = 'off';
app.im_mat_21_uni.Position = [291 344 25 23];
app.im_mat_21_uni.ImageSource = 'mat_21.PNG';

% Create im_mat_20_uni
app.im_mat_20_uni = uiimage(app.AadirmaterialesTab);
app.im_mat_20_uni.Visible = 'off';
app.im_mat_20_uni.Position = [291 370 25 19];
app.im_mat_20_uni.ImageSource = 'Mat_20.PNG';

% Create im_mat_19_uni
app.im_mat_19_uni = uiimage(app.AadirmaterialesTab);
app.im_mat_19_uni.Visible = 'off';
app.im_mat_19_uni.Position = [291 390 25 23];
app.im_mat_19_uni.ImageSource = 'Mat_19.PNG';

% Create im_mat_18_uni
app.im_mat_18_uni = uiimage(app.AadirmaterialesTab);
app.im_mat_18_uni.Visible = 'off';
app.im_mat_18_uni.Position = [292 413 24 19];
app.im_mat_18_uni.ImageSource = 'Mat_18.PNG';

% Create im_mat_17_uni
app.im_mat_17_uni = uiimage(app.AadirmaterialesTab);
app.im_mat_17_uni.Visible = 'off';
app.im_mat_17_uni.Position = [291 441 25 20];
app.im_mat_17_uni.ImageSource = 'Mat_17.PNG';

% Create im_mat_16_uni
app.im_mat_16_uni = uiimage(app.AadirmaterialesTab);
app.im_mat_16_uni.Visible = 'off';
app.im_mat_16_uni.Position = [291 461 25 25];
app.im_mat_16_uni.ImageSource = 'Mat_16.PNG';

% Create im_mat_15_uni
app.im_mat_15_uni = uiimage(app.AadirmaterialesTab);
app.im_mat_15_uni.Visible = 'off';
app.im_mat_15_uni.Position = [291 489 25 21];
```

```
            app.im_mat_15_uni.ImageSource = 'Mat_15.PNG';

            % Create im_mat_14_uni
            app.im_mat_14_uni = uiimage(app.AadirmaterialesTab);
            app.im_mat_14_uni.Visible = 'off';
            app.im_mat_14_uni.Position = [291 510 26 26];
            app.im_mat_14_uni.ImageSource = 'Mat_14.PNG';

            % Create ImportarDatosButton
            app.ImportarDatosButton = uibutton(app.AadirmaterialesTab, 'push');
            app.ImportarDatosButton.ButtonPushedFcn = createCallbackFcn(app, @ImportarDatosButtonPushed,
true);
            app.ImportarDatosButton.FontWeight = 'bold';
            app.ImportarDatosButton.Position = [14 695 101 22];
            app.ImportarDatosButton.Text = 'Importar Datos';

            % Create CargandoLabel
            app.CargandoLabel = uilabel(app.AadirmaterialesTab);
            app.CargandoLabel.Visible = 'off';
            app.CargandoLabel.Position = [137 695 68 22];
            app.CargandoLabel.Text = 'Cargando...';

            % Create DireccinLabel
            app.DireccinLabel = uilabel(app.AadirmaterialesTab);
            app.DireccinLabel.Visible = 'off';
            app.DireccinLabel.Position = [14 668 55 22];
            app.DireccinLabel.Text = 'Dirección';

            % Create direccion_excelLabel
            app.direccion_excelLabel = uilabel(app.AadirmaterialesTab);
            app.direccion_excelLabel.Visible = 'off';
            app.direccion_excelLabel.Position = [77 668 255 22];
            app.direccion_excelLabel.Text = 'direccion_excel';

            % Create NombreLabel
            app.NombreLabel = uilabel(app.AadirmaterialesTab);
            app.NombreLabel.Visible = 'off';
            app.NombreLabel.Position = [14 639 51 22];
            app.NombreLabel.Text = 'Nombre ';

            % Create nombre_excelLabel
            app.nombre_excelLabel = uilabel(app.AadirmaterialesTab);
            app.nombre_excelLabel.Visible = 'off';
            app.nombre_excelLabel.Position = [77 639 255 22];
            app.nombre_excelLabel.Text = 'nombre_excel';

            % Create Material9_torCheckBox
            app.Material9_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material9_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material9_torCheckBoxValueChanged, true);
            app.Material9_torCheckBox.Visible = 'off';
            app.Material9_torCheckBox.Text = 'Material 9_tor';
            app.Material9_torCheckBox.Position = [16 57 95 22];

            % Create Material12_torCheckBox
            app.Material12_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material12_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material12_torCheckBoxValueChanged, true);
            app.Material12_torCheckBox.Visible = 'off';
            app.Material12_torCheckBox.Text = 'Material 12_tor';
            app.Material12_torCheckBox.Position = [181 250 102 22];

            % Create Material14_torCheckBox
            app.Material14_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material14_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material14_torCheckBoxValueChanged, true);
            app.Material14_torCheckBox.Visible = 'off';
            app.Material14_torCheckBox.Text = 'Material 14_tor';
            app.Material14_torCheckBox.Position = [181 201 102 22];

            % Create Material15_torCheckBox
            app.Material15_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material15_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material15_torCheckBoxValueChanged, true);
```

```
            app.Material15_torCheckBox.Visible = 'off';
            app.Material15_torCheckBox.Text = 'Material 15_tor';
            app.Material15_torCheckBox.Position = [181 177 102 22];

            % Create Material20_torCheckBox
            app.Material20_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material20_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material20_torCheckBoxValueChanged, true);
            app.Material20_torCheckBox.Visible = 'off';
            app.Material20_torCheckBox.Text = 'Material 20_tor';
            app.Material20_torCheckBox.Position = [181 57 102 22];

            % Create Material21_torCheckBox
            app.Material21_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material21_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material21_torCheckBoxValueChanged, true);
            app.Material21_torCheckBox.Visible = 'off';
            app.Material21_torCheckBox.Text = 'Material 21_tor';
            app.Material21_torCheckBox.Position = [181 33 102 22];

            % Create Material22_torCheckBox
            app.Material22_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material22_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material22_torCheckBoxValueChanged, true);
            app.Material22_torCheckBox.Visible = 'off';
            app.Material22_torCheckBox.Text = 'Material 22_tor';
            app.Material22_torCheckBox.Position = [181 9 102 22];

            % Create Material16_torCheckBox
            app.Material16_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material16_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material16_torCheckBoxValueChanged, true);
            app.Material16_torCheckBox.Visible = 'off';
            app.Material16_torCheckBox.Text = 'Material 16_tor';
            app.Material16_torCheckBox.Position = [181 153 102 22];

            % Create Material17_torCheckBox
            app.Material17_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material17_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material17_torCheckBoxValueChanged, true);
            app.Material17_torCheckBox.Visible = 'off';
            app.Material17_torCheckBox.Text = 'Material 17_tor';
            app.Material17_torCheckBox.Position = [181 129 102 22];

            % Create Material18_torCheckBox
            app.Material18_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material18_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material18_torCheckBoxValueChanged, true);
            app.Material18_torCheckBox.Visible = 'off';
            app.Material18_torCheckBox.Text = 'Material 18_tor';
            app.Material18_torCheckBox.Position = [181 105 102 22];

            % Create Material19_torCheckBox
            app.Material19_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material19_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material19_torCheckBoxValueChanged, true);
            app.Material19_torCheckBox.Visible = 'off';
            app.Material19_torCheckBox.Text = 'Material 19_tor';
            app.Material19_torCheckBox.Position = [181 81 102 22];

            % Create Material10_torCheckBox
            app.Material10_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material10_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material10_torCheckBoxValueChanged, true);
            app.Material10_torCheckBox.Visible = 'off';
            app.Material10_torCheckBox.Text = 'Material 10_tor';
            app.Material10_torCheckBox.Position = [16 33 102 22];

            % Create Material11_torCheckBox
            app.Material11_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material11_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material11_torCheckBoxValueChanged, true);
            app.Material11_torCheckBox.Visible = 'off';
            app.Material11_torCheckBox.Text = 'Material 11_tor';
```

```
            app.Material11_torCheckBox.Position = [16 9 101 22];

            % Create Material1_torCheckBox
            app.Material1_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material1_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material1_torCheckBoxValueChanged, true);
            app.Material1_torCheckBox.Visible = 'off';
            app.Material1_torCheckBox.Text = 'Material 1_tor';
            app.Material1_torCheckBox.Position = [16 250 95 22];

            % Create Material2_torCheckBox
            app.Material2_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material2_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material2_torCheckBoxValueChanged, true);
            app.Material2_torCheckBox.Visible = 'off';
            app.Material2_torCheckBox.Text = 'Material 2_tor';
            app.Material2_torCheckBox.Position = [16 225 95 22];

            % Create Material3_torCheckBox
            app.Material3_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material3_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material3_torCheckBoxValueChanged, true);
            app.Material3_torCheckBox.Visible = 'off';
            app.Material3_torCheckBox.Text = 'Material 3_tor';
            app.Material3_torCheckBox.Position = [16 201 95 22];

            % Create Material4_torCheckBox
            app.Material4_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material4_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material4_torCheckBoxValueChanged, true);
            app.Material4_torCheckBox.Visible = 'off';
            app.Material4_torCheckBox.Text = 'Material 4_tor';
            app.Material4_torCheckBox.Position = [16 177 95 22];

            % Create Material5_torCheckBox
            app.Material5_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material5_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material5_torCheckBoxValueChanged, true);
            app.Material5_torCheckBox.Visible = 'off';
            app.Material5_torCheckBox.Text = 'Material 5_tor';
            app.Material5_torCheckBox.Position = [16 153 95 22];

            % Create Material6_torCheckBox
            app.Material6_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material6_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material6_torCheckBoxValueChanged, true);
            app.Material6_torCheckBox.Visible = 'off';
            app.Material6_torCheckBox.Text = 'Material 6_tor';
            app.Material6_torCheckBox.Position = [16 129 95 22];

            % Create Material7_torCheckBox
            app.Material7_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material7_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material7_torCheckBoxValueChanged, true);
            app.Material7_torCheckBox.Visible = 'off';
            app.Material7_torCheckBox.Text = 'Material 7_tor';
            app.Material7_torCheckBox.Position = [16 105 95 22];

            % Create Material8_torCheckBox
            app.Material8_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material8_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material8_torCheckBoxValueChanged, true);
            app.Material8_torCheckBox.Visible = 'off';
            app.Material8_torCheckBox.Text = 'Material 8_tor';
            app.Material8_torCheckBox.Position = [16 81 95 22];

            % Create Material13_torCheckBox
            app.Material13_torCheckBox = uicheckbox(app.AadirmaterialesTab);
            app.Material13_torCheckBox.ValueChangedFcn = createCallbackFcn(app,
@Material13_torCheckBoxValueChanged, true);
            app.Material13_torCheckBox.Visible = 'off';
            app.Material13_torCheckBox.Text = 'Material 13_tor';
            app.Material13_torCheckBox.Position = [181 225 102 22];
```

```
% Create im_mat_1_tor
app.im_mat_1_tor = uiimage(app.AadirmaterialesTab);
app.im_mat_1_tor.Visible = 'off';
app.im_mat_1_tor.Position = [125 249 25 23];
app.im_mat_1_tor.ImageSource = 'Mat_1.PNG';

% Create im_mat_2_tor
app.im_mat_2_tor = uiimage(app.AadirmaterialesTab);
app.im_mat_2_tor.Visible = 'off';
app.im_mat_2_tor.Position = [125 223 25 23];
app.im_mat_2_tor.ImageSource = 'Mat_2.PNG';

% Create im_mat_3_tor
app.im_mat_3_tor = uiimage(app.AadirmaterialesTab);
app.im_mat_3_tor.Visible = 'off';
app.im_mat_3_tor.Position = [125 198 25 23];
app.im_mat_3_tor.ImageSource = 'Mat_3.PNG';

% Create im_mat_4_tor
app.im_mat_4_tor = uiimage(app.AadirmaterialesTab);
app.im_mat_4_tor.Visible = 'off';
app.im_mat_4_tor.Position = [125 177 25 23];
app.im_mat_4_tor.ImageSource = 'Mat_4.PNG';

% Create im_mat_5_tor
app.im_mat_5_tor = uiimage(app.AadirmaterialesTab);
app.im_mat_5_tor.Visible = 'off';
app.im_mat_5_tor.Position = [125 152 25 23];
app.im_mat_5_tor.ImageSource = 'Mat_5.PNG';

% Create im_mat_6_tor
app.im_mat_6_tor = uiimage(app.AadirmaterialesTab);
app.im_mat_6_tor.Visible = 'off';
app.im_mat_6_tor.Position = [125 129 25 23];
app.im_mat_6_tor.ImageSource = 'Mat_6.PNG';

% Create im_mat_7_tor
app.im_mat_7_tor = uiimage(app.AadirmaterialesTab);
app.im_mat_7_tor.Visible = 'off';
app.im_mat_7_tor.Position = [125 104 25 23];
app.im_mat_7_tor.ImageSource = 'Mat_7.PNG';

% Create im_mat_8_tor
app.im_mat_8_tor = uiimage(app.AadirmaterialesTab);
app.im_mat_8_tor.Visible = 'off';
app.im_mat_8_tor.Position = [125 80 25 23];
app.im_mat_8_tor.ImageSource = 'Mat_8.PNG';

% Create im_mat_9_tor
app.im_mat_9_tor = uiimage(app.AadirmaterialesTab);
app.im_mat_9_tor.Visible = 'off';
app.im_mat_9_tor.Position = [125 55 25 23];
app.im_mat_9_tor.ImageSource = 'Mat_9.PNG';

% Create im_mat_10_tor
app.im_mat_10_tor = uiimage(app.AadirmaterialesTab);
app.im_mat_10_tor.Visible = 'off';
app.im_mat_10_tor.Position = [125 33 25 23];
app.im_mat_10_tor.ImageSource = 'Mat_10.PNG';

% Create im_mat_11_tor
app.im_mat_11_tor = uiimage(app.AadirmaterialesTab);
app.im_mat_11_tor.Visible = 'off';
app.im_mat_11_tor.Position = [125 6 25 25];
app.im_mat_11_tor.ImageSource = 'Mat_11.PNG';

% Create im_mat_12_tor
app.im_mat_12_tor = uiimage(app.AadirmaterialesTab);
app.im_mat_12_tor.Visible = 'off';
app.im_mat_12_tor.Position = [291 250 25 23];
app.im_mat_12_tor.ImageSource = 'Mat_12.PNG';

% Create im_mat_13_tor
app.im_mat_13_tor = uiimage(app.AadirmaterialesTab);
```

```
            app.im_mat_13_tor.Visible = 'off';
            app.im_mat_13_tor.Position = [290 223 27 26];
            app.im_mat_13_tor.ImageSource = 'Mat_13.PNG';

            % Create im_mat_22_tor
            app.im_mat_22_tor = uiimage(app.AadirmaterialesTab);
            app.im_mat_22_tor.Visible = 'off';
            app.im_mat_22_tor.Position = [291 11 26 22];
            app.im_mat_22_tor.ImageSource = 'Mat_22.PNG';

            % Create im_mat_21_tor
            app.im_mat_21_tor = uiimage(app.AadirmaterialesTab);
            app.im_mat_21_tor.Visible = 'off';
            app.im_mat_21_tor.Position = [291 33 25 23];
            app.im_mat_21_tor.ImageSource = 'mat_21.PNG';

            % Create im_mat_20_tor
            app.im_mat_20_tor = uiimage(app.AadirmaterialesTab);
            app.im_mat_20_tor.Visible = 'off';
            app.im_mat_20_tor.Position = [291 59 25 19];
            app.im_mat_20_tor.ImageSource = 'Mat_20.PNG';

            % Create im_mat_19_tor
            app.im_mat_19_tor = uiimage(app.AadirmaterialesTab);
            app.im_mat_19_tor.Visible = 'off';
            app.im_mat_19_tor.Position = [291 79 25 23];
            app.im_mat_19_tor.ImageSource = 'Mat_19.PNG';

            % Create im_mat_18_tor
            app.im_mat_18_tor = uiimage(app.AadirmaterialesTab);
            app.im_mat_18_tor.Visible = 'off';
            app.im_mat_18_tor.Position = [292 102 24 19];
            app.im_mat_18_tor.ImageSource = 'Mat_18.PNG';

            % Create im_mat_17_tor
            app.im_mat_17_tor = uiimage(app.AadirmaterialesTab);
            app.im_mat_17_tor.Visible = 'off';
            app.im_mat_17_tor.Position = [291 130 25 20];
            app.im_mat_17_tor.ImageSource = 'Mat_17.PNG';

            % Create im_mat_16_tor
            app.im_mat_16_tor = uiimage(app.AadirmaterialesTab);
            app.im_mat_16_tor.Visible = 'off';
            app.im_mat_16_tor.Position = [291 150 25 25];
            app.im_mat_16_tor.ImageSource = 'Mat_16.PNG';

            % Create im_mat_15_tor
            app.im_mat_15_tor = uiimage(app.AadirmaterialesTab);
            app.im_mat_15_tor.Visible = 'off';
            app.im_mat_15_tor.Position = [291 178 25 21];
            app.im_mat_15_tor.ImageSource = 'Mat_15.PNG';

            % Create im_mat_14_tor
            app.im_mat_14_tor = uiimage(app.AadirmaterialesTab);
            app.im_mat_14_tor.Visible = 'off';
            app.im_mat_14_tor.Position = [291 199 26 26];
            app.im_mat_14_tor.ImageSource = 'Mat_14.PNG';

            % Create BorrardatosButton
            app.BorrardatosButton = uibutton(app.AadirmaterialesTab, 'push');
            app.BorrardatosButton.ButtonPushedFcn = createCallbackFcn(app, @BorrardatosButtonPushed, true);
            app.BorrardatosButton.Position = [218 696 100 22];
            app.BorrardatosButton.Text = 'Borrar datos';

            % Show the figure after all components are created
            app.pag_principal.Visible = 'on';
        end
    end

    % App creation and deletion
    methods (Access = public)

        % Construct app
        function app = pagina_principal
```

```
            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.pag_principal)

            % Execute the startup function
            runStartupFcn(app, @startupFcn)

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.pag_principal)
        end
    end
end
```

Funciones auxiliares

"bisección_crossland_alterna"

```
function [x_0] = biseccion_crossland_alterna(alfa_cross,s_m_cross,beta_cross,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_crossland_alterna(a_0,alfa_cross,s_m_cross,beta_cross);
f_sup=funcion_crossland_alterna(b_0,alfa_cross,s_m_cross,beta_cross);
%     f_inf=subs(funcion,a_0);
%     f_sup=subs(funcion,b_0);
    cont=0;
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_crossland_alterna(a_0,alfa_cross,s_m_cross,beta_cross);
        f_sup=funcion_crossland_alterna(b_0,alfa_cross,s_m_cross,beta_cross);
        cont=cont+1;
        if cont>3
            break
        end
    end
    if cont>=3
        x_0=x_0_in;
    else
        %metodo iterativo
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_crossland_alterna(xm,alfa_cross,s_m_cross,beta_cross);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
    end

end
```

"bisección_crossland_media"

```
function [x_0] = biseccion_crossland_media(alfa_cross,beta_cross,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_crossland_media(a_0,alfa_cross,beta_cross);
f_sup=funcion_crossland_media(b_0,alfa_cross,beta_cross);

    cont=0;
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_crossland_media(a_0,alfa_cross,beta_cross);
        f_sup=funcion_crossland_media(b_0,alfa_cross,beta_cross);

        cont=cont+1;
        if cont>3
            break
        end
    end
```

```
    if cont>=3
        x_0=x_0_in;
    else
        %metodo iterativo
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_crossland_media(xm,alfa_cross,beta_cross);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
    end

end
```

"bisección_dangvan_alterna"

```
function [x_0] = biseccion_dangvan_alterna(alfa_dv,s_m_dv,beta_dv,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_dangvan_alterna(a_0,alfa_dv,s_m_dv,beta_dv);
f_sup=funcion_dangvan_alterna(b_0,alfa_dv,s_m_dv,beta_dv);

    cont=0;
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_dangvan_alterna(a_0,alfa_dv,s_m_dv,beta_dv);
        f_sup=funcion_dangvan_alterna(b_0,alfa_dv,s_m_dv,beta_dv);
        cont=cont+1;
        if cont>3
            break
        end
    end
    if cont>=3
        x_0=x_0_in;
    else
        %metodo iterativo
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_dangvan_alterna(xm,alfa_dv,s_m_dv,beta_dv);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
    end

end
```

"bisección_dangvan_media"

```
function [x_0] = biseccion_dangvan_media(alfa_dv,beta_dv,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_dangvan_media(a_0,alfa_dv,beta_dv);
```

```
f_sup=funcion_dangvan_media(b_0,alfa_dv,beta_dv);

    cont=0;
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_dangvan_media(a_0,alfa_dv,beta_dv);
        f_sup=funcion_dangvan_media(b_0,alfa_dv,beta_dv);
        cont=cont+1;
        if cont>3
            break
        end
    end
    if cont>=3
        x_0=x_0_in;
    else
        %metodo iterativo
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_dangvan_media(xm,alfa_dv,beta_dv);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
    end

end
```

"bisección_findley_ax_alterna"

```
function [x_0] = biseccion_findley_ax_alterna(alfa_fin,s_m_fin,beta_fin,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_findley_ax_alterna(a_0,alfa_fin,s_m_fin,beta_fin);
f_sup=funcion_findley_ax_alterna(b_0,alfa_fin,s_m_fin,beta_fin);

    cont=0;
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_findley_ax_alterna(a_0,alfa_fin,s_m_fin,beta_fin);
        f_sup=funcion_findley_ax_alterna(b_0,alfa_fin,s_m_fin,beta_fin);
        cont=cont+1;
        if cont>3
            break
        end
    end
    if cont>=3
        x_0=x_0_in;
    else
        %metodo iterativo
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_findley_ax_alterna(xm,alfa_fin,s_m_fin,beta_fin);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
```

```
    end

end
```

"bisección_findley_ax_media"

```
function [x_0] = biseccion_findley_ax_media(alfa_fin,beta_fin,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_findley_ax_media(a_0,alfa_fin,beta_fin);
f_sup=funcion_findley_ax_media(b_0,alfa_fin,beta_fin);

    cont=0;
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_findley_ax_media(a_0,alfa_fin,beta_fin);
        f_sup=funcion_findley_ax_media(b_0,alfa_fin,beta_fin);
        cont=cont+1;
        if cont>3
            break
        end
    end
    if cont>=3
        x_0=x_0_in;
    else
        %metodo iterativo
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_findley_ax_media(xm,alfa_fin,beta_fin);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
    end

end
```

"bisección_findley_tor_alterna"

```
function [x_0] = biseccion_findley_tor_alterna(alfa_fin,t_m_fin,beta_fin,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_findley_tor_alterna(a_0,alfa_fin,t_m_fin,beta_fin);
f_sup=funcion_findley_tor_alterna(b_0,alfa_fin,t_m_fin,beta_fin);

    cont=0;
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_findley_tor_alterna(a_0,alfa_fin,t_m_fin,beta_fin);
        f_sup=funcion_findley_tor_alterna(b_0,alfa_fin,t_m_fin,beta_fin);
        cont=cont+1;
        if cont>3
            break
        end
    end
    if cont>=3
        x_0=x_0_in;
    else
```

```
        %metodo iterativo
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_findley_tor_alterna(xm,alfa_fin,t_m_fin,beta_fin);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
    end

end
```

"bisección_findley_tor_media"

```
function [x_0] = biseccion_findley_tor_media(alfa_fin,beta_fin,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_findley_tor_media(a_0,alfa_fin,beta_fin);
f_sup=funcion_findley_tor_media(b_0,alfa_fin,beta_fin);

    cont=0;
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_findley_tor_media(a_0,alfa_fin,beta_fin);
        f_sup=funcion_findley_tor_media(b_0,alfa_fin,beta_fin);
        cont=cont+1;
        if cont>3
            break
        end
    end
    if cont>=3
        x_0=x_0_in;
    else
        %metodo iterativo
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_findley_tor_media(xm,alfa_fin,beta_fin);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
    end

end
```

"bisección_findley_zer"

```
function [x_0] = biseccion_findley_zer(lf_tr_pul,lf_ax_alt,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_findley_zer(a_0,lf_tr_pul,lf_ax_alt);
f_sup=funcion_findley_zer(b_0,lf_tr_pul,lf_ax_alt);

    cont=0;
```

```
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_findley_zer(a_0,lf_tr_pul,lf_ax_alt);
        f_sup=funcion_findley_zer(b_0,lf_tr_pul,lf_ax_alt);
        cont=cont+1;
        if cont>3
            break
        end
    end
    if cont>=3
        x_0=x_0_in;
    else
        %metodo iterativo
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_findley_zer(xm,lf_tr_pul,lf_ax_alt);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
    end

end
```

"bisección_matake_alterna"

```
function [x_0] = biseccion_matake_alterna(alfa_mat,s_m_mat,beta_mat,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_matake_alterna(a_0,alfa_mat,s_m_mat,beta_mat);
f_sup=funcion_matake_alterna(b_0,alfa_mat,s_m_mat,beta_mat);

    cont=0;
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_matake_alterna(a_0,alfa_mat,s_m_mat,beta_mat);
        f_sup=funcion_matake_alterna(b_0,alfa_mat,s_m_mat,beta_mat);
        cont=cont+1;
        if cont>3
            break
        end
    end
    if cont>=3
        x_0=x_0_in;
    else
        %metodo iterativo
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_matake_alterna(xm,alfa_mat,s_m_mat,beta_mat);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
    end
```

```
end
```

"bisección_matake_media"

```
function [x_0] = biseccion_matake_media(alfa_mat,beta_mat,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_matake_media(a_0,alfa_mat,beta_mat);
f_sup=funcion_matake_media(b_0,alfa_mat,beta_mat);

    cont=0;
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_matake_media(a_0,alfa_mat,beta_mat);
        f_sup=funcion_matake_media(b_0,alfa_mat,beta_mat);
        cont=cont+1;
        if cont>3
            break
        end
    end
    if cont>=3
        x_0=x_0_in;
    else
        %metodo iterativo
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_matake_media(xm,alfa_mat,beta_mat);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
    end

end
```

"bisección_papuga_ax_alterna"

```
function [x_0] =
biseccion_papuga_ax_alterna(a_p,s_m_pap,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_papuga_ax_alterna(a_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,s_m_pap);
f_sup=funcion_papuga_ax_alterna(b_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,s_m_pap);

    cont=0;
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_papuga_ax_alterna(a_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,s_m_pap);
        f_sup=funcion_papuga_ax_alterna(b_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,s_m_pap);
        cont=cont+1;
        if cont>3
            break
        end
    end
    if cont>=3
        x_0=x_0_in;
    else
        %metodo iterativo
```

```
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_papuga_ax_alterna(xm,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,s_m_pap);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
    end

end
```

"bisección_ papuga_ax_media"

```
function [x_0] =
biseccion_papuga_ax_media(a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_papuga_ax_media(a_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt);
f_sup=funcion_papuga_ax_media(b_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt);

    cont=0;
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_papuga_ax_media(a_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt);
        f_sup=funcion_papuga_ax_media(b_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt);
        cont=cont+1;
        if cont>3
            break
        end
    end
    if cont>=3
        x_0=x_0_in;
    else
        %metodo iterativo
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_papuga_ax_media(xm,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
    end

end
```

"bisección_papuga_tor_alterna"

```
function [x_0] =
biseccion_papuga_tor_alterna(a_p,t_m_pap,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_papuga_tor_alterna(a_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,t_m_pap);
f_sup=funcion_papuga_tor_alterna(b_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,t_m_pap);
```

```
    cont=0;
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_papuga_tor_alterna(a_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,t_m_pap);
        f_sup=funcion_papuga_tor_alterna(b_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,t_m_pap);
        cont=cont+1;
        if cont>3
            break
        end
    end
    if cont>=3
        x_0=x_0_in;
    else
        %metodo iterativo
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_papuga_tor_alterna(xm,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,t_m_pap);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
    end

end
```

"bisección_papuga_tor_media"

```
function [x_0] =
biseccion_papuga_tor_media(a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_papuga_tor_media(a_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt);
f_sup=funcion_papuga_tor_media(b_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt);

    cont=0;
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_papuga_tor_media(a_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt);
        f_sup=funcion_papuga_tor_media(b_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt);
        cont=cont+1;
        if cont>3
            break
        end
    end
    if cont>=3
        x_0=x_0_in;
    else
        %metodo iterativo
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_papuga_tor_media(xm,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
```

```
    end

end
```

"bisección_ sines_alterna"

```
function [x_0] = biseccion_sines_alterna(alfa_sin,beta_sin,s_m_sin,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_sines_alterna(a_0,alfa_sin,beta_sin,s_m_sin);
f_sup=funcion_sines_alterna(b_0,alfa_sin,beta_sin,s_m_sin);

    cont=0;
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_sines_alterna(a_0,alfa_sin,beta_sin,s_m_sin);
        f_sup=funcion_sines_alterna(b_0,alfa_sin,beta_sin,s_m_sin);
        cont=cont+1;
        if cont>3
            break
        end
    end
    if cont>=3
        x_0=x_0_in;
    else
        %metodo iterativo
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_sines_alterna(xm,alfa_sin,beta_sin,s_m_sin);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
    end

end
```

"bisección_sines_media"

```
function [x_0] = biseccion_sines_media(alfa_sin,beta_sin,a_0,b_0,tol_biseccion)
%   Obtiene una aproximacion inicial a partir del método de la biseccion
%   Criterio de parada: "Longitud"

f_inf=funcion_sines_media(a_0,alfa_sin,beta_sin);
f_sup=funcion_sines_media(b_0,alfa_sin,beta_sin);

    cont=0;
    x_0_in=a_0;
    while sign(f_inf)==sign(f_sup)
        a_0=a_0-80;
        b_0=b_0+80;
        f_inf=funcion_sines_media(a_0,alfa_sin,beta_sin);
        f_sup=funcion_sines_media(b_0,alfa_sin,beta_sin);
        cont=cont+1;
        if cont>3
            break
        end
    end
    if cont>=3
        x_0=x_0_in;
    else
```

```
        %metodo iterativo
        error_longitud=tol_biseccion+1;
        while error_longitud>tol_biseccion
            xm=(a_0+b_0)/2;
            f_sus=funcion_sines_media(xm,alfa_sin,beta_sin);
            if f_inf*f_sus<0
                b_0=xm;
            else
                a_0=xm;
            end
            error_longitud =abs(b_0-a_0);
        end

        x_0=xm;
    end

end
```

"calcula_alfa_findley_zer"

```
function [alfa_findley_zer] = calcula_alfa_findley_zer(lf_ax_alt,lf_tr_pul, tolerancia)
% Calcula alfa de Findley para s_-1 - s_0

    a_0=-10;
    b_0=+10;

    %Bolzano
    tol_biseccion=tolerancia*50;
    [x_0] = biseccion_findley_zer(lf_tr_pul,lf_ax_alt,a_0,b_0,tol_biseccion);

    %Newton-Raphson
    alfa_findley_zer=newton_raphson_findley_zer(lf_tr_pul,lf_ax_alt,x_0,tolerancia);

end
```

"calcula_crossland_axial"

```
function [s_m_cross,s_a_cross,entra_cross,puntos_cross] =
calcula_crossland_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa_cross,beta_cross)
% Calcula crossland

    puntos_cross=0;

    s_m_cross=zeros(puntos_discretizacion,1); %Parte de traccion y compresion
    s_a_cross=zeros(puntos_discretizacion,1);

    a_0=lf_ax_alt-10;
    b_0=lf_ax_alt+10;

    entra_cross=0;
    %Parte traccion y compresión
    for ii=1:puntos_discretizacion
        if ii==1
            s_m_cross(ii)=-s_ut;
        elseif ii==(puntos_discretizacion)
            s_m_cross(ii)=s_ut;
        else
            s_m_cross(ii)=s_m_cross(ii-1)+avance;
        end

        if ii~=1 && s_m_cross(ii)~=0
            if s_m_cross(ii-1)<0 && s_m_cross(ii)>0
                s_m_cross(ii)=0;
            end
        end

        %aproximacion inicial
        if s_m_cross~=0
            a_0=s_a_cross(ii-1)-10;
            b_0=s_a_cross(ii-1)+10;
```

```
        end

        %Bolzano
        tol_biseccion=tolerancia*50;
        [x_0] = biseccion_crossland_alterna(alfa_cross,s_m_cross(ii),beta_cross,a_0,b_0,tol_biseccion);

        %Newton-Raphson
        s_a_cross(ii)=newton_raphson_crossland_alt(alfa_cross,s_m_cross(ii),beta_cross,x_0,tolerancia);

    end

    % Cuando S_m/s_ut=1 y la curva no es horizontal, se calculan puntos hasta que la funcion corte la
horizontal (nuevo_numero_puntos)
    diferen=abs(s_a_cross(1)-s_a_cross(puntos_discretizacion));
    if diferen>10 && s_a_cross(puntos_discretizacion)>0
        entra_cross=1;
        %solo en este caso, se va a buscar el valor de la S_m
        a_0=s_m_cross(puntos_discretizacion)-10;
        b_0=s_m_cross(puntos_discretizacion)+10;

        %Bolzano
        tol_biseccion=tolerancia*50;
        [x_0] = biseccion_crossland_media(alfa_cross,beta_cross,a_0,b_0,tol_biseccion);

        %Newton-Raphson
        s_m_final=double(newton_raphson_crossland_m(alfa_cross,beta_cross,x_0,tolerancia));

        %Se calculan los nuevos puntos crossland
        s_rep=s_m_final-s_ut;

        if mod(s_rep,avance)==0
           puntos_cross=round(s_rep/avance); %+1; no es +1 porque el punto de S_m/s_ut=1 ya esta
        else
            puntos_cross=round(s_rep/avance)+1; %+2;
        end

        zer=zeros(puntos_cross,1);
        s_m_cross=[s_m_cross ; zer];
        s_a_cross=[s_a_cross ; zer];

        puntos_total=(puntos_discretizacion)+puntos_cross;
        for ii=(puntos_discretizacion):puntos_total

            if ii==puntos_total
                s_m_cross(ii)=s_m_final;
                s_a_cross(ii)=0;
            else
                s_m_cross(ii)=s_m_cross(ii-1)+avance;

                %aproximacion inicial
                if s_m_cross(ii)~=0
                    a_0=s_a_cross(ii-1)-10;
                    b_0=s_a_cross(ii-1)+10;
                end

                %Bolzano
                tol_biseccion=tolerancia*50;
                [x_0] =
biseccion_crossland_alterna(alfa_cross,s_m_cross(ii),beta_cross,a_0,b_0,tol_biseccion);
                %Newton-Raphson

s_a_cross(ii)=newton_raphson_crossland_alt(alfa_cross,s_m_cross(ii),beta_cross,x_0,tolerancia);
            end
        end
        if puntos_cross==0
            entra_cross=0;
        end
    end

         s_m_cross=s_m_cross/s_ut;
         s_a_cross=s_a_cross/lf_ax_alt;
end
```

"calcula_dangvan_axial"

```
function [s_m_dv,s_a_dv,entra_dv,puntos_dv] =
calcula_dangvan_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa_dv,beta_dv)
% Calcula Dang Van

    puntos_dv=0;

    s_m_dv=zeros(puntos_discretizacion,1); %Parte de traccion y compresion
    s_a_dv=zeros(puntos_discretizacion,1);

    a_0=lf_ax_alt-10;
    b_0=lf_ax_alt+10;

    %Parte traccion y compresión
    for ii=1:puntos_discretizacion
        entra_dv=0;
        if ii==1
            s_m_dv(ii)=-s_ut;
        elseif ii==(puntos_discretizacion)
            s_m_dv(ii)=s_ut;
        else
            s_m_dv(ii)=s_m_dv(ii-1)+avance;
        end

        if ii~=1 && s_m_dv(ii)~=0
            if s_m_dv(ii-1)<0 && s_m_dv(ii)>0
                s_m_dv(ii)=0;
            end
        end

        %aproximacion inicial
        if s_m_dv~=0
            a_0=s_a_dv(ii-1)-10;
            b_0=s_a_dv(ii-1)+10;
        end

        %Bolzano
        tol_biseccion=tolerancia*50;
        [x_0] = biseccion_dangvan_alterna(alfa_dv,s_m_dv(ii),beta_dv,a_0,b_0,tol_biseccion);

        %Newton-Raphson
        s_a_dv(ii)=newton_raphson_dangvan_alt(alfa_dv,s_m_dv(ii),beta_dv,x_0,tolerancia);

    end

    % Cuando S_m/s_ut=1 y la curva no es horizontal, se calculan puntos hasta que la funcion corte la
horizontal (nuevo_numero_puntos)
    diferen=abs(s_a_dv(1)-s_a_dv(puntos_discretizacion));
    if diferen>10 && s_a_dv(puntos_discretizacion)>0
        entra_dv=1;
        %solo en este caso, se va a buscar el valor de la S_m
        a_0=s_m_dv(puntos_discretizacion)-10;
        b_0=s_m_dv(puntos_discretizacion)+10;

        %Bolzano
        tol_biseccion=tolerancia*50;
        [x_0] = biseccion_dangvan_media(alfa_dv,beta_dv,a_0,b_0,tol_biseccion);

        %Newton-Raphson
        s_m_final=double(newton_raphson_dangvan_m(alfa_dv,beta_dv,x_0,tolerancia));

        %Se calculan los nuevos puntos dangvan
        s_rep=s_m_final-s_ut;

        if mod(s_rep,avance)==0
           puntos_dv=round(s_rep/avance); %+1; no es +1 porque el punto de S_m/s_ut=1 ya esta
        else
            puntos_dv=round(s_rep/avance)+1; %+2;
        end

        zer=zeros(puntos_dv,1);
        s_m_dv=[s_m_dv ; zer];
```

```
        s_a_dv=[s_a_dv ; zer];

        puntos_total=(puntos_discretizacion)+puntos_dv;
        for ii=(puntos_discretizacion):puntos_total

            if ii==puntos_total
                s_m_dv(ii)=s_m_final;
                s_a_dv(ii)=0;
            else
                s_m_dv(ii)=s_m_dv(ii-1)+avance;

                %aproximacion inicial
                if s_m_dv(ii)~=0
                    a_0=s_a_dv(ii-1)-10;
                    b_0=s_a_dv(ii-1)+10;
                end

                %Bolzano
                tol_biseccion=tolerancia*50;
                [x_0] = biseccion_dangvan_alterna(alfa_dv,s_m_dv(ii),beta_dv,a_0,b_0,tol_biseccion);

                %Newton-Raphson
                s_a_dv(ii)=newton_raphson_dangvan_alt(alfa_dv,s_m_dv(ii),beta_dv,x_0,tolerancia);

            end
        end
        if puntos_dv==0
            entra_dv=0;
        end
    end

    %Por último, se normalizan los datos
        %Eje vertical: S_a/S_e
        %eje horizontal: S_m/s_ut
        s_m_dv=s_m_dv/s_ut;
        s_a_dv=s_a_dv/lf_ax_alt;
end
```

"calcula_findley_axial"

```
function [s_m_fin,s_a_fin,entra_fin,puntos_fin] =
calcula_findley_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa_fin,beta_fin)
% Calcula findley

    puntos_fin=0;

    s_m_fin=zeros(puntos_discretizacion,1); %Parte de traccion y compresion
    s_a_fin=zeros(puntos_discretizacion,1);

    a_0=lf_ax_alt-10;
    b_0=lf_ax_alt+10;

    %Parte traccion y compresión
    for ii=1:puntos_discretizacion
        entra_fin=0;
        if ii==1
            s_m_fin(ii)=-s_ut;
        elseif ii==(puntos_discretizacion)
            s_m_fin(ii)=s_ut;
        else
            s_m_fin(ii)=s_m_fin(ii-1)+avance;
        end

        if ii~=1 && s_m_fin(ii)~=0
            if s_m_fin(ii-1)<0 && s_m_fin(ii)>0
                s_m_fin(ii)=0;
            end
        end

        %aproximacion inicial
        if s_m_fin~=0
            a_0=s_a_fin(ii-1)-10;
```

```
            b_0=s_a_fin(ii-1)+10;
        end

        %Bolzano
        tol_biseccion=tolerancia*50;
        [x_0] =biseccion_findley_ax_alterna(alfa_fin,s_m_fin(ii),beta_fin,a_0,b_0,tol_biseccion);

        %Newton-Raphson
        s_a_fin(ii)=newton_raphson_findley_ax_alt(alfa_fin,s_m_fin(ii),beta_fin,x_0,tolerancia);

    end

    % Cuando S_m/s_ut=1 y la curva no es horizontal, se calculan puntos hasta que la funcion corte la
horizontal (nuevo_numero_puntos)
    diferen=abs(s_a_fin(1)-s_a_fin(puntos_discretizacion));
    if diferen>10 && s_a_fin(puntos_discretizacion)>0
        entra_fin=1;
        %solo en este caso, se va a buscar el valor de la S_m
        a_0=s_m_fin(puntos_discretizacion)-10;
        b_0=s_m_fin(puntos_discretizacion)+10;

        %Bolzano
        tol_biseccion=tolerancia*50;
        [x_0] = biseccion_findley_ax_media(alfa_fin,beta_fin,a_0,b_0,tol_biseccion);

        %Newton-Raphson
        s_m_final=double(newton_raphson_findley_ax_m(alfa_fin,beta_fin,x_0,tolerancia));

        %Se calculan los nuevos puntos findley
        s_rep=s_m_final-s_ut;

        if mod(s_rep,avance)==0
           puntos_fin=round(s_rep/avance); % no es +1 porque el punto de S_m/s_ut=1 ya esta
        else
            puntos_fin=round(s_rep/avance)+1;
        end

        zer=zeros(puntos_fin,1);
        s_m_fin=[s_m_fin ; zer];
        s_a_fin=[s_a_fin ; zer];

        puntos_total=(puntos_discretizacion)+puntos_fin;
        for ii=(puntos_discretizacion):puntos_total

            if ii==puntos_total
                s_m_fin(ii)=s_m_final;
                s_a_fin(ii)=0;
            else
                s_m_fin(ii)=s_m_fin(ii-1)+avance;

                %aproximacion inicial
                if s_m_fin(ii)~=0
                    a_0=s_a_fin(ii-1)-10;
                    b_0=s_a_fin(ii-1)+10;
                end

                %Bolzano
                tol_biseccion=tolerancia*50;
                [x_0] =biseccion_findley_ax_alterna(alfa_fin,s_m_fin(ii),beta_fin,a_0,b_0,tol_biseccion);

                %Newton-Raphson
                s_a_fin(ii)=newton_raphson_findley_ax_alt(alfa_fin,s_m_fin(ii),beta_fin,x_0,tolerancia);

            end

        end
        if puntos_fin==0
            entra_fin=0;
        end
    end

    %Por último, se normalizan los datos
        %Eje vertical: S_a/S_e
        %eje horizontal: S_m/s_ut
```

```
        s_m_fin=s_m_fin/s_ut;
        s_a_fin=s_a_fin/lf_ax_alt;
end
```

"calcula_findley_torsion"

```
function [t_m_fin,t_a_fin,entra_fin,puntos_fin] =
calcula_findley_torsion(puntos_discretizacion,t_ut,avance_t,lf_tors_alt,tolerancia,alfa_fin,beta_fin)
% Calcula findley

    puntos_fin=0;

    t_m_fin=zeros(puntos_discretizacion,1); %Parte de traccion y compresion
    t_a_fin=zeros(puntos_discretizacion,1);

    a_0=lf_tors_alt-10;
    b_0=lf_tors_alt+10;

    %Parte traccion y compresión
    for ii=1:puntos_discretizacion
        entra_fin=0;
        if ii==1
            t_m_fin(ii)=-t_ut;
        elseif ii==(puntos_discretizacion)
            t_m_fin(ii)=t_ut;
        else
            t_m_fin(ii)=t_m_fin(ii-1)+avance_t;
        end

        if ii~=1 && t_m_fin(ii)~=0
            if t_m_fin(ii-1)<0 && t_m_fin(ii)>0
                t_m_fin(ii)=0;
            end
        end

        %aproximacion inicial
        if t_m_fin~=0
            a_0=t_a_fin(ii-1)-10;
            b_0=t_a_fin(ii-1)+10;
        end

        %Bolzano
        tol_biseccion=tolerancia*50;
        [x_0] = biseccion_findley_tor_alterna(alfa_fin,t_m_fin(ii),beta_fin,a_0,b_0,tol_biseccion);

        %Newton-Raphson
        t_a_fin(ii)=newton_raphson_findley_tor_alt(alfa_fin,t_m_fin(ii),beta_fin,x_0,tolerancia);

    end

    % Cuando t_m/t_ut=1 y la curva no es horizontal, se calculan puntos hasta que la funcion corte la
horizontal (nuevo_numero_puntos)
    diferen=abs(t_a_fin(1)-t_a_fin(puntos_discretizacion));
    if diferen>10 && t_a_fin(puntos_discretizacion)>0
        entra_fin=1;
        %solo en este caso, se va a buscar el valor de la t_m
        a_0=t_m_fin(puntos_discretizacion)-10;
        b_0=t_m_fin(puntos_discretizacion)+10;


        %Bolzano
        tol_biseccion=tolerancia*50;
        [x_0] = biseccion_findley_tor_media(alfa_fin,beta_fin,a_0,b_0,tol_biseccion);

        %Newton-Raphson
        t_m_final=double(newton_raphson_findley_tor_m(alfa_fin,beta_fin,x_0,tolerancia));

        %Se calculan los nuevos puntos findley
        s_rep=t_m_final-t_ut;

        if mod(s_rep,avance_t)==0
           puntos_fin=round(s_rep/avance_t); %+1; no es +1 porque el punto de t_m/t_ut=1 ya esta
```

```
        else
            puntos_fin=round(s_rep/avance_t)+1; %+2;

        end

        zer=zeros(puntos_fin,1);
        t_m_fin=[t_m_fin ; zer];
        t_a_fin=[t_a_fin ; zer];

        puntos_total=(puntos_discretizacion)+puntos_fin;


        for ii=(puntos_discretizacion):puntos_total

            if ii==puntos_total
                t_m_fin(ii)=t_m_final;
                t_a_fin(ii)=0;
            else
                t_m_fin(ii)=t_m_fin(ii-1)+avance_t;

                %aproximacion inicial
                if t_m_fin(ii)~=0
                    a_0=t_a_fin(ii-1)-10;
                    b_0=t_a_fin(ii-1)+10;
                end

                %Bolzano
                tol_biseccion=tolerancia*50;
                [x_0] = biseccion_findley_tor_alterna(alfa_fin,t_m_fin(ii),beta_fin,a_0,b_0,tol_biseccion);

                %Newton-Raphson
                t_a_fin(ii)=newton_raphson_findley_tor_alt(alfa_fin,t_m_fin(ii),beta_fin,x_0,tolerancia);

            end

        end
        if puntos_fin==0
            entra_fin=0;
        end
    end

    %Por último, se normalizan los datos
        %Eje vertical: t_a/S_e
         %eje horizontal: t_m/t_ut
         t_m_fin=t_m_fin/t_ut;
         t_a_fin=t_a_fin/lf_tors_alt;
end
```

"calcula_matake_axial"

```
function [s_m_mat,s_a_mat,entra_mat,puntos_mat] =
calcula_matake_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa_mat,beta_mat)
% Calcula matake

    puntos_mat=0;

    s_m_mat=zeros(puntos_discretizacion,1);
    s_a_mat=zeros(puntos_discretizacion,1);

    a_0=lf_ax_alt-10;
    b_0=lf_ax_alt+10;

    %Parte traccion y compresión
    for ii=1:puntos_discretizacion
        entra_mat=0;
        if ii==1
            s_m_mat(ii)=-s_ut;
        elseif ii==(puntos_discretizacion)
            s_m_mat(ii)=s_ut;
        else
            s_m_mat(ii)=s_m_mat(ii-1)+avance;
        end
```

```
        if ii~=1 && s_m_mat(ii)~=0
            if s_m_mat(ii-1)<0 && s_m_mat(ii)>0
                s_m_mat(ii)=0;
            end
        end

        %aproximacion inicial
        if s_m_mat~=0
            a_0=s_a_mat(ii-1)-10;
            b_0=s_a_mat(ii-1)+10;
        end

        %Bolzano
        tol_biseccion=tolerancia*50;
        [x_0] = biseccion_matake_alterna(alfa_mat,s_m_mat(ii),beta_mat,a_0,b_0,tol_biseccion);

        %Newton-Raphson
        s_a_mat(ii)=newton_raphson_matake_alt(alfa_mat,s_m_mat(ii),beta_mat,x_0,tolerancia);

    end

    % Cuando S_m/s_ut=1 y la curva no es horizontal, se calculan puntos hasta que la funcion corte la
horizontal (nuevo_numero_puntos)
    diferen=abs(s_a_mat(1)-s_a_mat(puntos_discretizacion));
    if diferen>10 && s_a_mat(puntos_discretizacion)>0
        entra_mat=1;
        %solo en este caso, se va a buscar el valor de la S_m
        a_0=s_m_mat(puntos_discretizacion)-10;
        b_0=s_m_mat(puntos_discretizacion)+10;


        %Bolzano
        tol_biseccion=tolerancia*50;
        [x_0] = biseccion_matake_media(alfa_mat,beta_mat,a_0,b_0,tol_biseccion);

        %Newton-Raphson
        s_m_final=double(newton_raphson_matake_m(alfa_mat,beta_mat,x_0,tolerancia));

        %Se calculan los nuevos puntos matake
        s_rep=s_m_final-s_ut;

        if mod(s_rep,avance)==0
           puntos_mat=round(s_rep/avance); %+1; no es +1 porque el punto de S_m/s_ut=1 ya esta
        else
            puntos_mat=round(s_rep/avance)+1; %+2;
        end

        zer=zeros(puntos_mat,1);
        s_m_mat=[s_m_mat ; zer];
        s_a_mat=[s_a_mat ; zer];

        puntos_total=(puntos_discretizacion)+puntos_mat;
        for ii=(puntos_discretizacion):puntos_total

            if ii==puntos_total
                s_m_mat(ii)=s_m_final;
                s_a_mat(ii)=0;
            else
                s_m_mat(ii)=s_m_mat(ii-1)+avance;

                %aproximacion inicial
                if s_m_mat(ii)~=0
                    a_0=s_a_mat(ii-1)-10;
                    b_0=s_a_mat(ii-1)+10;
                end

                %Bolzano
                tol_biseccion=tolerancia*50;
                [x_0] = biseccion_matake_alterna(alfa_mat,s_m_mat(ii),beta_mat,a_0,b_0,tol_biseccion);

                %Newton-Raphson
                s_a_mat(ii)=newton_raphson_matake_alt(alfa_mat,s_m_mat(ii),beta_mat,x_0,tolerancia);
```

```
            end

        end
        if puntos_mat==0
            entra_mat=0;
        end
    end

    %Por último, se normalizan los datos
        %Eje vertical: S_a/S_e
        %eje horizontal: S_m/s_ut
        s_m_mat=s_m_mat/s_ut;
        s_a_mat=s_a_mat/lf_ax_alt;
end
```

"calcula_papuga_torsion"

```
function [t_m_pap,t_a_pap,entra_t_pap] =
calcula_papuga_torsion(puntos_discretizacion,t_ut,avance_t,lf_ax_alt,lf_tors_alt,lf_tr_pul,tolerancia,a_p,b_p,
delta_angulo)
%Calcula Papuga a torsioón

    entra_t_pap=0;

    t_m_pap=zeros(puntos_discretizacion,1); %Parte de traccion y compresion
    t_a_pap=zeros(puntos_discretizacion,1);

    for ii=1:puntos_discretizacion
        if ii==1
            t_m_pap(ii)=-t_ut;
        elseif ii==(puntos_discretizacion)
            t_m_pap(ii)=t_ut;
        else
            t_m_pap(ii)=t_m_pap(ii-1)+avance_t;
        end
        if ii~=1 && t_m_pap(ii)~=0
            if t_m_pap(ii-1)<0 && t_m_pap(ii)>0
                t_m_pap(ii)=0;
            end
        end
    end

    %% Buscar el plano crítico para cada valor de t_m

    for jj=1:puntos_discretizacion
        dos_theta=0;
        ii=0;
        tension_alterna_min=lf_tors_alt*100;

        while dos_theta<=(180-delta_angulo)
            ii=ii+1;
            %aproximacion inicial
                a_0=lf_tors_alt-10;
                b_0=lf_tors_alt+10;

            %Se define la funcion
            % sumando 1 --> sum_1=(x^2)*sind(dos_theta)^2;
            % sumando 2 --> sum_2=(x)*cosd(dos_theta);
            % sumando 3 --> sum_3=((lf_tors_alt*tension_media)/(2*lf_tr_pul))*(cosd(dos_theta));

            %Bolzano
            tol_biseccion=tolerancia*50;
            [x_0] =
biseccion_papuga_tor_alterna(a_p,t_m_pap(jj),b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,a_0,b_0,tol_bisecci
on);
            %Newton-Raphson

tension_alterna=newton_raphson_papuga_tor_alt(a_p,t_m_pap(jj),b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,x_
0,tolerancia);

            if ii==1 && tension_alterna>1
                tension_alterna_min=tension_alterna;
```

```
                t_a_pap(jj)=tension_alterna;
            elseif tension_alterna<tension_alterna_min && tension_alterna>1
                tension_alterna_min=tension_alterna;
                %plano_critico=dos_theta; %Angulo donde esta el plano critico
                t_a_pap(jj)=tension_alterna;
            end

            dos_theta=dos_theta+delta_angulo;
        end
    end


    tension_media_min=t_ut*1000;
    % Cuando S_m/t_ut=1 y la curva no es horizontal, se calculan puntos hasta que la funcion corte la
horizontal (nuevo_numero_puntos)
    diferen=abs(t_a_pap(1)-t_a_pap(puntos_discretizacion));
    if diferen>10 && t_a_pap(puntos_discretizacion)>0
        entra_t_pap=1;
        %solo en este caso, se va a buscar el valor de la t_m
        dos_theta=0;
        ii=0;

        while dos_theta<=(180-delta_angulo)
        ii=ii+1;

            %aproximacion inicial
            a_0=t_m_pap(puntos_discretizacion)-10;
            b_0=t_m_pap(puntos_discretizacion)+10;

            %Bolzano
            tol_biseccion=tolerancia*50;
            [x_0] =
biseccion_papuga_tor_media(a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,a_0,b_0,tol_biseccion);
            %Newton-Raphson

tension_media=newton_raphson_papuga_tor_m(a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,x_0,tolerancia);

            if ii==1 && tension_media>1
                tension_media_min=tension_media;
                s_m_final=tension_media;
            elseif tension_media<tension_media_min && tension_media>1
                tension_media_min=tension_media;
                %plano_critico=dos_theta; %Angulo donde esta el plano critico
                s_m_final=tension_media;
            end

            dos_theta=dos_theta+delta_angulo;
        end

        %Se calculan los nuevos puntos papuga
        s_rep=s_m_final-t_ut;
        if mod(s_rep,avance_t)==0
            puntos_fin=round(s_rep/avance_t); %+1; no es +1 porque el punto de S_m/t_ut=1 ya esta
        else
            puntos_fin=round(s_rep/avance_t)+1; %+2;
        end

        zer=zeros(puntos_fin,1);
        t_m_pap=[t_m_pap ; zer];
        t_a_pap=[t_a_pap ; zer];

        puntos_total=(puntos_discretizacion)+puntos_fin;
        tension_alterna_min=lf_tors_alt*100;

        t_m_pap(puntos_total)=s_m_final;
        t_a_pap(puntos_total)=0;

        for jj=(puntos_discretizacion+1):(puntos_total-1)
            dos_theta=0;
            ii=0;

            t_m_pap(jj)=t_m_pap(jj-1)+avance_t;
            if t_m_pap(jj)>s_m_final
                t_m_pap(jj)=(0.5)*(t_m_pap(jj-1)+t_m_pap(jj+1));
```

```
            end

            while dos_theta<=(180-delta_angulo)
                ii=ii+1;
                %aproximacion inicial
                    a_0=t_a_pap(jj-1)-10;
                    b_0=t_a_pap(jj-1)+10;

                    %Bolzano
                    tol_biseccion=tolerancia*50;
                    [x_0] =
biseccion_papuga_tor_alterna(a_p,t_m_pap(jj),b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,a_0,b_0,tol_bisecci
on);
                    %Newton-Raphson

tension_alterna=newton_raphson_papuga_tor_alt(a_p,t_m_pap(jj),b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,x_
0,tolerancia);

                if ii==1 && tension_alterna>1
                    tension_alterna_min=tension_alterna;
                    t_a_pap(jj)=tension_alterna;
                elseif tension_alterna<tension_alterna_min && tension_alterna>1
                    tension_alterna_min=tension_alterna;
                    %plano_critico=dos_theta; %Angulo donde esta el plano critico
                    t_a_pap(jj)=tension_alterna;
                end

                dos_theta=dos_theta+delta_angulo;
            end
        end
        if puntos_fin==0
            entra_t_pap=0;
        end
    end

    %Por último, se normalizan los datos
        %Eje vertical: S_a/S_e
        %eje horizontal: S_m/t_ut
         t_m_pap=t_m_pap/t_ut;
         t_a_pap=t_a_pap/lf_tors_alt;

end
```

"calcula_papuga_uniaxial"

```
function [s_m_pap,s_a_pap,entra_pap] =
calcula_papuga_uniaxial(puntos_discretizacion,s_ut,avance,lf_ax_alt,lf_tors_alt,lf_tr_pul,tolerancia,a_p,b_p,d
elta_angulo)
%Calcula Papuga Uniaxial

    entra_pap=0;

    s_m_pap=zeros(puntos_discretizacion,1); %Parte de traccion y compresion
    s_a_pap=zeros(puntos_discretizacion,1);

    for ii=1:puntos_discretizacion
        if ii==1
            s_m_pap(ii)=-s_ut;
        elseif ii==(puntos_discretizacion)
            s_m_pap(ii)=s_ut;
        else
            s_m_pap(ii)=s_m_pap(ii-1)+avance;
        end
        if ii~=1 && s_m_pap(ii)~=0
            if s_m_pap(ii-1)<0 && s_m_pap(ii)>0
                s_m_pap(ii)=0;
            end
        end
    end

    %% Buscar el plano crítico para cada valor de s_m
```

```
    for jj=1:puntos_discretizacion

        dos_theta=0;
        ii=0;
        tension_alterna_min=lf_ax_alt*100;
        while dos_theta<=(180-delta_angulo)

            ii=ii+1;
            %aproximacion inicial
                a_0=lf_ax_alt-10;
                b_0=lf_ax_alt+10;

            %Se define la funcion
            % sumando 1 --> sum_1=(x^2/4)*sind(dos_theta)^2;
            % sumando 2 --> sum_2=(x/2)*(1+cosd(dos_theta));
            % sumando 3 --> sum_3=((lf_tors_alt*tension_media)/(4*lf_tr_pul))*(1+cosd(dos_theta));

            %Bolzano
            tol_biseccion=tolerancia*50;
            [x_0] =
biseccion_papuga_ax_alterna(a_p,s_m_pap(jj),b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,a_0,b_0,tol_biseccio
n);
            %Newton-Raphson

tension_alterna=newton_raphson_papuga_ax_alt(a_p,s_m_pap(jj),b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,x_0
,tolerancia);

            if ii==1 && tension_alterna>1
                tension_alterna_min=tension_alterna;
                s_a_pap(jj)=tension_alterna;
            elseif tension_alterna<tension_alterna_min && tension_alterna>1
                tension_alterna_min=tension_alterna;
                %plano_critico=dos_theta; %Angulo donde esta el plano critico
                s_a_pap(jj)=tension_alterna;
            end

            dos_theta=dos_theta+delta_angulo;
        end
    end

    tension_media_min=s_ut*1000;
    % Cuando S_m/s_ut=1 y la curva no es horizontal, se calculan puntos hasta que la funcion corte la
horizontal (nuevo_numero_puntos)
    diferen=abs(s_a_pap(1)-s_a_pap(puntos_discretizacion));
    if diferen>10 && s_a_pap(puntos_discretizacion)>0
        entra_pap=1;
        %solo en este caso, se va a buscar el valor de la S_m
        dos_theta=0;
        ii=0;

        while dos_theta<=(180-delta_angulo)

            ii=ii+1;
            %aproximacion inicial
            a_0=s_m_pap(puntos_discretizacion)-10;
            b_0=s_m_pap(puntos_discretizacion)+10;

            %Bolzano
            tol_biseccion=tolerancia*50;
            [x_0] =
biseccion_papuga_ax_media(a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,a_0,b_0,tol_biseccion);
            %Newton-Raphson

tension_media=newton_raphson_papuga_ax_m(a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,x_0,tolerancia);

            if ii==1 && tension_media>1
                tension_media_min=tension_media;
                s_m_final=tension_media;
            elseif tension_media<tension_media_min && tension_media>1
                tension_media_min=tension_media;
                %plano_critico=dos_theta; %Angulo donde esta el plano critico
                s_m_final=tension_media;
            end
```

```
            dos_theta=dos_theta+delta_angulo;
        end

        %Se calculan los nuevos puntos papuga
        s_rep=s_m_final-s_ut;
        if mod(s_rep,avance)==0
            puntos_fin=round(s_rep/avance); %+1; no es +1 porque el punto de S_m/s_ut=1 ya esta
        else
            puntos_fin=round(s_rep/avance)+1; %+2;
        end
        zer=zeros(puntos_fin,1);
        s_m_pap=[s_m_pap ; zer];
        s_a_pap=[s_a_pap ; zer];

        puntos_total=(puntos_discretizacion)+puntos_fin;
        tension_alterna_min=lf_ax_alt*100;

        s_m_pap(puntos_total)=s_m_final;
        s_a_pap(puntos_total)=0;

        for jj=(puntos_discretizacion+1):(puntos_total-1)
            dos_theta=0;
            ii=0;

            s_m_pap(jj)=s_m_pap(jj-1)+avance;
            if s_m_pap(jj)>s_m_final
                s_m_pap(jj)=(0.5)*(s_m_pap(jj-1)+s_m_pap(jj+1));
            end

            while dos_theta<=(180-delta_angulo)
                ii=ii+1;
                %aproximacion inicial
                    a_0=s_a_pap(jj-1)-10;
                    b_0=s_a_pap(jj-1)+10;

                %Bolzano
                tol_biseccion=tolerancia*50;
                [x_0] =
biseccion_papuga_ax_alterna(a_p,s_m_pap(jj),b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,a_0,b_0,tol_biseccio
n);
                %Newton-Raphson

tension_alterna=newton_raphson_papuga_ax_alt(a_p,s_m_pap(jj),b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,x_0
,tolerancia);

                if ii==1 && tension_alterna>1
                    tension_alterna_min=tension_alterna;
                    s_a_pap(jj)=tension_alterna;
                elseif tension_alterna<tension_alterna_min && tension_alterna>1
                    tension_alterna_min=tension_alterna;
                    %plano_critico=dos_theta; %Angulo donde esta el plano critico
                    s_a_pap(jj)=tension_alterna;
                end

                dos_theta=dos_theta+delta_angulo;
            end
        end
        if puntos_fin==0
            entra_pap=0;
        end
    end

    %Por último, se normalizan los datos
        %Eje vertical: S_a/S_e
        %eje horizontal: S_m/s_ut
        s_m_pap=s_m_pap/s_ut;
        s_a_pap=s_a_pap/lf_ax_alt;

end
```

"calcula_sines_axial"

```
function [s_m_sin,s_a_sin,entra_sin,puntos_sin] =
calcula_sines_axial(puntos_discretizacion,s_ut,avance,lf_ax_alt,tolerancia,alfa_sin,beta_sin)
% Calcula sines

    puntos_sin=0;

    s_m_sin=zeros(puntos_discretizacion,1); %Parte de traccion y compresion
    s_a_sin=zeros(puntos_discretizacion,1);

    a_0=lf_ax_alt-10;
    b_0=lf_ax_alt+10;

    %Parte traccion y compresión
    for ii=1:puntos_discretizacion
        entra_sin=0;
        if ii==1
            s_m_sin(ii)=-s_ut;
        elseif ii==(puntos_discretizacion)
            s_m_sin(ii)=s_ut;
        else
            s_m_sin(ii)=s_m_sin(ii-1)+avance;
        end

        if ii~=1 && s_m_sin(ii)~=0
            if s_m_sin(ii-1)<0 && s_m_sin(ii)>0
                s_m_sin(ii)=0;
            end
        end

        %aproximacion inicial
        if s_m_sin~=0
            a_0=s_a_sin(ii-1)-10;
            b_0=s_a_sin(ii-1)+10;
        end


        %Bolzano
        tol_biseccion=tolerancia*50;
        [x_0] = biseccion_sines_alterna(alfa_sin,beta_sin,s_m_sin(ii),a_0,b_0,tol_biseccion);

        %Newton-Raphson
        s_a_sin(ii)=newton_raphson_sines_alt(alfa_sin,beta_sin,s_m_sin(ii),x_0,tolerancia);

    end

    % Cuando S_m/s_ut=1 y la curva no es horizontal, se calculan puntos hasta que la funcion corte la
horizontal (nuevo_numero_puntos)
    diferen=abs(s_a_sin(1)-s_a_sin(puntos_discretizacion));
    if diferen>=10 && s_a_sin(puntos_discretizacion)>0
        entra_sin=1;
        %solo en este caso, se va a buscar el valor de la S_m
        a_0=s_m_sin(puntos_discretizacion)-10;
        b_0=s_m_sin(puntos_discretizacion)+10;

        %Bolzano
        tol_biseccion=tolerancia*50;
        [x_0] = biseccion_sines_media(alfa_sin,beta_sin,a_0,b_0,tol_biseccion);

        %Newton-Raphson
        s_m_final=double(newton_raphson_sines_m(alfa_sin,beta_sin,x_0,tolerancia));

        %Se calculan los nuevos puntos sines
        s_rep=s_m_final-s_ut;

        if mod(s_rep,avance)==0
           puntos_sin=round(s_rep/avance); %+1; no es +1 porque el punto de S_m/s_ut=1 ya esta
        else
            puntos_sin=round(s_rep/avance)+1; %+2;
        end

        zer=zeros(puntos_sin,1);
```

```
        s_m_sin=[s_m_sin ; zer];
        s_a_sin=[s_a_sin ; zer];

        puntos_total=(puntos_discretizacion)+puntos_sin;
        for ii=(puntos_discretizacion):puntos_total

            if ii==puntos_total
                s_m_sin(ii)=s_m_final;
                s_a_sin(ii)=0;
            else
                s_m_sin(ii)=s_m_sin(ii-1)+avance;

                %aproximacion inicial
                if s_m_sin(ii)~=0
                    a_0=s_a_sin(ii-1)-10;
                    b_0=s_a_sin(ii-1)+10;
                end

                %Bolzano
                tol_biseccion=tolerancia*50;
                [x_0] = biseccion_sines_alterna(alfa_sin,beta_sin,s_m_sin(ii),a_0,b_0,tol_biseccion);

                %Newton-Raphson
                s_a_sin(ii)=newton_raphson_sines_alt(alfa_sin,beta_sin,s_m_sin(ii),x_0,tolerancia);

            end

        end
        if puntos_sin==0
            entra_sin=0;
        end
    end

    %Por último, se normalizan los datos
        %Eje vertical: S_a/S_e
        %eje horizontal: S_m/s_ut
        s_m_sin=s_m_sin/s_ut;
        s_a_sin=s_a_sin/lf_ax_alt;
end
```

"comprueba_datos"

```
function [datos_uniaxial,continuar_uniaxial,datos_torsion,continuar_torsion] =
comprueba_datos(NUM,columna)
%Comprueba si todos los datos introducidos son numéricos y reales

    lim_inf_uniax=columna-1;
    lim_sup_uniax=columna;
    lim_inf_tors=columna+1;
    lim_sup_tors=columna+2;

    continuar_uniaxial=1;
    continuar_torsion=1;
    datos_uniaxial=0;
    datos_torsion=0;

    if isnan(NUM(5,columna))==1 || isnan(NUM(7,columna))==1
        continuar_uniaxial=0;
    else
        datos_para_uniax=NUM(9:end,lim_inf_uniax:lim_sup_uniax);
        pos_falta_dato=find(isnan(datos_para_uniax));
        [fil,~]=ind2sub(size(datos_para_uniax),pos_falta_dato);
        if isempty(fil)==1
            datos_uniaxial=datos_para_uniax;
        else
            if fil(1)~=1
                f_i_u=1;
                f_s_u=fil(1)-1;
                datos_uniaxial=datos_para_uniax(f_i_u:f_s_u,:);

                tam=size(datos_uniaxial);
                n=tam(1);
                for cc=1:2
```

```
                    for ff=1:n
                       if  isnan(datos_uniaxial(ff,cc))==1
                           continuar_uniaxial=0;
                           ff=n;
                           cc=2; %Para salir del bucle
                       end
                    end
                end

            else
                continuar_uniaxial=0;
            end
        end
    end


    if isnan(NUM(6,columna))==1 || isnan(NUM(7,columna))==1
        continuar_torsion=0;
    else

        tam=size(NUM);
        if lim_sup_tors>tam(2)
            continuar_torsion=0;
        else

            datos_para_tors=NUM(9:end,lim_inf_tors:lim_sup_tors);
            pos_falta_dato=find(isnan(datos_para_tors));
            [fil,~]=ind2sub(size(datos_para_tors),pos_falta_dato);
            if isempty(fil)==1
                datos_torsion=datos_para_tors;
            else
                if fil(1)~=1
                    f_i_t=1;
                    f_s_t=fil(1)-1;
                    datos_torsion=datos_para_tors(f_i_t:f_s_t,:);

                    tam=size(datos_torsion);
                    n=tam(1);
                    for cc=1:2
                        for ff=1:n
                           if  isnan(datos_torsion(ff,cc))==1
                               continuar_torsion=0;
                               ff=n;
                               cc=2; %Para salir del bucle
                           end
                        end
                    end

                else
                    continuar_torsion=0;
                end
            end
        end
    end

    if continuar_uniaxial==1
        s_ut=NUM(7,columna);
        t_ut=s_ut*0.75;
        lf_ax_alt=NUM(5,columna);
        datos_uniaxial(:,1)=datos_uniaxial(:,1)/s_ut;
        datos_uniaxial(:,2)=datos_uniaxial(:,2)/lf_ax_alt;
    end

    if continuar_torsion==1
        s_ut=NUM(7,columna);
        t_ut=s_ut*0.75;
        lf_tors_alt=NUM(6,columna);
        datos_torsion(:,1)=datos_torsion(:,1)/t_ut;
        datos_torsion(:,2)=datos_torsion(:,2)/lf_tors_alt;
    end

end
```

"funcion_crossland_alterna"

```
function [resul] = funcion_crossland_alterna(x_a,alfa_cross,s_m_cross,beta_cross)
%Evalúa la funcion de crossland
resul=sqrt(2)/3*x_a+alfa_cross/3*(s_m_cross+x_a)-beta_cross; %Se deja la ecuacion en funcion de "x" (tension
alterna)

end
```

"funcion_crossland_media"

```
function [resul] = funcion_crossland_media(x_m,alfa_cross,beta_cross)
%Evalúa la funcion de crossland
    %funcion=sqrt(2)/3*x+alfa_cross/3*(s_m_cross(ii)+x)-beta_cross; %Se sustituye s_a=0
    resul=sqrt(2)/3*0+alfa_cross/3*(x_m+0)-beta_cross;
end
```

"funcion_dangvan_alterna"

```
function [resul] = funcion_dangvan_alterna(x_a,alfa_dv,s_m_dv,beta_dv)
%Evalúa la funcion de Dang Van alterna
resul=0.5*x_a+alfa_dv*(s_m_dv+x_a)/3-beta_dv;

end
```

"funcion_dangvan_media"

```
function [resul] = funcion_dangvan_media(x_m,alfa_dv,beta_dv)
%Evalúa la funcion de Dang Van media
    resul=0.5*0+alfa_dv*(x_m+0)/3-beta_dv;
end
```

"funcion_findley_ax_alterna"

```
function [resul] = funcion_findley_ax_alterna(x_a,alfa_fin,s_m_fin,beta_fin)
%Evalúa la funcion de Findley axial alterna
resul=x_a^2+4*beta_fin*alfa_fin*(s_m_fin+x_a)-4*beta_fin^2;

end
```

"funcion_findley_ax_media"

```
function [resul] = funcion_findley_ax_media(x_m,alfa_fin,beta_fin)
%Evalúa la funcion de Findley axial media
    resul=0^2+4*beta_fin*alfa_fin*(x_m+0)-4*beta_fin^2;
end
```

"funcion_findley_tor_alterna"

```
function [resul] = funcion_findley_tor_alterna(x_a,alfa_fin,t_m_fin,beta_fin)
%Evalúa la funcion de Findley torsion alterna
resul=x_a^2*(alfa_fin^2+1)+x_a*(2*alfa_fin^2*t_m_fin)+t_m_fin^2*alfa_fin^2-beta_fin^2;
end
```

"funcion_findley_tor_media"

```
function [resul] = funcion_findley_tor_media(x_m,alfa_fin,beta_fin)
```

```
%Evalúa la funcion de Findley torsion media
    resul=0*(alfa_fin^2+1)+0*(2*alfa_fin^2*x_m)+x_m^2*alfa_fin^2-beta_fin^2;
end
```

"funcion_findley_zer"

```
function [resul] = funcion_findley_zer(x_a,lf_tr_pul,lf_ax_alt)
%Evalúa la funcion de Sines alterna
resul=lf_tr_pul*(sqrt((2*x_a)^2+1)+2*x_a)-lf_ax_alt*(sqrt(x_a^2+1)+x_a);

end
```

"funcion_matake_alterna"

```
function [resul] = funcion_matake_alterna(x_a,alfa_mat,s_m_mat,beta_mat)
%Evalúa la funcion de Matake alterna
resul=0.5*x_a+alfa_mat*(s_m_mat+x_a)/2-beta_mat;

end
```

"funcion_matake_media"

```
function [resul] = funcion_matake_media(x_m,alfa_mat,beta_mat)
%Evalúa la funcion de Matake media
    resul=0.5*0+alfa_mat*(x_m+0)/2-beta_mat;
end
```

"funcion_papuga_ax_alterna"

```
function [resul] = funcion_papuga_ax_alterna(x_a,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,s_m_pap)
%Evalúa la funcion de Papuga axial alterna
resul=a_p*((x_a^2/4)*sind(dos_theta)^2) + b_p*((x_a/2)*(1+cosd(dos_theta)) +
((lf_tors_alt*s_m_pap)/(4*lf_tr_pul))*(1+cosd(dos_theta)) ) - lf_ax_alt^2;

end
```

"funcion_papuga_ax_media"

```
function [resul] = funcion_papuga_ax_media(x_m,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt)
%Evalúa la funcion de Papuga axial media
    resul=a_p*((0^2/4)*sind(dos_theta)^2) + b_p*((0/2)*(1+cosd(dos_theta)) +
((lf_tors_alt*x_m)/(4*lf_tr_pul))*(1+cosd(dos_theta)) ) - lf_ax_alt^2;
end
```

"funcion_papuga_tor_alterna"

```
function [resul] = funcion_papuga_tor_alterna(x_a,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,t_m_pap)
%Evalúa la funcion de Papuga torsión alterna
resul=a_p*((x_a^2)*sind(dos_theta)^2) + b_p*((x_a)*cosd(dos_theta) +
((lf_tors_alt*t_m_pap)/(2*lf_tr_pul))*(cosd(dos_theta)) ) - lf_ax_alt^2;

end
```

"funcion_papuga_tor_media"

```
function [resul] = funcion_papuga_tor_media(x_m,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt)
%Evalúa la funcion de Papuga torsión media
    resul=a_p*((0^2)*sind(dos_theta)^2) + b_p*((0)*cosd(dos_theta) +
((lf_tors_alt*x_m)/(2*lf_tr_pul))*(cosd(dos_theta)) ) - lf_ax_alt^2;
end
```

"funcion_sines_alterna"

```
function [resul] = funcion_sines_alterna(x_a,alfa_sin,beta_sin,s_m_sin)
%Evalúa la funcion de Sines alterna
resul=sqrt(2)/3*x_a+alfa_sin/3*s_m_sin-beta_sin;

end
```

"funcion_sines_media"

```
function [resul] = funcion_sines_media(x_m,alfa_sin,beta_sin)
%Evalúa la funcion de Sines media
    resul=sqrt(2)/3*0+alfa_sin/3*x_m-beta_sin;
end
```

"newton_raphson_crossland_alt"

```
function [y] = newton_raphson_crossland_alt(alfa_cross,s_m_cross,beta_cross,x_0,tolerancia)
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"


    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
    cont=0;
    error_relativo=tolerancia+1;
    while error_relativo>=tolerancia && cont<=5

        sus_x0_en_f=double(funcion_crossland_alterna(x_0,alfa_cross,s_m_cross,beta_cross));
        x_eps=x_0+epsilon;
        sus_x0_en_df=double((funcion_crossland_alterna(x_eps,alfa_cross,s_m_cross,beta_cross)-
funcion_crossland_alterna(x_0,alfa_cross,s_m_cross,beta_cross))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```

"newton_raphson_crossland_m"

```
function [y] = newton_raphson_crossland_m(alfa_cross,beta_cross,x_0,tolerancia)
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"

    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
    cont=0;
    error_relativo=tolerancia+1;
    while error_relativo>=tolerancia && cont<=5

        sus_x0_en_f=double(funcion_crossland_media(x_0,alfa_cross,beta_cross));
        x_eps=x_0+epsilon;
        sus_x0_en_df=double((funcion_crossland_media(x_eps,alfa_cross,beta_cross)-
funcion_crossland_media(x_0,alfa_cross,beta_cross))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```

"newton_raphson_dangvan_alt"

```
function [y] = newton_raphson_dangvan_alt(alfa_dv,s_m_dv,beta_dv,x_0,tolerancia)
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"
```

```
    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
    cont=0;
    error_relativo=tolerancia+1;
    while error_relativo>=tolerancia && cont<=5

        sus_x0_en_f=double(funcion_dangvan_alterna(x_0,alfa_dv,s_m_dv,beta_dv));
        x_eps=x_0+epsilon;
        sus_x0_en_df=double((funcion_dangvan_alterna(x_eps,alfa_dv,s_m_dv,beta_dv)-
funcion_dangvan_alterna(x_0,alfa_dv,s_m_dv,beta_dv))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```

"newton_raphson_dangvan_m"

```
function [y] = newton_raphson_dangvan_m(alfa_dv,beta_dv,x_0,tolerancia)
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"

    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
    cont=0;
    error_relativo=tolerancia+1;
    while error_relativo>=tolerancia && cont<=5

        sus_x0_en_f=double(funcion_dangvan_media(x_0,alfa_dv,beta_dv));
        x_eps=x_0+epsilon;
        sus_x0_en_df=double((funcion_dangvan_media(x_eps,alfa_dv,beta_dv)-
funcion_dangvan_media(x_0,alfa_dv,beta_dv))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```

"newton_raphson_findley_ax_alt"

```
function [y] = newton_raphson_findley_ax_alt(alfa_fin,s_m_fin,beta_fin,x_0,tolerancia)
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"

    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
    cont=0;
    error_relativo=tolerancia+1;
    while error_relativo>=tolerancia && cont<=5

        sus_x0_en_f=double(funcion_findley_ax_alterna(x_0,alfa_fin,s_m_fin,beta_fin));
        x_eps=x_0+epsilon;
        sus_x0_en_df=double((funcion_findley_ax_alterna(x_eps,alfa_fin,s_m_fin,beta_fin)-
funcion_findley_ax_alterna(x_0,alfa_fin,s_m_fin,beta_fin))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```

"newton_raphson_findley_ax_m"

```
function [y] = newton_raphson_findley_ax_m(alfa_fin,beta_fin,x_0,tolerancia)
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"
```

```
    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
    cont=0;
    error_relativo=tolerancia+1;
    while error_relativo>=tolerancia && cont<=5

        sus_x0_en_f=double(funcion_findley_ax_media(x_0,alfa_fin,beta_fin));
        x_eps=x_0+epsilon;
        sus_x0_en_df=double((funcion_findley_ax_media(x_eps,alfa_fin,beta_fin)-
funcion_findley_ax_media(x_0,alfa_fin,beta_fin))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```

"newton_raphson_findley_tor_alt"

```
function [y] = newton_raphson_findley_tor_alt(alfa_fin,t_m_fin,beta_fin,x_0,tolerancia)
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"

    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
    cont=0;
    error_relativo=tolerancia+1;
    while error_relativo>=tolerancia && cont<=5

        sus_x0_en_f=double(funcion_findley_tor_alterna(x_0,alfa_fin,t_m_fin,beta_fin));
        x_eps=x_0+epsilon;
        sus_x0_en_df=double((funcion_findley_tor_alterna(x_eps,alfa_fin,t_m_fin,beta_fin)-
funcion_findley_tor_alterna(x_0,alfa_fin,t_m_fin,beta_fin))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```

"newton_raphson_findley_tor_m"

```
function [y] = newton_raphson_findley_tor_m(alfa_fin,beta_fin,x_0,tolerancia)
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"

    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
    cont=0;
    error_relativo=tolerancia+1;
    while error_relativo>=tolerancia && cont<=5

        sus_x0_en_f=double(funcion_findley_tor_media(x_0,alfa_fin,beta_fin));
        x_eps=x_0+epsilon;
        sus_x0_en_df=double((funcion_findley_tor_media(x_eps,alfa_fin,beta_fin)-
funcion_findley_tor_media(x_0,alfa_fin,beta_fin))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```

"newton_raphson_findley_zer"

```
function [y] = newton_raphson_findley_zer(lf_tr_pul,lf_ax_alt,x_0,tolerancia)
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"

    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
```

```
    cont=0;
    error_relativo=tolerancia+1;
    while error_relativo>=tolerancia && cont<=5

        sus_x0_en_f=double(funcion_findley_zer(x_0,lf_tr_pul,lf_ax_alt));
        x_eps=x_0+epsilon;
        sus_x0_en_df=double((funcion_findley_zer(x_eps,lf_tr_pul,lf_ax_alt)-
funcion_findley_zer(x_0,lf_tr_pul,lf_ax_alt))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```

"newton_raphson_matake_alt"

```
function [y] = newton_raphson_matake_alt(alfa_mat,s_m_mat,beta_mat,x_0,tolerancia)
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"

    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
    cont=0;
    error_relativo=tolerancia+1;
    while error_relativo>=tolerancia && cont<=5

        sus_x0_en_f=double(funcion_matake_alterna(x_0,alfa_mat,s_m_mat,beta_mat));
        x_eps=x_0+epsilon;
        sus_x0_en_df=double((funcion_matake_alterna(x_eps,alfa_mat,s_m_mat,beta_mat)-
funcion_matake_alterna(x_0,alfa_mat,s_m_mat,beta_mat))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```

"newton_raphson_matake_m"

```
function [y] = newton_raphson_matake_m(alfa_mat,beta_mat,x_0,tolerancia)
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"

    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
    cont=0;
    error_relativo=tolerancia+1;
    while error_relativo>=tolerancia && cont<=5

        sus_x0_en_f=double(funcion_matake_media(x_0,alfa_mat,beta_mat));
        x_eps=x_0+epsilon;
        sus_x0_en_df=double((funcion_matake_media(x_eps,alfa_mat,beta_mat)-
funcion_matake_media(x_0,alfa_mat,beta_mat))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```

"newton_raphson_papuga_ax_alt"

```
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"

    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
    cont=0;
    error_relativo=tolerancia+1;
```

```
    while error_relativo>=tolerancia && cont<=5


sus_x0_en_f=double(funcion_papuga_ax_alterna(x_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,s_
m_pap));
        x_eps=x_0+epsilon;

sus_x0_en_df=double((funcion_papuga_ax_alterna(x_eps,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_al
t,s_m_pap)-
funcion_papuga_ax_alterna(x_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,s_m_pap))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```

"newton_raphson_papuga_ax_m"

```
function [y] =
newton_raphson_papuga_ax_m(a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,x_0,tolerancia)
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"

    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
    cont=0;
    error_relativo=tolerancia+1;
    while error_relativo>=tolerancia && cont<=5


sus_x0_en_f=double(funcion_papuga_ax_media(x_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt));
        x_eps=x_0+epsilon;

sus_x0_en_df=double((funcion_papuga_ax_media(x_eps,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt)
-funcion_papuga_ax_media(x_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```

"newton_raphson_papuga_tor_alt"

```
function [y] =
newton_raphson_papuga_tor_alt(a_p,t_m_pap,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,x_0,tolerancia)
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"

    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
    cont=0;
    error_relativo=tolerancia+1;
    while error_relativo>=tolerancia && cont<=5


sus_x0_en_f=double(funcion_papuga_tor_alterna(x_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,t_m_pap));
        x_eps=x_0+epsilon;

sus_x0_en_df=double((funcion_papuga_tor_alterna(x_eps,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,t_m_pa
p)-funcion_papuga_tor_alterna(x_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,t_m_pap))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```

"newton_raphson_papuga_tor_m"

```
function [y] = newton_raphson_papuga_tor_m(a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt,x_0,tolerancia)
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"

    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
    cont=0;
    error_relativo=tolerancia+1;
    while error_relativo>=tolerancia && cont<=5

        sus_x0_en_f=double(funcion_papuga_tor_media(x_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt));
        x_eps=x_0+epsilon;

sus_x0_en_df=double((funcion_papuga_tor_media(x_eps,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt)-
funcion_papuga_tor_media(x_0,a_p,b_p,dos_theta,lf_ax_alt,lf_tr_pul,lf_tors_alt))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```

"newton_raphson_sines_alt"

```
function [y] = newton_raphson_sines_alt(alfa_sin,beta_sin,s_m_sin,x_0,tolerancia)
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"

    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
    cont=0;
    error_relativo=tolerancia+1;
    while error_relativo>=tolerancia && cont<=5

        sus_x0_en_f=double(funcion_sines_alterna(x_0,alfa_sin,beta_sin,s_m_sin));
        x_eps=x_0+epsilon;
        sus_x0_en_df=double((funcion_sines_alterna(x_eps,alfa_sin,beta_sin,s_m_sin)-
funcion_sines_alterna(x_0,alfa_sin,beta_sin,s_m_sin))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```

"newton_raphson_sines_m"

```
function [y] = newton_raphson_sines_m(alfa_sin,beta_sin,x_0,tolerancia)
%Newton-Raphson de una funcion. Criterio de parada: "error relativo"

    epsilon=0.05; %Valor muy pequeño para evaluar la pendiente numericamente
    cont=0;
    error_relativo=tolerancia+1;
    while error_relativo>=tolerancia && cont<=5

        sus_x0_en_f=double(funcion_sines_media(x_0,alfa_sin,beta_sin));
        x_eps=x_0+epsilon;
        sus_x0_en_df=double((funcion_sines_media(x_eps,alfa_sin,beta_sin)-
funcion_sines_media(x_0,alfa_sin,beta_sin))/epsilon);

        x_i=x_0-sus_x0_en_f/sus_x0_en_df;
        error_relativo =abs(x_i-x_0)/abs(x_i);
        x_0=x_i;
        cont=cont+1;
    end
    y=x_i;
end
```