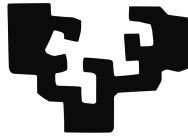


eman ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Fisikako Gradua

Gradu Amaierako Lana

Simulazioak fisikaren irakaskuntzan

Egilea

Aitor Iraola Galarza

Zuzendaria

Juan M. Aguirregabiria



ZTF-FCT

2014ko irailaren 3an

Laburpena

Edozein zientzia arlotan, jakintzaren muturrera iristeko ezinbestekoa da oinarria sendoa izatea, ulerpen orokorra. Fisikaren kasuan, berau irakasteko bide egokiak aurkitzea ez da lan erraza; testu-liburu, ariketa eta problema mila ditugu eskuragarri, baina ideien barneratze sakona lortzeko, askotan ezinbestekoa zaigu soluzioa *ikustea*. Horixe izango da proiektu honen helburua, **simulazioen bidez fisikako kontzeptuak *ikusi* eta horiek hobetu errotu ahal izatea**. Modu horretan, proiektuan zehar, fisikako problema ugari aztertu, simulatu eta garatuko ditugu.

Gaien aurkibidea

Laburpena	i
Gaien aurkibidea	iii
Irudien aurkibidea	v
1 Sarrera	1
2 Easy Java Simulations (EJS)	3
2.1 Zer da <i>EJS</i> ?	3
2.2 <i>EJS</i> n simulatzen	4
3 Proiektuaren garapena	7
3.1 DBH eta Batxilergoko simulazioak	8
3.1.1 Pendulu sinplea	8
3.1.2 Tiro parabolikoa	12
3.2 Unibertsitate mailako simulazioak	17
3.2.1 Uhin-grabitatorioak	17
3.2.2 Pendulu bikoitza	25
3.2.3 Talka-uhinak	32
4 Ondorioak	39
Bibliografia	43

Irudien aurkibidea

3.1	<i>Pendulu sinplearen</i> indarren eskema	8
3.2	<i>SIM_PenduluSinplea.jar</i> simulazioaren pantaila orokorra	10
3.3	<i>Tiro Parabolikoaren</i> indarren eskema	12
3.4	<i>SIM_TiroParabolikoa.jar</i> simulazioaren pantaila nagusia	13
3.5	<i>Mendian talka</i> modu gehigarria	14
3.6	<i>Hasierako altuera</i> eta <i>Mendian talka</i> elkarlanean	15
3.7	<i>SIM_UhinGrabitorioak.jar</i> simulazioaren hasierako pantaila (2D)	21
3.8	<i>SIM_UhinGrabitorioak.jar</i> simulazioaren 3D moduaren pantaila	23
3.9	<i>Pendulu Bikoitzaren</i> indarren-eskema	25
3.10	<i>SIM_PenduluBikoitza.jar</i> simulazioaren pantaila	29
3.11	<i>Talka-uhinen</i> indarren-eskema	32
3.12	<i>SIM_TalkaUhinak.jar</i> simulazioaren hasierako pantaila	35

1. KAPITULUA

Sarrera

Fisikaren irakaskuntza ez da erraza; eta are zailagoa da fisikan lehen pausoak ematen ari den ikaslearentzat; zailtasun nabariena, askotan, lortzen diren soluzio matematikoak interpretatzean datza; ikasleak, ebazten ari den problema hori ikustea, mugitzen ikustea, behar baitu kontzeptuak barneratuko baditu; aukeratu dudan *soluzioak ikusteko* bidea simulazio didaktikoena izan da. Hala, proiektu honetan, *Easy Java Simulations* programa erabiliko da simulazio horiek sortu eta fisika ikaslearen bizitza errazten saiatzeko. Ikusiko duzuen moduan, simulazio hauek interaktiboak, aldakorrak eta bisualki erakargarriak izango dira; tartean, bi pendulu dantzan ikusiko dituzue, uhin grabitatorioak eta talka-uhinak aztertuko ditugu eta, gainera, fisikan hasten direnei hainbeste buruhauste ematen dien *tiro parabolikoaren* simulazio bat ere izango da.

Lanean zehar, egitura ordenatu eta sinple bat jarraitzea erabaki dut, irakurterraza eta orobat, didaktikoa izateko helburuarekin. Protagonistak diren simulazioak jaisteko, atal bakoitzaren amaieran esteka bat aurkituko duzue; nahikoa da bertan klik egin eta agertzen den leihoan *descargar* botoia sakatzea. Horretaz gain, simulazioa abiarazteko, beharrezkoa da *JAVA* instalatua edukitzea; ala ez bada, egin klik [hemen](#) azken bertsioa deskargatzeko.

2. KAPITULUA

Easy Java Simulations (EJS)

Kapitulu honetan simulazioak egiterako orduan oinarri izan dudan programaren inguruan arituko naiz. Hala, programaren ezaugarriak azalduko ditut, abantailak eta desabantailak agertuz eta simulazioak egiteko eskaintzen duen laguntza itzela ulertarazten saiatuz.

2.1 Zer da *EJS*?

Kapitulua izenak iradokitzen duen moduan, *Easy Java Simulations* programaren akronimoa da *EJS*¹, eta, akronimoaren hitzak nahikoak dira programa hau bikain deskribatzeko: **Java** hizkuntzan **Simulazioak Erraz** egiteko programa. Sarreran aipatu bezala, fisika ikasleak ebazten dituen problemak *dinamiko* eta *interaktibo* bihurtzea da lan honen helburuetako bat. Badakigu *simulazioak* horretarako tresna egokiak direla; hala ere, askotan arazo larri baten aurrean aurkitzen da fisikaria: Nola joan fisikatik simulazioetara? Zubi hau igarotzea ez da lan erraza; izan ere, simulazioak zerotik egin ahal izateko, fisikariak lan izugarria egin behar du; ingurune grafiko bat diseinatu behar du, bertan simulazioa ikusteko moduan ipini, apaindu, optimizatu,... askotan, pauso horren zailtasuna da simulazio didaktiko asko ez egitearen arrazoietakoa bat. Eta horixe da *EJS* programak egiten duena, fisikariari lana erraztu; erabiltzaileak higidura ekuazioak lortu eta *sandbox* moduko batean objektu grafikoak kokatu besterik ez du egin behar, eta, sistema bera arduratzen da ekuazio diferentzialak ebatzi (lehen mailakoak), elementu ikusgaiak (malgukiak, bolatxoak, kaxak,...) mugitu eta simulazio didaktiko eta bisualki erakargarri bat sortzeaz. Abantaila gehiago ere aipatu ditzakegu: hasteko, JAVA erabiltzen denez, simulazioa sistema eragile guztietan exekutatu daitezke, nahikoa da JAVAren azken bertsioa deskargatzea. Gainera, simulazioak HTML orri batean ere ipini daitezke, eta, beraz, erraza da web orrietan integratzea. Azkenik, aipatu, idatzitako kodea bera simulazioaren barnean jarri daitekeela, erabiltzaileari,

¹Proiektuan zehar *EJS* akronimoa erabiliko dut programari buruz nabilenean.

nahi badu, simulazioa aldatu eta hobetzea ahalbidetuz.

Hurrengo atalean, laburki azalduko ditut simulazio bat egiteko eman beharreko pausoak.

2.2 EJSn simulatzen

EJS programaren egitura sinplea eta efektiboa da. Hiru atal nagusi bereizten dira bertan:

1. **Vista.** Atal honetan simulazioaren ingurune grafikoa diseinatzen da; aukera ugari ditu, eta, bertan grafikoen leihoak, malgukiak, bolatxoak, 3D inguruneak,... aurkitu daitezke. **Modelo** atalarekin batera, gehien erabiltzen den atala da.
2. **Modelo.** Bigarren atal honetan simulazioaren funtzionamenduarekin erlazionatutako guztia kontrolatzen da; 5 azpiatal bereizten dira bertan:

Variables Simulazioan erabiltzen diren aldagai guztiak definitzeko gunea da hau; aldagaiaren izena, hasierako balioa eta zein motatakoa (double, string, boolean,...) den aukeratzen dira.

Inicialización Simulazioa abiatzen denean, orri honetan idatzitako kodea irakurtzen du lehenik. Erabilgarria da, adibidez, objektuen banaketa konkretu bat agertzea nahi bada pantailan simulazioa hasi bezain pronto. (3.2.1 uhin-grabitatorioen simulazioan, bolatxoak zirkunferentzian zehar banatuta agertzeko, orrialde honetan idatzi da hasierako kodea)

Evolución Simulazioak, martxan dabilenean, egin behar duena azaltzen da; bi aukera daude bertan: alde batetik, lehen ordenako ekuazio diferentzialak ebazteko interfaze bat dago (Aldagai independentea zehaztu, ekuazioa idatzi eta programa bera arduratzen da zenbakizko metodoak aplikatzen); eta bestetik, JAVA kodea zuzenean idazteko orrialde bat ere badago.

Relaciones fijas Orrialde honetako kodigoa etengabe ebaluatzen da; simulazioaren pauso bakoitzean exekutatzen da bertan dagoen kodea; adibidez, penduluen simulazioetan (3.1.1 eta 3.2.2), soken luzeren kalkulua ($l = \sqrt{x^2 + y^2}$) orrialde honetan ipini da, erabiltzaileak pendulua saguarekin mugitu bezain pronto soken luzerak eguneratu daitezten.

Propio Azken azpiatal honetan zuk nahi dituzun funtzioak definitu daitezke JAVA hizkuntza erabiliz; gero, programaren beste edozein ataletan erabiltzeko.

3. **Descripción.** Azken atal honetan, simulazioaren azalpena egiteko aukera ematen du programak. Azalpen hori simulazioa irekitzean agertzen da leiho independente batean, eta erabilgarria da, era laburtuan, simulazioaren oinarri teorikoa, jarduerak,... azaltzeko.

Beraz, funtsean, hauexek dira simulazio bat *EJSn* eraikitzeke pausoak: problema fisikoa ebatzi; higidura ekuazioak lehen ordeneko ekuazio diferentzial moduan adierazi eta *EJSko Modelo* atala

osatu; ingurune-grafikoaren eskematxoa paperean egin eta *Vista* atala eraiki; simulazioa txukuna geratzeko azken ukituak eman, eta azkenik, garapen teorikoa eta jarduerak *Descripción* atalean adierazi.

Jadanik, prest gaude. Har ditzagun fisikako problemak; ulertu, ebatzi, sakondu, eta... simulatu!

3. KAPITULUA

Proiektuaren garapena

*Kapitulu honetan proiektuaren oinarri diren simulazioak aurkeztuko dira. Bi atal nagusi bereiztuko ditugu: alde batetik, **DBH eta Batxilergo mailako simulazioak** izango ditugu; horietan, fisikako problema arrunt batzuk garatu, ebatzi eta simulatuko ditugu, ikasleak izan ditzakeen arazo didaktikoak aztertuz eta hauei konponbidea eskeintzen saiatuz. Bestetik, **unibertsitate mailako** hiru simulazio izango ditugu esku artean: **uhin grabitatorioak, pendulu bikoitza eta talka-uhinak**.*

Kapitulu honetan egitura konstante bat erabiliko dugu simulazioen nondik norakoak azaltzeko; hala, problema bakoitzarekin atal hauek jorratuko ditugu:

1. **Garapen teorikoa:** problemaren azpian dagoen fisika laburtuko da.
2. **Simulazioa:** *EJS* programan simulazioa egiteko eman diren pausoak laburtuko dira eta simulazioak berak nola funtzionatzen duen azalduko da.
3. **Jarduera didaktikoak:** Erabiltzaileak, simulazioaren zentzu didaktikoa barneratzeko, egin ditzakeen zenbait jarduera azalduko dira.

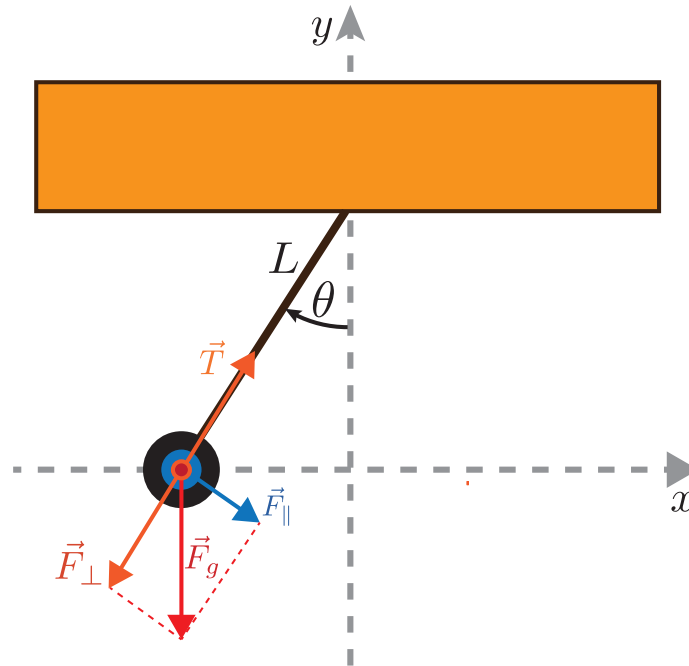
3.1 DBH eta Batxilergoko simulazioak

3.1.1 Pendulu sinplea

Batxilergo mailan aski ezaguna den problema dugu pendulu sinplearena. Maila horretan, lehen mailako hurbilketa bat egiten da osziladore harmoniko bat lortzeko; ikusiko dugunez, gure simulazioa gai izango da ekuazio diferentziala numerikoki ebatzi eta edozein angelurako higidura lortzeko.

Garapen teorikoa

Higidura-ekuazioa lortzea izango da gure helburua; pendulu sinplearen kasuan indarren-eskematik abiatuz ia zuzenean lor dezakegu. Hasteko, beraz, hauxe da gure problemaren irudia:



3.1 Irudia: *Pendulu sinplearen indarren eskema*

Grabitate-indarra da higidura eragingo duen kanpo-indarra. Irudian ikus daitekeenez, indar hori osagai paralelo (\parallel) eta perpendikularretan (\perp) banatzean, osagai perpendikularra sokak eragingandako tentsioarekin konpentsatuko da; aldez, urdinez margotutako osagai paraleloak higiarazi egingo du pendulua. Jarraian, Newtonen bigarren legea aplikatuko dugu θ koordinatu orokortua

erabiliz:

$$(\perp) \quad \vec{T} + \vec{F}_{\perp} = 0 \quad (3.1)$$

$$(\parallel) \quad F_{\parallel} = -mg \sin(\theta) = -ma = -mL\ddot{\theta} \quad (3.2)$$

Osagai paraleloko $(-)$ zeinua erraz ulertzen da; izan ere, pendulua θ angelua ezkererantz mugitzen dugunean, agertzen den indarra aurkako noranzkoan doa (eskuinerantz). Hala, gure pendulu sinplearen higidura-ekuazioa hauxe da:

$$\ddot{\theta} + \frac{g}{L} \sin(\theta) = 0 \quad (3.3)$$

Agertzen den $\sin(\theta)$ gaiak ekuazioa oso ez-lineala bihurtzen du eta zaila da angelu guztietarako soluzio analitiko bat aurkitzea; gure helburu didaktikoa dela-eta, hasierako angelu txikien¹ limitera jo genezake gauzak argiago ikusteko; bertan, $\sin(\theta) \sim \theta$ hurbilketa egin dezakegu lehen ordenean eta ekuazioa linealizatu; modu horretan,

$$\ddot{\theta} + \frac{g}{L}(\theta) = 0 \quad (3.4)$$

izango dugu: *osziladore harmonikoaren* ekuazioa! hala, $\omega = \sqrt{\frac{g}{L}}$ moduan identifikatuz, angelu txikietarako egokia den soluzio hurbildu analitikoa lortzen dugu:

$$\theta(t) = A \sin(\omega t) + B \cos(\omega t) \quad (3.5)$$

Hala ere, gure simulazioan zenbakizko metodoak erabiltzen ditugunez, gai izango gara (3.3) ekuazioa ebatzi eta penduluaren higidura hasierako angelu guztietan simulatzeko.

Simulazioa

2.kapituluan azaldu den moduan, *EJS* programa gai da lehen ordenako ekuazio diferentzialak numerikoki ebazteko. Gure penduluaren eboluzioa zehazten duen ekuazioa(3.3) bigarren ordenekoa denez, lehenengo lana lehen ordenako bi ekuazio lortzea da; horrela, $\omega = \frac{d\theta}{dt}$ moduan definituz,

$$\frac{d\omega}{dt} = \dot{\omega} = -\frac{g}{L} \sin(\theta) \quad (3.6)$$

$$\frac{d\theta}{dt} = \dot{\theta} = \omega \quad (3.7)$$

Orain, *EJS* gai da hasierako baldintzak² eman eta ekuazio diferentzial horiek askatzeko; behin gure penduluaren $\theta(t)$ dakigunean, egin behar den gauza bakarra koordinatu kartesiarretarako

¹Hurbilketa ongi dabil $\theta_0 \sim 20^\circ$ inguru arte.

²Hasierako konfigurazioa L sokaren luzera eta θ_0 hasierako angeluaren bidez definitu dira, eta, hasierako abiadura beti da zero.

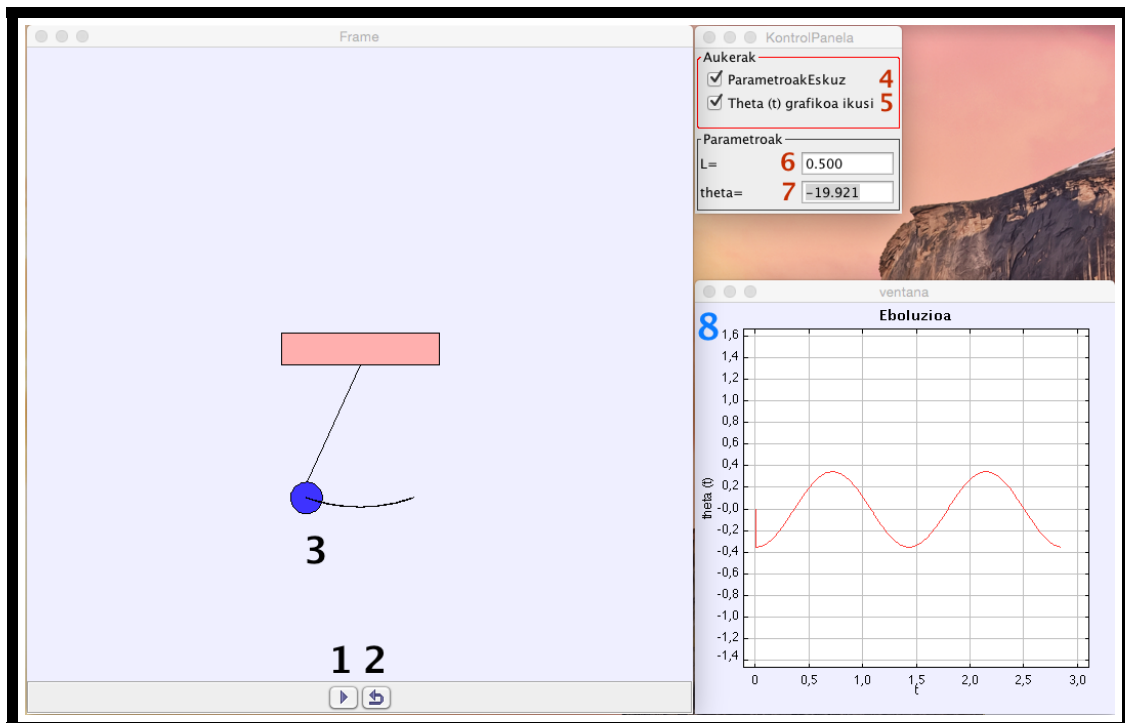
aldaketa da; hala, gure penduluaren koordinatu kartesiarrak (x, y) badira eta penduluaren hariak L neurtzen badu,

$$x = L \sin(\theta) \quad (3.8)$$

$$y = -L \cos(\theta) \quad (3.9)$$

Berriro ere $(-)$ zeinua agertzen da y koordinatuan; 3.3 irudian ikusten den moduan, y ardatzaren norantza positiboa gorraldera doa eta gure penduluaren higidura aurkako noranzkoan.

Jarraian, simulazio-leihoaren deskribapena egingo dugu; deskribapeneko zenbakiak hurrengo orrialdean dagoen irudiari dagozkie.



3.2 Irudia: *SIM_PenduluSimplea.jar* simulazioaren pantaila orokorra

- Kontrol-gunea (**beltzez**)
 1. **Play/pause botoia:** Simulazioa *abiarazi/pausarutzen* du.
 2. **Reset botoia:** Simulazioa hasierako egoerara itzularazten du.
 3. **Simulazio eremua:** Eremu interaktibo honetan gertatzen da simulazioa.
- Simulazioaren parametroak (**gorriz**)
 4. **Parametroak Eskuz botoia:** Penduluaren hasierako egoera zehazteko bi modu eskaintzen ditu; sakatua badago, hasierako baldintzak teklatura erabiliz sartu behar dira;

aldiz, sakatu gabe badago, saguaren bidez penduluaren muturra sakatu eta edozein puntutan utzi genezake.

5. **Theta(t) grafikoa ikusi botoia:** [Grafikoen Leihoa](#) ikusten den ala ez kontrolatzen du.
6. **L, sokaren luzera:** Penduluaren luzera aldatzen du.
7. **theta, hasierako angelua:** Hasierako angeluaren θ_0 balioa aldatzen du.

- Grafikoen leihoa ([urdinez](#))

8. **$\theta(t)$ grafikoa:** Simulazioa abian denean θ angeluaren balioak denboraren menpe marrazten dira leiho honetan.

Simulazioa jaitsi nahi izan ezkerre egin klik hurrengo estekan:

[SIM_PenduluSimplea.jar](#)

Jarduera didaktikoak

1. jarduera: Pendulu simplea

Jarduera didaktiko honetan, pendulu sinplearen simulazioa erabili eta aztertuko dugu. Beraz, ireki simulazioa eta...

- (a) Hurbilketa lineala aztertuko dugu lehenik; sakatu pendulua eta mugitu apur bat ($\theta_0 < 10^\circ$); ondoren, sakatu **Theta (t) grafikoa ikusi** aukera eta abiarazi simulazioa; hurbilketa linealean bazaude, grafiko sinusoidal bat ikusi beharko zenuke.
- (b) Saiatu hurbilketa linealaren limitea aurkitzen; horretarako, joan geroz eta hasierako angelu handiagoak ezartzen eta ikusi nola grafikoak sinusoidal izaera galtzen duen.
- (c) Orain, aukeratu **parametroak eskuz** aukera eta ipini $\theta = 180^\circ$; teorian, puntu hori penduluaren oreka egoera ezegonkorra da; hasieran, ikusiko duzunez, pendulua bertan geratzen da; hala ere, errore numerikoak direla medio, oreka egoeratik apur bat aterako da eta ezegonkorra denez, erori egingo da.

ditugun ekuazioak,

$$\dot{v}_x = 0 \quad (3.12)$$

$$\dot{x} = v_x \quad (3.13)$$

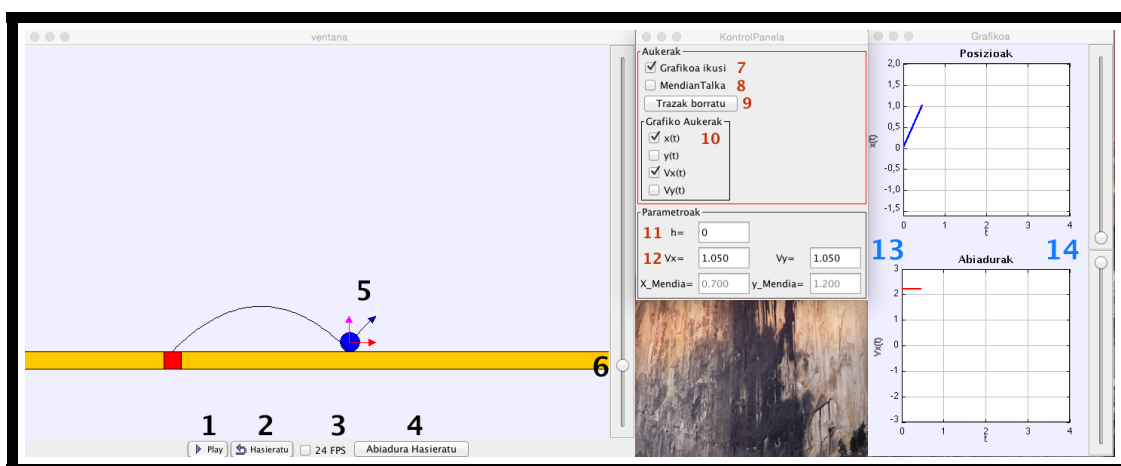
$$\dot{v}_y = -g \quad (3.14)$$

$$\dot{y} = v_y \quad (3.15)$$

Eboluzio-ekuazioak ezagututa, simulazioak nola funtzionatzen duen azalduko dugu jarraian.

Simulazioa

Nahiko erraza izan da simulazioa abian jartzea; funtsean, programak, hasierako abiaduraren balioa hartu, simulazio modua zehaztu³ eta (3.12)-(3.15) ekuazioak ebatzen ditu, eta, bolatxoak lurra jotzen duenean simulazioa gelditu egiten da. Hurrengo irudian, simulazioaren pantaila nagusia agertzen da; jarraian, deskribapena.



3.4 Irudia: SIM_TiroParabolikoa.jar simulazioaren pantaila nagusia

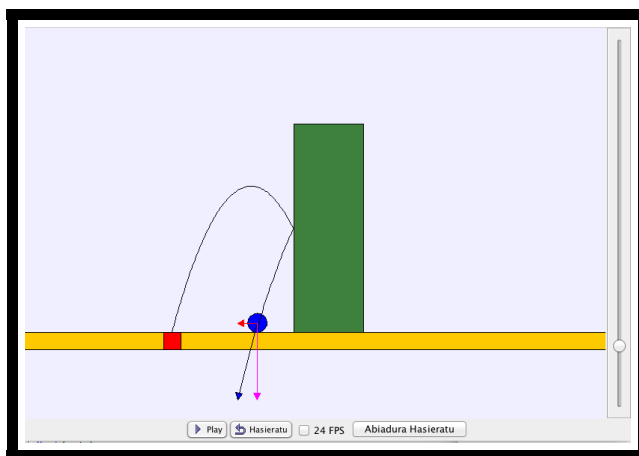
- Kontrol-gunea (beltzez)

1. **Play/pause botoia:** Simulazioa abiarazi/pausaraten du.
2. **Reset botoia:** Simulazioa hasierako egoerara itzularazten du.
3. **24 FPS botoia:** Sakatzen denean, simulazioa 24 fotograma/segundo-ko abiaduraz ejekutatzen da.
4. **Abiadura Hasieratu botoia:** \vec{v}_0 bolatxoaren abiadura hasieratu egiten du⁴. Goiko

³Gero agertuko denez, gehigarri batzuk ditu simulazioak.

⁴Hasierako abiadura hori $(v_x, v_y) = (1, 1)$ da.

- irudian botoi hau sakatu da bolatxoak lurra jo ondoren, horregatik agertzen dira bektoreen geziak hasieratuta.
5. **Hasierako abiaduren bektoreak:** ikusten diren hiru bektoreak interaktiboak dira; gezietan klik egin, mugitu, eta hasierako abiaduraren balioa nahi bezala zehaztu daiteke.
 6. **Zoom scroll-a:** Simulazio-leihoaren eskala aldatzeko balio du.
- Simulazioaren parametroak (**gorriz**)
7. **Grafikoa ikusi botoia:** Grafikoen Leihoa ikusten den ala ez kontrolatzen du.
 8. **Mendian talka botoia:** Simulazioaren lehenengo modu gehigarria da; klik egitean oztopo laukizuzen bat agertzen da pantailan; honek, ariketa ezberdin gehiago ebaztea ahalbidetzen du⁵. Itxura hau dauka simulazioak modu hori aktibatzean:

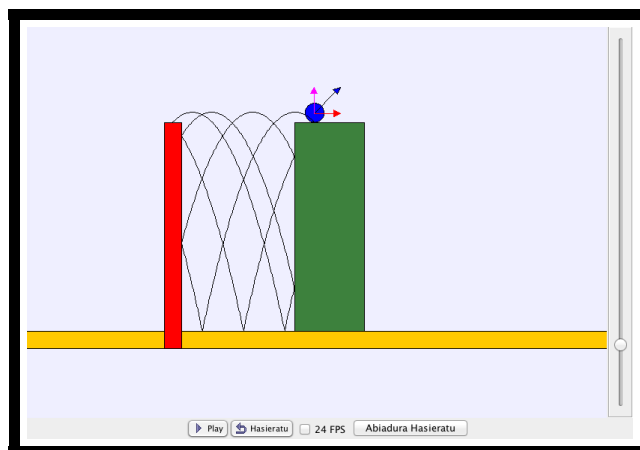


3.5 Irudia: *Mendian talka* modu gehigarria

- Ikus daitekeenez, programa gai da bolatxoak mendia noiz jo duen antzemateko; eta, talka ostean, $v'_x = -v_x$ aplikatzen da simulazio errealistagoa lortuz. Gainera, abiaduraren bektorea eta bere osagai kartesiarrak simulazioarekin batera aldatzen dira, didaktikoagoa bihurtuz. Mendiaren parametroak (ezker paretaren x_m posizioa eta mendiaren altuera) 11.puntuako gunean aldatu daitezke.
9. **Trazak ezabatu botoia:** Bolatxoaren ibilbidea adierazten duten lerro beltzak ezabatzen ditu.
 10. **Grafikoen aukerak:** Grafikoen leihoan marrazten diren grafikoa aukeratzen dira gune honetan.

⁵Adibidez, kalkulatu bolatxoaren hasierako abiadura mendiaren bestaldera igarotzeko. moduko problemetan lagungarria izan daiteke.

11. **Datuak eskuz sartzeko gunea:** Hasierako baldintzak kontrola daitezke dagozkien eremuak aldatuz.
12. **h (hasierako altuera) botoia:** Simulatzeko bigarren modu gehigarria dugu kontrol hau; aldagai honek bolatxoari hasierako altuera ematen dio, eta, horrela, ariketa mota gehiago ebatzi daitezke. Hauxe da modu gehigarri honen adibide bat:



3.6 Irudia: *Hasierako altuera eta Mendiaren talka* elkarlanean

Kasu honetan, bolatxoa h altueratik abiatu da; ikus daitekeenez, sortu berri den bloke gorriarekin ere talka egin dezake. Gainera, bolatxoak lurra jo ondoren berriro **play** botoia sakatzen badugu, hasierako abiadura berriaren $v'_y = -v_y$ egiten da. Azkenik, aipatu, mendiaren modua edota $h > 0$ aktibatzean zorua berdefinitu egiten dela (bolatxoa mendiaren gailurrean geratu da)

– Grafikoen leihoa (**urdinez**)

13. **Bolatxoaren posizio/abiadura grafikoak** Goiko grafikoan bolatxoaren $x(t)$ eta $y(t)$ (aukeraren arabera) posizioen grafikoa egiten da; eta, behekoan, $v_x(t)$ eta $v_y(t)$ abiadurena.
14. **Grafikoen zoom-ak (scroll moduan):** Grafiko bakoitzaren alboan dagoen *scroll*-a mugituz eskala aldatu daiteke.

Simulazioa jaisteko, egin klik hurrengo estekan:

[SIM_TiroParabolikoa.jar](#)

Jarduera didaktikoak

1. jarduera: Tiro Parabolikoa

Ireki simulazioa eta jarraitu ondoko pausoak:

- (a) Sakatu hasierako abiaduraren bektoreak eta mugitu nahieran; ikusiko duzunez, x eta y ardatzeko proiektzioak dagozkien norabideetan soilik mugitu daitezke. Hasierako abiadura aukeratu ondoren, abiarazi simulazioa.
- (b) Kontrol-panelean, egin orain $v_x = 0$ eta ikusi y ardatzean gertatzen den Higidura Zuzen Uniformeki Azeleratua (HZUA).
- (c) Aldatu h , altueraren kontrola, eta ipini zero ez den balio bat; orain, ziurtatu v_y zero dela eta eman hasierako abiadura horizontala (v_x). Paperean irismena kalkula dezakezu eta gero grafikoan emaitza egiaztatu.

2. jarduera: Mendi gainaren bila!

Jarduera honetan, **Mendian Talka** aukerarekin problema erraz bat ebatziko dugu. Hasteko, hartu papera eta ebatzi hurrengo problema: *Kalkulatu bolatxoaren hasierako abiadura mendiaren magalera iristeko (Mendia 0.4m zabal da)*. Soluzio ugari daudenez, modu didaktikoan planteatu daiteke ikasgela batean, eta gero ikasleen erantzun ezberdinak simulatu.

- (a) Sakatu **Mendian Talka** aukera eta aukeratu zure mendiaren parametroak (**X_mendia** eta **Y_mendia**). Jarraian, kalkulatu duzun hasierako abiadura ezarri pilotari eta abiarazi simulazioa. Asmatu ezker, pilota mendiaren magalean geratuko da.
- (b) Egin gauza bera, baina oraingoan bolatxoari hasierako altuera bat eman.

3.2 Unibertsitate mailako simulazioak

Sarreran azaldu bezala, unibertsitate mailako hiru simulazio garatu eta aztertuko ditugu atal honetan.

3.2.1 Uhin-grabitatorioak

Uhin grabitatorio lauak *Einsteinen Ekuazioetan* hurbilketa lineala egitean lortzen den erantzun naturala dira.

Garapen teorikoa

Esan bezala, hurbilketa lineala da gure abiapuntua; Horrela, eremu grabitatorioa ahula dela onartuko dugu, baina ez estatikoa. Matematikoki aztertzeko, metrika berria zehaztu beharra dago. Metrika hori *ia-Minkowskiarra* izango da, baina $h_{\mu\nu}$ gai txiki bat (perturbazioa) gehituko diogu:

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} \quad \text{non } |h_{\mu\nu}| \ll 1 \text{ den} \quad (3.16)$$

Hemen, $\eta_{\mu\nu}$ Minkowskiren metrika⁶ dugu eta $g_{\mu\nu}$ problema ebazteko erabiliko dugun metrika aldatua.

Metrikaz gain, **transformazio infinitesimal lokalak** ere kontsideratuko ditugu; horietan, lehen ordeneko aldaketa txiki bat eragingo diegu koordenatuei eta bertatik **geometria linealizatua** lortuko dugu. Has gaitzen koordenatuen aldaketa infinitesimal hori kontsideratzen:

$$x^{\mu'} = x^\mu + \xi^\mu(x) \quad \text{non } |\xi^\mu(x)| \ll 1 \text{ den} \quad (3.17)$$

Geometria linealizatu hau lortze aldera, hurrengo lana, (3.16) metrikaren koordenatu aldaketekiko jarrera nolakoa den zehaztea izango da; horretarako, koordenatuen arteko deribatuak kalkulatu ditugu,

$$\frac{\partial x^{\mu'}}{\partial x^\rho} = \delta_\rho^\mu + \xi_{,\rho}^\mu(x) \quad (3.18)$$

$$\frac{\partial x^\rho}{\partial x^{\mu'}} = \delta_\mu^\rho - \xi_{,\mu}^\rho(x) \quad (3.19)$$

⁶Espazio-denbora laua deskribatzen duen metrika; bere adierazpen matriziala osagai kartesiarretan hauxe da:

$$\eta_{\mu\nu} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Ondoren, metrikaren koordenatuak aldatuz,

$$\begin{aligned} g_{\mu'\nu'} &= \frac{\partial x^\rho}{\partial x^{\mu'}} \frac{\partial x^\sigma}{\partial x^{\nu'}} g_{\rho\sigma} = (\delta_\mu^\rho - \xi_{\nu,\mu}^\rho)(\delta_\nu^\sigma - \xi_{\mu,\nu}^\sigma)(\eta_{\rho\sigma} + h_{\rho\sigma}) \\ &= \eta_{\mu\nu} + \underbrace{h_{\mu\nu} - \xi_{\nu,\mu} - \xi_{\mu,\nu}}_{h_{\mu'\nu'}} = \eta_{\mu'\nu'} + h_{\mu'\nu'} \end{aligned} \quad (3.20)$$

Ikus daitekeenez, $h_{\mu'\nu'} = h_{\mu\nu} - \xi_{\nu,\mu} - \xi_{\mu,\nu}$ moduan transformatzen da hurbilketa lineala egitean. Einsteinen ekuazioetan hurbilketa lineala egiterakoan gomendagarria da $h_{\mu\nu}$ -ren aztarnarik gabeko zatia definitzea; hala, $\bar{h}_{\mu\nu} \equiv h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}h$ definituko dugu, $h = \eta^{\mu\nu}h_{\mu\nu}$ izanik. Hurbilketa egitean, koordenatu sisteman askatasun bat dagoela ikusten da orduan; askatasun horri **Lorenzen gaugea** deitzen diogu eta funtsean $\bar{h}_{\mu\nu}{}^{,\nu} = 0$ egitea ahalbidetzen digu. Hori guztia kontuan hartuz, honelaxe geratzen dira Einsteinen ekuazioak hurbilketa lineala eta Lorenzen gaugea kontsideratu ondoren:

$$\text{Energia momentuaren tentsorea: } \square^2 \bar{h}_{\mu\nu} = -2\kappa T_{\mu\nu} \quad (3.21)$$

$$\text{Lorenzen gaugea: } \bar{h}_{\mu\nu}{}^{,\nu} = 0 \quad (3.22)$$

$$\text{Metrika: } g_{\mu\nu} = \eta_{\mu\nu} + \bar{h}_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}\bar{h} \quad (3.23)$$

Uhin grabitatorioen soluzioa lortzeko, espazio hutsa kontsideratuko dugu; dakigunez, bertan, energia-momentuaren tentsorea nulua da ($T_{\mu\nu} = 0$) eta, orduan, Einsteinen ekuazioak:

$$\square^2 \bar{h}_{\mu\nu} = \left(\frac{1}{c^2} \frac{\partial^2}{\partial t^2} - \nabla^2 \right) \bar{h}_{\mu\nu} = 0 \quad (3.24)$$

$$\bar{h}_{\mu\nu}{}^{,\nu} = 0 \quad (3.25)$$

Ekuazio hauen soluzio dira modu honetan definituriko **uhin-lauak**:

$$\text{Uhin-lauak: } \bar{h}_{\mu\nu} = A_{\mu\nu} e^{ik_\rho x^\rho} \quad (3.26)$$

$$\text{Ortogonalitate baldintza: } k_\rho k^\rho = 0 \quad (3.27)$$

$$\text{Zeharkako anplitude tentsorea } A_{\mu\nu} k^\nu = 0 \quad (3.28)$$

Jarraian, x norabidean hedatzen diren uhinak kontsideratuko ditugu $k^\mu = (k, k, 0, 0)$ uhin-bektorea dutenak. (3.28) erabiliz ondoko baldintza lortzen dugu:

$$A_{00} + A_{0x} = A_{x0} + A_{xx} = A_{y0} + A_{yx} = A_{z0} + A_{zx} = 0 \quad (3.29)$$

Koordenatu minkowskietan jarraitzen badugu ordea, ez dugu emaitza argirik lortuko⁷; hala ere, lehen frogatu dugunez, koordenatu aldaketa *infinitesimal* bat egiten badugu badakigu

⁷10 polarizazio ezberdin gainezartzen baitira.

metrika berria (3.20) ekuazioaren arabera aldatuko dela; hala, modu honetan aukeratuko dugu ϵ_μ :

$$\epsilon_\mu = \frac{i}{4k} (A_{yy} + A_{zz} + 2A_{00}, A_{yy} + A_{zz} - 2A_{00}, 4A_{0y}, 4A_{0z}) e^{ik(x-ct)} \quad (3.30)$$

Koordenatu aukeraketa hau *TT gauge* moduan ezagutzen da; eta bere abantaila argi ikusten da perturbazio matrizea ($h_{\mu\nu}$) begiratzen badugu:

$$h_{\mu\nu}^{(TT)} = \bar{h}_{\mu\nu}^{(TT)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & A_+ & A_\times \\ 0 & 0 & A_\times & -A_+ \end{pmatrix} e^{ik(x-ct)} \quad (3.31)$$

Ikus daitekeenez, gure perturbazioak azterna nulua du eta simetrikoa da; gainera, bi polarizazio bereizten dira bertan: + polarizazioa (\leftrightarrow eta \updownarrow norabideetan) eta \times polarizazioa (\nearrow eta \searrow norabideetan).

Amaitzeko, kontsidera dezagun koordenatu hauetan gelditu dauden partikula hurbil askeen multzo bat eta aukera dezagun horietako bat erreferentzia bezala; koordenatu horietan partikula guztien posizio erlatiboak konstanteak dira baina ez beraien arteko distantziak⁸. Ikuspegi intuitibo hobe lortzeko koordenatu aldaketa bat egingo dugu, eta koordenatu berri horietan distantziak *euklidearrak* izango dira eta beraz, *ikus*i egin ahalko dugu partikulen higidura nolakoa den; hauxe da egin beharreko koordenatu aldaketa:

$$X^i \equiv \zeta^i + \frac{1}{2} \delta^{ij} h_{jk}^{(TT)} \zeta^k \quad (3.32)$$

eta, orduan,

$$X^x = \zeta^x \quad (3.33)$$

$$X^y = \zeta^y + \frac{1}{2} (A_+ \zeta^y + A_\times \zeta^z) e^{ik(x-ct)} \quad (3.34)$$

$$X^z = \zeta^z + \frac{1}{2} (A_\times \zeta^y - A_+ \zeta^z) e^{ik(x-ct)} \quad (3.35)$$

Horrela, x finko bakoitzerako zeharkako sekzioen eboluzioa jadanik kalkulatu dugunez prest gaude *EJS*ra itzuli eta simulazioa egiten hasteko.

Simulazioa

Uhin grabitatorioak bi modutan simulatu dira: 2D moduan eta 3D moduan.

⁸Distantziak $\sqrt{g_{ij}\xi^i\xi^j}$ moduan kalkulatu dira eta badakigu g_{ij} metrika ez dela konstantea; beraz, distantziak ere ez.

2Dko simulazioa

Simulazioa egiteko lehenengo lana gure uhin grabitazionalaren zeharkako sekzioa irudikatzen duen puntuz osaturiko zirkunferentzia sortzea izan da. Horretarako, n osagaiko bi bektore sortu dira, $X0[i]$ eta $Y0[i]$, eta bakoitzean partikula bakoitzaren hasierako koordinatuak ezarri dira unitate-zirkunferentzia bat osatuz:

```

1  for (int i=0; i<n ; i++) {
2      X0[i]=Math.cos(2*Math.PI/n*i);
3      Y0[i]=Math.sin(2*Math.PI/n*i);
4  }
```

Puntu horiek, uhin grabitatorioaren oreka egoera adierazten dute; hurrengo lana $h_{\mu\nu}$ perturbazioak eragiten duen higidura simulatzea da. Soluzio analitikoa badakigunez, $EJSn$ denbora diskretizatu eta modu honetan idatzi da puntuen denbora-eboluzioa. ($c = 1$ egin da ekuazioetan)

```

1  if (D3==false){
2  for (int i=0; i<n; i++){
3      X[i]=X0[i]+Nm*(0.5*Ap*X0[i]*Math.cos(v*t)+0.5*Ax*Y0[i]*Math.cos(theta0
          +v*t));
4      Y[i]=Y0[i]+Nm*(0.5*Ax*X0[i]*Math.cos(theta0+v*t)-0.5*Ap*Y0[i]*Math.cos
          (v*t));
5  }
6  t=t+dt;
7  }
```

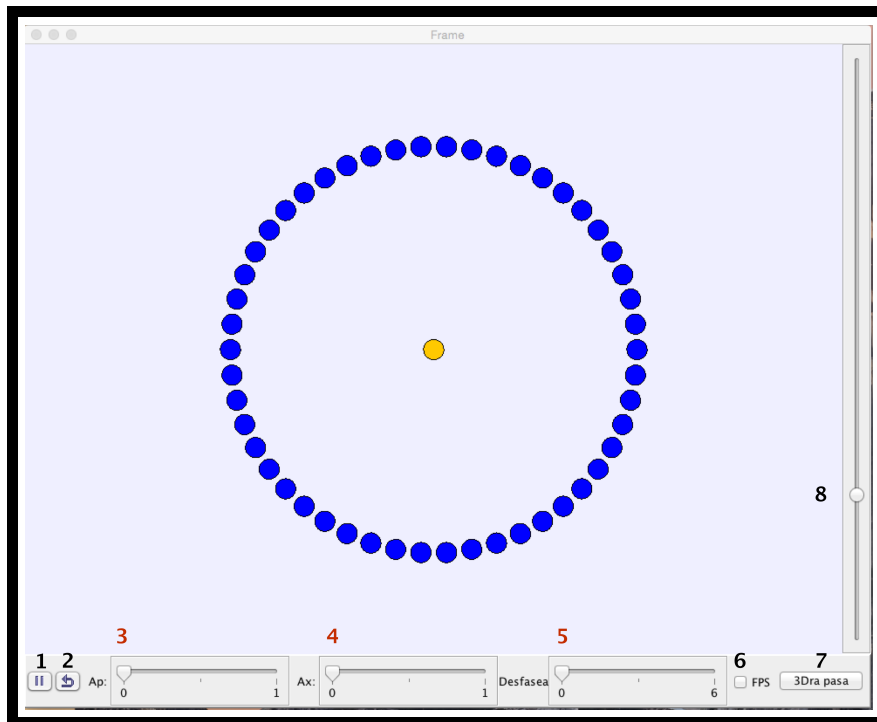
Hasieran ikusten den $D3==false$ baldintzak 2Dko uhinak simulatu behar dituela zehazten dio programari. Bestetik, $e^{ik(x-ct)}$ atalaren zentzu oszilakorra kosinuaren bidez adierazi da (zati erreala) eta θ_0 fase-diferentzia⁹ bat gehitu da bi polarizazio egoerak koherenteki gainezartzen direlako. Ekuazioetan agertzen diren A_x eta A_p konstanteak, berriz, uhinaren polarizazioen A_x eta A_+ anplitudeak dira, hurrenez hurren. Azkenik, denbora eguneratzen da.

Hauxe da simulazioaren 2D irudia eta kontrolen deskribapena:

- Kontrol-gunea (beltzez)

1. **Play/pause botoia:** Simulazioa *abiarazi/pausaritzen* du.
2. **Reset botoia:** Simulazioa hasierako egoerara itzularazten du.

⁹Desfasea ulertzeko bi uhin hedatzen direla onartuko dugu (polarizazio bakoitzeko uhin bat): lehen A_+ polarizazioarekin soilik ($A_x = 0$), eta, bigarrena, A_x polarizazioarekin ($A_+ = 0$) eta **lehenengoarekiko θ_0 desfasetuta**.



3.7 Irudia: *SIM_UhinGrabitorioak.jar* simulazioaren hasierako pantaila (2D)

6. **FPS botoia:** Sakatu gabe dagoenean simulazioa 24 fotograma/segundoko abiaduraz ejiakutatzen da. Sakatzen denean berriz, simulazioak segundoko ahalik eta fotograma gehien agertzen ditu.
 7. **3Dra pasa botoia:** 3D modua aktibatzen da.
 8. **Zoom barra:** *Zoom-barra* gora edo behera mugituz simulazioa urrundu edo gerturatu egin daiteke.
- Simulazioaren parametroak (gorriz)
 3. **Ap kontrola:** A_+ polarizazioaren anplitudea kontrolatzen du $[0,1]$ tartean.
 4. **Ax kontrola:** A_x polarizazioaren anplitudea kontrolatzen du $[0,1]$ tartean.
 5. **Desfasea:** A_+ eta A_x zenbaki konplexuen arteko desfase erlatiboa kontrolatzen da $[0,2\pi]$ tartean.

3Dko simulazioa

Simulazioan *3Dra pasa* botoia sakatzean, espazioan hedatzen den uhin grabitatorioa ikus daiteke hiru dimentsiotan. Kontzeptualki eman beharreko pausoa erraza da: z -ardatzean zehar mugitzean, jatorrizko oszilazioaren beste fase bat ikusten dugu. Hau programatzeko, z -ardatza diskretizatu egin da d luzerako tartetean; gero, $Z[i]$ bakoitzean zirkunferentzia bat marraztu da, uhinaren oreka-sekzioa adieraziz:

```

1  for (int i=0; i<m ; i++) {
2      for(int j=0; j<n ;j++){
3          X30[i*n+j]=Math.cos(2*Math.PI/n*j);
4          Y30[i*n+j]=Math.sin(2*Math.PI/n*j);
5          Z3[i*n+j]=d*(i-m/2);
6      }
7  }

```

Bertan agertzen den n zenbakiak sekzio bakoitzean dagoen partikula kopurua adierazten du; m zenbakiak, berriz, z -ardatzean zenbat sekzio ezberdin dauden. Hurrengo lana, $n \cdot m$ puntu horiek banaka-banaka kokatzea da. Horretarako, lehenik $z = 0$ zirkunferentzia n puntuz betetzen da ($i = 0$ eta j 0-tik n -ra); jarraian, bigarren zirkunferentzia osatzen da $z = d(1 - m/2)$ ¹⁰ puntuan. Prozesua errepikatu egiten da d distantziaz banatutako n puntuko m zirkunferentzia izan arte.

Amaitzeko, sekzio bakoitzean denbora-eboluzioa gehitzen da, baina, orain, $\cos(Z[i] + \theta_0 - ct)$ moduko gaiak agertzen dira, uhinaren 3D izaera agerian utziz.

```

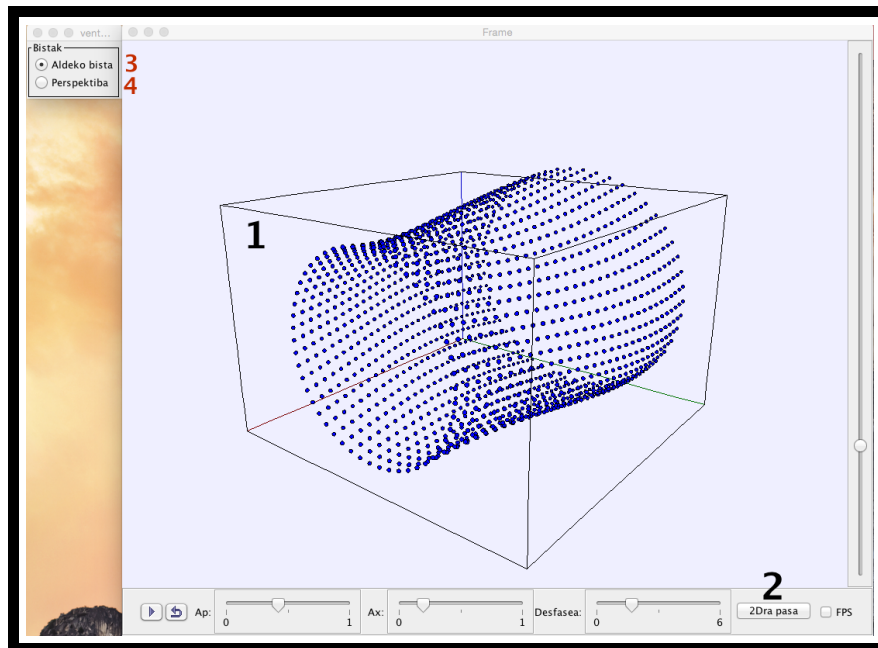
1  if (D3==true){
2  for (int i=0; i<n*m; i++){
3      X3[i]=X30[i]+Nm*(0.5*Ap*X30[i]*Math.cos(v*t+Z3[i])+0.5*Ax*Y30[i]*Math.
4          cos(theta0+v*t+Z3[i]));
5      Y3[i]=Y30[i]+Nm*(0.5*Ax*X30[i]*Math.cos(theta0+v*t+Z3[i])-0.5*Ap*Y30[i]
6          ]*Math.cos(v*t+Z3[i]));
7  }
8  t=t+dt;
9  }

```

Hauxe da 3D moduaren pantaila orokorra: Ikus daitekeenez, kontrolak 2D moduaren oso antzekoak direnez, desberdintasunak soilik azalduko ditugu:

- Leiho nagusia (beltzez)
 1. **3D pantaila interaktiboa:** 3D moduaren leihoan saguarekin klik egin eta mugitzen bada, simulazioaren ikuspegia kontrolatzen da.
 2. **2Dra pasa botoia:** 2D modua aktibatzen da.
- Ikuspegiaren aukerak (gorriz)
 3. **Aldeko bista aukera:** Simulazioaren ikuspegia zilindroaren alde bat ikustera finkatzen du.

¹⁰Simulazioko bista $-m \cdot d$ tik $m \cdot d$ -ra doanez, $z = d \cdot i$ moduan definituko bagenu pantaila erdia hutsik agertuko litzateke.



3.8 Irudia: *SIM_UhinGrabitorioak.jar* simulazioaren 3D moduaren pantaila

4. **Perspektiba aukera:** Aldeko bista aukeratu aurretik zenuen ikuspegira itzultzen du simulazioa

Simulazioa jaisteko, egin klik hurrengo estekan:

[SIM_UhinGrabitorioak.jar](#)

Jarduera didaktikoak

Orain, uhin-grabitatorioen simulazioarekin egin daitezkeen zenbait jarduera proposatuko dira.

1. jarduera: 2Dko uhin-grabitatorioen polarizazioa

Jarduera honetan, 2Dko uhinen simulazioa erabiliko dugu, eta agertzen diren bi polarizazioak aztertu.

- (a) Aldatu A_p scroll-a; teorian ikusi bezala, (+) polarizazioa kontrolatzen da horrela. Abiarazi simulazioa eta ikusi nola jokatzen duen uhin-grabitatorioak. Ba al du zentzurik polarizazio-egoera horri (+) deitzeak?
- (b) Jarri zeroan A_+ eta egin proba orain A_x scroll-a aldatzen; zer nabaritzen da orain? nolakoa da uhin berria? Egiaztatu nolabaiteko \times forma bat ikusten dela.

2. jarduera: Desfasearen gorabeherak

Jarduera honetan, **desfasea** parametroak zer egiten duen aztertuko dugu.

- Hasteko, ipini simulazioa A_+ polarizazioarekin eta **desfasea** = 0. Orain, mugitu **desfasea** scroll-a; gertatu al da zerbait? Desfaseak \times polarizazio-egoeraren $+$ -ekiko fase-aldaketa kontrolatzen duenez, ez luke ezer gertatu behar A_+ polarizazioa soilik duen uhinean.
- Jarraian, pausatu simulazioa eta mugitu A_\times polarizazio-egoeraren scroll-a. Orain, ikusi desfaseak duen eragina. Horretarako; abiarazi simulazioa **desfasea** = 0 izanik eta gogoan hartu gertatzen dena; gero, mugitu **desfasea**-ren scroll-a eta sakatu berriz ere play botoia.
- Egin proba ezberdinak A_\times eta A_+ polarizazioen balio ezberdinetarako.

3. jarduera: Uhin-grabitatorioak 3 dimentsiotan

Uhin-grabitatorioak 3Dtan aztertuko ditugu orain.

- Sakatu 3Dra **pasa** botoia; leiho berri bat ireki da, puntuen distribuzio zilindriko batekin. Hasteko, ipini A_p polarizazioaren balio bat; egiaztatu 3Dra pasatzearen eragin bakarra elkarrekin desfasatuta dauden^a 2Dko simulazio asko bata bestearen atzean jartzea dela.
- Sakatu saguarekin leihoan eta mugitu kamara uhinari parez pare begiratzeko moduan ipiniz. Gai al zara zeharkako-sekzioa ikusteko? Sakatu 2Dra **pasa** botoia eta egiaztatu higidura berdina ikusten duzula.
- Aukeratu goi-ezker muturrean dagoen **aldeko bista** aukera; zilindroaren alde bat ari zara ikusten orain. Aldatu **Desfasea** scroll-a eta egiaztatu uhinaren aldeko forman eraginik ez duela. (Lehenago bistara itzultzeko sakatu **perspektiba** botoia)
- Jolastu polarizazio-parametroak aldatzen eta itxura ezberdinetako uhinak sortzen.

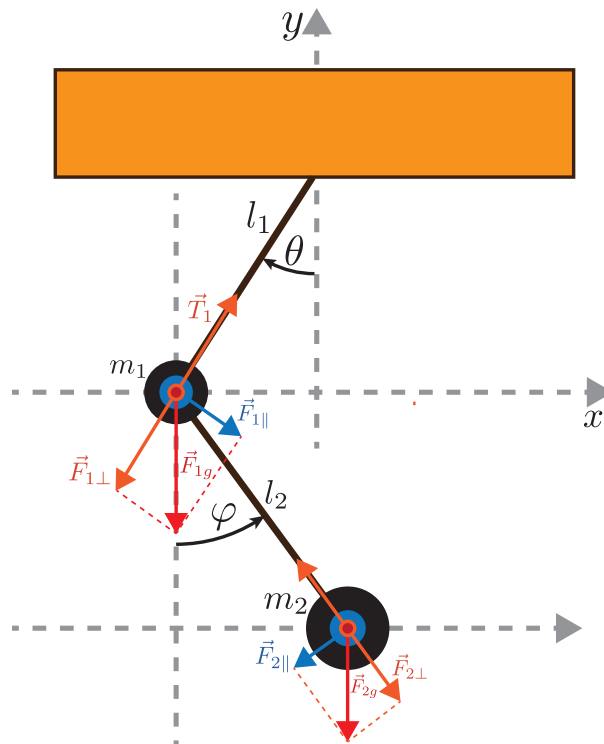
^a $e^{i(kz-\omega t)}$ ko z posizioa aldatzen delako.

3.2.2 Pendulu bikoitza

Hurrengo simulazioan 3.1.1 ataleko pendulu sinpleari bigarren pendulu bat gehituko diogu eta **pendulu bikoitzaren** problema ebartziko dugu; ikusiko dugun moduan, bi pendulu akoplatu hauek asko konplikatuko dituzte ebartzi beharreko ekuazioak eta lortuko dugun higidura kaotikoa izango da, orokorrean.

Oinarri teorikoa

Betiko moduan, lehenik eta behin pendulu bikoitzaren higidura-ekuazioak lortzea izango dugu helburu; horretarako, egin dezagun indarren-eskema. Irudian agertzen denez, gure bi penduluek



3.9 Irudia: Pendulu Bikoitzaren indarren-eskema

masa ezberdinak izan ahalko dituzte (m_1 eta m_2), eta soken luzera ere gure aukerakoa izango da. Hala ere, pendulu sinplearekin ez-bezala, kasu honetan ezin ditugu Newtonen ekuazioak zuzenean aplikatu dinamika konplexuegia baita. Orduan, zer egin?

Gure helburua problemaren atzean dagoen ekuazio diferentziala lortzea denez mekanika analitikoaz balia gaitetzke. Gure kasuan, *EJS* sistemak lehen mailako ekuazio diferentzialak askatzen

dituenez, formulazio hamiltondarra¹¹ erabiliko dugu problema ebazteko. 3.9 irudian ikusten denez, bi askatasun gradu ditu sistemak: θ , lehenengo penduluaren angelua eta φ , bertikalak bigarren penduluarekin osatzen duen angelua. Horiek horrela, lehen ordenako 4 ekuazio diferentzial aurkitu eta ebatzi beharko ditugu.

Hasteko, gure penduluen koordenatu kartesiarrak θ eta φ koordenatu orokortuen menpe idatziko ditugu:

$$\text{1.pendulua} \begin{cases} x_1 = l_1 \sin(\theta) \\ y_1 = -l_1 \cos(\theta) \end{cases} \quad (3.36)$$

$$\text{2.pendulua} \begin{cases} x_2 = \underbrace{l_2 \sin(\theta)}_{x_1} + l_2 \sin(\varphi) \\ y_2 = \underbrace{-l_1 \cos(\theta)}_{y_1} - l_2 \cos(\varphi) \end{cases} \quad (3.38)$$

$$(3.39)$$

Bi angeluen noranzko positiboa modu berean definitu da: erlojuaren orratzen kontra.

Gure penduluen ekuazioak lortu aurretik, azal ditzagun formulazio hamiltondarra aplikatzeko behar ditugun kontzeptu matematikoak.

Sistema fisiko baten *hamiltondarra* modu honetan definitzen da:

$$H = \sum_i \dot{q}_i p_i - L \quad (3.40)$$

Bertan, \dot{q}_i gaiak koordenatu orokortuen denborarekiko deribatuak dira, p_i gaiak momentu orokortuak eta, L , sistemaren lagrangearra¹². Momentu orokortuak lagrangearra deribatuz lor daitezke,

$$p_j = \frac{\partial L}{\partial \dot{q}_j} \quad (3.41)$$

eta gure sistema fisikoaren hamiltondarra ezagutzen dugunean, higidura ekuazioak zuzenean lortzen dira (3.40) deribatuz:

$$\dot{p}_j = -\frac{\partial H}{\partial q_j} \quad (3.42)$$

$$\dot{q}_j = \frac{\partial H}{\partial p_j} \quad (3.43)$$

Orain, prest gaude pendulu bikoitzaren higidura ekuazioak lortzeko. Hasteko, sistemaren energia zinetikoa (T) eta potentziala (V) kalkulatu ditugu koordenatu orokortuetan, gero lagrangearra

¹¹Dakigunez, mekanika analitikoan bi joera nagusi bereiz ditzakegu: *formulazio lagrangearra* eta *formulazio hamiltondarra*. Ebatzi beharreko problemak n askatasun gradu baditu, *formulazio lagrangearrak* bigarren ordenako n ekuazio diferentzial ematen ditu, eta, *formulazio hamiltondarrak*, berriz, lehen ordenako $2n$ ekuazio.

¹²Dakigunez, $L = T - V$.

($L = T - V$) lortzeko.

Energia potentziala:

$$\begin{aligned} V &= m_1 g y_1 + m_2 g y_2 \\ &= - (m_1 + m_2) g l_1 \cos(\theta) - m_2 g l_2 \cos(\varphi) \end{aligned} \quad (3.44)$$

Energia zinetikoa:

$$\begin{aligned} T &= \frac{1}{2} m_1 (\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2} m_2 (\dot{x}_2^2 + \dot{y}_2^2) \\ &= \frac{1}{2} m_1 l_1^2 \dot{\theta}^2 \underbrace{(\sin^2(\theta) + \cos^2(\theta))}_1 + \frac{1}{2} m_2 [l_1^2 \dot{\theta}^2 \underbrace{(\sin^2(\theta) + \cos^2(\theta))}_1 \\ &\quad + l_2^2 \dot{\varphi}^2 \underbrace{(\sin^2(\varphi) + \cos^2(\varphi))}_1 + 2l_1 l_2 \dot{\theta} \dot{\varphi} \underbrace{(\sin(\theta) \sin(\varphi) + \cos(\theta) \cos(\varphi))}_{\cos(\theta - \varphi)}] \\ &= \frac{1}{2} l_1^2 (m_1 + m_2) \dot{\theta}^2 + \frac{1}{2} l_2^2 m_2 \dot{\varphi}^2 + l_1 l_2 m_2 \dot{\theta} \dot{\varphi} \cos(\theta - \varphi) \end{aligned} \quad (3.45)$$

Orduan, **lagrangearra:**

$$\begin{aligned} L = T - V &= \overbrace{\frac{1}{2} l_1^2 (m_1 + m_2) \dot{\theta}^2 + \frac{1}{2} l_2^2 m_2 \dot{\varphi}^2 + l_1 l_2 m_2 \dot{\theta} \dot{\varphi} \cos(\theta - \varphi)}^T \\ &\quad + \underbrace{(m_1 + m_2) g l_1 \cos(\theta) + m_2 g l_2 \cos(\varphi)}_V \end{aligned} \quad (3.46)$$

Behin lagrangearra eskuartean dugula, momentu orokortuak kalkulatuko ditugu.

Momentu orokortuak:

$$p_\theta = (m_1 + m_2) l_1^2 \dot{\theta} + m_2 l_1 l_2 \dot{\varphi} \cos(\theta - \varphi) \quad (3.47)$$

$$p_\varphi = m_2 l_2^2 \dot{\varphi} + m_2 l_1 l_2 \dot{\theta} \cos(\theta - \varphi) \quad (3.48)$$

Hamiltondarra lortzeko, (3.40) ekuazioa erabiliko dugu; ikus daitekeenez, Hamiltonen ekuazioa lagrangearraren Legendreren transformazio bat besterik ez da.

$$H = \dot{\theta} p_\theta + \dot{\varphi} p_\varphi - L \quad (3.49)$$

Momentu orokortuak hamiltondarrean ordezkatzuz

$$\begin{aligned} H(\theta, \varphi, \dot{\theta}, \dot{\varphi}) &= \frac{1}{2} l_1^2 (m_1 + m_2) \dot{\theta}^2 + \frac{1}{2} l_2^2 m_2 \dot{\varphi}^2 + l_1 l_2 m_2 \dot{\theta} \dot{\varphi} \cos(\theta - \varphi) \\ &\quad - (m_1 + m_2) g l_1 \cos(\theta) - m_2 g l_2 \cos(\varphi) \end{aligned} \quad (3.50)$$

Hauxe da gure hamiltondarra. Hala ere, (3.42) higidura-ekuazioak kalkulatzeko $H(\theta, \varphi, p_\theta, p_\varphi)$ behar dugu; aldagai aldaketa bat egin beharra dago. Hori lortzeko, (3.47) eta (3.48) momentu

orokortuetatik $\dot{\theta}$ eta $\dot{\varphi}$ askatuko ditugu,

$$\dot{\theta} = \frac{l_2 p_\theta - l_1 p_\varphi \cos(\theta - \varphi)}{l_1^2 l_2 (m_1 + m_2 \sin^2(\theta - \phi))} \quad (3.51)$$

$$\dot{\varphi} = \frac{l_1 (m_1 + m_2) p_\varphi - l_2 m_2 p_\theta \cos(\theta - \phi)}{l_1 l_2^2 m_2 (m_1 + m_2 \sin^2(\theta - \phi))} \quad (3.52)$$

horiek hamiltondarrean sartu eta sinplifikatu ondoren, hauxe lortzen da:

$$H(\theta, \varphi, p_\theta, p_\varphi) = \frac{l_2^2 m_2 p_\theta^2 + l_1^2 (m_1 + m_2) p_\varphi^2 - 2 m_2 l_1 l_2 p_\theta p_\varphi \cos(\theta - \varphi)}{2 l_1^2 l_2^2 m_2 [m_1 + m_2 \sin^2(\theta - \varphi)]} \quad (3.53)$$

Amaitzeko, lor ditzagun higidura ekuazioak:

$$\dot{\theta} = \frac{\partial H}{\partial p_\theta} = \frac{l_2 p_\theta - l_1 p_\varphi \cos(\theta - \varphi)}{l_1^2 l_2 (m_1 + m_2 \sin^2(\theta - \phi))} \quad (3.54)$$

$$\dot{\varphi} = \frac{\partial H}{\partial p_\varphi} = \frac{l_1 (m_1 + m_2) p_\varphi - l_2 m_2 p_\theta \cos(\theta - \phi)}{l_1 l_2^2 m_2 (m_1 + m_2 \sin^2(\theta - \phi))} \quad (3.55)$$

$$\dot{p}_\theta = -\frac{\partial H}{\partial \theta} = -(m_1 + m_2) g l_1 \sin(\theta) - C_1 + C_2 \quad (3.56)$$

$$\dot{p}_\varphi = -\frac{\partial H}{\partial \varphi} = -m_2 g l_2 \sin(\varphi) + C_1 - C_2 \quad (3.57)$$

$$(3.58)$$

Agertzen diren C_1 eta C_2

$$C_1 \equiv \frac{p_\theta p_\varphi \sin(\theta - \varphi)}{l_1 l_2 (m_1 + m_2 \sin^2(\theta - \varphi))} \quad (3.59)$$

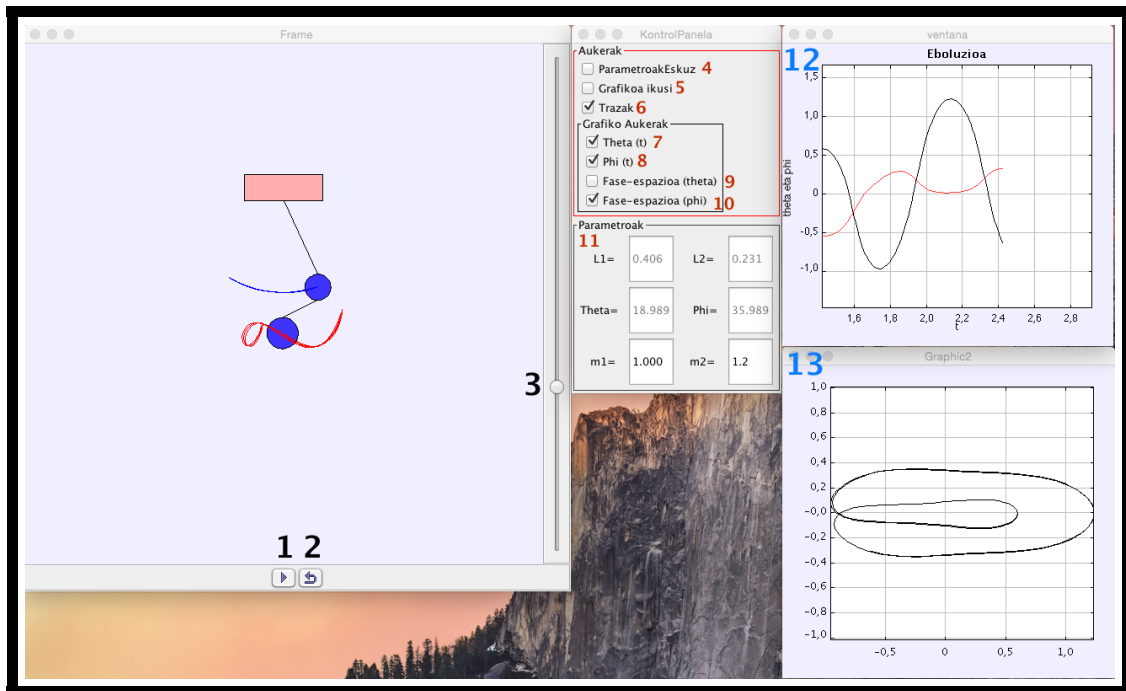
$$C_2 \equiv \frac{l_2^2 m_2 p_\theta^2 + l_1^2 (m_1 + m_2) p_\varphi^2 - l_1 l_2 m_2 p_\theta p_\varphi \cos(\theta - \varphi)}{2 l_1^2 l_2^2 (m_1 + m_2 \sin^2(\theta - \varphi))} \sin(2(\theta - \varphi)) \quad (3.60)$$

izanik.

Modu honetan, jadanik badauzkagu pendulu bikoitzaren eboluzio-ekuazioak, eta prest gaude *EJS* programan simulatzen hasteko.

Simulazioa

Nahiz eta teoria eta konplexutasun aldetik pendulu sinpletik bikoitzerako jauzia itzela izan, *EJS*n simulatzea ez da hain zaila; berau egiteko, jadanik amaitua nuen pendulu sinplearen simulazioan oinarritu naiz. Lehenik eta behin, bigarren pendulua gehitu diot lehenengoari; gero, φ aldagaia eta (p_θ, p_φ) momentuak berdefinitu dira, eta, azkenik, (3.54)-(3.57) eboluzio-ekuazioak gehitu. Behin aldagaien angeluarren denboraren menpeko eboluzioa kalkulatu ondoren oinarri-aldaketa egin da osagai kartesiarretara (3.36) eta (3.38) ekuazioetan oinarrituz. Hurrengo irudian simulazioaren hasierako leihoa ikus daiteke, eta, ondoren, deskribapena.



3.10 Irudia: *SIM_PenduluBikoitza.jar* simulazioaren pantaila

- Kontrol-gunea (**beltzez**)
 1. **Play/pause botoia**: Simulazioa *abiarazi/pausarutzen* du.
 2. **Reset botoia**: Simulazioa hasierako egoerara itzularazten du.
 3. **Zoom scroll-a**: Simulazio-leihoaren eskala aldatzeko balio du.
- Simulazioaren parametroak (**gorriz**)
 4. **Parametroak Eskuz botoia**: Pendulu sinplearen kasuan bezala, aktibatu gabe dagoenean saguarekin mugitu ditzakegu pendulu biak; bestela, hasierako baldintzak eskuz sartu behar dira.
 5. **Grafikoa ikusi: botoia** **Grafikoen Leihoa** ikusten den ala ez kontrolatzen du.
 6. **Trazak botoia**: Penduluek egiten duten ibilbidea margotu edo ez kontrolatzen du.
 - **Grafikoen leihoaren aukerak**
 7. **Theta (t) botoia**: $\theta(t)$ lehen penduluaren denborarekiko eboluzio-angeluarra marrazten du grafikoki.
 8. **Phi (t) botoia**: $\varphi(t)$ bigarren penduluaren denborarekiko eboluzio-angeluarra marrazten du grafikoki.
 9. **Fase-espazioa theta botoia**: Lehen penduluaren fase-espazioa marrazten du grafikoki; hots, $p_\theta(\theta)$

- 10. **Fase-espazioa ϕ botoia:** Bigarren penduluaren fase-espazioa marrazten du grafikoki; hots, $p_\varphi(\varphi)$
- 11. **Datuak eskuz sartzeko gunea:** Hasierako baldintzak kontrola daitezke dagozkien eremuak aldatuz. Gogoratu *Parametroak Eskuz* botoia aktibatua egon behar duela.
- Grafikoen leihoa (**urdinez**)
 - 12. **Denboraren menpeko grafikoen leihoa:** Leiho honetan marrazten $\theta(t)$ eta $\varphi(t)$ funtzioak marrazten dira.
 - 13. **Fase-espazio motako grafikoen leihoa:** Penduluen fase-espazioak ($p_\theta(\theta)$ eta $p_\varphi(\varphi)$) marrazteko leihoa da hau.

Betiko moduan, simulazioa jaisteko, egin klik hurrengo estekan:

[SIM_PenduluBikoitza.jar](#)

Jarduera didaktikoak

1. jarduera Pendulu bikoitza ezagutuz

Lehen jarduera honetan, *SIM_PenduluBikoitza.jar* simulazioari lehen begiratu bat ematea proposatzen da.

- (a) Hasteko, sakatu eta mugitu penduluak nahieran, gehien gustoko duzun hasierako konfigurazioa lortu arte; gero, sakatu kontrol-panelean **trazak** aukera eta abiarazi simulazioa. Bi penduluak dantzan hasiko dira. Ikusten duzun higidura, periodikoa al da? ala kaotikoa?
- (b) Pausatu simulazioa eta aukeratu **grafikoa ikusi**. Gero, aukeratu ikusi nahi dituzun grafiko ezberdinak eta abiarazi berriro simulazioa. Askotan, gase-espazioen grafikoan irudi argiagoak lortzen dira.
- (c) Azkenik, aldatu bigarren penduluaren masa erdira, ($m_2 = 0.5$); ipini hasierako baldintzak nahieran eta abiarazi simulazioa. Egin proba bestelako konfigurazioekin ($m_1 = 0.2$ eta $m_2 = 0.8, \dots$) ^a

^aKonputazio-erroreen ondorioz simulazioa ez dabil ondo masak oso handiak baldin badira; hala ere, masen eragina $\frac{m_1}{m_2}$ zatiduraren menpekoea denez, beti aukera daiteke masetako bat unitatekoa eta bestea ratioaren araberakoa.

2. jarduera Pendulu bikoitza simple bihurtzen

Ariketa honetan pendulu bikoitzetik sinplea lortzea proposatzen da. Limite hori aurkitzeko bigarren penduluaren eragina ezabatu behar da.

- (a) Aukeratu **parametroak** eskuz botoia eta egin $m_2 = 0$ eta $l_2 = 0$. Zer gertatu da? Simulazioak errorea ematen du $l_2 = 0$ egiten bada.
- (b) Orain, egin $m_2 = 0$ eta $l_2 = 0.0001^a$. Eman lehen pendulari hasierako angelu bat (θ) eta abiarazi simulazioa. Ikus daitekeenez, lortzen den soluzioa pendulu sinplearena da.

^aZero-ak arriskutsuak izan daitezke konputazionalki, baina erraz ekidin ditzakegu arazoak.

3. jarduera Kaosaren erdian, ordena

Orain arte, hasierako baldintzak penduluak saguarekin mugituz zehaztu badituzu, ia gehienetan erantzun kaotiko bat ematen duela ikusiko zenuen; jarduera honetan, kaos horren artean aurki daitekeen pendulu bikoitzaren soluzio periodikoetako bat aztertuko dugu.

- (a) Aukeratu **parametroak** eskuz botoia eta ipini hasierako baldintza hauek: $l_1 = 0.25$, $l_2 = 0.75$, $\theta = -30$ eta $\phi = 60$. Aukeratu baita **grafikoa** ikusi eta eman **fase-espazioa** (ϕ) botoiari. Jarraian, abiarazi simulazioa eta itxaron. Une batean, pendulua mutur batean geratu eta lehen egindako bide berbera errepikatzen hasiko da; eta ibilbide hori etengabe errepikatuko du; soluzio periodiko bat da!
- (b) Orain, hasieratu simulazioa eta ipini berriro (a) ataleko hasierako baldintza berdinak. Oraingoan, baina, desaktibatu **parametroak** eskuz aukera eta mugitu pendulu bikoitza apur bat (1mm baino gutxiago) eta abiarazi simulazioa. Ikusiko duzunez, jadanik soluzioa ez da periodikoa! Bigarren penduluaren higidura konfinatua dago fase-espazioaren eremu batean baina galdu egin du periodikotasuna. Honek, aurkitu dugun soluzio periodikoa ezegonkorra dela adierazten digu.

eta uhin-fronte bat (maximo bat) igortzen duela; orduan, t denbora igaro ondoren, partikula A puntura iritsiko da $|\vec{OA}| = v_p t$ izanik; eta uhin-fronte zirkularrak $|\vec{OB}| = v_u t$ erradioa izango du. Irudian ikus daitekeen bezala, uhinak konoa ukitzen duen puntuan lerro perpendikular bat marraztu ezker, iruki angeluzuzen bat lortzen dugu, eta bertatik zuzenean erlazioa dezakegu $\alpha/2$ angelua:

$$\sin(\alpha/2) = \frac{|\vec{OA}|}{|\vec{OB}|} = \frac{v_u t}{v_p t} = \frac{v_u}{v_p} \quad (3.61)$$

Ekuzio simple honetatik hainbat ondorio atera genitzake; alde batetik, partikularen abiadura handitu ahala konoa geroz eta estuagoa egiten da. Bestetik, $v_p = v_u$ egoera kontsideratzen badugu, erraz ikusten da $\alpha/2 = 90^\circ$ dela; hau da, konoa desagertu egingo da, eta uhin-fronte guztiak partikularekin higituko dira. Horixe da talka-uhinaren kasu limitea, hortik aurrera, uhinaren hedapen-abiadura handiagoetarako, talka-uhinik ez da agertuko. Nahiz eta teoria aldetik sinplea izan, orain ikusiko dugunez, simulazioa egitea ez da hain erraza.

Simulazioa

Simulazioa eraikitzeko gakoa, uhin-zirkularren igorle bat egitean datza. Ziur bide errazagorik badela, baina hauxe izan da nire programazio aukera: *EJS* sistemak n elementuko **objektu multzoak** sortzeko aukera ematen du; kasu honetan, n zirkulu sortzeko esango diogu sistemari. Zirkulu horien zentroen posizioak n elementuko $X[i]$ bektore batean gordeko ditugu. Hurrengo lana, uhina bera sortzea da; horretarako, hauxe da erabilitako estrategia:

1. $T[i]$ bektore berri bat sortu dut; bere elementu bakoitzean i .uhina¹³ **noiz** atera behar den adierazi da; horretarako, T_0 konstantea definitu da, hots, ondoz ondoko uhinen arteko denbora tartea (Periodoa).

```

1  for (int i=0; i<n; i++) {
2      T[i] = -i*T0;
3  }
```

Ikus daitekeenez, uhin bakoitzari denbora negatibo bat esleitu zaio; simulazioaren pauso berri bakoitzeko dt denbora gehitzen joango zaio $T[i]$ elementu bakoitzari ($T[i]^{berri} = T[i]^{zahar} + dt$) eta nahikoa izango da $T[i] > 0$ den ala ez ikustea uhin berriak atera behar duen jakiteko.

2. Modu honetan, higitzen dabilen gure *partikulak* bere azpian dagoen ingurunea perturbatuko du T_0 denbora igarotzen den bakoitzean; eta gero, sortzen den uhina bere kabuz hedatuko da ingurunearen bereizgarri den v_u abiadurarekin. Konputazionalki egiteko, T_0 aldiunero uhin-igorle bat finkatu behar da partikula dagoen lekuan; eta noski, programak jakin egin behar

¹³Uhina esaten denean, uhinaren maximoetaz ari naiz.

du $T[i]$ bektorean dauden uhin-igorleak jadanik finkatuak dauden ala ez. Hori lortzeko; boolean moduko bektore bat sortu da; bektore horretan i .elementua jadanik atera den ala ez adieraziko da; hasiera batean

```

1  for (int i=0; i<n; i++) {
2      lehen[i]=false;
3  }
```

egingo dugu; honek, uhin-igorlerik oraindik finkatu ez dela adierazten digu.

3. Simulazioa martxan jartzen denean, hauxe da programak egiten duena:

(a) Elementu bakoitzeko $T[i] < 0$ den begiratzen du:

```

1  for (int i=0; i<n; i++) {
2      if (T[i]<0) {
3          r[i]=0; //negatiboa den heinean ez dago uhinik.
4      }
```

$T[i] < 0$ bada, i zirkuluaren erradioa 0 egiten du (uhin-fronte horrek ez du atera behar oraindik)

(b) $T[i] > 0$ bada, bi kasu daude: lehenengo kasuan, $T[i]$ lehenengoz egin da zero baino handiagoa eta orduan, uhin-igorlearen $X[i]$ posizioa une horretan partikula higikaria dagoen lekuan finkatu behar da. Bigarren kasuan, $T[i]$ uhina lehendik sortua da; hala bada, erradioa handitu besterik ez da egin behar:

```

1      else if (T[i]>0) {
2          if (lehen[i]==false) {
3              x[i]=xp;
4              lehen[i]=true;
5          }
```

Ikus daitekeen moduan, programak `lehen[i]` bektorea begiratzen du uhina aurretik sortu den edo ez jakiteko; Aurretik sortu ez bada (*false* kasua) uhinaren zentroa finkatzen du, eta, nola ez, `lehen[i]=true` bihurtu, uhina jadanik sortua izan dela zehaztuz.

(c) Gero, erradioak eguneratzen dira:

```

1      r[i]=(Math.abs(v0)+Math.abs(vp))*T[i];
```

Ageri den bezala, i uhinaren erradioa $r(i) \propto vT[i]$ da; baina zein da behar dugun v ? Kontuan izan behar dugu bi erreferentzia sistema ditugula; partikularen erreferentzia

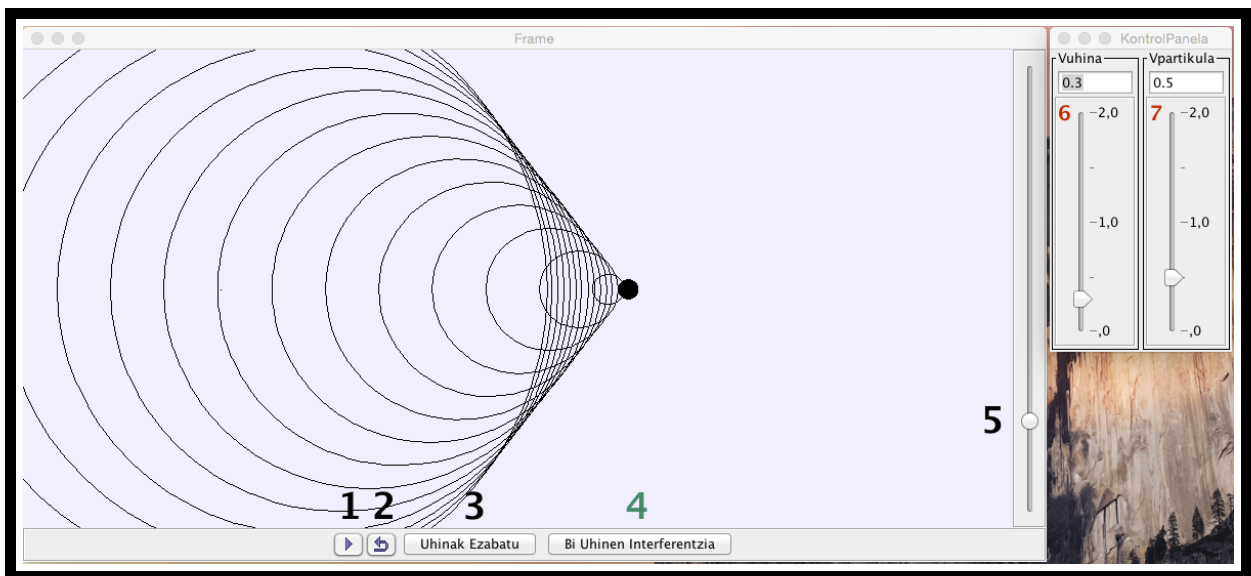
sistematik ateratzen diren uhin guztiak v_u abiaduraz higitzen dira; hala ere, laborategitik neurketak egitean, erabili behar dugun abiadura $v^{lab} = v^* + V$ da; gure kasuan, erreferentzia sistema higikorra partikula denez $v = v_u + v_p$ erabili beharko da (programan ipini den moduan). Kontuan izan $T[i]$ bektoreen elementuak T_0 periodoaz bereizturik daudela; hala, uhin izaera

4. Azkenik, partikularen posizioa eta simulazioko denborak eguneratzen dira:

```

1 // Partikula aurrera mugitu
2 xp=xp0+vp*t;
3 // Uhin faseei denbora gehitzen zaie
4 for (int i=0;i<n;i++){
5     T[i]=T[i]+dt;
6 }
7 //Simulazioaren denbora berritu
8 t+=dt;
```

Horixe da simulazio programatzeko egin dena; orain simulazioaren kontrolen eskema egingo dugu:



3.12 Irudia: *SIM_TalkaUhinak.jar* simulazioaren hasierako pantaila

- Kontrol-gunea (beltzez)

1. **Play/pause botoia:** Simulazioa *abiarazi/pausaritzen* du.
2. **Reset botoia:** Simulazioa hasierako egoerara itzularazten du.

3. **Uhinak Ezabatu botoia:** Marrazturik dauden uhinak ezabatzen ditu partikularen posizioa hasieratu gabe.
 4. **Bi Uhin Interferentzia botoia:** Aukera honekin fisikoki guztiz ezberdina baina simulazio aldetik antzekoa den problema aztertzen da: bi uhin interferentzia. Kuriositate moduan inplementatu da modu hau, *EJS* sistemaren eta simulazioen erabilera aniztasuna agerian uzteko. Informazio gehiagorako, jo B eranskinera.
 5. **Zoom scroll-a:** Simulazio-leihoaren eskala aldatzeko balio du.
- Simulazioaren parametroak (**gorriz**)
 6. **Vuhina kontrol gunea:** v_u uhinaren hedapen-abiadura alda daiteke gune honetan, zuzenean idatziz edota scroll-a erabiliz.
 7. **Vpartikula kontrol gunea:** v_p partikularen abiadura alda daiteke gune honetan, zuzenean idatziz edota scroll-a erabiliz.

Simulazioa jaitsi nahi izan ezkerre, erabili hurrengo esteka:

[SIM_TalkaUhinak.jar](#)

Jarduera didaktikoak

Simulazioa erabiltzerakoan ondokoa

1. jarduera: abiadurekin jolasean

Talka-uhinen fenomenoak ulertzeko asmoz, ireki simulazioa eta egin frogak hasierako baldintzak aldatuz:

- (a) Egin $v_p > v_u$; talka-uhinak agertzeko baldintza. Nola aldatzen da α angelua v_p geroz eta handiagoa egitean?
- (b) Aztertu $v_p = v_u$ kasu limitea. Zer gertatzen da? Zein da kasu honetan α angeluaren balioa? Egiaztatu uhin-fronteen maximo guztiak partikularekin batera doazela.
- (c) Azkenik, egin $v_u > v_p$; ze fenomeno fisikoren antzekoa da gertatzen dena? Laborategiko erreferentzia-sisteman bi behatzaile ipintzen baditugu, bata partikularen aurrean eta bestea atzealdean, zeinek neurtuko luke uhinaren maiztasun altuena?

2. jarduera: Uhin interferentzia

Sakatu *Bi Uhin Interferentzia* botoia; orain, kolore ezberdinetako bi bolatxo agertzen dira pantailan. Egin klik eta mugitu bi bolatxoak nahieran eta ikusi zer gertatzen den `play` sakatzean.

- (a) Gai al zara sortzen den interferentzia patroia ikusteko? Saiatu simulazioa pausatzen eta interferentziaren maximo eta minimoak antzematen
- (b) Zer gertatzen da uhin-igorleen arteko distantzia aldatzeko? Ipin bi bolatxoak gertu; antzematen al dituzu orain maximo eta minimoak? Zer ondorio atera dezakezu? (Simulazioaren zoom-a aldatu hobeto ikusi ahal izateko)

4. KAPITULUA

Ondorioak

Iritsi gara amaierara; bidean, simulazio ugari aztertu eta fisikaren alor ezberdinak ukitu ditugu. Azken kapitulu honetan, proiektua amaitu ondoren atera ditudan ondorioak azalduko dira.

Ondorioetan murgildu aurretik, proiektuaren helburuaren inguruan idatzi nahi dut. Sarreran aipatu bezala, proiektuaren helburu nagusia fisikako problemen simulazio didaktikoak egitea izan da; hau da, ikasleen fisika-ulermena hobetu eta kontzeptuak errotzera bideratutako simulazioak egitea. Horrela, proiektuan zehar fisikako gai ugari landu dira, maila ezberdinetako problemak garatu (tiro parabolikotik uhin grabitatorioetara) eta simulazioak interaktiboak eta didaktikoak egitea lortu da, besteak beste, jarduera ezberdinak proposatuz. Hori guztia dela eta, proiektuak bere helburua bete duela esango nuke. Eta helburua betetzearekin batera, beste helburu ugari sortu zaizkit; mundu zabal bat baitago simulazioak eta didaktika uztartzen.

Jarraian, aztertu diren problemetan ateratako ondorio nagusiak jorratuko ditut:

Pendulu Sinplea izan da landutako lehen simulazioa. Kontzeptu teorikoak oinarritzekoak izan arren, simulazioa interaktibo egitea ez da guztiz trivialea izan. Bi arazo nagusi izan ditut: alde batetik, pendulua saguarekin mugitu eta hasierako baldintzak eguneratzearen kontua, eta, bestetik, angeluak pantailan modu egokian¹ agertzea. Hala ere, simulazio hau lehenengo lantzeak asko lagundu dit *pendulu bikoitzaren* simulazioa egiterako garaian.

Tiro Parabolikoa izan da bigarrena. Izan dudan partikular-irakasle esperientzian buruhauste asko antzeman ditut ikasleengan problema honen inguruan, eta gogoan nuen berau lantzeko. Hala, simulaziorik didaktikoena haxe denaren irudipena dut; adibidez, bolatxoan abiadura-bektore kartesiarrak marrazteak altuera maximoko puntuan $v_y = 0$ nola egiten den *ikustea* ahalbidetzen du; gainera, simulatzeko modu ezberdinak ezarri dira (hasierako

¹Egoki dionadanean, angelua graduetan emateaz, `Math.arctan()` funtzioaren funtzionamendu *karakteristikoa* ulertu eta moldatzeaz ari naiz.

altuera eta *mendian talka* modua), eta, horrela, ariketa mota gehiegi ebatzi eta ikaslego zabalago batera iritsi daiteke. Simulatzerako orduan, kontrol-egitura diseinatzeak eman dit lan gehien; aukera ugari daudenez, zaila izan da elementu guztiak ondo uztartzea. Adibide moduan, mendirik ez dagoenean pilotak ez du ezeren kontra talka egin behar; eta mendia dagoenean, berriz, magalean geratu beharra du eta ez mendian barrena igaro eta zorua jo. Horrelako arazo ugari sortzen dira simulazioak geroz eta konplexuago, eta, orokorrean, errealagoak egiten saiatu ezker.

Uhin grabitatorioena izan da unibertitate maila izan duen lehen problema. Teorikoki eta matematikoki ulertzea gehien kosta zaidan problema izan da hau; izan ere, niretzat nahiko berria den matematika kobariantea erabili behar izan da. Hala ere, teoria zaila egin bazait ere, *EJS*n simulatzea askoz ere sinpleagoa izan da; horiek horrela, behin 2D-tako zeharkako sekzioen eboluzioa egin ondoren, animatuta sentitu eta 3D-tara pasatzea erabaki nuen. Agian, didaktikoki mami gutxien duen simulazioa izan da; baina, fisikoki, ez dugu ahaztu behar Einsteinen Ekuazioen hutsaren soluzio hurbildua ari garela ikusten gure pantailetan.

Pendulu Bikoitza landu da gero. Simulazio hau izan da nire kuttunena; mekanika ikasten nabilela beti sentitzen nuen lortzen genituen Lagrangeren ekuazioak nire kabuz simulatzeko grina; azkenean, karrera amaitzean, lortu dut grin hori asetzea. Teoria pisutsua izan arren (Hamiltonen ekuazioak deribatzea ez da lan erraza izan), simulazioa bera egitea ez da zaila izan, aurretik egin nuen *pendulu simplea* izan baitut eredu. Baina, problemaren benetako mamia simulazioaren emaitzetan dago. Izan ere, eskuartean duguna sistema kaotiko bat da; ez nuen lehenago sistema kaotikorik erabili eta zenbait ondorio interesgarri atera ditut. Hala, pendulu bikoitzaren kasuan, emaitza kaotikoak hasierako konfigurazioaren *energia* batetik aurrera agertzen direnaren sententzia izan dut; horiek horrela, penduluen hasierako angeluak txikiak ipini ezker higidura periodikoagoak lortu ditut, eta, aldiz, angelu handiagoko (eta beraz, energia potentzial gehiago) konfigurazioek erantzun oso kaotikoak eman dituzte. Bestalde, jardueretan agertu den hasierako konfigurazio periodiko gehiago ere aurkitu ditut, baina denak izan dira ezegonkorak².

Talka-uhinak aztertu ditugu amaitzeko. Hauxe izan da JAVA programazio aldetik lan gehien eman didan simulazioa. Ebatzi beharreko ekuazio diferentzialik ez egotean, eskuz idatzi behar izan dut simulazioaren eboluzioa. Hala ere, egindako lana erabilgarria izan da, inoiz uhin zirkularren iturri bat behar izan ezker, simulazio honetan egindako lana erabil baitezaket. Fisikaren ikuspuntutik, gainera, talka-uhinen kontzeptua elektrodinamikara ere hedatu daiteke, *Cherenkov*-en erradiazioa³ azaltzeko, adibidez.

Gainera, \LaTeX ingurunea erabiltzen ikasi dut, eta, proiektuko eskema grafikoak egiteko *Adobe Illustrator* programarekin ere aritu naiz.

²Hasierako baldintzak aldatu, eta, penduluen higidura bornatua, baina, ez-periodikoa, bihurtzen baitzen.

³Kasu honetan, baina, talka-uhinak 3D-tan gertatzen dira, eta, beraz, konikoak izango dira.

Azken finean, proiektu hau amaitu baino, hasi besterik egin ez dudalaren irudipena dut. Beti izan naiz zientzia irakastearen maitalea; pozgarria baita gauzak azaldu eta jendeak ulertzea; eta ildo horretatik, bikaina izan da *EJS* programa ezagutu eta simulazioen mundu honetan lehen pausoak ematen hastea. Amaitzeko, aipatu, idatzizko lanaren luzera dela medio, ezin izan ditudala egindako simulazio guztiak ipini; hala ere, horiek jaitsi nahi izan ezker, hurrengo estekak erabili ditzakezue:

Karga erlatibista eremu elektromagnetiko batean: [SIM_KargaErlatibista.jar](#)
 q karga baten higidura klasikoa eremu elektromagnetiko batean: [SIM_EremuEB.jar](#)

Bibliografía

- [1] FRANCISCO ESQUEMBRE, *Creación de Simulaciones Interactivas en Java*, Pearson Prentice Hall, Madrid (2005). Web orria: <http://fem.um.es/Ejs/>
- [2] JOHN T. TAYLOR, *Classical Mechanics*, University Science Books, Sausalito CA (2005).
- [3] H. J. PAIN, *The Physics of Vibration and Waves*, 4th Ed., Wiley, Chichester (1993).
- [4] M. P. HOBSON, G. EFSTATHIOU and A. N. LASENBY, *General Relativity. An introduction for Physicists*, Cambridge University Press, Cambridge (2006).