



GRADO EN INGENIERÍA MECÁNICA

TRABAJO FIN DE GRADO

2014 / 2015

*CARACTERIZACIÓN A IMPACTO DE CAUCHO RECICLADO MEDIANTE
ELEMENTOS FINITOS*

ANEXO 2: VISCOELASTICIDAD

DATOS DE LA ALUMNA O DEL ALUMNO

NOMBRE: ANE

APELLIDOS: ESCRIBANO CASTRO

FDO.:

FECHA: 10-02-2015

DATOS DEL DIRECTOR O DE LA DIRECTORA

NOMBRE: IRANTZU

APELLIDOS: URIARTE GALLASTEGI

DEPARTAMENTO: INGENIERÍA MECÁNICA

FDO.:

FECHA: 10-02-2015

Contenido

Capítulo 1. Introducción	1
1.1. Estructura del Algoritmo	1
Capítulo 2. Algoritmo	2
2.1. P.S.O.....	2
2.2. Coste.....	7
2.3. Simular	8
2.4. AnalizarDatos	8
Capítulo 3. Ensayos	10
3.1. Probetas grandes	10
3.1.1. Densidad = 0.7 g/cm ³	10
3.1.2. Densidad = 0.8 g/cm ³	11
3.1.3. Densidad = 0.9 g/cm ³	12
3.1.4. Densidad = 1g/cm ³	12
3.2. Probetas pequeñas	13
3.2.1. Densidad = 0.7 g/cm ³	13
3.2.2. Densidad = 0.9 g/cm ³	14
3.2.3. Densidad = 1g/cm ³	15
Capítulo 4. Resultados obtenidos	17

Índice de tablas

Tabla 1. Constantes viscoelásticas para densidades 0.7, 0.8, 0.9 y 1 g/cm ³ ajustadas para cada curva (probetas grandes).	17
Tabla 2. Constantes viscoelásticas para densidades 0.7, 0.9 y 1 g/cm ³ ajustadas para cada curva (probetas pequeñas).	17

Índice de figuras

Fig. 1. Resultados obtenidos para probetas grandes con densidad 0.7 g/cm ³ .	10
Fig. 2. Gráfica para probetas grandes con densidad 0.7 g/cm ³ .	11
Fig. 3. Resultados obtenidos para probetas grandes con densidad 0.8 g/cm ³ .	11
Fig. 4. Gráfica para probetas grandes con densidad 0.8 g/cm ³ .	11
Fig. 5. Resultados obtenidos para probetas grandes con densidad 0.9 g/cm ³ .	12
Fig. 6. Gráfica para probetas grandes con densidad 0.9 g/cm ³ .	12
Fig. 7. Resultados obtenidos para probetas grandes con densidad 1 g/cm ³ .	12
Fig. 8. Gráfica para probetas grandes con densidad 1 g/cm ³ .	13
Fig. 9. Resultados obtenidos para probetas pequeñas con densidad 0.7 g/cm ³ .	13
Fig. 10. Gráfica para probetas pequeñas con densidad 0.7 g/cm ³ .	14
Fig. 11. Resultados obtenidos para probetas pequeñas con densidad 0.9 g/cm ³ .	14
Fig. 12. Gráfica para probetas pequeñas con densidad 0.9 g/cm ³ .	15
Fig. 13. Resultados obtenidos para probetas pequeñas con densidad 1 g/cm ³ .	15
Fig. 14. Gráfica para probetas pequeñas con densidad 1 g/cm ³ ..	16

Capítulo 1. Introducción

Para la obtención de parámetros de la parte viscoelástica se ha utilizado un Algoritmo de Optimización por nube de partículas (PSO). Este algoritmo, tal y como se ha explicado en la memoria, se basa en el comportamiento de sistemas naturales tales como abejas, aves, hormigas...

Un Algoritmo PSO trabaja con una población de partículas, las cuales se desplazan en un espacio de búsqueda conforme a ciertas reglas matemáticas. El movimiento de cada partícula depende de su mejor posición obtenida, así como la mejor posición global hallada en todo el espacio de búsqueda. Las partículas se van orientando conforme se encuentran mejores posiciones. El proceso se va repitiendo con el objetivo, no garantizado, de hallar en algún movimiento, una solución lo suficientemente satisfactoria.

A continuación, se explica el proceso que hemos utilizado para la obtención de los 3 parámetros que hemos obtenido para caracterizar el modelo viscoelástico.

1.1. Estructura del Algoritmo

En la elaboración de éste proceso, se han utilizado los siguientes procesos:

- 1- PSO: Programa principal con el Algoritmo PSO.
- 2- Datos: Conjunto de todos los datos experimentales hallados en trabajos previos. Se han usado los correspondientes a Tiempo, Lambda y Cauchy.
- 3- Coste: Subprograma que calcula el error matemático en el ajuste.
- 4- Simular: Subprograma para utilizar las funciones del modelo a estudiar.
- 5- AnalizarDatos: Subprograma para la obtención de las gráficas.

Capítulo 2. Algoritmo

Se expone el proceso en cuestión para poder analizarlo. Como ejemplo, se usará el de la densidad 0.7 para el caso de probetas pequeñas.

2.1. P.S.O.

Se limpia todo lo que se haya hecho previamente:

```
clc
clear all
close all
```

Se nombra los parámetros globales que se vayan a usar durante el programa:

```
global Nmuestras tiempo L Sigmaexp Lpunto I2 sigmaeinter
MSigmaexp DatosPSO dT
```

Se les da valores a dichos parámetros globales:

```
%Parametros PSO
MuestrasporEje=5;
Nparticulas=MuestrasporEje^3; [Número de valores en los ejes X, Y y Z, (>0)]
Inerciamin=0.4;
Inerciamax=0.9;
phi1=0.5;
phi2=0.5;
Niter=100;%20
dt=1;
```

Se llama a los datos que se tienen de densidades, en este caso, para 0.7, de los cuales se disponen los siguientes:

- P_0.004_0.71 (hiperelástico)
- P_0.04_0.72_A
- P_0.04_0.72_B
- P_0.4_0.72_A

- P_0.4_0.72_B

Todos ellos quedan reunidos en:

```
Datos_07_Cauchy;
```

Como se tienen 5 conjuntos de datos de densidades del mismo tipo, K va desde 1 a 5. Con ello, se coge toda la longitud de los Excel y se define la longitud total de las variables:

```
for k=1:5
Comando=['Nmuestras(', num2str(k), '=length(DatosEnsayo', num2str(k), '(:,1));'];
eval(Comando);
end

tamanhomax=max(Nmuestras);
tiempo=zeros(tamanhomax,5);
L=zeros(tamanhomax,5);
Sigmaexp=zeros(tamanhomax,5);
lpunto=zeros(tamanhomax,5);
I2=zeros(tamanhomax,5);
sigmaeinter=zeros(tamanhomax,5);
```

Se crean 3 matrices en las cuales se meten los tiempos de cada ensayo en una, las lambdas de cada ensayo en otra, y las tensiones Cauchy de cada ensayo en la última.

```
for k=1:5
Comando=['tiempo(1:Nmuestras(', num2str(k), '),', num2str(k), ')=DatosEnsayo', num2str(k), '(:,1);'];
eval(Comando);
Comando=['L(1:Nmuestras(', num2str(k), '),', num2str(k), ')=DatosEnsayo', num2str(k), '(:,2);'];
eval(Comando);
Comando=['Sigmaexp(1:Nmuestras(', num2str(k), '),', num2str(k), ')=DatosEnsayo', num2str(k), '(:,3);'];
eval(Comando);
end
```

Se calcula λ e I_2 :

```
for k=1:5
lpunto(1:Nmuestras(k),k)=[0;diff(L(1:Nmuestras(k),k))./diff(tiempo(1:Nmuestras(k),k))];
I2(:,k)=L(:,k).^(-2+2*L(:,k));
MSigmaexp(k)=mean(Sigmaexp(1:Nmuestras(k),k));
dT(1:Nmuestras(k),k)=[tiempo(2,k)-tiempo(1,k);diff(tiempo(1:Nmuestras(k),k))];
```

Para la hiperelástica no se hace, por lo que para $k > 1$, se procede a realizar la siguiente ecuación:


```

if k>1
    sigmaeinter(1:Nmuestras(k),k)=interp1(L(1:Nmuestras(1),1),
    Sigmaexp(1:Nmuestras(1),1),L(1:Nmuestras(k),k),'linear','e
    xtrap');
end
end

```

Con todo esto, se crea una matriz que contenga todos los datos que se necesitan. En total 12 columnas divididas de la siguiente manera:

- En las columnas 1 y 2 → Coste PSO y R2
- Columnas 3, 4, 5 → Variables que contienen la posición POS.
- Columnas 6, 7, 8 → 3 valores de velocidades ya que se pueden mover en 3 direcciones
- Columnas 9, 10, 11 → Posición óptima de cada partícula.
- Columna 12 → Coste óptimo de cada partícula.

La primera vez que se arranca el Algoritmo, es importante establecer los siguientes “límites” para encontrar los parámetros. Gracias a esto, los parámetros que se desean obtener girarán en torno a unos determinados valores.

```

% bloA4=5e+05;
% bloA5=-8e+05;
% bloA6=1e-2;
% bupA4=0;
% bupA5=0;
% bupA6=0;

```

Una vez arrancado el programa, se establecen los límites en torno a los parámetros hallados en el primer arranque. Si éstos no se cumplen, se procede a otra vuelta hasta que se consigan los valores de los límites, y unos parámetros que figuren dentro de ellos.

```

bloA4=1e5;
bloA5=-9e5;
bloA6=0.5;
bupA4=3e6;
bupA5=-1e5;
bupA6=1.5;

```

Se crea una distribución uniforme entre los límites detallados arriba:

```

PA4=random('Uniform',bloA4,bupA4,1,MuestrasporEje);
PA5=random('Uniform',bloA5,bupA5,1,MuestrasporEje);
PA6=random('Uniform',bloA6,bupA6,1,MuestrasporEje);

```

```

VA4=random('Uniform',-abs(bupA4-bloA4),abs(bupA4-
bloA4),1,MuestrasporEje);
VA5=random('Uniform',-abs(bupA5-bloA5),abs(bupA5-
bloA5),1,MuestrasporEje);
VA6=random('Uniform',-abs(bupA6-bloA6),abs(bupA6-
bloA6),1,MuestrasporEje);

DatosPSO=zeros(Nparticulas,12);
i=0;

for iA4=1:MuestrasporEje
    for iA5=1:MuestrasporEje
        for iA6=1:MuestrasporEje
            i=i+1;

DatosPSO(i,:)=[0,0,PA4(iA4),PA5(iA5),PA6(iA6),VA4(iA4),VA5(iA5),
VA6(iA6),PA4(iA4),PA5(iA5),PA6(iA6),0];
            end
        end
    end
end
end

```

Se llama al subprograma Coste:

```

for i=1:Nparticulas
    [DatosPSO(i,1), DatosPSO(i,2)]=Coste_07(DatosPSO(i,3:5));
end
DatosPSO(:,12)=DatosPSO(:,1);
[CosteMin,ref]=min(DatosPSO(:,1));
PosOptGlobal=DatosPSO(ref,3:5);

```

Con esto, queda hallada la posición global óptima inicial. Ahora, lo que se procede a hacer es calcular todas las combinaciones posibles para cada valor de A4 con los valores de A5 y A6. Esto es, si A4, A5 y A6 son cada uno un vector de 50 elementos, hay 125000 combinaciones posibles.

```

DatosHistoricos_07=[];

for k=1:Niter
    [DatosPSO(i,1), DatosPSO(i,2)]=Coste_07(DatosPSO(i,3:5));
    for i=1:Nparticulas
        r1=random('Uniform',0,1,1,1);
        r2=random('Uniform',0,1,1,1);
        Inercia=(Inerciamax-Inerciamin)*k^2/Niter^2+(Inerciamin-
Inerciamax)*2*k/Niter+Inerciamax;

```

Fórmula de la viscoelasticidad para caso de Yang:

```

    DatosPSO(i,6:8)=Inercia*DatosPSO(i,6:8)+
    phi1*r1*(DatosPSO(i,9:11)-DatosPSO(i,3:5))+
    phi2*r2*(PosOptGlobal-DatosPSO(i,3:5));
    DatosPSO(i,3:5)=DatosPSO(i,3:5)+ DatosPSO(i,6:8)*dt;
    [DatosPSO(i,1),
DatosPSO(i,2)]=Coste_07(DatosPSO(i,3:5));

```

Se almacenan los datos que se han ido obteniendo:

```
DatosHistoricos_07=[DatosHistoricos_07, [DatosPSO(i,3);DatosPSO(i,4);Da
tosPSO(i,5);DatosPSO(i,1);DatosPSO(i,2)]];
    if DatosPSO(i,1)<DatosPSO(i,12)
        DatosPSO(i,12)=DatosPSO(i,1);
        DatosPSO(i,9:11)=DatosPSO(i,3:5);
    end
end
[CosteMin,ref]=min(DatosPSO(:,12));
PosOptGlobal=DatosPSO(ref,9:11);
plot3(DatosPSO(:,3),DatosPSO(:,4),DatosPSO(:,5),'xk')
grid on
xlabel('C2 [Pa]')
ylabel('C3 [Pa]')
zlabel('C4 [s]')
hold on
end
```

Se almacena la mejor posición óptima local para cada parámetro:

```
A4=PosOptGlobal(1);
A5=PosOptGlobal(2);
A6=PosOptGlobal(3);
```

Como se ha mencionado antes, sólo nos interesan los valores viscoelásticos, por lo que se salta los primeros grupos de datos, es decir, los hiperelásticos.

Se guardan los datos:

```
todaslassigmas_07 = zeros(4,860);
for k=[2,3,4,5]

    sigmainter=Simular_07(Nmuestras(k),tiempo(1:Nmuestras(k),k),L(1:
Nmuestras(k),k),A4,A5,A6,lpunto(1:Nmuestras(k),k),sigmaeinter(1:
Nmuestras(k),k),I2(1:Nmuestras(k),k),dT(1:Nmuestras(k),k));
        todaslassigmas_07(k-1,1:length(sigmainter)) = sigmainter;
end

save('DatosHistoricos_07.mat', 'DatosHistoricos_07')
save('todaslassigmas_07.mat','todaslassigmas_07')
```

Se llama al subprograma AnalizarDatos:

```
AnalizarDatos_07
```

2.2. Coste

```
function [C,R2medio]=Coste_07(X)
```

Se definen las variables globales:

```
global Nmuestras tiempo L Sigmaexp Lpunto I2 sigmaeinter
MSigmaexp DatosPSO dT

A4=X(1);
A5=X(2);
A6=X(3);
```

Se definen los valores iniciales de las variables:

```
Ntot=0;
clear Error;
Error=[];
NumError=[];
DenError=[];
```

Para todos los casos, menos el hiperelástico, se llama al subprograma Simular:

Se calcula el coeficiente de determinación R^2 . Con ello, se halla el error del ajuste:

```
for k=[2,3,4,5]

    sigmainter=Simular_07(Nmuestras(k),tiempo(1:Nmuestras(k),k),L(1:
Nmuestras(k),k),A4,A5,A6,Lpunto(1:Nmuestras(k),k),sigmaeinter(1:
Nmuestras(k),k),I2(1:Nmuestras(k),k),dT(1:Nmuestras(k),k));
    Msigmainter=MSigmaexp(k);

    Error=[Error,sum((sigmainter-
Sigmaexp(1:Nmuestras(k),k)).^2./((Msigmainter-
Sigmaexp(1:Nmuestras(k),k)).^2))];

    NumError=[NumError,sum((sigmainter-
Sigmaexp(1:Nmuestras(k),k)).^2)];

    DenError=[DenError,sum((Msigmainter-
Sigmaexp(1:Nmuestras(k),k)).^2)];

end
```

Se coge el valor que dé menos error. *NumError*. Quiere decir que va término a término.

```
R2=1-NumError./DenError;
%save('R2.mat','R2')
R2medio=mean(R2);
EG=NumError./DenError;
EGtotal=sum(EG);

C=EGtotal;
```

2.3. Simular

Utilizamos la integral de convolución para viscoelasticidad no lineal:

```
function
[Sigma]=Simular_07(Nmuestras,tiempo,L,A4,A5,A6,Lpunto,sigmae,I2,dT)

f1= L.*Lpunto.*(A4+A5*(I2-3));
f2= exp(-tiempo/A6).*dT;
f3= L.^-2.*Lpunto.*(A4+A5*(I2-3));

integrall1=conv(f1,f2);   %%longitud 2n, solo nos interesa hasta
length(tiempo)
integrall1cortado=integrall1(1:Nmuestras);
integral2=conv(f3,f2);
integral2cortado=integral2(1:Nmuestras);

Sigma=sigmae+L.^2.*integrall1cortado+L.^-1/2.*integral2cortado;
```

2.4. AnalizarDatos

Con este subprograma, se puede sacar las gráficas de cada conjunto de datos, con λ en el eje X y σ en el eje Y. dependiendo de los datos que se quieran poner en la gráfica, o de cuantos datos se dispongan se usarán unos u otros comandos. Para este caso, que solo se disponen de 5 conjuntos de datos, dejamos sin usar el correspondiente al sexto (para la velocidad más alta).

```
figure
plot(DatosEnsayo2(:,2),DatosEnsayo2(:,3),'-k','linewidth',2)
grid on
xlabel('\lambda')
ylabel('\sigma [Pa]')

hold on
plot(DatosEnsayo3(:,2),DatosEnsayo3(:,3),'--k','linewidth',2)
plot(DatosEnsayo4(:,2),DatosEnsayo4(:,3),'k','linewidth',2.5)
plot(DatosEnsayo5(:,2),DatosEnsayo5(:,3),'-ok','linewidth',0.5)
% plot(DatosEnsayo6(:,2),DatosEnsayo6(:,3),'k','linewidth',1.5)

% -. edo -- edo +k

plot(DatosEnsayo2(:,2),todaslassigmas_07(1,1:length(DatosEnsayo2
(:,2))), '-r', 'linewidth', 0.5)
plot(DatosEnsayo3(:,2),todaslassigmas_07(2,1:length(DatosEnsayo3
(:,2))), '--r', 'linewidth', 0.5)
plot(DatosEnsayo4(:,2),todaslassigmas_07(3,1:length(DatosEnsayo4
(:,2))), ':r', 'linewidth', 0.5)
plot(DatosEnsayo5(:,2),todaslassigmas_07(4,1:length(DatosEnsayo5
(:,2))), '-sr', 'linewidth', 0.5)
```

```

%
plot(DatosEnsayo6(:,2),todaslassigmas_07(5,1:length(DatosEnsayo6
(:,2))), 'r', 'linewidth', 0.5)

%(1)    2,3,4,5,6
%    h=legend('Experimental    0.57s-1    (a)', 'Experimental    0.57s-1
(b)', 'Experimental 5.7s-1 (a)', 'Experimental 5.7s-1 (b)', 'Experimental
57s-1', 'Estimated 0.57s-1 (a)', 'Estimated 0.57s-1 (b)', 'Estimated
5.7s-1 (a)', 'Estimated 5.7s-1 (b)', 'Estimated 57s-1');

%(8)    2,3, 4,5
h=legend('Experimental    0.57s-1    (a)', 'Experimental    0.57s-1
(b)', 'Experimental    5.7s-1    (a)', 'Experimental    5.7s-1
(b)', 'Estimated 0.57s-1 (a)', 'Estimated 0.57s-1 (b)', 'Estimated
5.7s-1 (a)', 'Estimated 5.7s-1 (b)');

```

Capítulo 3. Ensayos

Una vez descrito todos los procesos, se muestran todos los resultados obtenidos para todas las densidades, tanto para probetas pequeñas como grandes.

3.1. Probetas grandes

3.1.1. Densidad = 0.7 g/cm³

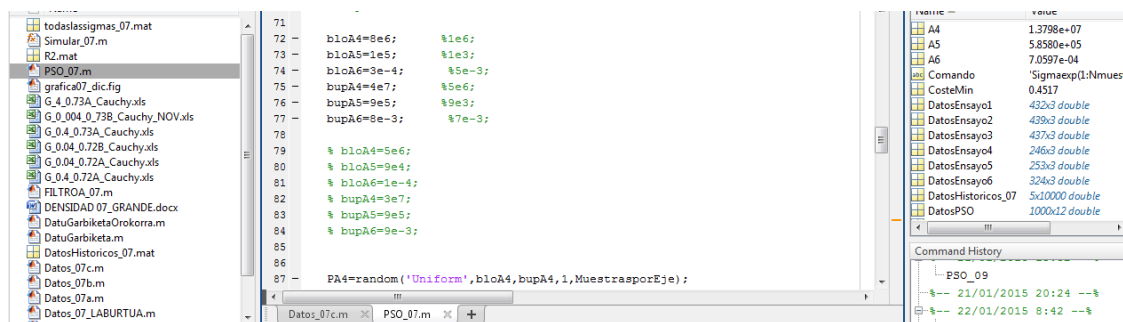


Fig. 1. Resultados obtenidos para probetas grandes con densidad 0.7 g/cm³.

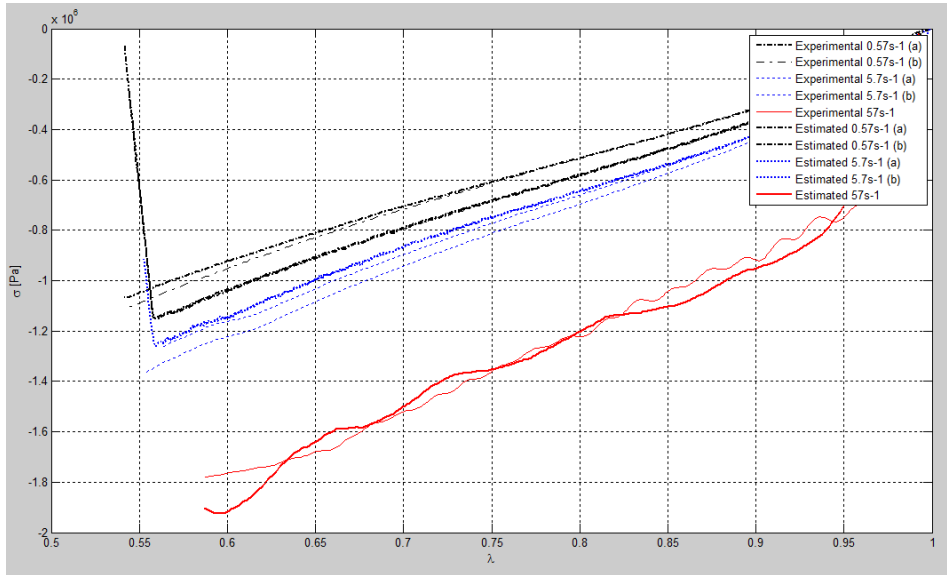


Fig. 2. Gráfica para probetas grandes con densidad 0.7 g/cm³.

3.1.2. Densidad = 0.8 g/cm³



Fig. 3. Resultados obtenidos para probetas grandes con densidad 0.8 g/cm³.

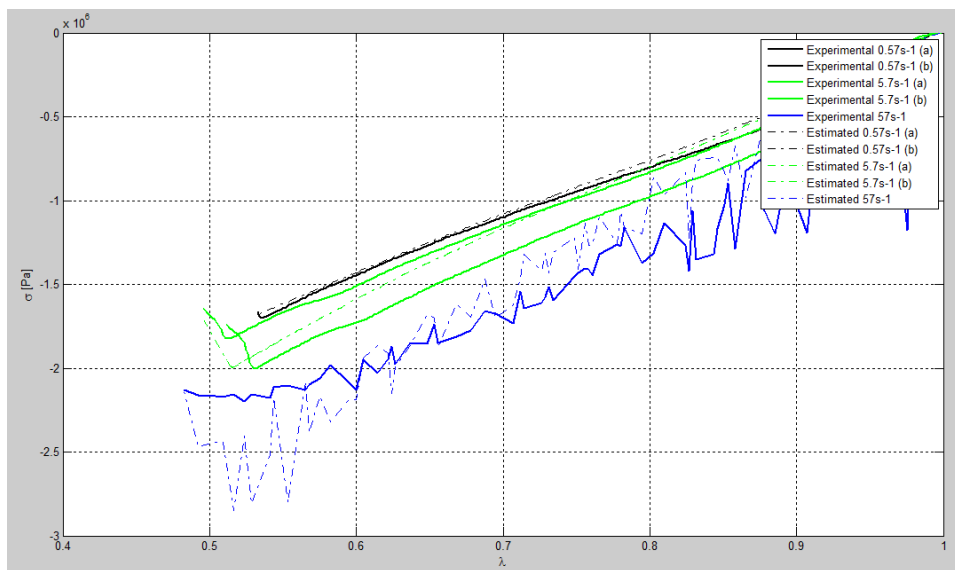


Fig. 4. Gráfica para probetas grandes con densidad 0.8 g/cm³.

3.1.3. Densidad = 0.9 g/cm³

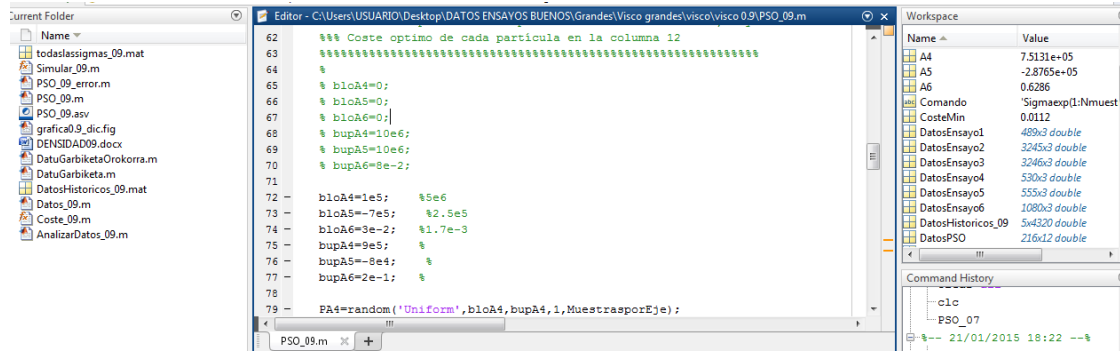


Fig. 5. Resultados obtenidos para probetas grandes con densidad 0.9 g/cm³.

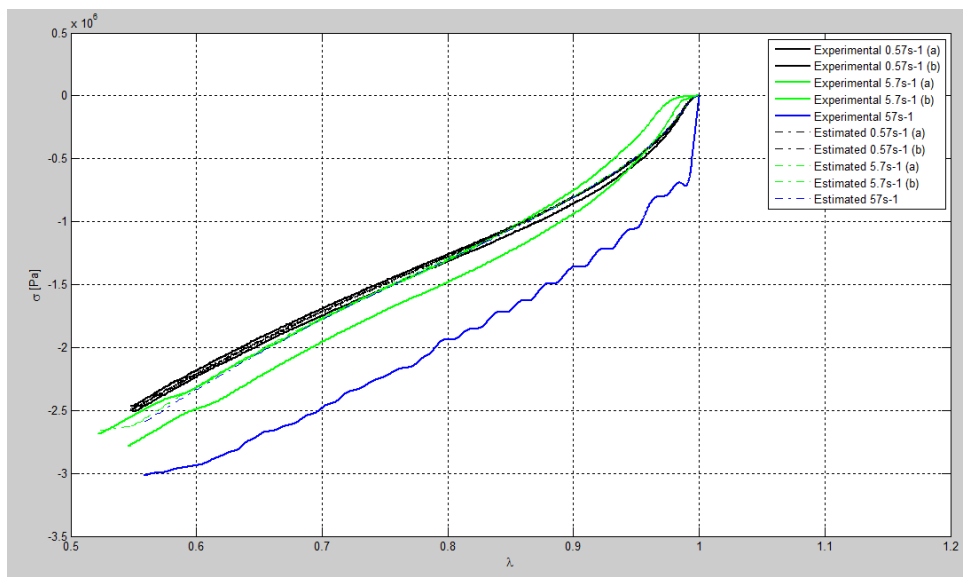


Fig. 6. Gráfica para probetas grandes con densidad 0.9 g/cm³.

3.1.4. Densidad = 1g/cm³

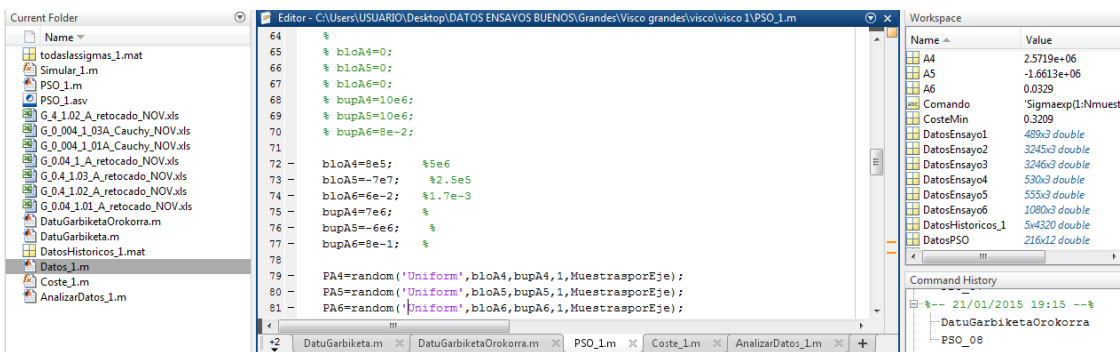


Fig. 7. Resultados obtenidos para probetas grandes con densidad 1 g/cm³.

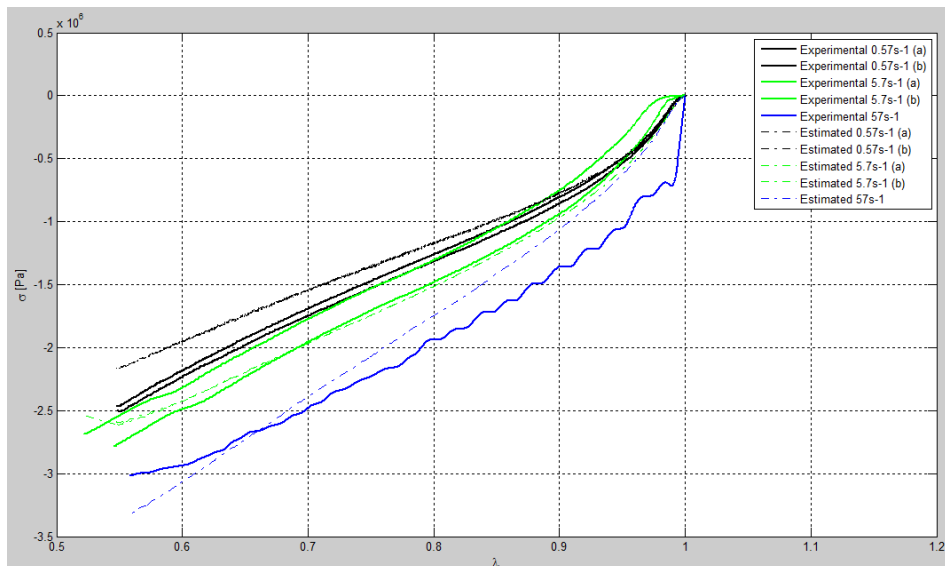


Fig. 8. Gráfica para probetas grandes con densidad 1 g/cm³.

3.2. Probetas pequeñas

3.2.1. Densidad = 0.7 g/cm³

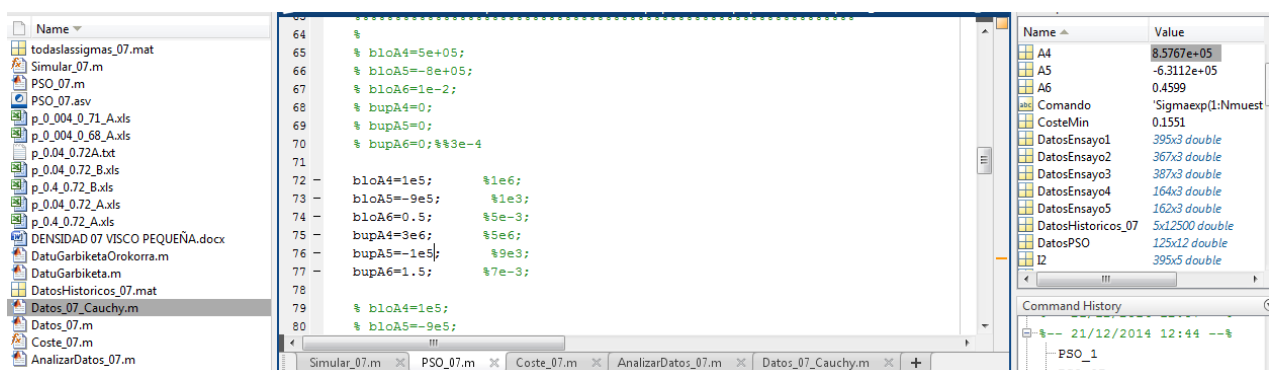
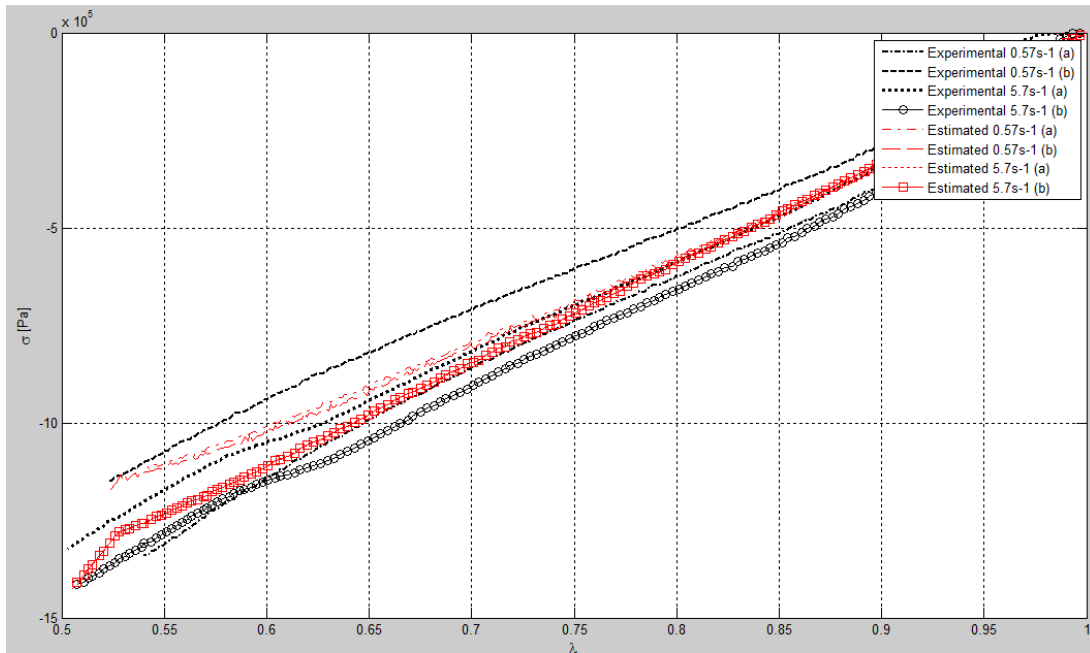
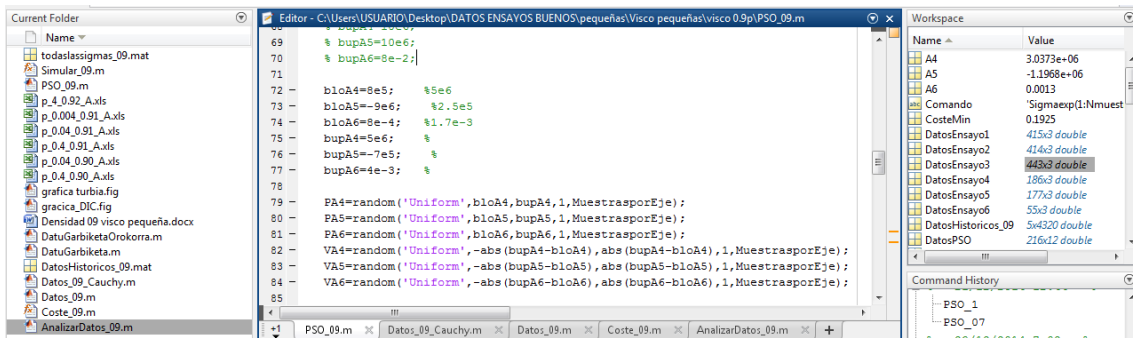


Fig. 9. Resultados obtenidos para probetas pequeñas con densidad 0.7 g/cm³.



3.2.2. Densidad = 0.9 g/cm^3



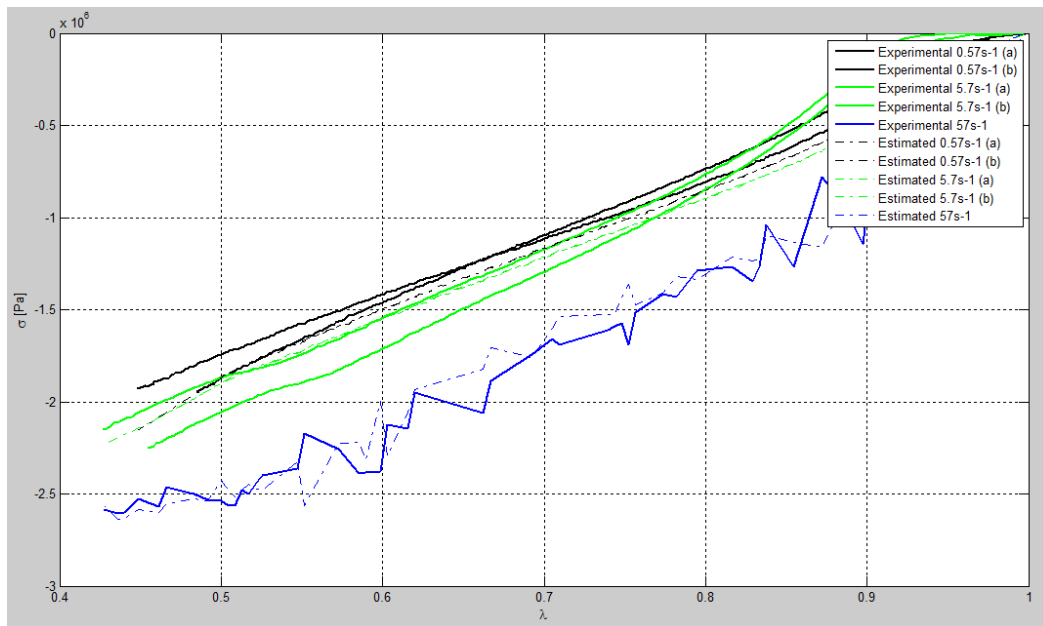


Fig. 12. Gráfica para probetas pequeñas con densidad 0.9 g/cm^3 .

3.2.3. Densidad = 1 g/cm^3



Fig. 13. Resultados obtenidos para probetas pequeñas con densidad 1 g/cm^3 .

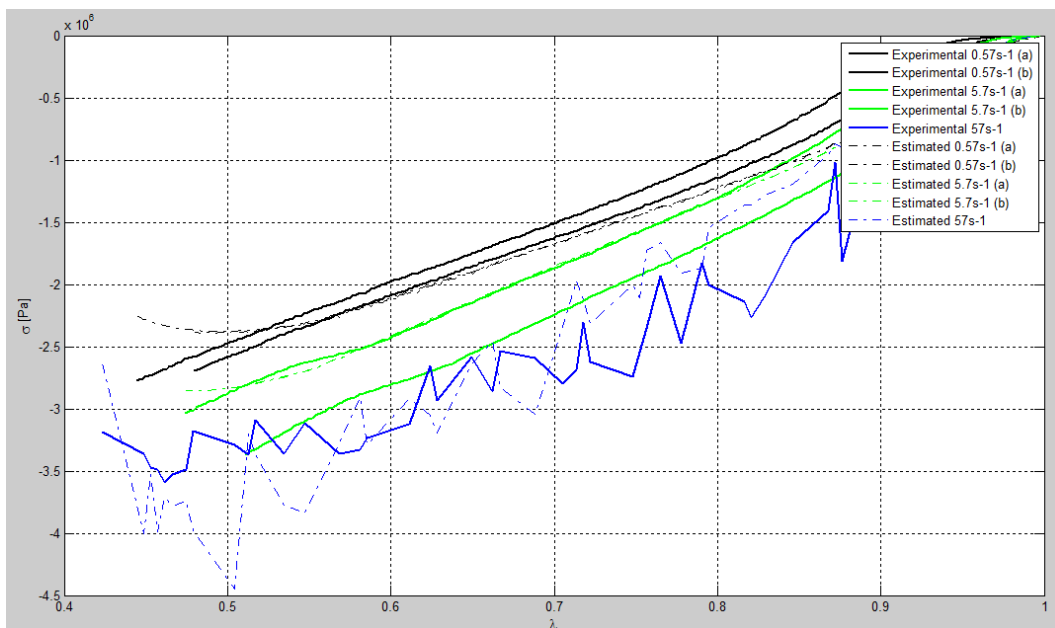


Fig. 14. Gráfica para probetas pequeñas con densidad 1 g/cm^3 .

Capítulo 4. Resultados obtenidos

A continuación, se muestran en dos tablas los parámetros obtenidos para cada densidad. Primero para probetas grandes, y luego para las probetas pequeñas.

<u>Pa</u>	<u>0.7 g/cm³</u>	<u>0.8 g/cm³</u>	<u>0.9 g/cm³</u>	<u>1 g/cm³</u>
C₂	1,3798E+07	1,003E+06	7,5131E+05	2,5179E+06
C₃	5,858E+05	-2,8045E+05	-2,8765E+05	-1,6613E+06
C₄	0.0007	0.7003	0,6286	0,0329

Tabla 1. Constantes viscoelásticas para densidades 0.7, 0.8, 0.9 y 1 g/cm³ ajustadas para cada curva (probetas grandes).

<u>Pa</u>	<u>0.7 g/cm³</u>	<u>0.9 g/cm³</u>	<u>1 g/cm³</u>
C₂	8.5767·10 ⁵	3.0373·10 ⁶	1.4193·10 ⁶
C₃	-6.3112·10 ⁵	-1.1968·10 ⁶	-9.3064·10 ⁶
C₄	0.4599	0.0013	0.2155

Tabla 2. Constantes viscoelásticas para densidades 0.7, 0.9 y 1 g/cm³ ajustadas para cada curva (probetas pequeñas).