

Unibertsitate Masterra

Konputazio Ingeniaritza eta Sistema Adimentsuak



Konputazio Zientziak eta Adimen Artifiziala Saila –
Departamento de Ciencias de la Computación e Inteligencia Artificial



Universidad del País Vasco
Euskal Herriko Unibertsitatea

K
I
S
A

I
C
S
I

Master Tesia

Soluzio periodikoen bilaketa edo jarraipena
kolisiotik gertuko egoeretan.

Mikel Antoñana Otaño.

Tutorea(k)

Ander Murua

Konputazio Zientziak eta Adimen Artifiziala saila
Informatika Fakultatea

Joseba Makazaga

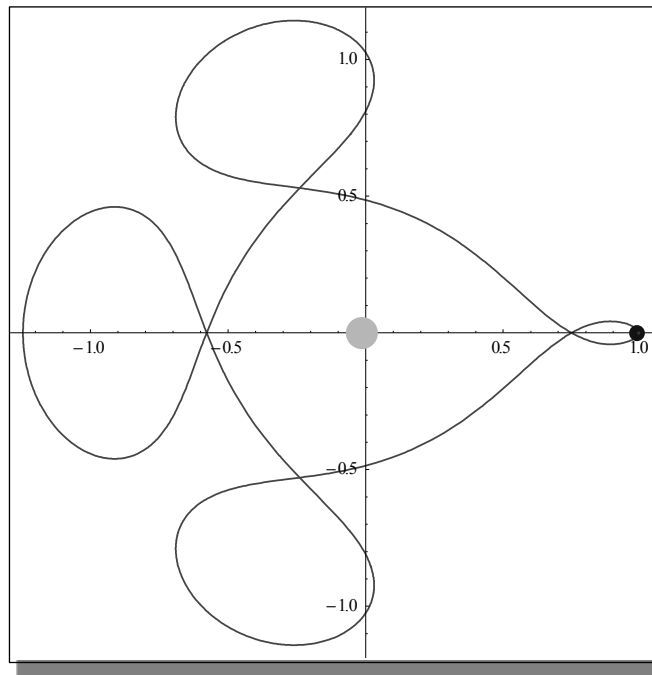
Konputazio Zientziak eta Adimen Artifiziala saila
Informatika Fakultatea



KZAA
/CCIA

2013ko irailan

Soluzio periodikoen bilaketa edo jarraipena kolisiotik gertuko egoeretan.



Soluzio periodikoen bilaketa edo jarraipena kolisiotik gertuko egoeretan.

Laburpena.

Kolisiotik gertuko egoera bat dugunean, N-gorputzen ekuazio diferentzialen singularitateek zenbakizko integrazioaren kalkuluen doitasun galera eta konputazio denbora handitzea eragiten dituzte. Ekuazio transformazioaren bidez singularitateak ezabatzeko teknikari erregularizazioa deritzen zaio. Lehen zatian, erregularizazio teknika deskribatu eta Taylor integrazio metodoan ikertuko dugu. Bigarren zatian, Urabe metodoan oinarritutako soluzio periodikoen jarraipen algoritmo berri bat inplementatuko dugu. Erregularizazioa soluzio periodikoen jarraipenean ikertu eta gure inplementazioa AUTO softwarearekin konparatuko dugu.

Sarrera.

N-gorputzen grabitazio problemetan zenbakizko integrazioa kalkulatzeko dugun zailtasun handiena, kolisiotik gertuko egoerek sortzen dute. Edozein bi gorputzen distantzia txikia denean, ekuazio diferentzialen singularitateek integrazioaren kalkuluen doitasun galera eta konputazio denbora handitzea eragiten dute.

Kolisio gertuko egoerak maiz gertatzen dira, esaterako kometa bat planeta bati gerturatzen zaionean. Une horretan aldaketak oso azkarrak dira. Zenbakizko integrazioan urrats oso txikiak beharrezkoak dira ekuazio diferentzialaren eskuineko aldeko aldaketak ondo jasotzeko. Integrazioan urrats kopurua handia izateak, doitasun galera eta konputazio denbora altua eragiten du. Arazo hauek gainditzeko eta sistema dinamiko hauek aztertu ahal izateko, ekuazioen erregularizazioa ezinbesteko dela ikusiko dugu. Erregularizazioa ekuazio transformazioaren bidez singularitateak ezabatzeko teknikari deitzen zaio.

Lan honetan bi zati bereizi dira: lehen zatian, erregularizazioa deskribatu eta Taylor integrazio metodoan ikertu dugu. Bigarren zatian, soluzio periodikoen jarraipen gaia landu dugu.

Lehen zatiko 1. atalean, sarrera moduan kolisio gertuko problemen zenbakizko integrazioan dauden zailtasunak erakutsi ditugu. Horretarako, bi adibide landu ditugu: hiru gorputzen problema murriztua (R3BP) eta 7 gorputzen Pleidades (PLEI) problema. Jarraian, 2. atalean, erregularizazio teknikaren oinarriak zehaztu ditugu eta aurreko bi problemetan aplikatu dugu. Lehen zati honen azken atalean, Taylor integrazio metodoan erregularizazioaren eraginkortasuna ikertu dugu. Taylor izeneko softwarean [6] esperimentuak egin ditugu.

Bigarren zatian, soluzio periodikoen jarraipena aztertu dugu. Lehenik, 4. atalean soluzio periodikoen jarraipenaren oinarriak azalduko ditugu. Ondoren, 5. eta 6. atalean, Uraberen metodoan oinarritutako, hau da, polinomio trigonometriko trunkatuaren bidezko soluzio periodikoen jarraipenaren algoritmo berri bat garatu dugu. 7. atalean, AUTO [1] erreferentziazko softwarea aztertu da; software hau zatikako polinomioen kolokazio metodoan oinarritzen da. Bi algoritmoetan R3BP problemaren erregularizazioa ikertu dugu eta bi inplementazioen arteko konparaketa egin dugu.

Dokumentuaren 8. atalean, ondorioak eta lan honi jarraipena emateko etorkizuneko ildo nagusiak aipatu dira.

Ikerketa honetan , esperimentuak egiteko eta irudiak sortzeko kodea eskuragarri jarri da. Irudiak sortzeko Mathematica softwarea erabili da eta kodea Git software biltegi batean antolatu da. Software honen kopia lortzeko:

```
$ cd  
$ git clone https://mikel1969@bitbucket.org/mikel1969/mastertesia.git
```

Aurkibide nagusia.

Laburpena	iii.
1-Zatia: Zenbakizko integrazioa.	
1- Sarrera	1.
2- Erregularizazioa	11.
3- Taylor metodoa	19.
2-Zatia: Soluzio periodikoen jarraipena.	
4- Soluzio periodikoen jarraipena	31.
5- Uraberen metodoa	37.
6- Gure Inplementazioa	45.
7- AUTO.	55.
8- Ondorioak.	71.
Bibliografia	73.
A-Eranskina.	75.

Aurkibidea.

Laburpena.	iii.
1-ZATIA: Zenbakizko integrazioa..	
1-Sarrera.	
11-Zenbakizko integrazio metodoak	1.
12- Kolisio gertuko egoerak.	3.
121- Hiru Gorputzen problema murriztua (R3BP).	3.
122- Zazpi Gorputzen problema: Pleidades (PLEI).	8.
2- Erregularizazioa.	
21-Sarrera.	11.
22-Erregularizazioa.	11.
23-Esperimentuak.	14.
3-Taylor metodoa.	
3.1- Sarrera.	19.
3.2-Taylor integrazio metodoa.	19.
3.3-Taylor Softwarearen deskribapena.	20.
3.4-Esperimentuak.	21.
3.4.1- Deskribapena.	22.
3.4.2- Emaitzak.	22.
3.4.3- Emaitzen azterketa.	24.
3.5- Eranskina.	25.
3.5.1- Ekuazioak modu eraginkorrean.	25.
3.5.2- Emaitza guztiak.	26.
2-ZATIA: Soluzio periodikoen jarraipena.	
4-Soluzio periodikoen jarraipena.	
4.1- Sarrera.	31.
4.2- Soluzioen Jarraipena.	31.
4.3- Soluzio periodikoen jarraipena	33.

5- Urabe metodoa.

5.1-Sarrera.	37.
5.2- Urabe metodoa I.	37.
5.3 -Urabe metodoa II .	40.
5.4- Soluzio periodikoen jarraipena.	43.

6-Gure inplementazioa.

6.1-M koefiziente kopurua.	45.
6.2- Errore neurketa.	46.
6.3- Mathematica bertsioa.	48.
6.4- C programa bertsioa.	49
6.4.1- Garapena.	49.
6.4.2- Esperimentuak.	51.

7-Auto.

7.1- Sarrera.	55.
7.2- Zenbakizko jarraipena.	55.
7.3- Kolokazio metodoa.	57.
7.4- Soluzio periodikoen jarraipena.	58.
7.5- AUTO erabilera.	60.
7.6- Esperimentuak.	61.
7.6.1- Errore neurketa.	61.
7.6.2- AUTO n R3BP problema.	62.
7.6.3-Esperimentua I.	64.
7.6.4- Esperimentua II	67.
7.6.5- AUTO eranskina.	70.

8-Ondorioak. 71.**Bibliografia.** 73.**A-Eranskina.** 75.

A.1- Ordenagailuaren ezaugarriak.	75.
A.2- R3BP ekuazioak.	75.
A.3- Fourier.	77.
A.4- Inplementazioaren oharrak.	82.

1-Sarrera.

1.1- Zenbakizko integrazio metodoak.

Lehen ordenako ekuazio diferentzial arrunten (EDA) hasierako baliodun problemak aztertuko ditugu.

$$u'(t) = f(u(t)), \quad u(t_0) = u_0$$

non $t \in [t_0, t_f]$, $u \in \mathbb{R}^n$, $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$.

Ekuazio diferentzial arrunt baten soluzio analitikoa era analitikoan emandako formula bat da. Beraz, soluzioa edozein unetan ezaguna da aldagai askeari balioak emanaz. EDA baten zenbakizko soluzioa $[t_0, t_f]$ tarteko puntu diskretu batzuetarako soluzio hurbildua da. Hau da, finkatutako N puntu diskretuetarako $u(t)$ soluzio analitikoaren hurbilpenak izango dira.

$$t_0 < t_1 < t_2 < \dots < t_N = t_f,$$

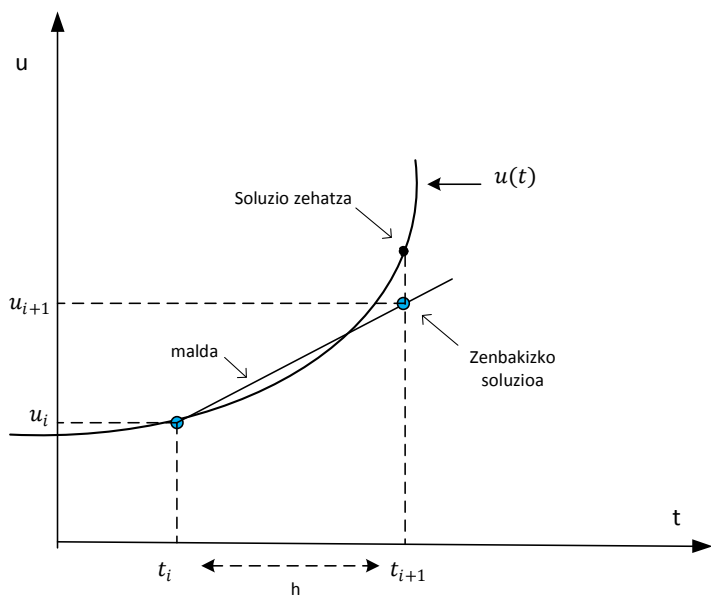
$$u(t_i) \approx u_i \quad i = 1, \dots, N.$$

Zenbakizko integrazio metodoetan soluzioa urratsez urrats kalkulatu dugu. Batetik, urrats bakarreko eta urrats anitzeko metodoak ditugu. Bestetik, metodo implizituak eta esplizituak bereiz ditzakegu. Urrats bakarreko metodo esplizituetan, (t_{i+1}, u_{i+1}) balioa aurreko (t_i, u_i) ezagunetik kalkulatu dugu ondoko moduan,

$$t_{i+1} = t_i + h_i,$$

$$u_{i+1} = u_i + \text{Malda } h_i,$$

non h_i urrats tamaina den eta Malda, du/dt hurbiltzen duen balioa.



Irudia 1.1: Urrats bakarreko metodo esplizitua.

Zenbakizko integrazio metodoen errorea.

Zenbakizko metodotan bi errore mota gertatzen dira: biribiltze eta trunkatze-errorea. Konputagailuan zenbakiak bit kopuru finitu baten bidez adierazten dira; zenbaki erreal baten mantisa adierazteko bit kopuru handiagoa behar denean, mantisa moztu beharra dago. Errore honi biribiltze errorea deitzen diogu. Bestalde, zenbakizko soluzioa erabilitako metodoaren arabera hurbilpen bat da eta errore honi trunkatze-errorea deritzogu.

Ekuazio diferentzialen zenbakizko soluzioa urratsez urrats kalkulatu dugu. Urrats bakoitzean egindako trunkatze erroreari trunkatze errore lokala diogu eta prozesuan zehar metatutako erroreari trunkatze errore metatua edo globala.

Inplementazio motak.

Zenbakizko integrazio implementazio desberdinen artean, sailkapen hau egin dezakegu:

1. Urratsa luzera finkoko implementazioa.

Soluzio diskretuaren urratsak $h_i = h$ balio finkoa duenean.

$$t_{i+1} - t_i = h, \quad i = 1, \dots, N - 1$$

2. Urrats aldakorreko implementazioa.

Soluzio diskretuaren urrats bakoitza h_i , ekuazio diferentzialen aldaketei egokituko zaio.

$$t_{i+1} - t_i = h_i, \quad i = 1, \dots, N - 1$$

3. Denbora erregularizazio implementazioa.

Ekuazioetan singularitateak izaten dira, hau da, balioak non $f(u(t))$ ekuazio diferentzialak infinitu balioa hartzen duen. Ekuazio diferentzialen transformazioen bidez singularitateak ezabatze teknika erregularizazioa deritza. Transformazio hauen ondorioz singularitate gabeko ekuazioak berriak integratu ditugu.

Urrats aldakorreko eta denbora erregularizazioaren bidezko implementazioetan, integrazio urrats luzera ekuazio diferentzialaren aldaketei egokitzen zaio: ekuazio diferentzialen aldaketa txikia denean urratsa luzera handia eta aldaketa handia denean urratsa txikia erabiliko da.

Bi metodoen arteko diferentzi handiena da, denbora erregularizazioaren bidezko implementazioetan ekuazio diferentzialak aldagai aske berri baten arabera berridazten direla eta berridatzitako ekuazioak berri hauek integratu ditugula urrats luzera finkoa erabiliz. Urrats aldakorreko metodoetan, aldiz, ekuazioak ez dira berridatziko eta beraz, ekuazio diferentzialen singularitatea arazoa ez da konponduko; kolisiotik gertuko unea azaltzen denean integrazio urrats kopurua handia beharrezkoa da eta integrazio kalkuluen erroreen (biribiltze errorea eta trunkatze errorea) ondorioz soluzio okerra lortuko da.

1.2- Kolisiotik gertuko egoerak.

Ekuazio diferentzial batzuk singularitateak azaltzen dituzte, hau da, $u(t)$ balioa non $f(u(t))$ ekuazio diferentziala infinitu den. N-gorputzen mugimenduaren ekuazioetan i . eta j . gorputzak talka egiten dutenean, hau da, $r_{ij} = 0$ denean ekuazio diferentziala singularra da.

$$x_i'' = \sum_{\substack{j=1 \\ j \neq i}}^n \frac{m_j (x_j - x_i)}{r_{ij}}, \quad y_i'' = \sum_{\substack{j=1 \\ j \neq i}}^n \frac{m_j (y_j - y_i)}{r_{ij}}$$

$$\text{non } r_{ij} = \left((x_i - x_j)^2 + (y_i - y_j)^2 \right)^{3/2}$$

N-gorputzen probleman bi gorputz elkar gerturatzen direnean, elkarrekiko indarrak asko handitzen dira. Singularitateak gertu dauden une honetan, aldaketak oso azkarrak dira; beraz, zenbakizko integrazioaren urratsak asko txikitu behar dira doitasun maila mantendu nahi bada.

Jarraian, kolisiotik gertuko egoerak azaltzen dituzten bi problema aztertuko ditugu: lehena Hiru Gorputzen Problema Murriztua (R3BP) eta bigarrena, Zazpi Gorputzen problema bat (Pleidades izenekoa) [5,245.orr].

1.2.1- Hiru gorputzen problema murriztua (R3BP).

Hiru gorputzen problema orokorra.

Masa ezberdineko hiru gorputz, elkarrekiko erakarpen eraginpean libreki mugitzen dira espazioan [13].

Gorputzen masak: m_1, m_2, m_3

Gorputzen posizio bektoreak: $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$

$$\mathbf{u}_1 = (x_1, y_1, z_1), \quad \mathbf{u}_2 = (x_2, y_2, z_2), \quad \mathbf{u}_3 = (x_3, y_3, z_3)$$

Gorputzen arteko distantziak:

$$|\mathbf{u}_{12}| = [(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2]^{1/2}$$

$$|\mathbf{u}_{23}| = [(x_2 - x_3)^2 + (y_2 - y_3)^2 + (z_2 - z_3)^2]^{1/2}$$

$$|\mathbf{u}_{13}| = [(x_1 - x_3)^2 + (y_1 - y_3)^2 + (z_1 - z_3)^2]^{1/2}$$

Mugimenduaren ekuazioak, 18 dimentsioko sistema dinamikoa dugu :

$$\ddot{\mathbf{u}}_1 = m_2 \frac{\mathbf{u}_2 - \mathbf{u}_1}{|\mathbf{u}_{12}|^3} + m_3 \frac{\mathbf{u}_3 - \mathbf{u}_1}{|\mathbf{u}_{13}|^3}$$

$$\ddot{\mathbf{u}}_2 = m_1 \frac{\mathbf{u}_1 - \mathbf{u}_2}{|\mathbf{u}_{12}|^3} + m_3 \frac{\mathbf{u}_3 - \mathbf{u}_2}{|\mathbf{u}_{23}|^3}$$

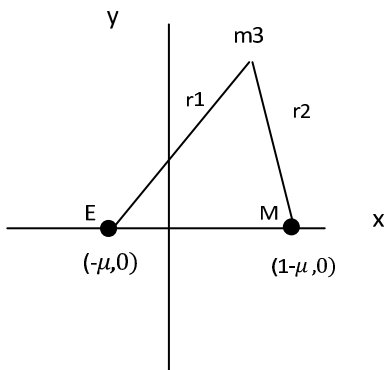
$$\ddot{\mathbf{u}}_3 = m_1 \frac{\mathbf{u}_1 - \mathbf{u}_3}{|\mathbf{u}_{13}|^3} + m_2 \frac{\mathbf{u}_2 - \mathbf{u}_3}{|\mathbf{u}_{23}|^3}$$

Hiru gorputzen problema murriztua.

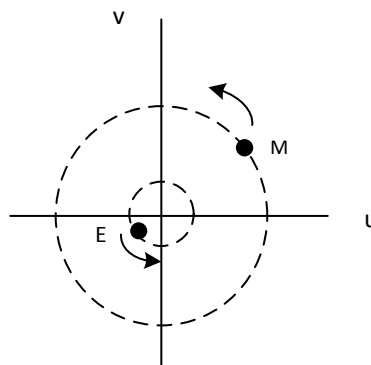
Problema orokorrari honako murriztapenak kontsideratuz definitzen da 3 gorputzen problema murriztua:

- Horietako bi gorputzei nagusiak (m_1 eta m_2) esaten zaie eta hauek, orbita zirkularrean mugitzen dira bere masa zentroaren inguruan. Adibidez lurra eta Ilargia.
- Problema orokorrean, hiru gorputzen masa edozein izan daiteke. Murriztuan aldiz, bi gorputzen masa edozein delarik, hirugarrenarena beste bi gorputzen masa baino askoz ere txikiagoa da. Adibidez asteroide edo satelitea.
- Hirugarren gorputzak (m_3) beste bi gorputz nagusien masarekin alderatuta oso masa txikia izanik, ez du eraginik m_1 eta m_2 gorputzen mugimenduan. Hirugarren gorputza, bi gorputz nagusien eraginpean hauen orbitaren plano berdinean mugitzen dela suposatuko dugu.

Problema ezberdinak deskribatzen dira μ parametroaren arabera. Guk aztertutako problema zehatzean, $\mu = 0.01277471$ balioa Lurra-Ilargia modeloari dagokio. Gorputz nagusi handiena, Lurra, $(-\mu, 0)$ eta gorputz nagusi txikiena, Ilargia, $(1 - \mu, 0)$ posizioan kokatzen dira. Ilargia orbita zirkularra du lurraren inguruan eta ekuazio diferentzialak satelitearen mugimendua deskribatzen dute.



Irudia 1.2. Errotazio koordenatuak



Irudia 1.2. Koordenatu finkoak

(x, y) masa txikiko gorputzaren kokapen koordenatuak izanik, hauek dira errotazio koordenatu sisteman (irudia 1.2) dagokion mugimenduaren ekuazioak (4 dimentsioko sistema dinamikoa),

$$\dot{x} = p_x + y$$

$$\dot{y} = p_y - x$$

$$\dot{p}_x = p_y - umu \frac{x + mu}{r_1^3} - mu \frac{x - (umu)}{r_2^3}$$

$$\dot{p}_y = -p_x - umu \frac{y}{r_1^3} - mu \frac{y}{r_2^3}$$

$$\text{non, } r_1 = ((x + mu)^2 + y^2)^{\frac{1}{2}}$$

$$r_2 = ((x - umu)^2 + y^2)^{\frac{1}{2}}$$

$$mu = 0.01277471, \quad umu = 1 - mu$$

Sistema dinamikoaren energia, soluzioan zehar konstante mantentzen dena, honakoa da,

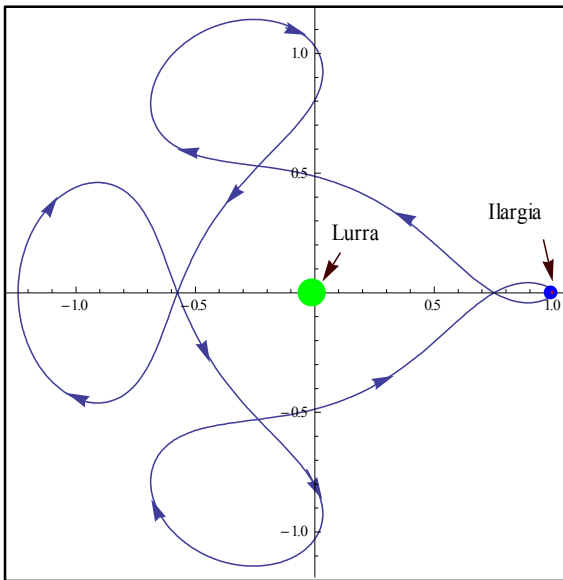
$$E = \frac{1}{2}(p_x^2 + p_y^2) + p_x y - p_y x - \left(\frac{1-\mu}{r_1}\right) - \frac{\mu}{r_2} - \frac{1}{2}\mu(1-\mu)$$

Ezaguna da hasierako balio hauetarako soluzioa periodikoa periodikoa [5, 129.orr]:

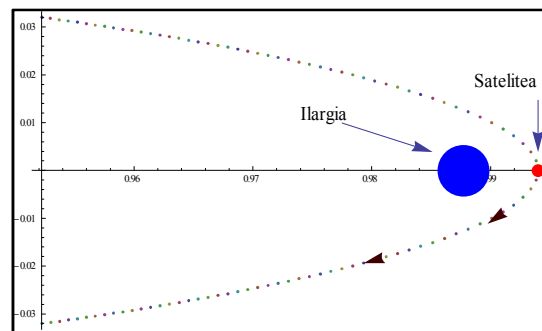
$$\begin{aligned} x(0) &= 0.994, & p_x(0) &= 0 \\ y(0) &= 0, & p_y(0) &= -2.00158510637908252240537862224 + 0.994 \\ t_{end} &= 17.0652165601579625588917206249 \end{aligned}$$

Kolisiotik gertuko unea

Honako bi irudi hauetan, soluzio periodikoa eta kolisiotik gertuko unea irudikatu ditugu.

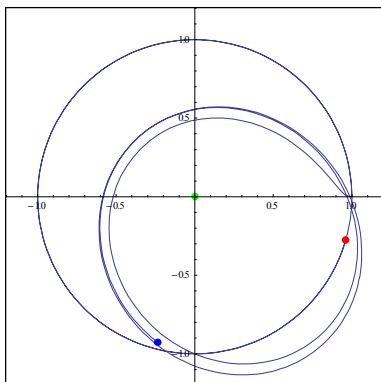


Irudia 1.3. Arenstorf orbita izeneko soluzio periodikoaren orbita.



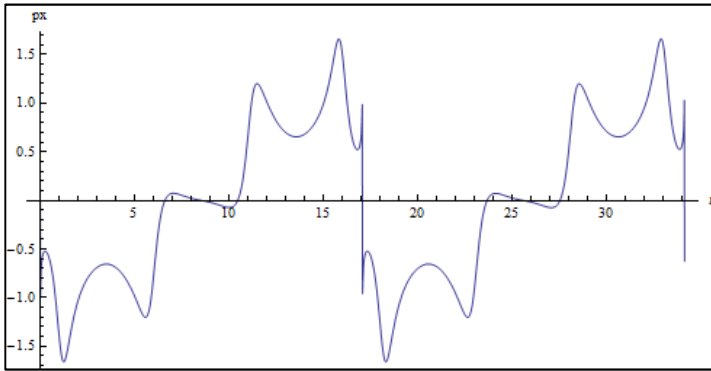
Irudia 1.4. $t=0$ unean kolisiotik gertuko egoera; Ilargia (0.987723, 0) eta satelitea (0.994, 0) posizioa.

Mugimendu erreala hobeto ulertzeko, interesgarria da mugimendua koordenatu finkoetan azaltzea: sateliteak lurraren inguruan 3 bira oso ematen ditu.

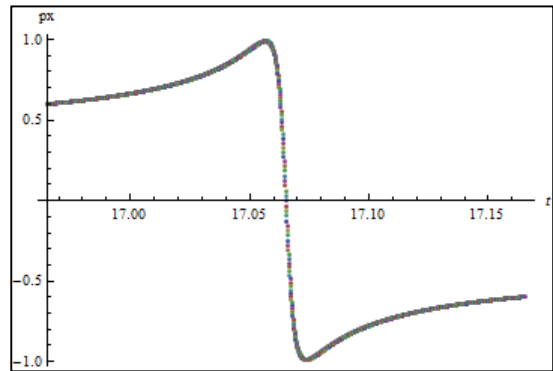


Irudia 1.5. Koordenatu finkoetan mugimenduaren grafikoa.

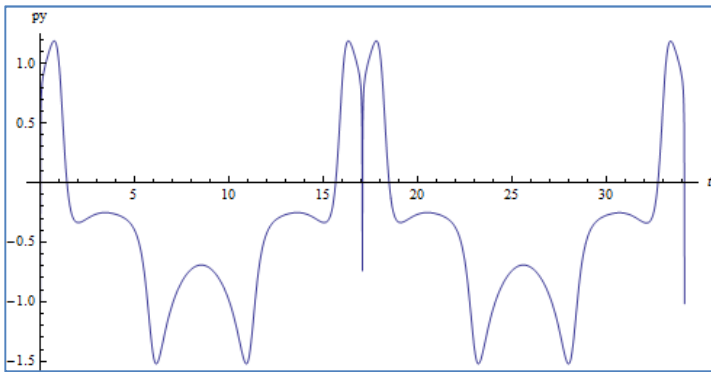
Momentuaren ($p_x(t)$, $p_y(t)$) eboluzioa irudikatu dugu, kolisiotik gertuko unean oso aldaketa azkarra gertatzen dela baieztatzeko; une honen grafikoaren detailea ere azaltzen dugu.



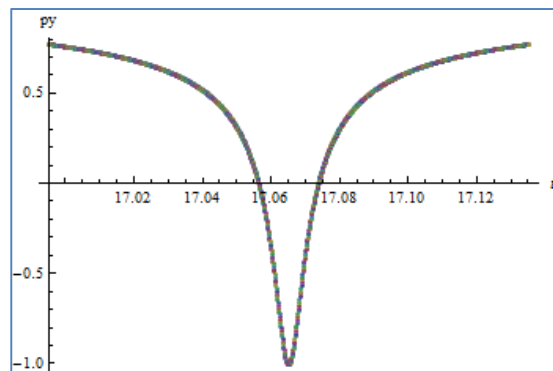
Irudia 1.6 $t=17.06521$ unean p_x momentuaren aldaketa azkarra.



Irudia 1.7 p_x detailea kolisiotik gertuko unean.



Irudia 1.8 $t=17.06521$ unean p_y momentuaren aldaketa azkarra.



Irudia 1.9 p_y detailea kolisiotik gertuko unean .

Kolisiotik gertuko momentuaren aldaketa azkarrak, urrats aldarreko metodoen beharra erakusten dute.

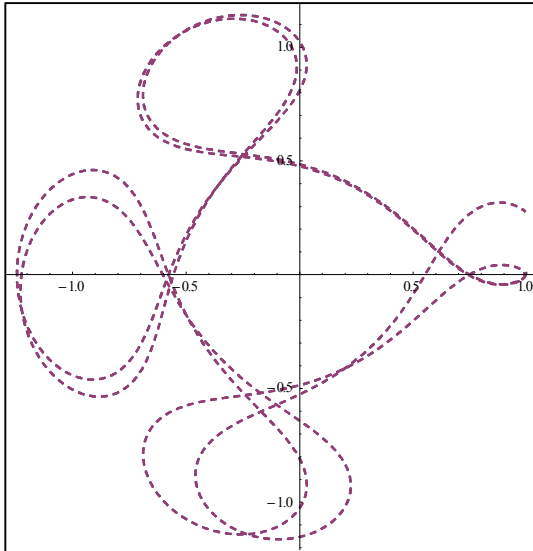
Hasierako balioekiko sentsibilitatea.

Beste ikuspuntu batetik ere uler daiteke kolisiotik gertuko egoerek duten zailtasuna. Hasierako balioen perturbazio txiki batek egoera aldagaien eboluzioan duen eragina aztertuko dugu. Aurreko hasierako balio berdinekin integrazioa egingo dugu baina orain x_0 pixka bat aldatua ($+10^{-7}$). Hau da,

$$\begin{aligned} x(0) &= 0.994 + 10^{-7}, & p_x(0) &= 0 \\ y(0) &= 0, & p_y(0) &= -2.00158510637908252240537862224 + 0.994 \end{aligned}$$

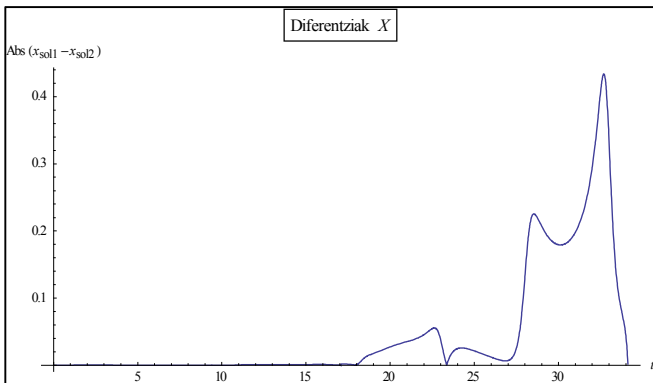
$$t_{end} = 17.0652165601579625588917206249$$

Aldatutako hasierako balio hauentzako zenbakizko integrazio kalkulatu dugu. Lehenik bi orbitak irudi berdinean irudikatu oso ezberdinak direla konprobatu dugu.

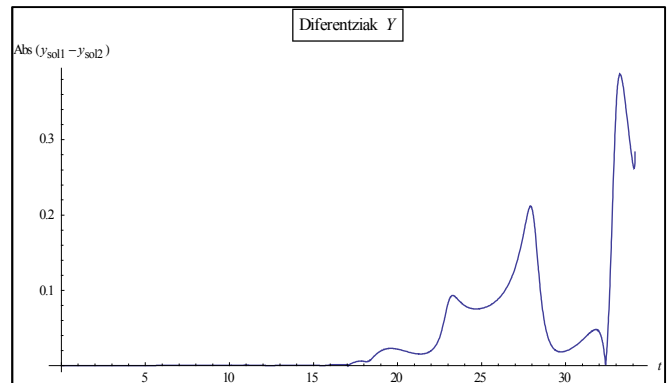


Irudia 1.10 Hasierako balioei sentsibilitatea.

Eta bigarrenik, x eta y koordenatuen balioen diferentzia irudikatuko ditugu.



Irudia 1.11 x koordenatuen diferentzia.



Irudia 1.12 y koordenatuen diferentzia.

Irudien azalpena: integrazioa $2T$ raino egin da. Hasieran bi orbiten eboluzioa oso antzekoa da, baina $t \approx 17$ unetik aurrera orbitak oso ezberdinak bilakatzen dira. Adibide honen bidez ikusi dugu problema honek hasierako balioekiko duen sentsibilitatea. Tarteko errore lokalak ere asko anplifikatzen dira.

Kodea.

- r3bp.nb: 1.2.1 ataleko irudiak eta simulazioa.

1.2.2- Zazpi gorputzen problema: Pleidades.

PLEI (“Pleidades”) izeneko zeruko mekanikako problema dugu [5,245.orr] ; 7 izar planoan mugitzen dira. x_i, y_i kokapen koordinatuak eta $m_i = i$ ($i = 1, \dots, 7$) masa izanik.

$$x'_i = vx_i, \quad y'_i = vy_i$$

$$vx'_i = \sum_{j \neq i}^7 m_j (x_j - x_i)/r_{ij}^3, \quad vy'_i = \sum_{j \neq i}^7 m_j (y_j - y_i)/r_{ij}^3$$

$$\text{non } r_{ij} = \left((x_i - x_j)^2 + (y_i - y_j)^2 \right)^{1/2}, \quad i, j = 1, \dots, 7.$$

Hasierako balioak,

$$x_1(0) = 3, \quad x_2(0) = 3, \quad x_3(0) = -1, \quad x_4(0) = -3,$$

$$x_5(0) = 2, \quad x_6(0) = -2, \quad x_7(0) = 2,$$

$$y_1(0) = 3, \quad y_2(0) = -3, \quad y_3(0) = 2, \quad y_4(0) = 0,$$

$$y_5(0) = 0, \quad y_6(0) = -4, \quad y_7(0) = 4,$$

$$vx_i(0) = vy_i(0) = 0, \quad i \text{ guztientzako honakoak ezik}$$

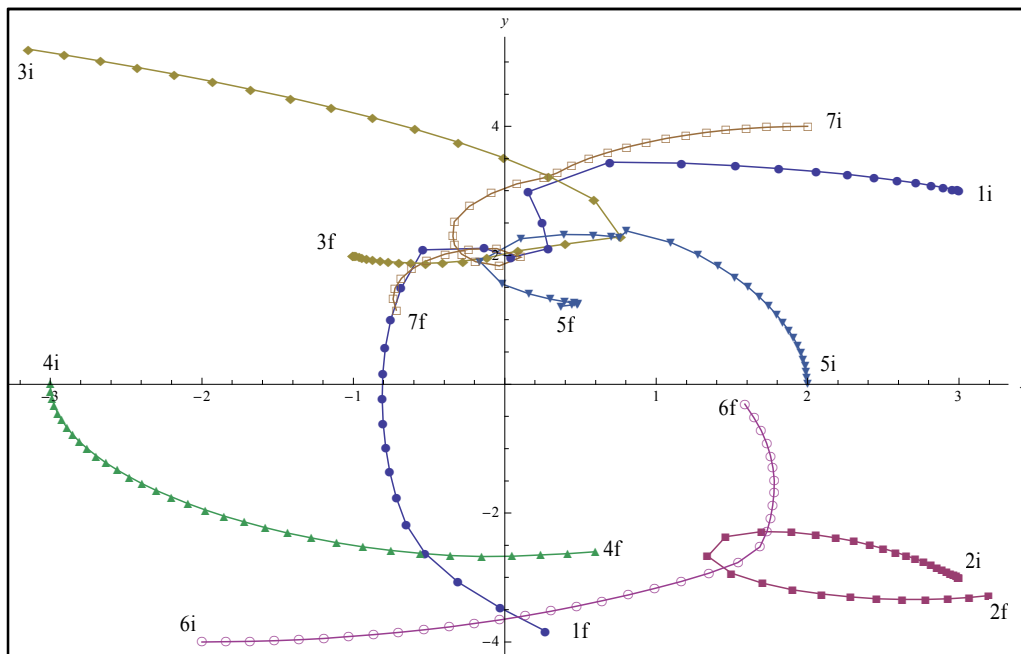
$$vx_6(0) = 1.75, \quad vx_7(0) = -1.5, \quad vy_4(0) = -1.25, \quad vy_5(0) = 1$$

$0 \leq t \leq t_{end} = 3$ tartean integratuko dugu.

Irudia 1.12 7 izarren mugimendua irudikatu dugu. Izar bakoitzaren hasierako posizioa “i” batez adierazten da eta azken posizio ($t_{end} = 3$), “f” batez.

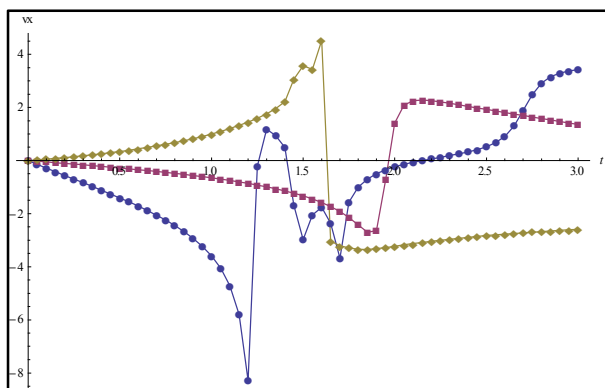
gorputza-1	1	1	3	1	2	5
gorputza-2	7	3	5	7	6	7
r_{ij}^2	0.0129	0.0193	0.0031	0.0011	0.1005	0.07
t	1.23	1.46	1.63	1.68	1.94	2.14

Taula 1.1 Taulan kolisiotik gertuko unek laburtzen dira.

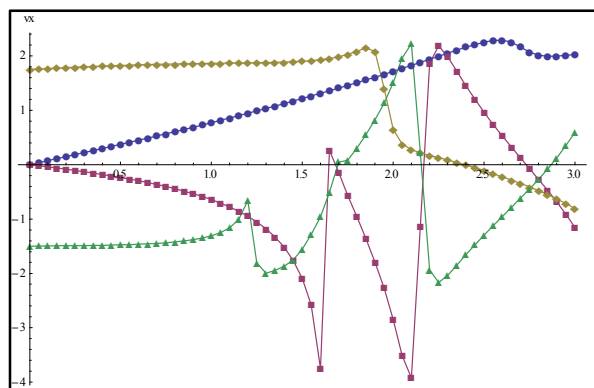


Irudia. 1.12 7 izarren mugimendua $0 \leq t \leq 3$.

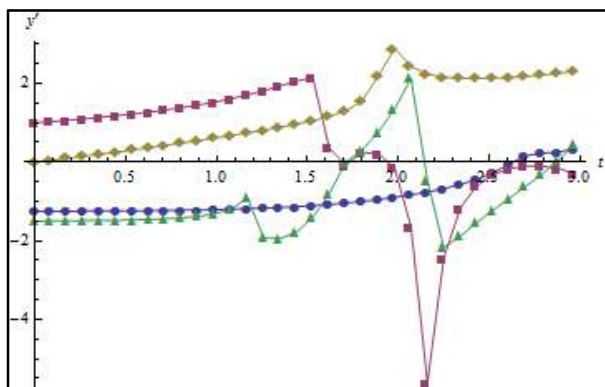
Jarraiko irudietan, $x'_i(t), y'_i(t)$ abiadurak irudikatzen ditugu; kolisiotik gertuko uneetan abiadura aldaketa azkarrak ikusi daitezke.



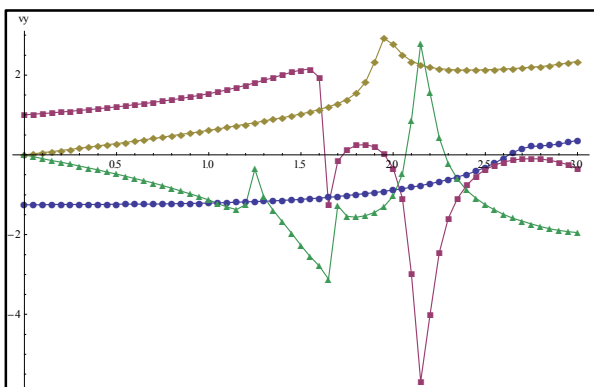
Irudia 1.13 $x'_1(t), x'_2(t), x'_3(t)$



Irudia 1.14 $x'_4(t), x'_5(t), x'_6(t), x'_7(t)$



Irudia 1.15 $y'_1(t), y'_2(t), y'_3(t)$



Irudia 1.16 $y'_4(t), y'_5(t), y'_6(t), y'_7(t)$

Gerhard Wanner matematikariak $t_{end} = 3$ unerako kalkulaturako egoera aldagaien balioak, soluzio zehatzetat kontsideratuko dugu (*).

$$\begin{aligned}x_1 &= 0.370613914397051290093 \\x_2 &= 3.237284092057233092803 \\x_3 &= -3.222559032418323347100 \\x_4 &= 0.6597091455775308359349 \\x_5 &= 0.3425581707156579790377 \\x_6 &= 1.562172101400631016045 \\x_7 &= 0.7003092922212495385147\end{aligned}$$

$$\begin{aligned}y_1 &= -3.943437585517392055277 \\y_2 &= -3.271380973972549928020 \\y_3 &= 5.225081843456544192438 \\y_4 &= -2.590612434977469510811 \\y_5 &= 1.198213693392274637514 \\y_6 &= -0.2429682344935823409161 \\y_7 &= 1.091449240428979747882\end{aligned}$$

$$\begin{aligned}vx_1 &= 3.417003806314314752291 \\vx_2 &= 1.354584501625501221476 \\vx_3 &= -2.590065597810775419618 \\vx_4 &= 2.025053734714241106485 \\vx_5 &= -1.155815100160449092711 \\vx_6 &= -0.8072988170223021725659 \\vx_7 &= 0.5952396354208718766607\end{aligned}$$

$$\begin{aligned}vy_1 &= -3.741244961234008471204 \\vy_2 &= 0.3773459685750629036558 \\vy_3 &= 0.9386858869551078886946 \\vy_4 &= 0.3667922227200569866696 \\vy_5 &= -0.3474046353808494366007 \\vy_6 &= 2.344915448180936923142 \\vy_7 &= -1.947020434263291900674\end{aligned}$$

Kodea

Plei.nb: pleidades problemaren ekuazioak eta 1.2.2 ataleko irudiak.

(*) Komunikazio pertsonala, Gerhard Wanner-ek 2013. Uztailean Donostian egindako egonaldian.

2- Erregularizazioa.

2.1-Sarrera.

Kolisiotik gertuko egoerek N-gorputzen grabitazio problemen zenbakizko soluzioa kalkulatzeko integrazioa burutzeko dugun zailtasun handiena sortzen dute. Ekuazioen transformazioen bidez singularitateak ezabatzeko teknikari erregularizazioa deritzo. Transformazio hauen ondorioz, singularitate gabeko ekuazio berriak integratuko ditugu, doitasuna hobetuz eta konputazio denbora azkartuz.

Hauek dira N-gorputzen mugimenduaren ekuazioak,

$$x_i'' = \sum_{\substack{j=1 \\ j \neq i}}^n \frac{m_j (x_j - x_i)}{r_{ij}}, \quad y_i'' = \sum_{\substack{j=1 \\ j \neq i}}^n \frac{m_j (y_j - y_i)}{r_{ij}}$$

$$\text{non } r_{ij} = \left((x_i - x_j)^2 + (y_i - y_j)^2 \right)^{3/2}$$

i. eta *j.* gorputzak oso hurbil daudenean, r_{ij} -k balio txikia hartzen du eta ekuazio diferentziala singularra da. Ekuazioen formulazio berri bat bilatu nahi dugu, non $r_{ij} \rightarrow 0$ den uneetan ekuazioak erregularrak diren.

2.2-Erregularizazioa.

Ez da posible ekuazio diferentzial guztien erregularizazio egitea; singularitatea ezabatu ezin daitekeenean, portaera singularra leuntzea bilatuko dugu. Ekuazio diferentzialen erregularizazio nahiz leuntzearen helburua, zenbakizko integrazio kalkulua hobetzea da.

Erregularizazioa eta leuntzea, ekuazio diferentzialen aldagaia aldaketaren bidez gauzatzen da. Aldagai aldaketa bi motatakoa izan daiteke:

- Erregularizazio orokorra.
Transformazioan menpeko aldagaiak eta aldagai askea aldatuko ditugu.
- Denboraren erregularizazioa.
Transformazioan soilik aldagai askea aldatuko da.

Jatorrizko ekuazio diferentziala honakoa izanik erregularizazio motak azalduko ditugu,

$$\frac{du}{dt} = f(u(t)), \quad u: \text{menpeko aldagaia}, \quad t: \text{aldagai askea}$$

Erregularizazio orokorra.

Ekuazioetatik singularitateak ezabatzeko, menpeko aldagaiak eta aldagai aske berriak gehituko ditugu,

$$u = g(z)$$

$$\tau = \int_{t_0}^t \frac{dt}{s(z)} \equiv \frac{dt}{d\tau} = s(z)$$

Sistema berrian, mugimendu $z = z(\tau)$ funtzioak deskribatzen du; z aldagai berria, τ aldagai askearen menpekoa da. Erregularizatu gabeko sisteman, mugimendua $u = u(t)$ funtzioak deskribatzen zuen.

Ekuazio diferentzialak modu honetan berridatziko ditugu,

$$u = g(z) \Rightarrow \frac{du}{dt} = \frac{dg}{dz} \frac{dz}{d\tau} \frac{d\tau}{dt} = \dot{u} = \frac{g' z'}{s(z)} \Rightarrow z' = \frac{s(z) \dot{u}}{g'}$$

$$\text{non, } \dot{u} = \frac{du}{dt}$$

Denboraren erregularizazioa.

Aldagai askea soilik transformatzen duen erregularizazioa azalduko dugu,

$$u = z$$

$$\frac{dt}{d\tau} = s(z)$$

Honako garapen hau egingo dugu aldagai berriarekiko ekuazioak lortzeko,

$$u = z \Rightarrow \frac{du}{dt} = \frac{dz}{d\tau} \frac{d\tau}{dt} = \dot{u} = \frac{z'}{s(z)} \Rightarrow z' = s(z) \dot{u}$$

$$\text{non, } \dot{u} = \frac{du}{dt}$$

Adibidea.

Erregularizazio metodoa 2-gorputzen adibide simple baten bidez azalduko dugu [2]; bi gorputz lerro zuzen batean elkar hurbiltzen den problema. Lehenik erregularizazio gabeko bertsioa eta bigarrenik erregularizazio aplikatuz lortutako singularitate gabeko planteamendua azalduko dugu.

Erregularizazio gabeko planteamendua.

x bi gorputzen distantzia izanik, hau da problema dinamikoa deskribatzen duen ekuazioa,

$$(1) \quad \frac{d^2 x}{dt^2} = -\left(\frac{1}{x^2}\right)$$

Eta energia integrala (inbariantea),

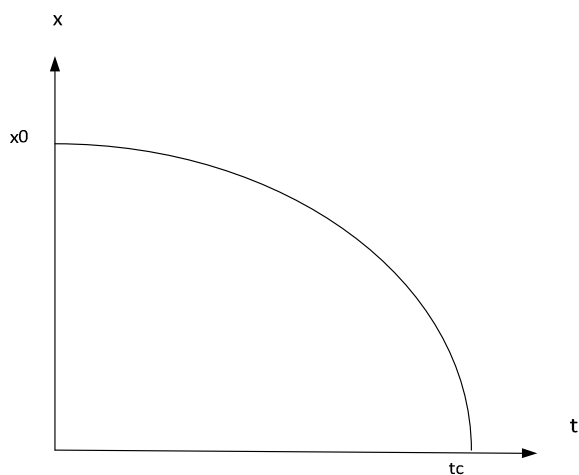
$$(2) \quad \frac{1}{2} \left(\frac{dx}{dt}\right)^2 = \left(\frac{1}{x}\right) + C$$

Hasierako balioak, $x(0) = x_0$, $x'(0) = 0 \Rightarrow C = -(1/x_0)$

Gorputza $t = 0$, $x(0) = x_0$ posizioan dago. Gorputza jatorrira hurbiltzen da abiadura negatiboarekin. t_c kolizio unea izanik, $0 < t < t_c$ eta $0 < x < x_0$.

Goiko bi ekuazioek $x \rightarrow 0$ hurbiltzen denean singularitatea azaltzen dute, ekuazio diferentzialak (abiadura eta azelerazioa) infinitu balioa hartzen du. Zenbakizko integrazio metodo estandarrek singularitate honen gertuko balioetarako okerrak dira, ez baita ekuazioen ebaluazio zehatza egiten.

Sistema honetan lortutako $x(t)$ soluzioaren irudia azaltzen dugu. Kolisio ondorengo, $t > t_c$, egoera ezin daiteke kalkulatu.



Irdia 2.1 $x(t)$.

Erregularizazio planteamendua.

Sundman-ek (1913), honako aldagai askearen erregularizazioa proposatu zuen,

$$(3) \quad d\tau = dt/x$$

eta denbora erregulazioaren azalpenetan esandakoaren arabera

$$(4) \quad \frac{dt}{d\tau} = s(x) \Rightarrow s(x) = x$$

(3) ekuazioaren arabera, τ "denbora" berria, t jatorrizko denbora eta x menpeko aldagaiaren arabera da. Erregularizazioa gehituz, urrats-tartea zenbakizko integrazioan zehar aldatzen da. x txikitzen denean, Δt ere txikitzen da ekuazioen eskuineko espresioen aldaketa handiei egokitzeko. x eta Δt biak batera txikitzen direnez, $\Delta t/x$ koefizientea konstantea mantentzen da. Ondorioz, τ aldagaia t aldagaiaren ordeztaraz, $\Delta\tau$ urratsak leunagoak dira.

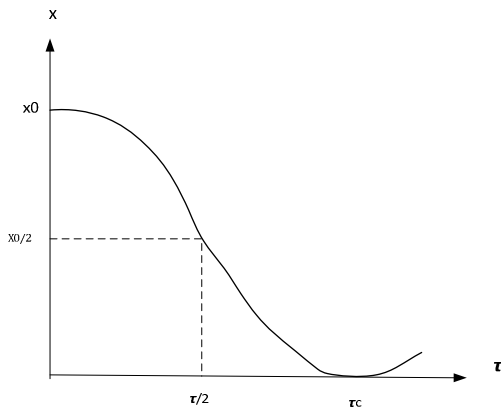
Urrats aldakorreko integratzaileak, (3) ekuazioaren moduko formula baten arabera urratsa egokitzen duten metodoak dira. Urrats aldakorreko metodoaren eta erregularizazioaren arteko diferentzia handiena da, azken honetan ekuazioak aldagai aske berriaren arabera berridatzi eta ekuazio berri hauek integratuko ditugula. Urrats aldakorreko metodoetan, ekuazioak ez dira aldatzen; singularitatea ez da ezabatzen. Bi gorputz elkar hurbiltzen direnean, integrazio urrats kopuru handia behar da. Kontutan hartuta urrats bakoitzeko kalkuluan biribiltze eta trunkatze-errorea dugula, urrats kopuru muga batetik aurrera emaitza okerra izango dugu.

Erregularizazioa (1) eta (2) ekuazioei aplikatuz, ekuazio berriak lortuko ditugu,

$$(5) \quad x'' - 2Cx - 1 = 0$$

$$(6) \quad (x')^2 = 2x + 2Cx^2$$

Lortu ditugun ekuazio berriak, $x \rightarrow 0$ erregularrak dira. Nabarmentzekoa da $1/x^2$ singularitatea duen jatorrizko ekuazio diferentziala, singularitate gabeko ekuazio diferentzial lineal batez ordezkatu dugula.



Hasiera : $t = 0 = \tau$, $x(0) = x_0$, $x'(0) = 0$.
 Kolisio unea: $t = t_c$, $\tau = \tau_c$, $x(0) = 0$, $x'(0) = 0$
 Hasierako posizioa itzuli : $t = 2t_c$, $\tau = 2\tau_c$, $x(2\tau_c) = x_0$

Irudia 2.2 $x(\tau)$ oszilazio-soluzioa dugu.
 x_0 eta 0 puntuen artean.

2.3- Esperimentuak.

Atal honetan, N-gorputzen problemaren denbora erregularizazioa aplikatuko dugu. N-gorputzen problemarentzat $s(z)$ funtzio ezberdinak proposatu dira [2, 394.orr]. Guk "Time-scale preprocessing in numerical integration of ordinary differential equations" [9] artikuluan definitutako $s(z)$ funtzioekin egingo ditugu esperimentuak.

Jarraian, R3BP eta PLEI problemarentzako erregularizazioa funtzioak azaldu eta aplikatuko ditugu.

2.3.1- Erregularizazioa: R3BP.

Proposatutako erregularizazio funtzioa honakoa da,

$$\frac{dt}{d\tau} = s(u) \quad \Rightarrow \quad s(u) = \left(\frac{umu}{r_1^3} + \frac{mu}{r_2^3} \right)^{-1/2}$$

R3BP problemaren ekuazioak gogoratzuz,

$$x' = p_x + y$$

$$y' = p_y - x$$

$$p_x' = p_y - umu \frac{x + mu}{r_1^3} - mu \frac{x - (umu)}{r_2^3}$$

$$p_y' = -p_x - umu \frac{y}{r_1^3} - mu \frac{y}{r_2^3}$$

eta hasierako balioak,

$$x(0) = 0.994, y(0) = 0, \quad p_x(0) = 0, p_y(0) = -2.00158510637908252240537862224 + 0.994$$

$$t_{end} = 17.0652165601579625588917206249$$

Eta erregularizaturiko ekuazio diferentzialak,

$$x' = s(u)(p_x + y)$$

$$y' = s(u)(p_y - x)$$

$$p_x' = s(u) \left(p_y - umu \frac{x + mu}{r_1^3} - mu \frac{x - (umu)}{r_2^3} \right)$$

$$p_y' = s(u) \left(-p_x - umu \frac{y}{r_1^3} - mu \frac{y}{r_2^3} \right)$$

$$t' = s(u)$$

hasierako balioak,

$$x(0) = 0.994, y(0) = 0, \quad p_x(0) = 0, p_y(0) = -2.00158510637908252240537862224 + 0.994$$

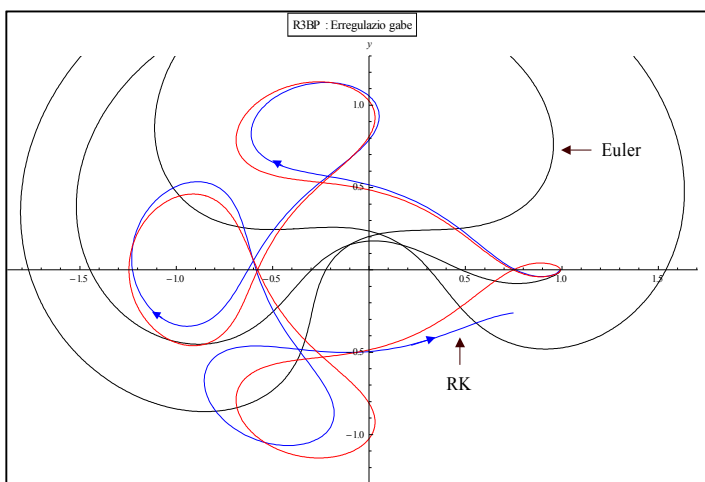
$$\tau_{end} = 29.80472130126875080466217100557$$

Aldagai aske berriarekiko periodoa baliokidea (τ_{end}) kalkulatzeko, jatorrizko ekuazio diferentzialei $d\tau/dt = 1/s(u)$ aldagaia gehituz eta t_{end} denboran integratuz lortu dugu.

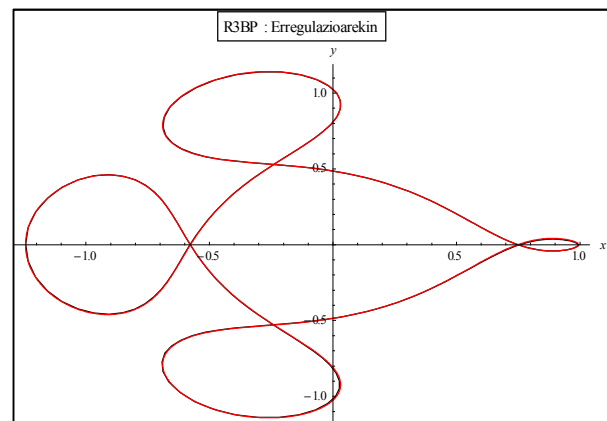
Ekuazioen erregularizatu gabeko eta erregularizatutako bertsioekin zenbakizko integrazio ezberdinak konparatu ditugu. Erregularizazioa erabili dugunean urrats kopuru txikiagoarekin errore txikiagoa lortu dugu.

	Metodoa	Kokapen errorea	Urratsak	Irudia
Soluzio zehatza		10^{-13}		Irudia 2.3 (gorriz)
Erregularizatu gabe	Euler	1.9×10^1	24.000	Irudia 2.3 (beltzez)
Erregularizatu gabe	RK4	3×10^{-1}	6.000	Irudia 2.3 (urdinez)
Erregularizatua	RK4	2×10^{-3}	200	Irudia 2.4 (urdinez)

Taula 2.1 R3BP erregularizazioa. Kokapen errorea: $\text{Norma}[u(T) - u(0)]$.



Irudia 2.3 Erregularizazio gabe.



Irudia 2.4 Erregularizatua.

Kodea.

- TauR3BP.nb: τ_{end} balioaren kalkulua.
- R3BP.nb: irudia 2.3 eta irudia 2.4 kalkulua.

2.3.2- Erregularizazioa-PLEI

Proposatutako erregularizazio funtzioa honakoa da,

$$\frac{dt}{d\tau} = s(u) \Rightarrow s(u) = \left(\sum_{i=1}^7 \sum_{j=1}^{i-1} 1/r_{ij}^2 \right)^{-3/4}$$

PLEI problemaren ekuazioak gogoratu,

$$x'_i = vx_i, \quad y'_i = vy_i$$

$$vx'_i = \sum_{j \neq i}^7 m_j (x_j - x_i)/r_{ij}^3, \quad vy'_i = \sum_{j \neq i}^7 m_j (y_j - y_i)/r_{ij}^3$$

$$\text{non } r_{ij} = \left((x_i - x_j)^2 + (y_i - y_j)^2 \right)^{1/2}, \quad i, j = 1, \dots, 7.$$

eta $0 \leq t \leq t_{end} = 3$ tartean integratuko dugu.

Eta erregularizaturiko ekuazio diferentzialak,

$$x'_i = s(u) vx_i, \quad y'_i = s(u) vy_i$$

$$vx'_i = s(u) \left(\sum_{j \neq i} m_j (x_j - x_i)/r_{ij}^3 \right), \quad vy'_i = s(u) \left(\sum_{j \neq i} m_j (y_j - y_i)/r_{ij}^3 \right)$$

$$t' = s(u)$$

Eta τ aldagai aske berriari dagokion integrazio tartea, $0 \leq \tau \leq \tau_{end} = 14.5719950134136$.

Aldagai aske berriarekiko integrazio denbora baliokidea (τ_{end}), kalkulatzeko jatorrizko ekuazio diferentzialei $d\tau/dt = 1/s(u)$ aldagaia gehituz eta t_{end} denboran integratuz lortu dugu.

Ekuazioen erregularizatu gabeko eta erregularizatutako bertsioekin zenbakizko integrazio ezberdinak konparatu ditugu. Emaitzetan argi geratzen zaigu erregularizatutako bertsioan, urrats kopuru txikiagoarekin emaitza hobeak lortu ditugula. Errorea kalkulatzeko 1.2.2. atalean azaldutako soluzioa ezaguna erabili dugu (u_{sol}).

	Metodoa	Urratsak	Errorea
Erreg. Gabe	RK4	10.000	3.8×10^{-4}
Erregularizazioa	RK4	1.000	8.7×10^{-6}

Taula 2.2 PLEI erregularizazioa.

$$Errorea = \text{Norma}[\{u_{sol} - u(\tau_{end})\}].$$

Kodea.

TauPLEI.nb: Tau-ren balioaren kalkulua

PLEI.nb: kalkuluen zehaztapenekin.

3-Taylor metodoa.

3.1-Sarrera.

Atal honetan Taylor zenbakizko integrazio metodoa aztertuko dugu. Gure helburua, erregularizazioaren eraginkortasuna Taylor integrazio metodoan ikertzea da. Lehenik, Taylor zenbakizko integrazio metodoaren azalpenak emango ditugu. Taylor integrazio metodoa probatzeko urrats aldakorrek egokitzen dituen softwarea [6] erabiliko dugu. Lehen atalean azaldutako R3BP eta PLEI problemenezako egingo ditugu esperimentuak.

3.2- Taylor integrazio metodoa.

Eman dezagun, hasierako baliodun lehen ordenako ekuazio diferentzial arruntak (EDA),

$$u'(t) = f(u(t)), \quad u(t_0) = u_0$$

$$\text{non } t \in [t_0, t_f], \quad u \in \mathbb{R}^n, \quad f: \mathbb{R}^n \rightarrow \mathbb{R}^n$$

Zenbakizko soluzioa $[t_0, t_f]$ tarteko puntu diskretu batzuetarako soluzio hurbildua da. Finkatutako M puntu diskretu batzuetarako, $t_0 < t_1 < t_2 < \dots < t_M = t_f$ kalkulatu da soluzioa. Urrats bakarrek metodo esplizituetan, (t_{i+1}, u_{i+1}) balioa aurreko soluzio ezagunetik (t_i, u_i) kalkulatu dugu. Taylor metodoan $u_{k+1} \approx u(t_{k+1})$, $u(t)$ -ren Taylor Seriea $u(t_k)$ balioan hurbilduz lortuko dugu.

$$u(t_{k+1}) = u(t_k) + u'(t_k)h + \frac{u''(t_k)}{2!}h^2 + \dots + \frac{u^{(N)}(t_k)}{N!}h^N + O(h^{N+1}), \quad k = 0, \dots, M-1$$

$u^{(j)}(t) = f^{(j-1)}(u(t))$, f funtzioaren t -rekiko $(j-1)$. deribatu totala izanik. Deribatuen formulak errekursiboki kalkulatu ditugu:

$$u'(t) = f$$

$$u''(t) = f_t + f_u u' = f_t + f_u f$$

$$u^{(3)}(t) = f_{tt} + 2f_{tu} u' + f_u u'' + f_{uu} (u')^2 = f_{tt} + 2f_{tu} f + f_{uu} f^2 + f_u (f_t + f_u f)$$

$$u^{(4)}(t) = \dots$$

Eta modu orokorrean, $u^{(j)}(t) = P^{(j-1)}f(u(t))$, non P deribatu eragiketa den, $P = \left(\frac{\partial}{\partial t} + f \frac{\partial}{\partial u}\right)$

Beraz metodoan lehen urratsa, $u^{(j)}(t)$ deribatuak kalkulatzeko izango da. Gero integrazioaren $(k+1)$. urrats bakoitzean $u^{(j)}(t_k)$ deribatuak ebaluatu behar dira, $u(t_{k+1})$ seriearen koefizienteak kalkulatzeko.

Laburtuz $[t_k, t_{k+1}]$ urrats bakoitzean. N ordeneko Taylor metodoaren k . urratsa honela definituko dugu,

$$u_{k+1} = u_k + d_1 h + \frac{d_2 h^2}{2!} + \frac{d_3 h^3}{3!} + \dots + \frac{d_N h^N}{N!}$$

$$d_j = u^{(j)}(t_k), \quad j = 1, 2, \dots, N, \quad k = 0, 1, 2, \dots, M-1$$

N ordeneko Taylorren metodoaren errore globala $O(h^{N+1})$ ordenekoa da. Ondorioz, N nahi bezain handia aukeratu daiteke lortu nahi dugun errore txikia lortzeko. N ordena finkoa bada, teorikoki lortu nahi den errore globalerako beharrezkoa dugun h urratsa zein den kalkulatu daiteke.

3.3-Softwarearen deskribapena.

Hauek dira erabiliko dugun softwarearen izena eta bere esteka nagusienak,

A software package for numerical integration of ODEs by means of high-order Taylor methods.
Angel Zorba eta Maorong Zou.

<http://www.math.utexas.edu/users/mzou/taylor/>
<http://www.maia.ub.es/~angel/taylor/>

Taylor EDA (ekuazio diferentzial arrunta) integratzeko kode sortzailea da. Software honen sarrera EDA sistema bat da eta irteera, Taylor metodoaren bidez EDAREN zenbakizko integrazioa kalkulatzeko duen C programa. Sortutako kodeak, iterazio bakoitzean errore lokala muga batean mantentzeko, bai ordena bai urratsa egokitzen ditu.

Jarraian software erabiltzeko laburpen bat azalduko dugu.

Ekuazioen definizioa.

eq.in: fitxategi batean gure ekuazio diferentzialen ekuazioak, hasierako balioak, errore tolerantziak, ... adieraziko ditugu. Garrantzitsua da ekuazio hauek modu eraginkorrean zehaztea

Adibidez R3BP problemari dagokion ekuazio diferentzialak.

```
/* ODE specification: rtbp.in */

mu=0.012277471;umu=1-mu;
a=x1-umu;b=x1+mu;
y2=x2*x2;
r1=y2+b*b; D1=r1^(-3./2);
r2=a*a+y2; D2=r2^(-3./2);
lag=umu*D1+mu*D2;

diff(x1,t)= x4+x2;
diff(x2,t)= x5-x1;
diff(x4,t)= x5-(x1+mu)*(umu*D1)-(x1-umu)*(mu*D2);
diff(x5,t)=-x4-x2*lag;

initial_values= 0.994,0.00,0.00,-1.00758510637908252240537862224;
start_time= 0.0;stop_time = 17.0652165601579625588917206249;
minimal_step = 0.0001; maximal_step = 0.1;
absolute_error_tolerance=0.1e-10;relative_error_tolerance=0.1e-10;
```

Kodea.

Softwareak gure ekuazio diferentzialak integratzen dituen programa sortzen du. Kodea sortzeko azpian idatzitako aginduak exekutatu behar ditugu (adibide gisa R3BP problemarentzat egindakoa esplikatuko dugu). Lehen bi aginduek honako fitxategiak sortzen dituzte: `taylor_rtbp.c` eta `taylor.h` fitxategiak, urrats kontrolerako eta deribatu multzoaren kodearekin; bestetik, `taylor_step_rtbp` fitxategia integrazio urrats bat egiten duen funtzioaren kodearekin. Hirugarren aginduak, `main_rtbp.c` programa nagusia sortzen du; azken programa nagusi hau adibidea bat da eta gure beharretara egokituko dugu.

Doitasun bikoitza.

```
taylor -name rtbp -o taylor_rtbp.c -step -jet -sqrt rtbp.in
taylor -name rtbp -o taylor.h -header
taylor -name rtbp -o main_rtbp.c -main_only rtbp.in
```

Doitasun laukoitza

```
taylor -name rtbp -o taylor_rtbp.c -step -jet -sqrt rtbp.in
taylor -name rtbp -o taylor.h -dd_real -header
taylor -name rtbp -o main_rtbp.c -main_only -dd_real rtbp.in
```

Exekutagarria.

Doitasun bikoitza

```
gcc -O3 main_rtbp.c taylor_rtbp.c -lm -s
```

Doitasun laukoitza

```
g++ -O3 main_rtbp_f.c taylor_rtbp.c -lqd -lm -s
```

Exekuzioa:

```
./a.out
```

Urrats kontrola.

Integrazioaren step_ctrl_method aldagaian, urratsa aukeratzeko metodoa zehaztuko dugu:

- 0: urrats finkoa.
- 1: urrats aldakorra (kontrol estandarra).
- 2: aurreko metodoaren antzekoa baina erroreak sortu ditzakeen urrats luzera handiak ekiditen dituen.

Bertsioa: Taylor 1.4.0 (2003. Maiatza).

3.4-Esperimentuak.**3.4.1-Deskribapena.**

R3BP eta PLEI problemak Taylor Softwarearekin aztertu ditugu. Bi problema hauek, 1.2. atalean deskribatu genituen eta laburbilduz esan, biak kolisio gertuko egoerak dituztenez erregularizazioa aplikatzeko egokiak ditugula.

R3BP: - 4 dimentsioko sistema dinamikoa (1 gorputz).
- Kolisio gertuko egoera bat, satellite eta ilargiaren artean.

PLEI: - 28 dimentsioko sistema dinamikoa (7 gorputz).
- 6 kolisio gertuko unek.

R3BP probleman, probak doitasun bikoitza (double) eta doitasun laukoitzeko (quadruple) datu motekin egin dira.

Doitasun bikoitza: 15 dezimal digitu.

Doitasun laukoitza (extended float type): 30 dezimal digitu (qd_real C++ liburutegia erabili dugu, non dd_real datu motan definitzen den).

Argitu beharra dago, PLEI probleman ezaguna dugun soluzioa 17 dezimal dituela eta horregatik ezin izan ditugula doitasun laukoitzari dagozkion esperimentuak egin.

Problema bakoitza hiru metodo erabiliz exekutatu dugu:

1. Metodoa. Ekuazio diferentzialak erregularizaziorik gabe eta urrats aldakorra.
2. Metodoa. Ekuazio diferentzialak erregularizazioarekin eta urrats aldakorra.
3. Metodoa. Ekuazio diferentzialak erregularizazioarekin eta urrats finkoa.
(1) metodoaren urrats kopuru eta orden berdinak errepikatuz.

Tolerantziak.

Modalitate bakoitza tolerantzi ezberdinetarako exekutatu dugu.

- Doitasun bikoitza: $Tol = 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}, 10^{-10}, 10^{-11}, 10^{-12}, 10^{-13}, 10^{-14}, 10^{-15}, 10^{-16}$.
- Doitasun laukoitza: $Tol = 10^{-21}, 10^{-22}, 10^{-23}, 10^{-24}, 10^{-25}, 10^{-26}, 10^{-27}, 10^{-28}, 10^{-29}, 10^{-30}$.

Denboren neurketa

Esperimentuetarako erabili dugun ordenagailuaren ezaugarriak (A-1)eranskinean deskribatuta daude. Gure problema ebazteko oso denbora gutxi behar denez, denborak neurtzeko prozesu luzeagoak simulatu ditugu. Horretarako 1000 aldiz prozesu berdina ebazteko behar duen denbora neurtu dugu.

time ./a.out.

Zutabeen azalpenak.

- Metodoa: goian izendatutakoaren arabera 1,2,3.
- Tolerantzia: Taylor exekutatzeke erabilitako errore absolutu eta erlatiboak.
- Errorea: kokapen errorea. Soluzioa periodikoa denez, hasierako eta bukaerako balioen arteko diferentzia.
R3BP : $err = Norm((x_{end}, y_{end}) - (x_0, y_0))$
PLEI : $err = Max(Norm((x_{i_{end}}, y_{i_{end}}) - (x_{i_{exac}}, y_{i_{exac}}))), i = 1, 2, \dots, 7$
- Urratsak: Taylor metodoaren urrats kopurua.
- Ordena: Taylor metodoak erabilitako deribatu ordena.
- Denbora: time aginduak itzulitako segundoak.

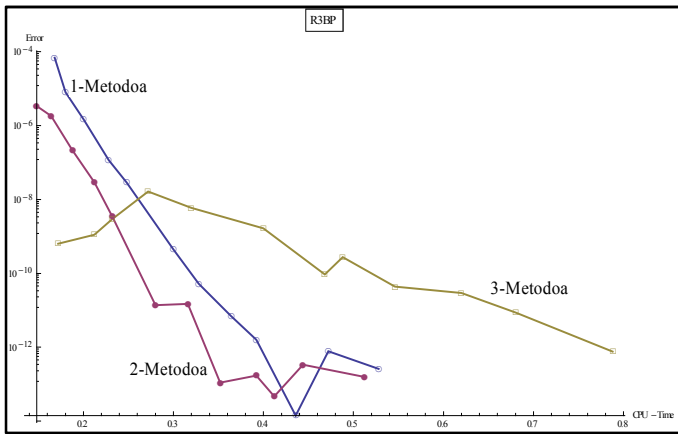
3.4.2-Emaitzak.

Atal honetan, esperimentuen azterketa egin dugu eta azalpenak emaitzen laburpen batean oinarrituz egin dira. Esperimentu guztien emaitzak 3.5.2. atalean kontsultatu daitezke.

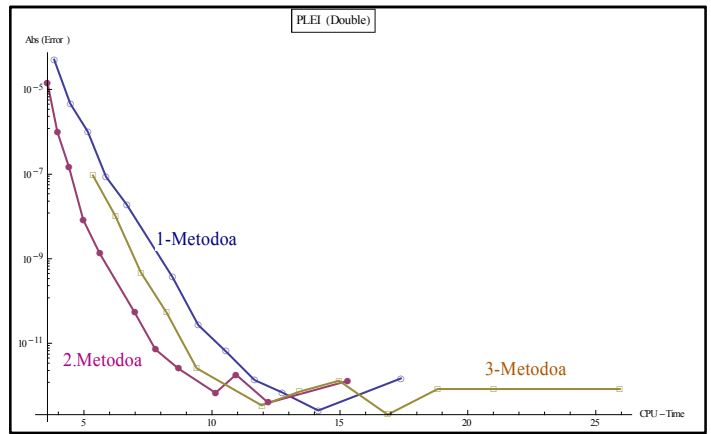
3.4.2.1- Doitasun bikoitza.

Problema	Metodoa	Tolerantzia	Errorea	Urratsak	Taylor-Ordena	Denbora
R3BP	1	10^{-10}	4×10^{-10}	189	13	0.3s
R3BP	2	10^{-10}	1×10^{-11}	119	13	0.28s
R3BP	3		3×10^{-10}	189	15	0.48s
R3BP	3		4×10^{-11}	190	16	0.54s
PLEI	1	10^{-11}	3×10^{-11}	271	14	9.46s
PLEI	2	10^{-11}	7×10^{-12}	155	14	7.78s
PLEI	3		3×10^{-12}	273	11	9.39s

Taula 3.1 Doitasun bikoitza, emaitzen laburpena.



Irudi 3.1 R3BP doitasun bikoitza.

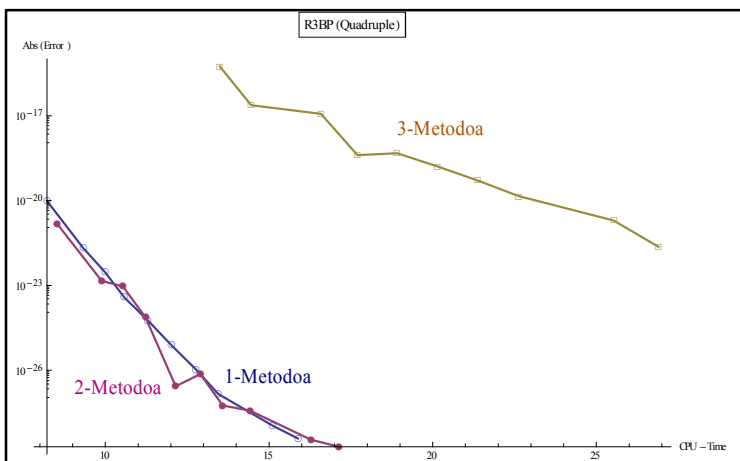


Irudi 3.2 PLEI doitasun bikoitza.

3.4.2.2- Doitasun laukoitza.

Problema	Metodoa	Tolerantzia	Errorea	Urratsak	Taylor-Ordena	Segundoak
R3BP	1	10^{-25}	6×10^{-25}	194	30	11.3
R3BP	2	10^{-24}	7×10^{-25}	122	29	11.2
R3BP	3		2×10^{-22}	195	36	26.9

Taula 3.2 Doitasun laukoitza, emaitzen laburpena.



Irudia 3.3 R3BP doitasun laukoitza.

3.4.3-Eraitzen azterketa.

Doitasun bikoitza.

- Bi problemetan, 2-metodoa erabiliz denborak onenak lortu dira; taulan azaltzen denez, denbora gutxiagoan errore txikiagoa lortu da.
- R3BP problemetan, 3-metodoarekin 1-metodoarekin baino denbora gehiago behar izan da errore berdina lortzeko. PLEI problemetan, aldiz, 3-metodoarekin 1-metodoarekin baino errore txikiagoa lortu da denbora berdinean.
- 3-metodoari dagokionez, urrats finkoa behar izanez gero ondo ordezkatzeko du urrats aldakorreko metodoa. Problema kontserbatutako integrazio luzeetan beharrezkoa izan ohi da kalkuluak urrats finkoarekin egitea.

Doitasun laukoitza.

- R3BP problemari dagokionez, 1-2 metodoen arteko denborak oso antzekoak dira. Kasu honetan ere, 3-metodoa 1-metodoa baino motelagoa da. Badirudi erregularizazioak ekuazioei gehitu dion konplexutasunak, doitasun laukoitzeko aritmetikan areagotu dela. Logikoa dirudi, konplexutasuna gehitzeak doitasun laukoitzeko aritmetikan kostu gehiago izatea.

Konplexutasun azterketa.

Erregularizazioa erabiltzen dugunean, nabarmenki urrats kopuru gutxiago behar da baina kalkulu denbora ez da proportzio berdinean txikitu.

	R3BP (bikoitza)	R3BP (laukoitza)	PLEI (bikoitza)
Urratsak%	63 %	63%	58%
Denbora%	94 %	106%	85%
(Urratsak/denbora)%	148 %	168%	146%

Taula 3.3 Erregularizazio bertsioaren (2.metodoa) laburpena.

Erregularizazio bertsioan ekuazioen konplexutasuna handitu dugu. Taylor metodoan biderkadura kostu handien duen eragiketa izanik, honakoa izan da gehitutako konplexutasuna:

- R3BP jatorrizko EDAk 12 biderkadura ditu eta erregularizazio bertsioan 22 (+%83).
- PLEI jatorrizko EDAk 126 biderkadura ditu eta erregularizazio bertsioan 148 (+%17).

Beraz, proportzionalki R3BP problemetan erregularizazioa gehitzean, PLEI problemetan baino konplexutasun handiagoa gehitu diogu. Honek esplikatuko luke, PLEI problemetan erregularizatutako bertsioan denborak gehiago hobetu izana. Dena den, Taula 3.3-ren datuak ez datoz bat arrazoiak honekin; erregularizazio bertsioan, R3BPn urratsa 148% eta PLEIn urratsa 146% garestitu baitira.

3.5- Eranskina.

3.5.1-Ekuazioak modu eraginkorrean.

Problemaren ekuazioak modu eraginkorrean zehaztu behar dira. R3BP probleman ekuazioak adierazteko modu bakarra kontsideratu da. PLEI aldiz ekuazioak adierazteko modu ezberdinak aztertu dira eta kalkulu eraginkorrenak lortu dituen aukera kontsideratu da.

-Erregularizazio gabe.

0A-eq7bp.

$$r_{12} = 1/(q_{a12} * q_{a12} + q_{b12} * q_{b12});$$

$$aux_{12} = r_{12}^{(3./2)};$$

$$diff(pa1,t)=(-m2*aux12*q_{a12}-m3*aux13*q_{a13}-m4*aux14*q_{a14}-m5*aux15*q_{a15}-m6*aux16*q_{a16}-m7*aux17*q_{a17});$$

0B-eq7bp

$$r_{12} = (q_{a12} * q_{a12} + q_{b12} * q_{b12});$$

$$aux_{12} = r_{12}^{(-3./2)};$$

$$diff(pa1,t)=(-m2*aux12*q_{a12}-m3*aux13*q_{a13}-m4*aux14*q_{a14}-m5*aux15*q_{a15}-m6*aux16*q_{a16}-m7*aux17*q_{a17});$$

Modalitate honetan "0B-eq7bp" izan da eraginkorragoa.

-Erregularizazioarekin.

0A-req7bp

$$r_{12} = 1/(q_{a12} * q_{a12} + q_{b12} * q_{b12});$$

$$aux_{12} = r_{12}^{(3./2)};$$

$$s_2 = r_{12};$$

$$s_3 = r_{13} + r_{23};$$

$$diff(pa1,t)=(-m2*aux12*q_{a12}-m3*aux13*q_{a13}-m4*aux14*q_{a14}-m5*aux15*q_{a15}-m6*aux16*q_{a16}-m7*aux17*q_{a17});$$

0B-req7bp

$$r_{12} = (q_{a12} * q_{a12} + q_{b12} * q_{b12});$$

$$aux_{12} = r_{12}^{(-3./2)};$$

$$s_2 = (1/r_{12});$$

$$s_3 = (1/r_{13}) + (1/r_{23});$$

$$diff(pa1,t)=(-m2*aux12*q_{a12}-m3*aux13*q_{a13}-m4*aux14*q_{a14}-m5*aux15*q_{a15}-m6*aux16*q_{a16}-m7*aux17*q_{a17});$$

Modalitate honetan bi aukerak antzekoak dira; 0A-req7bp denbora gutxiago behar duenez hau aukeratu da integratzeko orduan. Bigarren aukera hartu da kalkuluetarako.

3.5.2- Emaitza guztiak.

Kodea.

3-Taylor azpidirektorioan aurkituko dituzue egindako esperimentuen fitxategiak. Esperimentu bakoitzari dagokion laburpena azterketa.nb fitxategian jaso dira.

3.5.2.1- Doitasun bikoitza.

3.5.2.1.1-R3BP.

1-Metodoa.

Tolerantzia	Urratsak	Taylor-Ordena	Denbora	Kokapen-Err	Inbariante-Err
1×10^{-5}	187	7	0.168s	7×10^{-5}	8×10^{-7}
1×10^{-6}	187	8	0.18s	8×10^{-6}	4×10^{-8}
1×10^{-7}	187	9	0.2s	1×10^{-6}	9×10^{-9}
1×10^{-8}	187	10	0.228s	1×10^{-7}	9×10^{-10}
1×10^{-9}	188	11	0.248s	3×10^{-8}	1×10^{-10}
1×10^{-10}	189	13	0.3s	4×10^{-10}	2×10^{-12}
1×10^{-11}	189	14	0.328s	5×10^{-11}	3×10^{-13}
1×10^{-12}	189	15	0.364s	7×10^{-12}	6×10^{-14}
1×10^{-13}	190	16	0.392s	2×10^{-12}	5×10^{-14}
1×10^{-14}	190	17	0.436s	1×10^{-14}	2×10^{-14}
1×10^{-15}	191	18	0.472s	8×10^{-13}	3×10^{-14}
1×10^{-16}	191	20	0.528s	3×10^{-13}	3×10^{-14}

Taula 3.4 R3BP doitasun bikoitza (1-metodoa).

2-Metodoa.

Tolerantzia	Urratsak	Taylor-Ordena	Denbora	Kokapen-Err	Inbariante-Err
1×10^{-5}	122	7	0.148	3×10^{-6}	6×10^{-6}
1×10^{-6}	121	8	0.164	2×10^{-6}	3×10^{-7}
1×10^{-7}	121	9	0.188	2×10^{-7}	6×10^{-8}
1×10^{-8}	120	10	0.212	3×10^{-8}	3×10^{-9}
1×10^{-9}	120	11	0.232	3×10^{-9}	5×10^{-10}
1×10^{-10}	119	13	0.28	1×10^{-11}	2×10^{-11}
1×10^{-11}	120	14	0.316	1×10^{-11}	6×10^{-12}
1×10^{-12}	120	15	0.352	1×10^{-13}	9×10^{-13}
1×10^{-13}	120	16	0.392	2×10^{-13}	1×10^{-13}
1×10^{-14}	120	17	0.412	5×10^{-14}	2×10^{-14}
1×10^{-15}	120	18	0.444	3×10^{-13}	2×10^{-14}
1×10^{-16}	120	20	0.512	2×10^{-13}	1×10^{-14}

Taula 3.5 R3BP doitasun bikoitza (2-metodoa).

3-Metodoa.

Urratsak	Taylor-Ordena	Denbora	Kokapen-Err	Inbariante-Err
187	7	0.172	6×10^{-10}	8×10^{-8}
187	8	0.212	1×10^{-9}	3×10^{-8}
187	9	0.232	3×10^{-9}	4×10^{-9}
187	10	0.272	2×10^{-8}	1×10^{-9}
188	11	0.32	6×10^{-9}	3×10^{-10}
189	13	0.4	2×10^{-9}	3×10^{-11}
189	14	0.468	9×10^{-11}	9×10^{-12}
189	15	0.488	3×10^{-10}	4×10^{-12}
190	16	0.546	4×10^{-11}	9×10^{-13}
190	17	0.62	3×10^{-11}	4×10^{-13}
191	18	0.68	9×10^{-12}	1×10^{-13}
191	20	0.788	8×10^{-13}	2×10^{-14}

Taula 3.6 R3BP doitasun bikoitza (3-metodoa).

3.5.2.1.2-PLEI.

1-Metodoa.

Tolerantzia	Urratsak	Taylor-Ordena	Denbora	Kokapen-Err
1×10^{-5}	284	7	3.832	5×10^{-5}
1×10^{-6}	280	8	4.464	5×10^{-6}
1×10^{-7}	277	9	5.152	1×10^{-6}
1×10^{-8}	275	10	5.864	9×10^{-8}
1×10^{-9}	273	11	6.66	2×10^{-8}
1×10^{-10}	271	13	8.465	4×10^{-10}
1×10^{-11}	271	14	9.469	3×10^{-11}
1×10^{-12}	270	15	10.553	6×10^{-12}
1×10^{-13}	270	16	11.665	1×10^{-12}
1×10^{-14}	269	17	12.749	7×10^{-13}
1×10^{-15}	269	18	14.157	3×10^{-13}
1×10^{-16}	269	20	17.385	1×10^{-12}

Taula 3.7 PLEI doitasun bikoitza (1-metodoa).

2-Metodoa.

Tolerantzia	Urratsak	Taylor-Ordena	Denbora	Kokapen-Err
1×10^{-5}	181	7	3.576	1×10^{-5}
1×10^{-6}	172	8	3.964	1×10^{-6}
1×10^{-7}	166	9	4.416	1×10^{-7}
1×10^{-8}	163	10	4.972	8×10^{-9}
1×10^{-9}	160	11	5.612	1×10^{-9}
1×10^{-10}	156	13	6.98	5×10^{-11}
1×10^{-11}	155	14	7.784	7×10^{-12}
1×10^{-12}	154	15	8.693	3×10^{-12}
1×10^{-13}	154	16	10.145	7×10^{-13}
1×10^{-14}	155	17	10.941	2×10^{-12}
1×10^{-15}	155	18	12.197	4×10^{-13}
1×10^{-16}	155	20	15.297	1×10^{-12}

Taula 3.8 PLEI doitasun bikoitza (2-metodoa).

3-Metodoa.

Urratsak	Taylor-Ordena	Denbora	Kokapen-Err
284	7	5.344	9×10^{-8}
280	8	6.228	1×10^{-8}
277	9	7.224	5×10^{-10}
275	10	8.221	5×10^{-11}
273	11	9.397	3×10^{-12}
271	13	11.945	3×10^{-13}
271	14	13.429	7×10^{-13}
270	15	14.969	1×10^{-12}
270	16	16.865	2×10^{-13}
269	17	18.821	8×10^{-13}
269	18	21.013	8×10^{-13}
269	20	25.938	8×10^{-13}

Taula 3.9 PLEI doitasun bikoitza (3-metodoa).

3.5.2.2-Doitasun laukoitza.

3.5.2.2.1-R3BP.

1-Metodoa.

Tolerantzia	Urratsak	Taylor-Ordena	Denbora	Kokapen-Err
1×10^{-21}	193	25	8.213	1×10^{-20}
1×10^{-22}	193	26	9.325	2×10^{-22}
1×10^{-23}	193	28	9.985	3×10^{-23}
1×10^{-24}	193	29	10.585	4×10^{-24}
1×10^{-25}	194	30	11.301	6×10^{-25}
1×10^{-26}	194	31	12.013	8×10^{-26}
1×10^{-27}	194	32	12.761	1×10^{-26}
1×10^{-28}	194	33	13.445	1×10^{-27}
1×10^{-29}	195	35	15.113	1×10^{-28}
1×10^{-30}	155	36	15.881	4×10^{-29}

Taula 3.10 R3BP doitasun laukoitza (1-metodoa).

2-Metodoa.

Tolerantzia	Urratsak	Taylor-Ordena	Denbora	Kokapen-Err
1×10^{-21}	121	25	8.533	"1." $\times 10^{-21}$
1×10^{-22}	122	26	9.889	1." $\times 10^{-23}$
1×10^{-23}	122	28	10.529	"9." $\times 10^{-24}$
1×10^{-24}	122	29	11.229	7." $\times 10^{-25}$
1×10^{-25}	123	30	12.149	3." $\times 10^{-27}$
1×10^{-26}	123	31	12.901	7." $\times 10^{-27}$
1×10^{-27}	123	32	13.585	6." $\times 10^{-28}$
1×10^{-28}	123	33	14.421	4." $\times 10^{-28}$
1×10^{-29}	124	35	16.281	4." $\times 10^{-29}$
1×10^{-30}	124	36	17.133	2." $\times 10^{-29}$

Taula 3.11 R3BP doitasun laukoitza (2-metodoa).

3-Metodoa.

Urratsak	Taylor-Ordena	Denbora	Kokapen-Err
193	25	13.493	6×10^{-16}
193	26	14.453	2×10^{-17}
193	28	16.589	1×10^{-17}
193	29	17.697	4×10^{-19}
194	30	18.913	5×10^{-19}
194	31	20.137	2×10^{-19}
194	32	21.365	5×10^{-20}
194	33	22.633	1×10^{-20}
195	35	25.526	2×10^{-21}
155	36	26.89	2×10^{-22}

Taula 3.12 R3BP doitasun laukoitza (3-metodoa).

4- Soluzio periodikoen jarraipena

4.1-Sarrera.

Zenbakizko integrazio metodoak sistema dinamikoaren portaera aztertzeko tresnak dira, hasierako balioei dagokien sistemaren eboluzioa kalkulatzeko. Soluzioen jarraipenean, sistemaren portaera parametro baten arabera nola aldatzen den aztertuko dugu; soluzio ezagun batetik abiatuta parametro baten aldaketa eginez soluzio adarraren jarraipena egingo dugu.

Lehenengo soluzioen jarraipen teknika orokorra azalduko dugu, gero soluzio periodikoen jarraipenean sakontzeko.

4.2-Soluzioen jarraipena.

Suposatu, honako ekuazio sistema dugula, non λ jarraipen parametroa dugun,

$$F(\mathbf{u}, \lambda) = 0, \quad \mathbf{u} \in \mathbb{R}^n, F(\cdot, \cdot) \in \mathbb{R}^n, \quad \lambda \in \mathbb{R}.$$

$\mathbf{x} \equiv (\mathbf{u}, \lambda)$, deitzen badugu honakoa da sistema

$$F(\mathbf{x}) = 0, \quad F: \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$$

Algoritmo orokorra.

Suposatu $\mathbf{x}_0 = (\mathbf{u}_0, \lambda_0)$ soluzioa ezaguna dugula

$$F(\mathbf{x}_0) = 0, \quad \mathbf{x}_0 = (\mathbf{u}_0, \lambda_0), \quad \mathbf{u}_0 \in \mathbb{R}^n, \quad \lambda_0 \in \mathbb{R}.$$

eta $\mathbf{x}_1 = (\mathbf{u}_1, \lambda_1)$ soluzio berria kalkulatu nahi dugula.

Hasierako soluzio honetatik abiatuta $\mathbf{x}_1^{(0)} = (\mathbf{u}_0, \lambda_0 + \Delta\lambda)$, hurrengo ekuazio sistema ebatzi nahi dugu

$$F(\mathbf{x}) = 0$$

$$\lambda_1 - \lambda_0 + \Delta\lambda = 0.$$

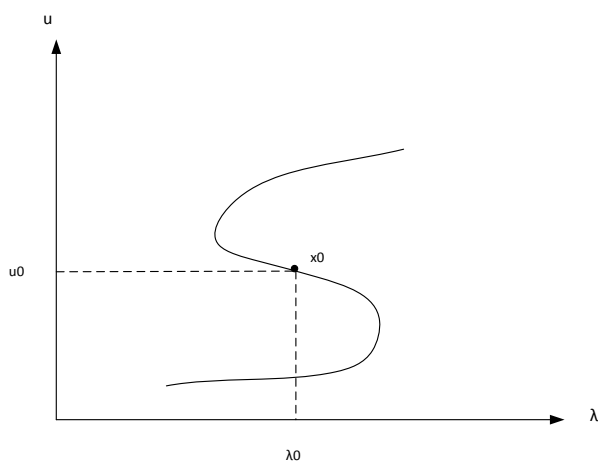
Newton-en metodoa aplikatuz kalkulatu dugu \mathbf{u}_1 soluzioa.

$$\begin{cases} \mathbf{F}_u(\mathbf{u}_1^{(k)}, \lambda_1) \Delta \mathbf{u}_1^{(k)} = -F(\mathbf{u}_1^{(k)}, \lambda_1) \\ \mathbf{u}_1^{(k+1)} = \mathbf{u}_1^{(k)} + \Delta \mathbf{u}_1^{(k)}, \quad k = 0, 1, 2, \dots \end{cases}$$

$$\mathbf{u}_1^{(0)} = \mathbf{u}_0 \text{ hasierako hurbilpena}$$

Aurreko metodoa iteratiboki aplikatuz, $\mathbf{x}(s)$ soluzio familia kalkulatu dugu.

Urratsa	λ_k	Hasierako hurbilpena	Soluzio berria
1	$\lambda_0 + \Delta\lambda$	\mathbf{u}_0	\mathbf{u}_1
2	$\lambda_0 + 2\Delta\lambda$	\mathbf{u}_1	\mathbf{u}_2
...	...		
k	$\lambda_0 + (k + 1)\Delta\lambda$	\mathbf{u}_k	\mathbf{u}_{k+1}



Irudia 4.1 $F(\mathbf{u}, \lambda) = 0$, soluzioa adarra.

Oinarri teorikoa.

Funtzio Implizituaren Teorema (FIT), soluzioaren jarraipena egiteko baldintzak ezartzen ditu. Ondoren azaltzen dugu, Funtzio Implizitu Teorema [3. 18.orr],

Teorema.

$\mathbf{x}_0 \equiv (\mathbf{u}_0, \lambda_0)$, $F(\mathbf{x}) = 0$ soluzio erregularra dugu. \mathbf{x}_0 soluziotik gertu, $\mathbf{x}(s)$ 1-dimentsioko soluzioen jarraitasun bakarra existitzen da, soluzio familia edo soluzio adarra deiturikoa, non $\mathbf{x}(\mathbf{0}) = \mathbf{x}_0$ den.

Soluzio erregularra.

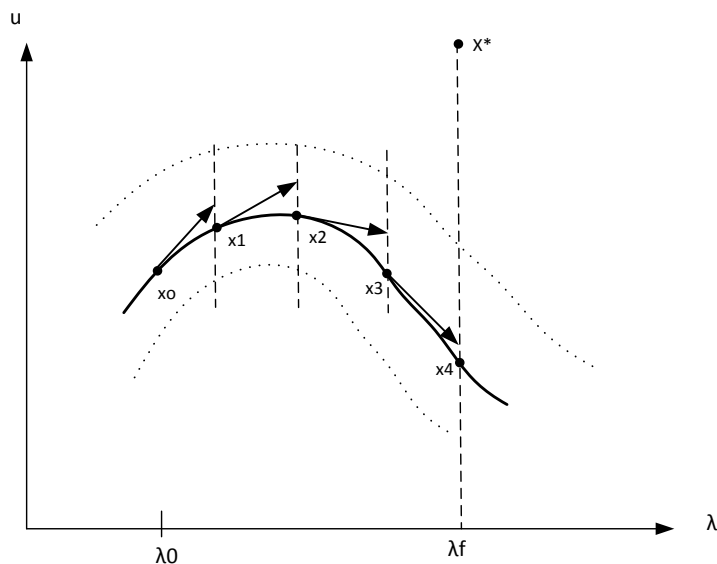
$\mathbf{x}_0 \equiv (\mathbf{u}_0, \lambda_0)$, $F(\mathbf{x}) = 0$ soluzio erregularra da baldin $F_x(\mathbf{x}_0) \equiv F_x^0$ matrizea ($dim = n \times (n + 1)$), $Rank(F_x(\mathbf{x}_0)) = n$ den.

$$Rank(F_x(\mathbf{x}_0)) = Rank(F_u^0 | F_\lambda^0) = n \Leftrightarrow \begin{cases} (i) & F_u^0 \text{ ez da singularra} \\ & \text{edo} \\ (ii) & \begin{cases} \dim N(F_u^0) = 1 \\ \text{eta} \\ F_\lambda^0 \notin \mathcal{R}(F_u^0) \end{cases} \end{cases}$$

non $N(F_u^0)$: F_u^0 -ren espazio hutsa.
 $\mathcal{R}(F_u^0)$ F_u^0 matrizearen rangoa.

Newton metodoaren erakarren eremua.

F funtzioak baldintza batzuk betetzen dituzenean, Newton metodoaren erakarren eremu batek inguratzen du soluzio adarra. Honela, soluzio adarraren hasierako balio batetik x_0 abiatuz, eta urratsak behar bezain txikiak izanez gero, urrats guztiak erakarren eremuaren barruan izango dira eta Newton metodoak hauek guztietan konbergituko du. Aldiz, urrats handiegia emango bagenu, ez dugu hurbilketa (x^*) on bat izango eta Newtonen konbergentzia ez da ziurra.



Irudia 4.2 Newton erakarren eremua.

4.3- Soluzio periodikoen jarraipena.

Suposatu honako EDA sistema dugula,

$$\mathbf{u}'(t) = \mathbf{f}(\mathbf{u}(t), \lambda), \quad \mathbf{u}(\cdot), \mathbf{f}(\cdot) \in \mathbb{R}^n, \quad \lambda \in \mathbb{R}$$

Ekuazio diferentzialaren periodikotasuna 2π izan dadin honako transformazioa aplikatuko dugu $t \mapsto \frac{2\pi t}{T}$ eta honela periodoa T esplizituki duen ekuazio baliokidea berridatziko dugu,

$$(1) \quad \mathbf{u}'(t) = \frac{T}{2\pi} \mathbf{f}(\mathbf{u}(t), \lambda) \Rightarrow \mathbf{F}(\mathbf{u}, \lambda) = \mathbf{u}'(t) - \frac{T}{2\pi} \mathbf{f}(\mathbf{u}(t), \lambda) = 0 \quad \mathbf{u}(\cdot) \in \mathbb{R}^n, \mathbf{f}(\cdot) \in \mathbb{R}^n, \lambda, T \in \mathbb{R}$$

$$(2) \quad \mathbf{u}(0) = \mathbf{u}(2\pi)$$

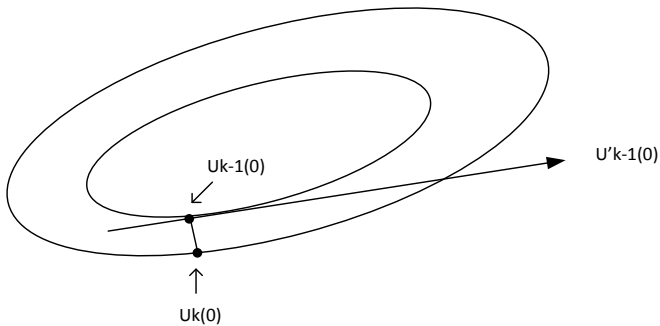
Periodikotasuna.

(1) eta (2) ekuazioek ez dute $\mathbf{u}(t)$ bakarria definitzen. $\mathbf{u}_k(t)$ soluzioa denboran desplazatuz adina soluzio ditugu; $\mathbf{u}_k(t)$ soluzioa periodikoa bada, $\mathbf{u}_k(t + \sigma)$ ere soluzio periodikoa izango da edozein σ -rentzat. Hau da, soluzio kandidatua finkatu behar dugu eta horretarako aingura baldintza zehaztu behar dugu.

Soluzio ezaguna \mathbf{u}_{k-1} eta soluzio kandidatuaren \mathbf{u}_k arteko distantzia $t = 0$ unean minimoa izatea eskatuko dugu:

$$\|\mathbf{u}_{k-1}(0) - \mathbf{u}_k(t)\|^2 \Rightarrow \min_t (\mathbf{u}_{k-1}(0) - \mathbf{u}_k(t))^T (\mathbf{u}_{k-1}(0) - \mathbf{u}_k(t)) \Rightarrow$$

$$(\mathbf{u}_k(t) - \mathbf{u}_{k-1}(0))^T \mathbf{u}'_{k-1}(t) = 0 \Rightarrow \boxed{(\mathbf{u}_k(0) - \mathbf{u}_{k-1}(0))^T \mathbf{f}(\mathbf{u}_{k-1}(0)) = 0}$$



Irudia 4.3 Aingura baldintza interpretazio geometrikoa.

Inbariantea.

Sistemak inbariante bat duenean, goian azaldutako jarraipen metodo estandarrek ez du ondo funtzionatzen. Ekuazio bat erredundantea da eta horregatik planteamendu estandarra okerra da.

Aukera bat, kontserbatzen den ekuazioa kontsideratu eta ezezagun bat ezabatzea litzateke; hainbat eragozpen aipatzen direnez [4, 272.orr], aukera hau baztertuko dugu.

Bigarren aukera bat, sisteman ezezagun bat gehitzea da (α). Ezezagun berri honek (disipazio terminoa) zero balioa izan behar du, soluzioa periodikoa izan dadin. Hau da, guk problemak planteatzerakoan erabiliko dugun aukera.

Demagun gure sistema dinamikoan inbariantea den kantitatea, $I: \mathbb{R}^n \rightarrow \mathbb{R}$, $I(\mathbf{u}(t)) = \text{konstantea}$, orbitaren edozein unetan.

Disipazio terminoa (α) gehituz, ekuazioa honela berriidatziko dugu,

$$\boxed{\mathbf{u}'(t) = \frac{T}{2\pi} (\mathbf{f}(\mathbf{u}(t)) + \alpha \nabla I(\mathbf{u}))}$$

Froga. Konprobatuko dugu, soluzioa existitzeko $\alpha = 0$ izan behar dela.

$(p_{k-1}, T_{k-1}, 0)$ gertu dagoen soluzio periodiko berri bat $(u(0), T, \alpha)$ bilatzen dugu.

Demagun $u_{k-1}(t)$ soluzio periodikoa, $p_{k-1} := u_{k-1}(0)$ hasierako puntua eta T_{k-1} periodoa.

$I(u(t))$ inbariantea denez, $\nabla I(u) \cdot \mathbf{f}(u) \equiv 0$

$$0 = I(u(2\pi)) - I(u(0)) = \int_0^{2\pi} \nabla I(u(t)) \cdot \dot{u}(t) dt = \alpha T \int_0^{2\pi} \|\nabla I(u(t))\|^2 dt$$

Integrala ezin daiteke zero izan, $\nabla I(u(0))$ gertu izanik $\nabla I(p_{k-1}) \neq 0$ bete behar delako.

Beraz, soluzio periodikoen jarraipenean eta sistemak inbariante bat duenean ekuazio sistema laburtuko dugu.

$\mathbf{x}_0 = (\mathbf{u}_0, T_0, 0)$ soluzioa ezaguturik, $\mathbf{x}_1 = (\mathbf{u}_1, T_1, \alpha_1)$ soluzioa kalkulatu nahi dugu.

T , jarraipen aldagaia izanik.

Hasierako soluzio honetatik abiatuta $\mathbf{x}_1^{(0)} = (\mathbf{u}_0, T_0 + \Delta T, 0)$, hurrengo ekuazio sistema ebatzi nahi dugu

$$\mathbf{u}'(t) = \frac{T}{2\pi} (\mathbf{f}(\mathbf{u}(t)) + \alpha \nabla I(\mathbf{u})), \quad \mathbf{u}, \mathbf{f}(\cdot), \nabla I(\mathbf{u}) \in \mathbb{R}^n, \quad T, \alpha \in \mathbb{R}$$

$$\mathbf{u}(0) = \mathbf{u}(2\pi)$$

$$(\mathbf{u}_1(0) - \mathbf{u}_0(0)) \mathbf{f}(\mathbf{u}_0(0)) = 0$$

5-Uraberaren metodoa.

5.1- Sarrera.

Atal honetan soluzio periodikoen jarraipena egiteko algoritmoa proposatuko da. Soluzio periodikoa M ordenako polinomio trigonometrikoaren bidez hurbilduko dugu eta jarraipena egiteko ekuazio sistema Uraberaren metodoaren ideietan oinarrituz ondorioztatuko dugu.

Lehenik, 5.2. atalean Uraberaren metodoa azalduko dugu; polinomio trigonometrikoaren koefizienteak kontsideratuko ditugu ezezagunak. Bigarrenik, 5.3. atalean Uraberaren metodo baliokidea azalduko dugu baina planteamendu honetan soluzio periodikoaren balio diskretuak ditugu ezezagunak. Azkenik, 5.4. atalean soluzioen jarraipena egiteko sistema laburtuko dugu.

5.2- Uraberaren metodoa I.

Orbiten zenbakizko kalkulua egiteko algoritmo bat deskribatuko dugu, Uraberaren metodoa Galerkinen prozedura aplikatzean oinarritutakoa. Algoritmo honetan, orden altuko polinomio trigonometrikoaren koefizienteak (Galerkinen hurbilketa) kalkulatu daitezke doitasun handiarekin, Newtonen iterazioak aplikatuz. Ekuazio diferentzial ez lineal eta periodikoetan erabil daitekeen algoritmoa da, soluzio periodikoen azterketarako.

5.2.1-Deskribapena

Atal honetan azalduetakoa [11] artikuluaen laburpen bat da. Artikuluaren notazioa mantendu da.

Ekuazio diferentzial periodikoa kontsideratuz,

$$\ddot{x} = X(x, \dot{x}, t), \quad x, \dot{x}, X(x, \dot{x}, t) \in \mathbb{R}^d$$

$X(x, \dot{x}, t)$, t aldagai independentearekiko 2π periodikoa izanik.

$T \neq 2\pi$ periododun soluzio periodikoak kalkulatu nahi bada (beti ere, $X(x, \dot{x}, t)$ t aldagai independentearekiko T periodikoa baldin bada), aldagai aldaketa baten bidez lor daiteke ($t \rightarrow \tau = 2\pi t/T$).

Soluzio periodikoa bilatzen dugu (2π periododuna) zein polinomio trigonometrikoaren bidez adierazten dugun

$$x_m(t) = a_0 + \sum_{n=1}^m (a_{2n-1} \sin(nt) + a_{2n} \cos(nt))$$

a_v koefiziente ezezagunak, honako berdintza aplikatuz kalkulatu ditugu,

$$\ddot{x}_m(t) = X_m(x_m(t), \dot{x}_m(t), t)$$

Batetik, $X_m(x_m(t), \dot{x}_m(t), t)$ m ordeneko Fourier seriearen bidez hurbilduko dugu,

$$X_m(x_m(t), \dot{x}_m(t), t) = A_0 + \sum_{n=1}^m (A_{2n-1} \sin(nt) + A_{2n} \cos(nt))$$

A_v , Fourier koefizienteei dagokien ekuazioak,

$$A_0 = \frac{1}{2\pi} \int_0^{2\pi} X[x_m(s), \dot{x}_m(s), s] ds$$

$$A_{2n-1} = \frac{1}{\pi} \int_0^{2\pi} X[x_m(s), \dot{x}_m(s), s] \sin(ns) ds$$

$$A_{2n} = \frac{1}{\pi} \int_0^{2\pi} X[x_m(s), \dot{x}_m(s), s] \cos(ns) ds, \quad n = 1, 2, \dots, m$$

Bestetik $x_m(t)$ deribatuz,

$$\dot{x}_m(t) = \sum_{n=1}^m (-a_{2n-1} n^2 \sin(nt) - a_{2n} n^2 \cos(nt))$$

Hortaz, $\dot{x}_m(t) = X_m(x_m(t), \dot{x}_m(t), t)$ ekuazioaren 1, $\sin nt$ eta $\cos nt$ terminoen koefizienteak berdinduz,

$$0 = A_0 \quad \Rightarrow \quad F_0(\alpha) \equiv \frac{1}{2\pi} \int_0^{2\pi} X[x_m(s), \dot{x}_m(s), s] ds = 0$$

$$-a_{2n-1} n^2 = A_{2n-1} \quad \Rightarrow \quad F_{2n-1}(\alpha) \equiv \frac{1}{\pi} \int_0^{2\pi} X[x_m(s), \dot{x}_m(s), s] \sin(ns) ds + a_{2n-1} n^2 = 0$$

$$-a_{2n} n^2 = A_{2n} \quad \Rightarrow \quad F_{2n}(\alpha) \equiv \frac{1}{\pi} \int_0^{2\pi} X[x_m(s), \dot{x}_m(s), s] \cos(ns) ds + a_{2n} n^2 = 0$$

$n = 1, 2, \dots, m$ eta $\alpha = (a_0, a_1, \dots, a_{2m-1}, a_{2m})$

Polinomio Trigonometrikoaren koefizienteei, m ordeneko Galerkinen hurbilketa esaten zaie eta koefizienteak kalkulatzeko metodoari Galerkin metodoa. Ekuazioak modu orokorrean berridatziz,

$$F_\mu(\alpha) = 0 \quad \mu = 0, 1, \dots, 2m$$

$\hat{\alpha} = (\hat{a}_0, \hat{a}_1, \dots, \hat{a}_{2m-1}, \hat{a}_{2m})$ soluzioaren hurbilpena dela suposatuz, Newtonen metodoa aplikatuko dugu ekuazio sistema ebazteko

$$F_\mu(\hat{\alpha}) + \sum_{v=0}^{2m} J_{\mu v}(\hat{\alpha}) h_v = 0 \quad \mu, v = 0, 1, \dots, 2m$$

$$h_v = a_v - \hat{a}_v, \quad J_{\mu v}(\hat{\alpha}) = \frac{\partial F_\mu}{\partial a_v}(\hat{\alpha})$$

Urabek F_μ eta $\partial F_\mu / \partial a_\nu$ balioen zenbakizko hurbilketa kalkulatzeko metodoa zehazten du. A_ν Fourier koefizienteak modu honetan kalkulatu ditugu,

$$A_0 = \frac{1}{2N} \sum_{i=1}^{2N} X[x_m(t_i), \dot{x}_m(t_i), t_i]$$

$$A_{2n-1} = \frac{1}{\pi} \sum_{i=1}^{2N} X[x_m(t_i), \dot{x}_m(t_i), t_i] \sin n t_i$$

$$A_{2n} = \frac{1}{\pi} \sum_{i=1}^{2N} X[x_m(t_i), \dot{x}_m(t_i), t_i] \cos n t_i$$

non $n = 1, 2, \dots, m$ $N > m$

$$t_i = \frac{2i-1}{2N} \pi \quad i = 1, 2, \dots, 2N.$$

5.2.2-Adibidea: Hill's Variation Orbit.

Hauek ditugu, Ilargiaren mugimendua deskribatzen duten Hill-en ekuazioak,

$$\ddot{x}_1 - 2\varepsilon \dot{x}_2 + \left(\frac{\lambda}{r^3} - 3\varepsilon^2 \right) x_1 = 0, \quad \ddot{x}_2 + 2\varepsilon \dot{x}_1 + \frac{\lambda}{r^3} x_2 = 0$$

$$\text{non } r = (x_1^2 + x_2^2)^2, \quad \lambda = (1 + \varepsilon)^2 / \beta^2$$

$$\varepsilon = 0.080848933808312 \text{ eta } \beta = 0.999093141975298$$

Parametroak $m = 13$ eta $N = 35$ balioekin finkatuz eta hasierako soluzioa honakoa kontsideratuz

$$\bar{a}_0 = 0 \quad \bar{a}_1 = 0 \quad \bar{a}_2 = 1 \quad \bar{a}_3 = \bar{a}_4 = \dots = \bar{a}_{2m} = 0$$

$$\bar{b}_0 = 0 \quad \bar{b}_1 = 0 \quad \bar{b}_2 = 1 \quad \bar{b}_3 = \bar{b}_4 = \dots = \bar{b}_{2m} = 0$$

Gure Mathematicako inplementazioan, Uraberen metodoa aplikatuz honako soluzioa lortu dugu,

$$x_1(t) = 0.9913042530384607 \text{Cos}[t] + 0.0015158712700485 \text{Cos}[3t] + 0.0000058811169710 \text{Cos}[5t] \\ + 3.00439151 \times 10^{-8} \text{Cos}[7t] + 1.753324 \times 10^{-10} \text{Cos}[9t] + 1.1069 \times 10^{-12} \text{Cos}[11t] + 7.3 \\ \times 10^{-15} \text{Cos}[13t]$$

$$x_2(t) = 1.0086957469615403 \text{Sin}[t] + 0.0015155436890769 \text{Sin}[3t] + 0.0000058761961858 \text{Sin}[5t] \\ + 3.00193478 \times 10^{-8} \text{Sin}[7t] + 1.752040 \times 10^{-10} \text{Sin}[9t] + 1.1061 \times 10^{-12} \text{Sin}[11t] + 7.3 \\ \times 10^{-15} \text{Sin}[13t]$$

Azpimarratu, soluzioa Newtonen 4 iterazio eginez lortzen dela eta Hill-ek kalkulatuakoaren soluzio berdina dela.

Kodea

HillVariations.nb

5.3-Uraberaren metodoa II.

Aurreko atalean azaldutako metodoa errepikatuz, orbita periodikoak kalkulatzeko ekuazio sistema baliokidea proposatuko da. Soluzioa adierazteko ezezagunak polinomio trigonometrikoaren koefizienteak erabili beharrean soluzioaren balio diskretuak erabiliko ditugu. Garapen osoa deskribatuko da eta bukaeran soluzio periodikoak aztertzeke ekuazio sistema laburtuko dugu.

5.3.1 Deskribapena.

Honako T periododun EDA izanik,

$$\frac{du}{dt} = f(u(t)), \quad u(\cdot), f(\cdot) \in \mathbb{R}^n$$

Ekuazio diferentziala berridatziko dugu 2π periododuna izan dadin,

$$\frac{du}{dt} = \frac{T}{2\pi} f(u(t)), \quad T \text{ periodoa izanik}$$

$u(t)$ soluzioa, uniformeki banatutako $(N + 1)$ puntu diskretuen bidez adieraziko dugu.

$$t_j = \frac{2\pi j}{N}, \quad u_j = u(t_j), \quad f_j = f(u_j), \quad j = 0, \dots, N$$

Soluzioa, M ordenako polinomio trigonometriko baten bidez hurbildu daiteke:

$$u_M(t) = a_0 + \sum_{k=1}^M (a_{2k-1} \sin(kt) + a_{2k} \cos(kt))$$

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} u_M(t) dt$$

$$a_{2k-1} = \frac{1}{\pi} \int_0^{2\pi} \sin(kt) u_M(t) dt, \quad k = 1, \dots, M$$

$$a_{2k} = \frac{1}{\pi} \int_0^{2\pi} \cos(kt) u_M(t) dt, \quad k = 1, \dots, M$$

Eta dagokion bertsio diskretua

$$u_j = a_0 + \sum_{k=1}^M (a_{2k-1} \sin(kt_j) + a_{2k} \cos(kt_j)), \quad j = 0, \dots, N$$

$$a_0 \cong \tilde{a}_0 = \frac{1}{N} \sum_{l=0}^{N-1} u_l$$

$$a_{2k-1} \cong \tilde{a}_{2k-1} = \frac{2}{N} \sum_{l=0}^{N-1} \sin(k t_l) u_l, \quad k = 1, \dots, M$$

$$a_{2k} \cong \tilde{a}_{2k} = \frac{2}{N} \sum_{l=0}^{N-1} \cos(k t_l) u_l, \quad k = 1, \dots, M$$

Soluzio periodikoaren u_V balio ezezagunak honako berdintasunetik kalkulatuko ditugu,

$$\frac{u_M(t)}{dt} = \frac{T}{2\pi} f(u_m(t))$$

Batetik EDA periodikoa M ordenako polinomio trigonometrikoa baten bidez hurbilduko dugu,

$$\frac{T}{2\pi} f(u_m(t)) = A_0 + \sum_{k=1}^M (A_{2k-1} \sin(kt) + A_{2k} \cos(kt))$$

$$A_0 = \frac{1}{2\pi} \int_0^{2\pi} \frac{T}{2\pi} f(u_m(t)) dt$$

$$A_{2k-1} = \frac{1}{\pi} \int_0^{2\pi} \sin(kt) \frac{T}{2\pi} f(u_m(t)) dt, \quad k = 1, \dots, M$$

$$A_{2k} = \frac{1}{\pi} \int_0^{2\pi} \cos(kt) \frac{T}{2\pi} f(u_m(t)) dt, \quad k = 1, \dots, M$$

Eta dagokion bertsio diskretua,

$$\frac{T}{2\pi} f(u_j) = A_0 + \sum_{k=1}^M (A_{2k-1} \sin(kt_j) + A_{2k} \cos(kt_j)), \quad j = 0, \dots, N$$

$$A_0 \cong \tilde{A}_0 = \frac{T}{2\pi N} \sum_{l=0}^{N-1} f_l$$

$$A_{2k-1} \cong \tilde{A}_{2k-1} = \frac{T}{\pi N} \sum_{l=0}^{N-1} \sin(k t_l) f_l, \quad k = 1, \dots, M$$

$$A_{2k} \cong \tilde{A}_{2k} = \frac{T}{\pi N} \sum_{l=0}^{N-1} \cos(k t_l) f_l, \quad k = 1, \dots, M$$

Bestetik $u_M(t)$ deribatuz,

$$\frac{u_M(t)}{dt} = \sum_{k=1}^M (k a_{2k-1} \cos(kt) - k a_{2k} \sin(kt))$$

$$\frac{du_j}{dt} = \sum_{k=1}^M (k a_{2k-1} \cos(kt_j) - k a_{2k} \sin(kt_j)), \quad j = 0, \dots, N$$

$\frac{u_M(t)}{dt} = \frac{T}{2\pi} f(u_M(t))$ berdintasunean 1, $\sin(kt)$ eta $\cos(kt)$ osagaien koefizienteak berdinuz

$$\sum_{k=1}^M (k a_{2k-1} \cos(kt_j) - k a_{2k} \sin(kt_j)) = A_0 + \sum_{k=1}^M (A_{2k-1} \sin(kt_j) + A_{2k} \cos(kt_j))$$

$$0 = \tilde{A}_0 \quad \Rightarrow \quad \frac{T}{2\pi N} \sum_{l=0}^{N-1} f_l = 0$$

$$-k a_{2k} = \tilde{A}_{2k-1} \quad \Rightarrow \quad a_{2k} = -\frac{T}{k\pi N} \sum_{l=0}^{N-1} \sin(k t_l) f_l$$

$$k a_{2k-1} = \tilde{A}_{2k} \quad \Rightarrow \quad a_{2k-1} = \frac{T}{k\pi N} \sum_{l=0}^{N-1} \cos(k t_l) f_l$$

Honako garapena egingo dugu, u_V ezezagunen menpeko ekuazio sistema lortzeko,

$$u_j = a_0 + \sum_{k=1}^M (a_{2k-1} \sin(kt_j) + a_{2k} \cos(kt_j)) \Rightarrow$$

$$u_j = a_0 + \sum_{k=1}^m \left(\frac{T}{\pi N k} \sum_{l=0}^{N-1} \cos(kt_l) f_l \sin(kt_j) - \frac{T}{\pi N k} \sum_{l=0}^{N-1} \sin(kt_l) f_l \cos(kt_j) \right)$$

$$u_j = a_0 + T \sum_{l=0}^{N-1} \mu_{jl} f_l$$

$$\text{non } \mu_{jl} = \frac{1}{\pi N k} \sum_{k=1}^m (\cos(kt_l) \sin(kt_j) - \sin(kt_l) \cos(kt_j)) \Rightarrow \mu_{jl} = -\frac{1}{\pi k} \sum_{k=1}^m (\sin(k(t_l - t_j)))$$

$$0 = \frac{T}{2\pi N} \sum_{l=0}^{N-1} f_l \Rightarrow \sum_{l=0}^{N-1} f_l = 0$$

$$t_j = \frac{2\pi j}{N}, \quad u_j = u(t_j), \quad j = 0, \dots, N-1, \quad k = 1, \dots, M$$

Laburtuz lortutako ekuazioak hauek dira,

$$F_j(\omega) = u_j - a_0 - T \sum_{l=0}^{N-1} \mu_{jl} f_l$$

$$G(\omega) = \sum_{l=0}^{N-1} f_l = 0$$

$$j = 0, \dots, N-1, \quad N > 2M, \quad \omega = (u_0, u_1, \dots, u_{N-1}, a_0)$$

$$\text{non } \mu_{jl} = -\frac{1}{\pi k} \sum_{k=1}^M (\sin(k(t_l - t_j)))$$

$$u_j = u(t_j), \quad f_j = f(u_j)$$

Ekuazioak: $|F_j| = N, |G| = 1 \Rightarrow N + 1$.

Ezezagunak: $|u_j| = N, |a_0| = 1 \Rightarrow N + 1$.

5.3.2-Adibidea: Hill's Variation Orbit

Planteamendu berriarekin, "Hill's Variation Orbit" adibidearen implementazioa Mathematican.

Kodea:

HillVariations.nb

5.4- Soluzio periodikoen jarraipena.

Aurreko ataleko soluzio periodikoak kalkulatzeko ekuazioetan oinarrituz, soluzio periodikoen jarraipena egiteko sistema zehatza definituko dugu. Aingura baldintza eta disipazio terminoa gehituz honakoa da ekuazio sistema,

$Hondarra(\omega) = (F_j(\omega), G(\omega), (\hat{u}_0 - u_0) f(u_0)), \quad \omega = (u_0, u_1, \dots, u_{N-1}, a_0, \alpha)$

$$1) \quad F_j(\omega) = u_j - a_0 - T \sum_{l=0}^{N-1} \mu_{jl} f_l$$

$$2) \quad G(\omega) = \sum_{l=0}^{N-1} f_l = 0$$

$$3) \quad (\hat{u}_0 - u_0) f(u_0) = 0 \text{ (aingura baldintza)}$$

$$j = 0, \dots, N-1, \quad N > 2M$$

$$\mu_{jl} = -\frac{1}{\pi k} \sum_{k=1}^M (\sin(k(t_l - t_j)))$$

$$u_j = u(t_j), \quad f_j = f(u_j) + \alpha g(u_j)$$

\hat{u}_0 : aurreko soluzioaren hasierako balioa

Ekuazio kopurua, $1) N * n, 2) n, 3) 1$ $(N + 1) * n + 1$.

Ezezagun kopurua, $u_j, a_0 \in \mathbb{R}^n, \alpha \in \mathbb{R}$ $(N + 1) * n + 1$.

Jarraipen parametroa. $T \in \mathbb{R}$.

Algoritmoa.

Hasierako $\mathbf{x}_0 = (\omega_0, T_0)$ soluzioa ezaguna dugu

$$Hondarra(\mathbf{x}_0) = 0, \quad \mathbf{x}_0 = (\omega_0, T_0), \quad \omega_0 \in \mathbb{R}^{(N+1)*n+1}, \quad T_0 \in \mathbb{R}.$$

Iterazioa (z)

$$z = z + 1;$$

$$T_z = T_{z-1} + \Delta T$$

$$\omega_z = \text{Newton}(Hondarra(T_z), \omega_{z-1})$$

Iterazio_amaiera

6-Gure inplementazioa.

Soluzio periodikoen jarraipenerako programa berri bat inplementatuko dugu. Uraberen metodoan oinarrituko gara eta C-lengoian garatuko dugu. R3BP problemarekin hainbat esperimentu egingo ditugu.

6.1- M koefiziente kopurua.

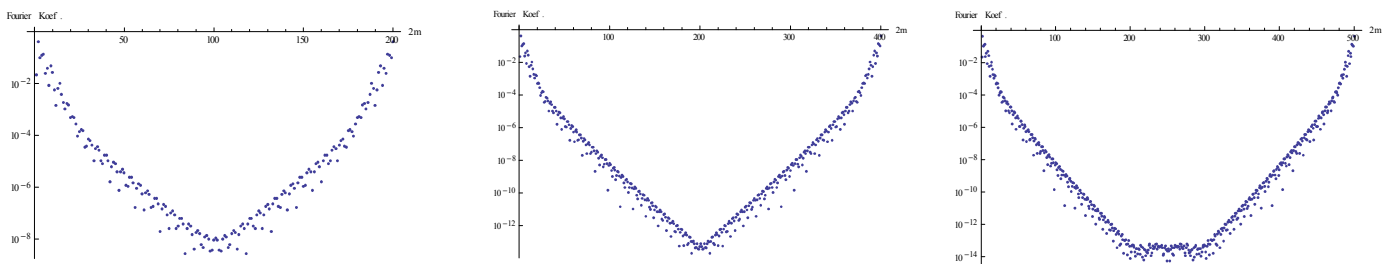
Uraberen metodoan azaldu dugunez, soluzioa M ordenako polinomio trigonometrikoa baten bidez hurbilduko dugu:

$$\mathbf{u}_j = \mathbf{a}_0 + \sum_{k=1}^M (a_{2k-1} \sin(kt_j) + a_{2k} \cos(kt_j)), \quad j = 0, \dots, N$$

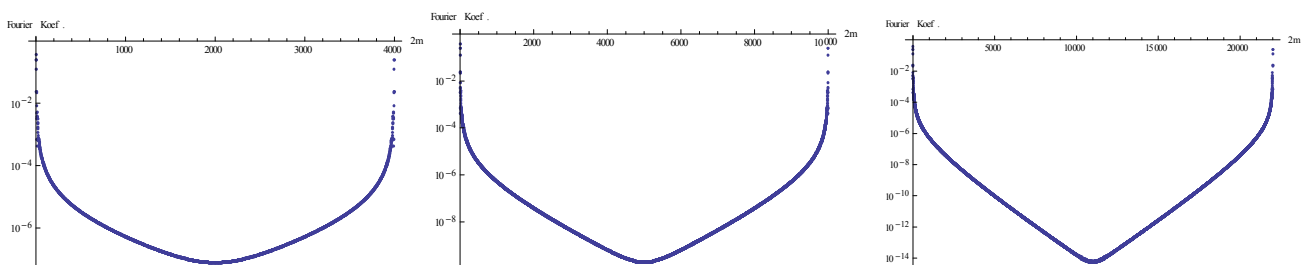
non $N = 2M + 1$

Hasierako soluzio ezagunaren hurbilketa on bat lortzeko zein den M balio egokia aztertuko dugu R3BP problemarentzat. M handitzen dugun neurrian, Fourieren koefizienteak gero eta txikiagoak dira. Koefizienteak zero balioa hartzen duten M balioarentzat, soluzioaren hurbilketa on bat egiten ari garela esan nahi du.

Erregularizazioa erabiltzen dugunean, $M = 200$ balioarekin koefizienteak zero balioa hartzen dute. Aldiz, erregularizazioa erabiltzen ez dugunean $M = 11000$ balioarekin lortzen dugu koefizienteak zero izatea. Alde batetik, erregularizazioaren dugun kasuan askoz ere M txikiagoa nahikoa dugula ikusten dugu. Bestalde, erregularizazio gabeko bertsioan M oso balio handia behar dugu hurbilpen ona lortzeko eta hainbeste faktore behar izanak, arazoak sortuko dizkigu.



Irudia 6.1 EDAn Erregularizazioa erabiliz, x koordenatuari dagozkion Fourieren koefizienteak $M=100, 200, 250$ balioetarako. `ListPlot[Log[10, Abs[IFFT[qa]]]]`



Irudia 6.2 EDAn Erregularizazioa gabeko kasua, x koordenatuari dagozkion Fourieren koefizienteak $M=2000, 5000, 10000$ balioetarako. `ListPlot[Log[10, Abs[IFFT[qa]]]]`

Guk proposatutako inplementazioan ezin ditugu erregularizazio gabeko bertsioaren probak egin. Hurbilpen on bat izateko, $M = 11.000$, kokatu dugu behar dugun koefiziente kopurua. Aztertu dezagun zein tamainako jakobiarra behar dugun R3BP problemarentzat (4 dimentsio):

$$M = 11.000 \Rightarrow N = 2M + 1 \Rightarrow N = 22.000$$

\Rightarrow Jakobiarraren dimentsioa ($4N \times 4N$) \Rightarrow (88.000×88.000) Matrizearen osagai kopurua.

Bestalde zenbaki erreala memorian adierazteko 8 byte (64bit) behar ditugula kontsideratuz, $8 * (88.000 \times 88.000) = 8 * 88 * (1.000 * 1.000) = 7.524 * 10^6 \Rightarrow$ 7.5 GB Jakobiarrarentzat beharrezko memoria.

Gure konputagailuen mugaren ondorioz, ezin ditugu erregularizazio gabeko bertsioaren frogak egin.

Kodea

- MAzterketa1.nb: erregularizatuko bertsioaren M koefizienteen kalkulua.
- * • MAzterketa2.nb: erregularizatu gabeko bertsioaren M koefizienteen kalkulua.

6.2- Errore neurketa.

Soluzio bakoitzaren errorearen neurketa 4 modutara egingo dugu: soluzioaren ordezkapena ekuazio diferentzian, orbitan energia aldaketa, kokapenaren periodikotasuna eta Fourier seriearen koefizienteak kalkulatu ditugu. Soluzioa lortzeko exekuzio denbora ere azalduko dugu.

- EDA ordezkapena.
- Energia integrala.
- Periodikotasuna.
- Fourier Seriearen koefizienteak.
- Denbora.

6.2.1- EDA ordezkapena.

Gure soluzio periodiko (2π) diskretuaren dagokion Fourierren koefizienteak kalkulatu ditugu

$$u_j, j = 0, \dots, N - 1$$

$$a_0 = \frac{1}{N} \sum_{l=0}^{N-1} u_l, a_{2k-1} = \frac{2}{N} \sum_{l=0}^{N-1} \sin(k t_l) u_l, a_{2k} = \frac{2}{N} \sum_{l=0}^{N-1} \cos(k t_l) u_l, \quad k = 1, \dots, M$$

Eta modu honetan M ordenako polinomio trigonometrikoa,

$$u(t) = a_0 + \sum_{k=1}^M (a_{2k-1} \sin(kt) + a_{2k} \cos(kt))$$

$u(t)$ soluzioak honako berdintasuna bete behar du,

$$\frac{u(t)}{dt} = \frac{T}{2\pi} f(u(t)) \Rightarrow \frac{u(t)}{dt} - \frac{T}{2\pi} f(u(t)) = 0$$

Integralaren bidez neurtuko dugu,

$$\text{Errore1} = \int_0^{2\pi} (u'(s))^2 - \left(\frac{T}{2\pi} f(u(s))\right)^2 ds$$

*Oharra:

Mathematican Errore1 kalkulatu dugunean integralak honako arazoak izan ditu:

A warning message is generated because the integral has a singularity inside the integration region.

A warning is generated because of the singularity at the origin for this highly oscillatory integral.

6.2.2- Energiaren integrala.

R3BP problemaren energia,

$$H(x(t), y(t), p_x(t), p_y(t)) = \frac{1}{2}(p_x^2 + p_y^2) + p_x y - p_y x - \left(\frac{1-\mu}{D_1^{1/3}}\right) - \frac{\mu}{D_2^{1/3}} - \frac{1}{2} \mu (1-\mu)$$

Ezaguna da energia R3BP problemaren integrala edo inbariantea dela, hau da, orbitaren zehar konstantea mantentzen den balioa,

$$H_0 = H(u_0), \quad H(u(t)) - H_0 = 0, \quad t_0 \leq t \leq t_f$$

Beraz, energiaren bidezko errorea,

$$\text{Errore2} = \text{Max}(H(u_j) - H_0), \quad t_j = \frac{2\pi j}{N}, u_j = u(t_j) \quad j = 0, \dots, N-1$$

6.2.3- Periodikotasuna.

Gure soluzioa $u(t)$ periodikoa denez, T periodoa izanik: $\text{Norma}(u(T) - u(0)) = 0$

Lortutako soluzioaren lehen balioa hartuz (u_0), Mathematica softwarean zehaztasun handiarekin integratuko dugu eta kokapen errorea kalkulatu dugu.

```
sol = NDSolve[eqnsedo[x0, y0, vx0, vy0, 0], {x, y, px, py, t}, {t, 0, T}, Method -> {"ImplicitRungeKutta", DifferenceOrder -> 20}, MaxSteps -> 10^6, WorkingPrecision -> prec, AccuracyGoal -> acc][[1]];
```

Kokapen errorea kontsideratuko dugu,

$$\text{Errore3} = \text{Norma}(\{x(T), y(T)\} - \{x(0), y(0)\})$$

6.2.4- Fourier seirearen koefizienteak.

$u(t)$ soluzio periodikoa bada, polinomio trigonometriko batez hurbil daiteke,

$$u(t) = a_0 + \sum_{k=1}^M (a_{2k-1} \sin(kt) + a_{2k} \cos(kt))$$

M handitzen dugun neurrian a_{2k-1}, a_{2k} zerora hurbiltzen dira. Azken lau koefizienteek dagokien norma kalkulatu dugu; hurbilketa ona izateko koefiziente hauek zero izan behar dute.

$$\text{Errore4} = \text{Norma}(a_{2M}, a_{2M-1}, a_{2M-2}, a_{2M-3})$$

6.3- Mathematica bertsioa.

Mathematica Softwarea erabiliz implementazio zuzena egin dugu. Garapen teorikoa, modu praktikoan zuzena dela ikusteko balio izan du. R3BP erregularizazio bertsioa probatu dugu:

- 1- Jarraipena aurrerantz : $M = 200$ eta $T_f = T_0 + 0.01$, $T_0 = 29.804721$
- 2- Jarraipena atzerantz : $M = 200$ eta $T_f = T_0 - 0.01$, $T_0 = 29.804721$

Egindako azterketan gure soluzio ezagunaren erroreak kalkulatu ditugu, erreferentzi gisa izateko.

	Errore1	Errore2	Errore3	Errore4	Denbora
1- Jarraipena aurrerantz	4×10^{-20}	1×10^{-10}	1×10^{-8}	3×10^{-13}	46 min
2- Jarraipena atzerantz:	3×10^{-20}	1×10^{-10}	1×10^{-8}	3×10^{-13}	46 min

M parametroaren balioa handitzean exekuzio denborak oso luzeak izan dira. Hauek izan dira lortu ditugun denborak,

M		Iterazioak	Denborak
100	$T_0 + 0.01$	10	13.11 minutu
200	$T_0 + 0.01$	10	46.45 minutu

Mathematica lengoia Interpretatua baino eraginkorragoa den lengoia konpilatu bat erabiltzeko beharra ikusi da. Implementaziorako C-lengoia aukeratu dugu.

Kodea.

JarraipenaR3BP.nb: 6.3. ataleko kodearen zehaztasuneekin.

6.4- C-programa bertsioa.

6.4.1- Garapena.

Soluzioaren jarraipenaren implementazio batean oinarrituko gara. Soluzio jarraipenaren ideia nagusia gogoratuko dugu. μ parametroaren arabera jarraipena,

$$F(x, \mu), x, F(\cdot) \in \mathbb{R}^n, \mu \in \mathbb{R}$$

$$\mu = \mu_0 \Rightarrow F(x, \mu_0) \quad \text{Problemaren soluzioa } (x_0) \text{ ezaguna dugu non } F(x_0, \mu_0) = 0$$

$$\mu = \mu_{end} \Rightarrow F(x, \mu_{end}) \quad \text{Problemaren soluzioa } (x_{end}) \text{ kalkulatu nahi dugu non } F(x_{end}, \mu_{end}) = 0$$

$\mu_0 \rightarrow \mu_{end}$ urrats ezberdinak emango ditu. i . iterazio izanik,

$$\mu_i = \mu_{i-1} + \Delta\mu_i$$

$$F(x, \mu_i) \text{ Newtonen metodoa aplikatuz, } x_i \text{ soluzioa } F(x_i, \mu_i) = 0$$

C-lengoaian egindako inplementazioari gure problemaren hondarra eta jakobiarra zehaztuz, μ parametroaren arabera soluzioaren jarraipena egingo dugu. 5.4. atalean definitutako ekuazio sistemaren honako funtzioak programatuko ditugu:

- *Hondarra* $\Rightarrow F(x, \mu)$
- *Jakobiarra* $\Rightarrow \text{Jac}(F(x, \mu))$
- *Hasierako soluzioa* $\Rightarrow x_0$

Uraberen metodoa.

Uraberen metodoari dagozkion. Hondarra eta Jakobiarra azalduko ditugu,

$\mu \equiv T$ (periodoa)

$$\text{Hondarra} = \left(u_j - a_0 - T \sum_{l=0}^{N-1} \mu_{jl} f_l, \quad qb_0, \quad \sum_{l=0}^{N-1} f_l \right), \quad f_l = f(u_l) + \alpha \nabla I(u_l)$$

$$\text{Jakobiarra} = \begin{pmatrix} u_0 & u_1 & u_2 & \dots & u_{N-1} & a_0 & \alpha \\ \blacksquare & \blacksquare & \blacksquare & \dots & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \dots & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \dots & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{pmatrix}$$

Jakobiarraren espresioak aztertuta Jakobiarra dentsoa dela ikus daiteke.

$$F_j(\omega) = u_j - a_0 - T \sum_{l=0}^{N-1} \mu_{jl} f_l, \quad \omega = (u_0, u_1, \dots, u_{N-1}, a_0, \alpha)$$

$$\frac{F_j(\omega)}{du_i} = \frac{du_j}{du_i} - T \mu_{ji} \frac{df_i}{du_i}, \quad i = 0, \dots, N-1$$

$$\frac{F_j(\omega)}{da_0} = -1$$

$$\frac{F_j(\omega)}{d\alpha} = -T \sum_{l=0}^{N-1} \mu_{jl} \nabla I(u_j)$$

$$qb_0 = 0$$

$$\frac{dq_{b_0}}{qb_0} = 1, \quad \text{gainontzekoak} = 0$$

$$G(\omega) = \sum_{l=0}^{N-1} f_l$$

$$\frac{dG(\omega)}{du_i} = \frac{df_i}{du_i}, \quad i = 0, \dots, N-1$$

$$\frac{dG(\omega)}{da_0} = 0$$

$$\frac{dG(\omega)}{d\alpha} = \sum_{l=0}^{N-1} \nabla I(u_j)$$

*Aingura baldintza.

5.4. atalean aingura baldintza $(\hat{u}_0 - u_0) f(u_0) = 0$ definitu genuen. R3BP probleman, aldiz, aingura baldintza modu errezago batean finkatu dugu: $qb_0 = y_0 = 0$

Kodea.

64-Cprograma: azpidirektorian aurkituko dituzu programaren fitxategi guztiak. Zehazki, "homotopiarakofuntzioak.c" fitxategian aurkituko dituzue funtzio hauen inplementazioak,

```
void hondarra(int dim, val_type *x, val_type mu, val_type T, val_type em[])
void japexac (int dim, val_type *x, val_type *jak, val_type mu, val_type T)
```

Eranskinean (E1-Homotopia.pdf) programa erabiltzeko argibideak ematen ditugu.

6.4.2- Esperimentuak.

6.4.2.1- Mathematica eta C-Programaren konparaketa.

Lehenik, Mathematica-ko bertsioarekin konparatzea pentsatu da. Gure C-programarekin eta Mathematicako bertsioarekin soluzio berdina lortu da eta gainera exekuzio denborak asko gutxitu dira.

Bi kasu frogatu dira:

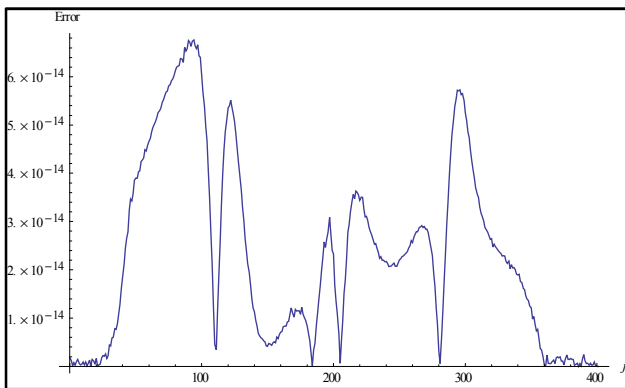
- 1- Jarraipena aurrerantz: $M = 200$ eta $T_f = T_0 + 0.01$, $T_0 = 29.804721$.
- 2- Jarraipena atzerantz: $M = 200$ eta $T_f = T_0 - 0.01$, $T_0 = 29.804721$.

Mathematica soluzioa **MathSol** eta C-programaren soluzioa **CPrgSol** badira, errorea era honetan kalkulatu dugu:

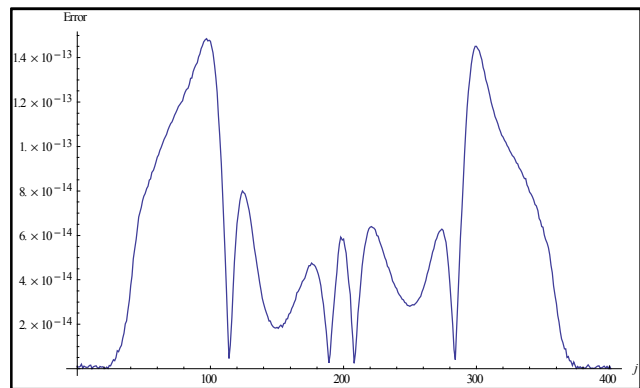
$$\left\{ \begin{array}{l} \text{MathSol} = \{x_m(j), y_m(j), px_m(j), py_m(j)\} \\ \text{CPrgSol} = \{x_c(j), y_c(j), px_c(j), py_c(j)\} \end{array} \right\} \Rightarrow \text{Errorea} = \text{Max}(\text{Abs}[\text{MathSol} - \text{CPrgSol}])$$

$$j = 0, \dots, N - 1$$

	ErrX	ErrY	ErrPx	ErrPy	Denbora
1-j. aurrerantz	6×10^{-14}	5×10^{-14}	1×10^{-13}	1×10^{-13}	1 min
2-j. atzerantz	1×10^{-13}	1×10^{-13}	2×10^{-13}	2×10^{-13}	1 min



Irudia 6.3 Jarraipena aurrerantz $\{x_m(j) - x_c(j)\}$



Irudia 6.4 Jarraipena atzerantz $\{x_m(j) - x_c(j)\}$

Kodea.

KonparaketaMatC.nb: atal honen kalkuluaren zehaztapenak.

6.4.2.2-Soluzio periodikoen jarraipena.

Bi esperimentu egin ditugu: lehen esperimentuan $M = 200$ finkatuta, tarte eta norabide ezberdineko jarraipenak frogatu dira; bigarren esperimentuan ordea, jarraipen tarte finkatuta $T_f = 31$, M balio ezberdinetarako frogak egin dira.

Esperimentu-1.

Lehen esperimentu taldean, jarraipen mota ezberdinak exekutatu ditugu: jarraipen tarte txikia eta handiak; jarraipena aurrerantz eta atzerantz.

- 1- $M = 200$ eta $T_f = T_0 + 0.01$, $T_0 = 29.804721$.
- 2- $M = 200$ eta $T_f = T_0 - 0.01$, $T_0 = 29.804721$.
- 3- $M = 200$ eta $T_f = 31$.
- 4- $M = 250$ eta $T_f = 31$.
- 5- $M = 200$ eta $T_f = 28$.

		Errore1	Errore2	Errore3	Errore4	CPU Time
M=200	1- J. aurrerantz $T_f = T_0 + 0.01$	4×10^{-20}	1×10^{-10}	3×10^{-9}	3×10^{-13}	1m4s
M=200	2- J. atzerantz: $T_f = T_0 - 0.01$	3×10^{-20}	1×10^{-10}	2×10^{-10}	3×10^{-13}	1m
M=200	3-Jarraipena $T_f = 31$	1×10^{-14}	2×10^{-7}	1×10^{-6}	1.10^{-10}	1m53s
M=250	4-Jarraipena $T_f = 31$	1×10^{-18}	2×10^{-9}	4×10^{-9}	1×10^{-12}	2m49.6s
M=200	5-Jarraipena $T_f = 28$	6×10^{-26}	2×10^{-14}	6×10^{-8}	1×10^{-16}	3m35s

Esperimentu hauetan azpimarratzekoa,

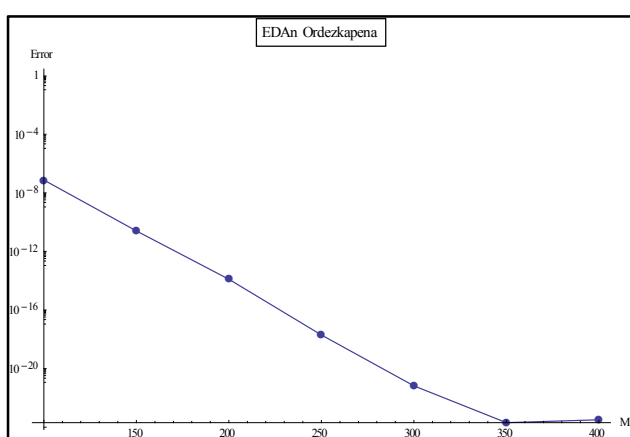
- Jarraitasun tarte txikia denean, doitasun handiko soluzioak lortu ditugu.
- Jarraitasun tarte handia denean, atzerantz emaitza hobekak izan ditugu.
- Jarraipena $T_f = 31$ (aurrerantz). Satelitea ilargira hurbiltzen da, hau da, zailtasuna handitzen ari gara. Hasierako egoerarentzat finkatutako koefiziente balioarekin ($M = 200$) jarraipenean lortutako soluzioaren $errore2 = 1 \times 10^{-7}$ izan da. Logikoa dirudi kolisio egoera areagotzen denez, koefiziente kopuru handiagoa behar izatea ($M = 250 \Rightarrow errore2 = 2 \times 10^{-9}$).
- Jarraipena $T_f = 28$ (atzerantz). Satelitea ilargitik urruntzen da, hau da, kolisio egoera gutxitzen ari gara. Hasierako egoerarentzat finkatutako koefiziente balioarekin ($M = 200$) jarraipenean lortutako soluzioaren $inbariante\ err = 2 \times 10^{-14}$ izan da.

Esperimentu-2.

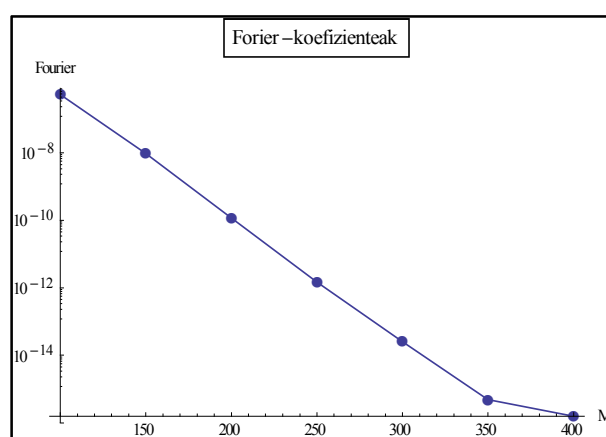
Bigarren esperimentu taldean, jarraipen zehatz bat ($T_0 = 29.804721 \rightarrow T_f = 31$) M koefiziente ezberinetarako exekutatu dugu.

- 1- $M = 100$ eta $T_f = 31$.
- 2- $M = 150$ eta $T_f = 31$.
- 3- $M = 200$ eta $T_f = 31$.
- 4- $M = 250$ eta $T_f = 31$.
- 5- $M = 300$ eta $T_f = 31$.
- 6- $M = 350$ eta $T_f = 31$.
- 7- $M = 400$ eta $T_f = 31$.

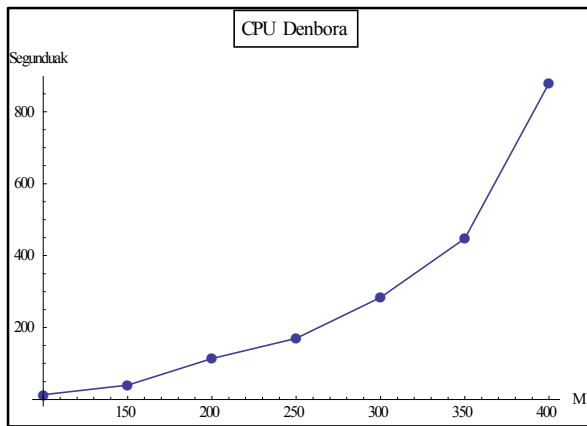
		Errore1	Errore2	Errore3	Errore4	CPU Time
M=100	Jarraipena T=31	7×10^{-8}	6×10^{-4}	3×10^{-3}	5×10^{-7}	12.6s
M=150	Jarraipena T=31	2×10^{-11}	8×10^{-6}	5×10^{-5}	9×10^{-9}	39.5s
M=200	Jarraipena T=31	1×10^{-14}	2×10^{-7}	1×10^{-6}	1×10^{-10}	1m53.9s
M=250	Jarraipena T=31	1×10^{-18}	2×10^{-9}	4×10^{-9}	1×10^{-12}	2m49.6s
M=300	Jarraipena T=31	6×10^{-22}	1×10^{-11}	1×10^{-8}	2×10^{-14}	4m43.2s
M=350	Jarraipena T=31	1×10^{-24}	2×10^{-12}	3×10^{-8}	5×10^{-16}	7m27.4s
M=400	Jarraipena T=31	3×10^{-24}	1×10^{-12}	8×10^{-9}	1×10^{-16}	14m39.1s



Irdia 6.5 Errore1 dagokion grafikoa.



Irdia 6.6 Errore4 dagokion grafikoa.



Irudia 6.7 Denboraren grafikoa.

Esperimentu hauetan azpimarratzekoa, koefiziente kopurua handituz errorea esponentzialki txikitzen dela.. Bestalde konputazio denbora M koefiziente kopuruaren proportzionala da.

Kodea.

ErroreAzterketaC.nb: 6.4.2.2. atalaren esperimentuen zehaztasunak.

Laburpena.nb: 6.4.2.2. atalaren irudien laburpenak.

7-AUTO

7.1-Sarrera.

AUTO softwarearen lehen bertsioa 1986. urtekoa dugu eta sistema dinamikoak aztertzeko *zenbakizko jarraipenen* algoritmo nagusia bilakatu da. AUTO ekuazio diferentzialen mugaldeko formulazioan oinarritzen da eta mugaldeko problema hauek ebazteko, zatikako polinomioen kolokazio metodoa (Orthogonal Collocation with piecewise polynomials) erabiltzen du.

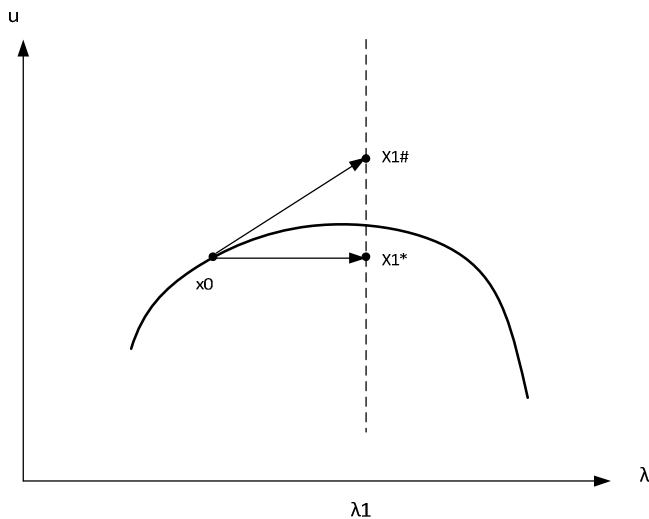
AUTOren oinarriak diren zenbakizko jarraipena eta kolokazio metodoak azalduko ditugu. Bigarrenik softwarea erabiltzeko hainbat zehaztapen emango ditugu. Ondoren, R3BP problemarentzat bi esperimentu ezberdin egingo ditugu. Azkenik ondorio batzuk aipatuko ditugu.

7.2-Zenbakizko jarraipena.

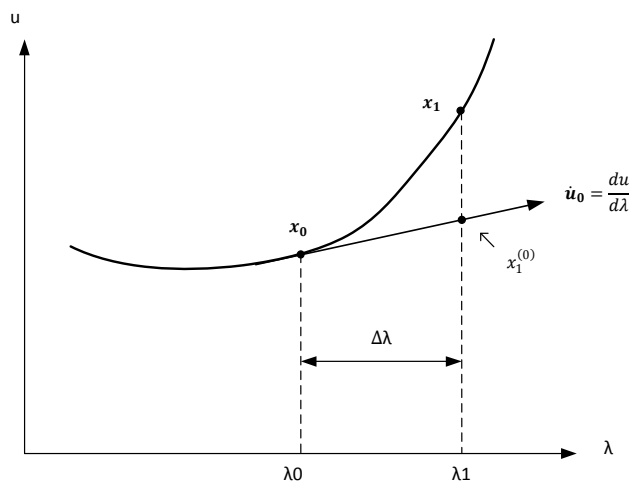
AUTOk jarraipena egiteko, Pseudo-Arku-Luzera teknika erabiltzen du. [3, 20.orr]

4.2. atalean soluzioen jarraipen algoritmoa azaldu genuen algoritmoari *parametro naturala jarraipena esaten zaio*. Hasierako hurbilpena ($x_1^{(0)}$) hobetzen duen beste metodoa honakoa dugu, *tangente jarraipena*. Bi metodoen adierazpen grafikoak argi uzten du bien arteko diferentzia,

- Parametro naturala jarraipena : $x_1^{(0)} = (\mathbf{u}_0, \lambda_0 + \Delta\lambda)$
- Tangente jarraipena: $x_1^{(0)} = (\mathbf{u}_0 + \Delta\lambda\dot{\mathbf{u}}_0, \lambda_0 + \Delta\lambda), \quad \dot{\mathbf{u}}_0 = \frac{d\mathbf{u}}{d\lambda}$



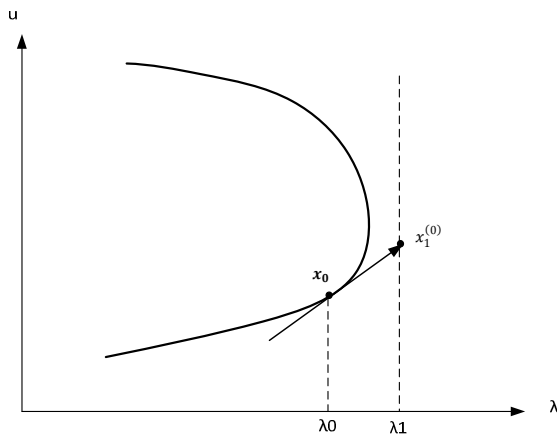
Irudia 7.1 x_1^* Jarraipen naturala.
 $x_{1\#}$ Tangente jarraipena.



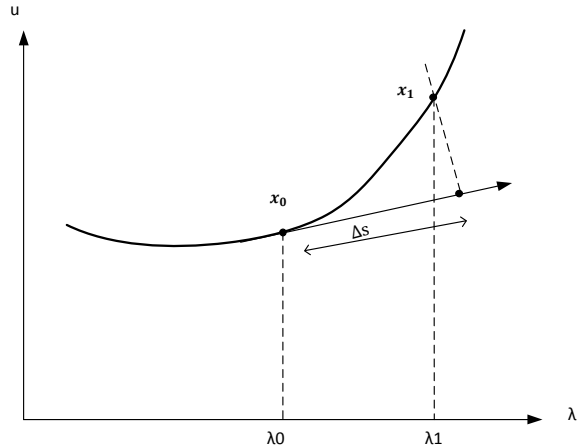
Irudia 7.2 Tangente jarraipenaren zehaztasunak.

Jarraipenak ez du ondo funtzionatzeko soluzio adarrean tolestura (fold) azaltzen denean (irudia 7.3). Tolestura uneak gainditu ahal izateko *Pseudo-Arku-Luzerako jarraipen* teknika proposatzen da. Metodo honetan, jarraipena arku luzeran egiten da (Δs), λ parametroaren ordean. Ekuazio berri bat gehitzen dugu, plano ekuazio hau, $(\dot{\mathbf{u}}_0, \dot{\lambda}_0)$, $(\mathbf{u}_0, \lambda_0)$ puntutik Δs distantziara dagoen tangentearen perpendikularra da.

$$(\mathbf{u}_1 - \mathbf{u}_0) \cdot \dot{\mathbf{u}}_0 + (\lambda_1 - \lambda_0) \dot{\lambda}_0 - \Delta s = 0$$



Irudia 7.3 Tolestura arazoa.



Irudia 7.4 Pseudo-Arku-Luzera.

$\mathbf{x}_0 = (\mathbf{u}_0, \lambda_0)$ $\mathbf{F}(\mathbf{x}) = 0$ sistemaren soluzioa eta soluzioa adarraren $(\dot{\mathbf{u}}_0, \dot{\lambda}_0)$ norabide bektorea ezagunak izanik, Pseudo-Arku-Luzerako jarraipenean, $\mathbf{x}_1 = (\mathbf{u}_1, \lambda_1)$ soluzio kalkulatzeko honako ekuazio sistema dugu,

$$\mathbf{F}(\mathbf{u}_1, \lambda_1) = 0$$

$$(\mathbf{u}_1 - \mathbf{u}_0)^* \dot{\mathbf{u}}_0 + (\lambda_1 - \lambda_0) \dot{\lambda}_0 - \Delta s = 0$$

eta Newton-en metodoa honela laburtu daiteke,

$$\mathbf{F}_{\mathbf{u}}(\mathbf{u}_1^{(k)}, \lambda_1^{(k)}) \Delta \mathbf{u}_1^{(k)} + \mathbf{F}_{\lambda}(\mathbf{u}_1^{(k)}, \lambda_1^{(k)}) \Delta \lambda_1^{(k)} = -\mathbf{F}(\mathbf{u}_1^{(k)}, \lambda_1^{(k)})$$

$$\dot{\mathbf{u}}_0^* \Delta \mathbf{u}_1^{(k)} + \dot{\lambda}_0 \Delta \lambda_1^{(k)} = (\mathbf{u}_1 - \mathbf{u}_0)^* \dot{\mathbf{u}}_0 + (\lambda_1 - \lambda_0) \dot{\lambda}_0 - \Delta s$$

$$\mathbf{u}_1^{(k+1)} = \mathbf{u}_1^{(k)} + \Delta \mathbf{u}_1^{(k)}$$

$$\lambda_1^{(k+1)} = \lambda_1^{(k)} + \Delta \lambda_1^{(k)}, \quad k = 0, 1, 2, \dots$$

Eta matrize notazio erabiliz,

$$\begin{pmatrix} \mathbf{F}_{\mathbf{u}}(\mathbf{u}_1^{(k)}, \lambda_1^{(k)}) & \mathbf{F}_{\lambda}(\mathbf{u}_1^{(k)}, \lambda_1^{(k)}) \\ \dot{\mathbf{u}}_0^* & \dot{\lambda}_0 \end{pmatrix} \begin{pmatrix} \Delta \mathbf{u}_1^{(k)} \\ \Delta \lambda_1^{(k)} \end{pmatrix} = \begin{pmatrix} -\mathbf{F}(\mathbf{u}_1^{(k)}, \lambda_1^{(k)}) \\ (\mathbf{u}_1 - \mathbf{u}_0)^* \dot{\mathbf{u}}_0 + (\lambda_1 - \lambda_0) \dot{\lambda}_0 - \Delta s \end{pmatrix}$$

Pseudo- Arku- Luzerako jarraipenaren Newton metodorako sistema honelakoa da,

$$\begin{pmatrix} A & c \\ b^* & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ h \end{pmatrix}$$

AUTOren inplementazioak modu eraginkorrean ebazten du ekuazio sistema. A matrizea, matrize bakana eta bere egitura berezian oinarritzen da inplementazio eraginkorra egiteko.

7.3-Kolokazio metodoa.

AUTOk EDA mugaldeko problemak ebazteko, zatikako polinomioen kolokazio metodoa erabiltzen du. Doitasun handiko metodoa dugu eta tarte aldakorreko diskretizazioa erabiltzen du.

Mugaldeko problema.

Suposatu honako EDA sistema dugula non λ jarraipen parametroa dugun

$$\mathbf{u}'(t) = \mathbf{f}(\mathbf{u}(t), \lambda), \quad \mathbf{u}(\cdot), \mathbf{f}(\cdot) \in \mathbb{R}^n, \quad \lambda \in \mathbb{R}$$

Mugaldeko baldintza,

$$\mathbf{b}(\mathbf{u}(0), \mathbf{u}(1), \lambda) = 0, \quad \mathbf{b} \in \mathbb{R}^n$$

Zatikako polinomioak.

Lehenik soluzioaren diskretizazioa finkatuko dugu,

$$0 = t_0 < t_1 < \dots < t_N = 1, \quad h_j = t_j - t_{j-1}, \quad j = 1, \dots, N$$

\mathbf{P}_h^m zatikako polinomio multzoa definituz, $\mathbf{P}_h^m = \{p_h \in C\{0,1\} \mid p_{h_{\{t_j, t_{j-1}\}}} \in P^m\}$ eta non P^m *gradua* $\leq m$ duten polinomio multzoa izanik-

Zatikako polinomioen kolokazio metodoak hurrengo kolokazio ekuazioak betetzen dituzten $p_h \in P_h^m$ bilatzen ditu

- $p_h'(z_{j,i}) = \mathbf{f}(p_h(z_{j,i}), \lambda), \quad j = 1, \dots, N, \quad i = 1, \dots, m$
- p_h mugaldeko baldintza betetzen duen.

$\{t_j, t_{j-1}\}$ tarte bakoitzari dagozkion $z_{j,i}$ kolokazio puntuak, Gaussen puntuak ditugu. m kolokazio puntu kopurua da.

Lagrange polinomio interpolatzaileak.

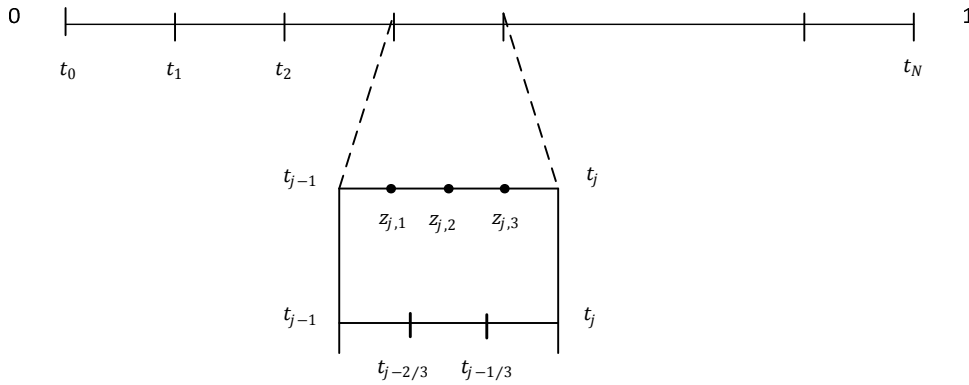
$\{t_j, t_{j-1}\}$ tarte bakoitzari polinomio bat dagokio eta Lagrangeren formulazioa erabiltzen da,

$$\{l_{j,i}(t)\}, \quad j = 1, \dots, N, \quad i = 1, \dots, m$$

$$l_{j,i}(t) = \prod_{k=0, k \neq i}^m \frac{t - t_{j-\frac{k}{m}}}{t_{j-\frac{i}{m}} - t_{j-\frac{k}{m}}}, \quad t_{j-\frac{i}{m}} = t_j - \frac{i}{m} h_j$$

$$p_j(t) = \sum_{i=0}^m l_{j,i}(t) \mathbf{u}_{j-\frac{i}{m}}$$

Beraz soluzio, $\mathbf{u}(t_j) = \mathbf{u}_j$ eta $\mathbf{u}\left(t_{j-\frac{i}{m}}\right) = \mathbf{u}_{j-\frac{i}{m}}$



Irudia 7.5 $m = 3$ kolokazio kopuruari dagokion eskema.

Kolokazio ekuazioak hauek dira,

$$\mathbf{p}'_j(z_{j,i}) = \mathbf{f}(\mathbf{p}_j(z_{j,i}), \lambda), \quad j = 1, \dots, N, \quad i = 1, \dots, m$$

$$b_i(\mathbf{u}_0, \mathbf{u}_N, \lambda) = 0, \quad i = 1, \dots, n$$

$$\int_0^1 (\mathbf{u}(t) - \mathbf{u}_0(t))^* \dot{\mathbf{u}}_0(t) dt + (\lambda_1 - \lambda_0) \lambda_0 - \Delta s = 0 \Rightarrow$$

$$\Rightarrow \sum_{j=1}^N \sum_{i=0}^m w_{j,i} \left[\mathbf{u}_{j-\frac{i}{m}} - (\mathbf{u}_0)_{j-\frac{i}{m}} \right]^* (\dot{\mathbf{u}}_0)_{j-\frac{i}{m}} + (\lambda_1 - \lambda_0) \lambda_0 - \Delta s = 0$$

Ekuazio kopurua: $mnN + n + 1$

Ezezagun kopurua: $\left\{ \mathbf{u}_{j-\frac{i}{m}} \right\} \in \mathbb{R}^{mnN+n}, \lambda \in \mathbb{R}$

Jarraipen aldagaia Δs

7.4 - Soluzio periodikoen jarraipena.

Problemaren mugaldeko formulazioa eginez ebatziko du,

$$\mathbf{u}'(t) = \mathbf{f}(\mathbf{u}(t), \lambda), \quad \mathbf{u}(\cdot), \mathbf{f}(\cdot) \in \mathbb{R}^n, \quad \lambda \in \mathbb{R}$$

Honako aldaketa eginez, $t \mapsto \frac{t}{T}$, periodoa finkatuko dugu.

$$\mathbf{u}'(t) = T \mathbf{f}(\mathbf{u}(t), \lambda), \quad \mathbf{u}(\cdot), \mathbf{f}(\cdot) \in \mathbb{R}^n, \quad T, \lambda \in \mathbb{R}$$

1 periodun soluzioa bilatuko dugu,

$$\mathbf{u}(0) = \mathbf{u}(1)$$

Azpirarratu soluzio periodikoen kasuan periodoa, T ezezagun bat gehitu dugula. Bestalde, 4.3. atalean azaldu genuen bezala soluzio periodikoen jarraipenean ekuazio bat gehitu behar dugu, aingura baldintza. AUTOk aingura baldintza finagoa erabiltzen du[E1, 31.orr].:

$$\int_0^1 \mathbf{u}_k(t)^* \mathbf{u}'_{k-1}(t) dt = 0$$

AUTOOn soluzio periodikoen jarraipenean, $\mathbf{u}_{k-1}, T_{k-1}, \lambda_{k-1}$ emanik honako ekuazio sistema ebatziko dugu,

$$\mathbf{u}'_k(t) = T_k \mathbf{f}(\mathbf{u}_k(t), \lambda_k)$$

$$\mathbf{u}_k(0) = \mathbf{u}_k(1)$$

$$\int_0^1 \mathbf{u}_k(t)^* \mathbf{u}'_{k-1}(t) dt = 0$$

$$\int_0^1 (\mathbf{u}_k(t) - \mathbf{u}_{k-1}(t))^* \dot{\mathbf{u}}_{k-1}(t) dt + (T_k - T_{k-1})\dot{T}_k + (\lambda_k - \lambda_{k-1}) \dot{\lambda}_{k-1} - \Delta s = 0$$

non $\mathbf{u}(\cdot), \mathbf{f}(\cdot) \in \mathbb{R}^n \quad T, \lambda \in \mathbb{R}$

Jakobiarraren egitura, honako adibiderako $n = 1, N = 3, m = 3$

$$\begin{pmatrix} u_0 & u_{1/3} & u_{2/3} & u_1 & & u_2 & & u_N & T & \lambda \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & & & & & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & & & & & \blacksquare & \blacksquare \\ \blacksquare & & & \blacksquare & \blacksquare & \blacksquare & \blacksquare & & \blacksquare & \blacksquare \\ & & & \blacksquare & \blacksquare & \blacksquare & \blacksquare & & \blacksquare & \blacksquare \\ & & & \blacksquare & \blacksquare & \blacksquare & \blacksquare & & \blacksquare & \blacksquare \\ & & & & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & & & & & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ & & & & & & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & & & & & & & & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{pmatrix}$$

Irudia 7.6 Jakobiarraren egitura orokorra. Azken hiru lerroak: mugaldeko baldintza, aingura baldintza eta Pseudo-Arku-Luzera ekuazioa.

7.5- AUTOren erabilera.

AUTO-07p bertsioarekin lan egin dugu eta hau erabiltzeko laburpen txiki bat prestatu dugu.

Problema zehaztu.

AUTO programan lau fitxategi zehaztu behar ditugu:

1- Problema.f90.

Problema definitzen duten ekuazioak eta baldintzak zehaztuko ditugu. Zehazki honako funtzioak adieraziko ditugu:

- func. EDAen ekuazioak.
- bcnd. Mugaldeko baldintzak.
- icnd. Integral baldintzak.
- stpnt. Parametro eta hasierako soluzioaren hasieratzea.
- pvls. Jarraipenean zehar aldagaien eta parametroen balioak erabiltzeko.
- fopt. Optimizazio problemetan erabilgarria.

2- c.Problema.

Problema nola ebazteko zehazten zaizkion hainbat parametro.

3- script.auto.

Gure problema ebazteko idatzitako Python scripta.

4- Problema.dat (hasierako soluzioa).

AUTOri hasierako soluzioa adierazteko aukera ezberdinak daude. Hasierako soluzioa ASCII fitxategi batean zehaztu daiteke.

Exekuzioa.

```
$ auto scripta.auto
$ auto
$ auto>plot('problema')
```

Emaitzak.

AUTO exekutatzeraoan jarraipenari buruzko informazio orokorra pantailaratzen du:

- EP. Hasierako soluzioari dagokiona.
- MX. AUTOk ez du konbergitu.
- UZ. Zehaztutako parametro bukaera lortu da.
- LP. Tolestura.
- BP. Adarkatze puntua.

Emaitzak honako fitxategietan jasotzen dira:

- b.problema (edo fort.7). . Jarraipen urrats bakoitzari dagokion informazioa.
- s.problema (edo fort.8). . Jarraipenaren soluzioak jasotzen dituen fitxategia.
- d.problema (edo fort.9). Jarraipenaren urrats bakoitzean egindako Newton-en iterazioei buruzko informazio osagarria. Adibidez IID=3 zehaztuz gero, Newtonen iterazioetan kalkulaturako hondarren balioak jasotzen dira.

“WriteRawFilename” aginduak soluzio zehatz bat ASCII fitxategi batera esportatzeko aukera ematen du. Egokiena fitxategi hau Mathematica edo antzeko software batean inportatu, gero soluzioaren azterketa egiteko.

7.6- Esperimentuak.

7.6.1- Errorearen neurtzea.

AUTOrekin lortutako soluzioaren errorea neurtzeko, honako azterketak egin ditugu:

- EDA ordezkapena.
- Energia integrala.
- Periodikotasuna.
- Denbora.

EDA ordezkapena.

AUTOk soluzioa m graduko zatikako polinomioen bidez adierazten du soluzioa. Polinomio interpolatzaileak kalkulatu ditugu eta hauek ekuazio diferentzialak betetzen dituzten aztertuko dugu.

$$0 = t_0 < t_1 < \dots < t_N = 1, \quad h_j = t_j - t_{j-1}, \quad j = 1, \dots, N$$

$$\mathbf{p}_h \in P_h^m \Rightarrow \mathbf{p}'_h(t) - \mathbf{f}(\mathbf{p}_h(t)) = 0$$

$$p_{h_{\{t_j, t_{j-1}\}}} \in P^m \quad j = 1, \dots, N \quad \text{Tarte bakoitzari dagokio polinomioa.}$$

Beraz, zatikako polinomioak EDA betetzen duten aztertzeko honako balioa kontsideratu dugu,

$$\text{Errore1} = \sum_{j=1}^N \left(\int_{t_{j-1}}^{t_j} \left(p'_{h_{\{t_j, t_{j-1}\}}}{}^2 - \mathbf{f}(p_{h_{\{t_j, t_{j-1}\}}})^2 \right) dt \right),$$

Energia integrala.

R3BP problemaren energia,

$$H(x(t), y(t), p_x(t), p_y(t)) = \frac{1}{2} (p_x^2 + p_y^2) + p_x y - p_y x - \left(\frac{1-\mu}{D_1^{1/3}} \right) - \frac{\mu}{D_2^{1/3}} - \frac{1}{2} \mu (1-\mu)$$

Ezaguna da energia R3BP problemaren integrala edo inbariantea dela, hau da, orbitaren zehar konstantea mantentzen den balioa,

$$H_0 = H(\mathbf{u}_0), \quad H(\mathbf{u}(t)) - H_0 = 0, \quad t_0 \leq t \leq t_f$$

Beraz, energiaren bidezko errorea,

$$\text{Errore2} = \text{Max}(H(\mathbf{u}_j) - H_0), \quad \mathbf{u}_j = \mathbf{u}(t_j), \quad \mathbf{u}\left(t_{j-\frac{i}{m}}\right) = \mathbf{u}_{j-\frac{i}{m}} \quad j = 1, \dots, N, \quad i = 1, \dots, m$$

Periodikotasuna.

Gure soluzioa $\mathbf{u}(t)$ periodikoa denez, T periodoa izanik: $\text{Norma}(\mathbf{u}(T) - \mathbf{u}(0)) = 0$

Lortutako soluzioaren lehen balioa hartuz (\mathbf{u}_0), Mathematica softwarean zehaztasun handiarekin integratuko dugu eta kokapen errorea kalkulatu dugu.

```
sol = NDSolve[eqnsedo[x0,y0,vx0,vy0,0],{x,y,px,py,t},{t,0,T},Method->{"ImplicitRungeKutta",DifferenceOrder->20},MaxSteps->10^6,WorkingPrecision->prec,AccuracyGoal->acc][[1]];
```

Kokapen errorea kontsideratu dugu,

$$\boxed{\text{Errore3} = \text{Norma}(\{x(T), y(T)\} - \{x(0), y(0)\})}$$

7.6.2- AUTOOn R3BP problema.

Erregularizatutako bertsioan erabilitako ekuazioak azalduko ditugu. Bi ohar eman behar ditugu, AUTOOn erabilitako formulazioa uler dadin:

1- Perturbazio funtzioa.

4.3.1. atalean esan genuenez, sistema dinamikoak inbariante bat duenean, α aldagai era honetan gehitu behar genuen,

$$\mathbf{u}'(t) = \frac{T}{2\pi} [f(\mathbf{u}(t)) + \alpha \nabla g(\mathbf{u})], \quad \alpha \in \mathbb{R}$$

AUTOren dokumentazioan [1] eta [8] artikuluan aipatzen denez, R3BP problema konkreturako egokiena $g(\mathbf{u}) = (0,0,0, v_x, v_y, v_z)$ erabiltzea dugu.

2- IPS=2.

Parametro honen bidez, soluzio periodikoak kalkulatzeko adierazten dugu. Horregatik ez ditugu, ez mugaldeko baldintzak ez aingura baldintza zehaztu behar; AUTOk definitutako aingura baldintza erabiliko du. Exekutatzeko beste aukera bat dugu IPS=4, mugaldeko problemetzat; orduan mugaldeko baldintzak eta aingura baldintza zehaztu behar dugu.

Jarraian R3BP probleman erabilitako ekuaziok deskribatuko ditugu:

1- r3b.f90

```
SUBROUTINE FUNC(NDIM,U,ICP,PAR,IJAC,F,DFDU,DFDP)
```

```
x = U(1); y = U(2); z = U(3);
xp = U(4); yp = U(5); zp = U(6);
t = U(7)
```

```
rl = PAR(1); rmu = PAR(2)
```

```
dE = SQRT((x+rmu)**2 + y**2 + z**2)
dM = SQRT((x-1+rmu)**2 + y**2 + z**2)
rnc = 1 - rmu
dE3 = 1./dE**3; dM3 = 1./dM**3
lag = SQRT(rnc*dE3+rmu*dM3)
lag2 = 1./lag
```

```

F(1)= lag2*(xp+y)
F(2)= lag2*(yp-x)
F(3)= lag2*(zp)
F(4)= lag2*(yp-(x+rmu)*(rmc*dE3)-(x-rmc)*(rmu*dM3))+r1*(xp+y)
F(5)= lag2*(-xp-y*(rmc*dE3+rmu*dM3))+r1*(yp-x)
F(6)= lag2*(-z*(rmc*dE3+rmu*dM3))+r1*zp
F(7)= lag2

```

```
END SUBROUTINE FUNC
```

```
!-----
SUBROUTINE STPNT(NDIM,U,PAR,T)
  PAR(1) = 0.    PAR(2) = 0.012277471d0;

```

```
END SUBROUTINE STPNT
```

```
!-----
SUBROUTINE PVLS(NDIM,U,PAR)

```

```

  rmu = PAR(2)
  x = U(1)  y = U(2)  z = U(3)
  xp = U(4) yp = U(5) zp = U(6)

  d1 = SQRT((x+rmu)**2 + y**2 + z**2)
  d2 = SQRT( (x-1+rmu)**2 + y**2 + z**2 )

  E = -(rmu/d2)-(1-rmu)/d1+y*xp-x*yp+(xp**2 + yp**2 + zp**2)/2
  PAR(3) = E

```

```
END SUBROUTINE PVLS
```

```
!-----
```

```

SUBROUTINE BCND  END SUBROUTINE BCND
SUBROUTINE ICND  END SUBROUTINE ICND
SUBROUTINE FOPT  END SUBROUTINE FOPT

```

2- c.r3b

```

dat = 'r3b'
NDIM= 6, IPS = 2, IRS = 0, ILP = 0
ICP = [1,11,3]
NTST= 200, NCOL= 4, IAD = 3, ISP = 1, ISW =1, IPLT= 0, NBC= 0, NINT= 0
NMX= 1000, NPR= 1, MXBF= 10, IID = 2, ITMX= 10, ITNW= 5, NWTN= 3, JAC= 0
EPSL= 1e-08, EPSU = 1e-08, EPSS = 1e-06
DS = 1e-3, DSMIN= 1e-08, DSMAX= 1e-1, IADS=1
NPAR = 11, THL = {}
UZSTOP = {11: 31.}

```

3- r3b.auto

```

r1=run('r3b',c='r3b')
save(r1,'r3b')
r1(70).writeRawFilename('out1')           # 70 soluzioa esportatu out1 ASCII fitxategi batera.

print "\n***Clean the directory***"
clean()

```

Exekuzioa.

```
>auto r3b.auto
```

```
>auto
```

```
auto>plot('r3b')
```

7.6.3-Esperimentua I.

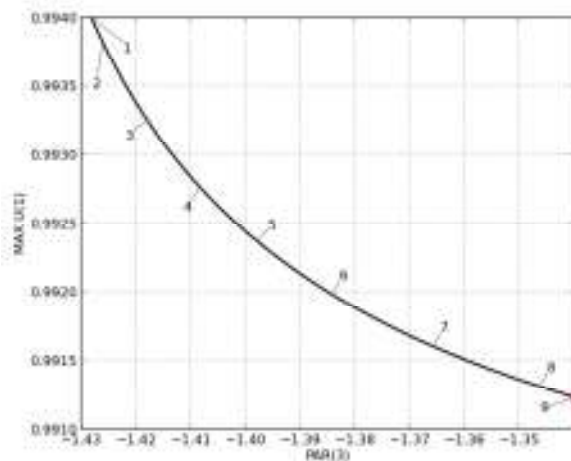
Esperimentu honen helburua erregularizazio eta erregularizazio gabeko kasuak konparatzea izan da. Jarraipen parametroa energia aukeratu da.

- Energia maila hasieran: -1.32821
- Energia maila bukaeran: -1.34

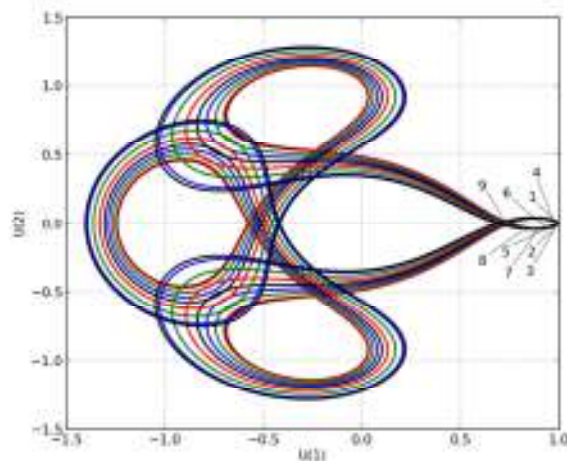
Auto programa diskretizazio ezberdinak erabiliz exekutatu da: NTST=50,100,150,200,250,300,350,400,500

Zehazki hauek izan dira AUTOn erabilitako parametroen balioak:

```
dat = 'r3b'
NDIM= 6, IPS = 2, IRS = 0, ILP = 0
ICP = [1,11,3]
NCOL= 4, IAD = 3, ISP = 1, ISW =1, IPLT= 0, NBC= 0, NINT= 0
NMX= 1000, NPR= 1, MXBF= 10, IID = 3, ITMX= 10, ITNW= 5, NWTN= 3, JAC= 0
EPSL= 1e-08, EPSU = 1e-08, EPSS = 1e-06
DS = 1e-3, DSMIN= 1e-08, DSMAX= 1e-1, IADS=1
NPAR = 11, THL = {},UZSTOP = {3: -1.34}
```



Irudia 7.7 AUTOn soluzioa adarraren grafikoa
PAR(3)=Energia



Irudia 7.8 AUTOn soluzioen grafikoa.

Irudiak. Honako araua erabili da grafikoak hobeto uler daitezen.

Erregularizazioa: Kolore urdinez eta "●" marka.

Erregular. gabea: Kolore moreez eta "□" marka.

Hurrengo tauletan, esperimentu honen exekuzio guztiak laburtzen ditugu.

NTST	Errore1	Errore2	Errore3	Denbora
50	1×10^{-4}	5×10^{-4}	1×10^{-5}	0.16s
100	3×10^{-7}	1×10^{-5}	2×10^{-7}	0.32s
150	1×10^{-8}	2×10^{-6}	6×10^{-8}	0.47s
200	1×10^{-9}	4×10^{-7}	2×10^{-8}	0.76s
250	2×10^{-10}	1×10^{-7}	2×10^{-9}	0.93s
300	4×10^{-11}	6×10^{-8}	5×10^{-8}	1.08s
350	1×10^{-11}	3×10^{-8}	1×10^{-8}	1.17s
400	4×10^{-12}	2×10^{-8}	2×10^{-8}	1.46s
500	7×10^{-13}	8×10^{-9}	4×10^{-9}	1.67s

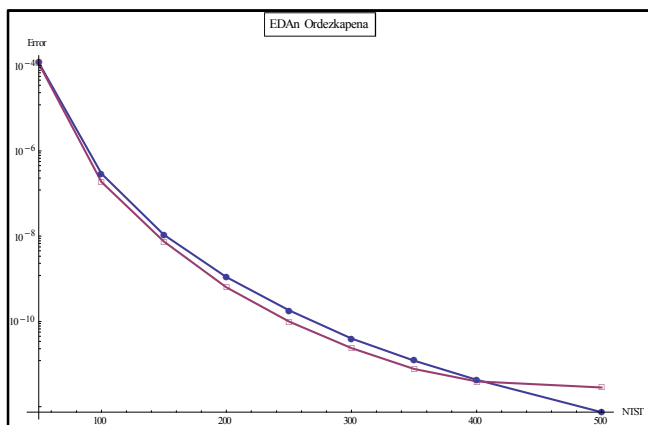
Taula 7.1. Erregularizazio bertsioaren datuen laburpena.

NTST	Errore1	Errore2	Errore3	Denbora
50	1×10^{-4}	8×10^{-3}	2×10^{-2}	0.28s
100	2×10^{-7}	5×10^{-5}	1×10^{-5}	0.68s
150	7×10^{-9}	8×10^{-6}	2×10^{-7}	0.86s
200	6×10^{-10}	2×10^{-6}	1×10^{-7}	1.03s
250	1×10^{-10}	5×10^{-7}	1×10^{-7}	1.16s
300	2×10^{-11}	2×10^{-7}	5×10^{-8}	1.47s
350	8×10^{-12}	9×10^{-8}	8×10^{-8}	1.73s
400	4×10^{-12}	6×10^{-8}	7×10^{-8}	1.95s
500	3×10^{-12}	2×10^{-8}	8×10^{-8}	2.28s

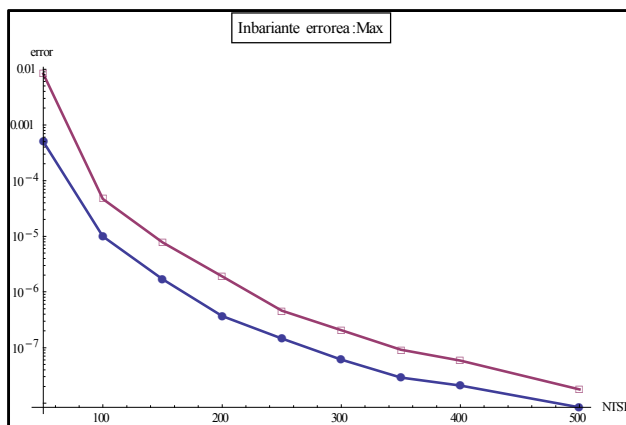
Taula 7.2. Erregularizazio gabeko bertsioaren datuen laburpena.

Erroreak.

Erregularizazio bertsioan eta erregularizatu gabeko bertsioan erroreak oso antzekoak izan dira. Gainera errorea tarte kopurua handitu heinean, errorea esponentzialki txikitzen da.



Irudia 7.9 Errore1.



Irudia 7.10 Errore2.

Denborak.

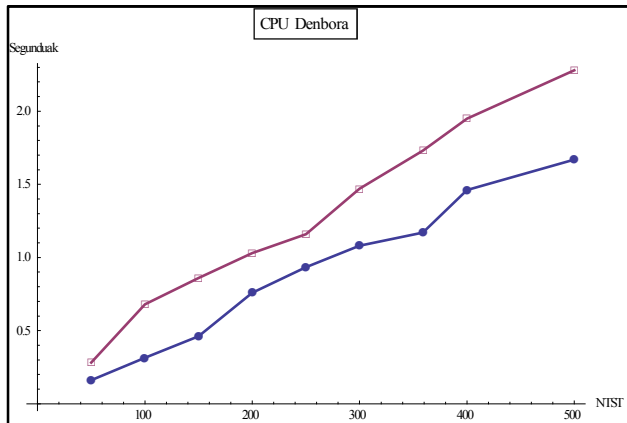
AUTOk exekuzioan pantailaratzen duen balioa kontsideratu da.

Erregularizazio bertsioan hasierako soluzioa adierazteko 400 balioko diskretizazioa erabili dugu; erregularizazio gabeko bertsioan 22.000 balioko diskretizazioa. AUTOk lehen urratsean ASCII fitxategia hauek inportatu behar ditu. Fitxategia handi hauek inportatzeko denbora kalkulatu dira eta exekuzio denboretatik deskontatu ditugu.

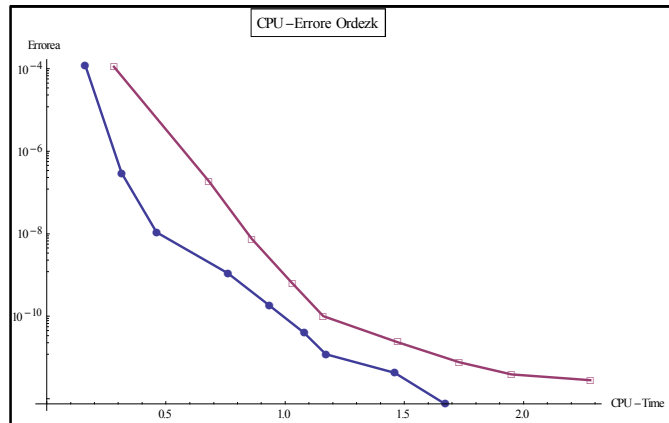
Inportazio denborak (NTST=200):

- R3BP erregularizazio: denbora=3,0E-02
- R3BP erregular. gabe: denbora=3,1E-01.

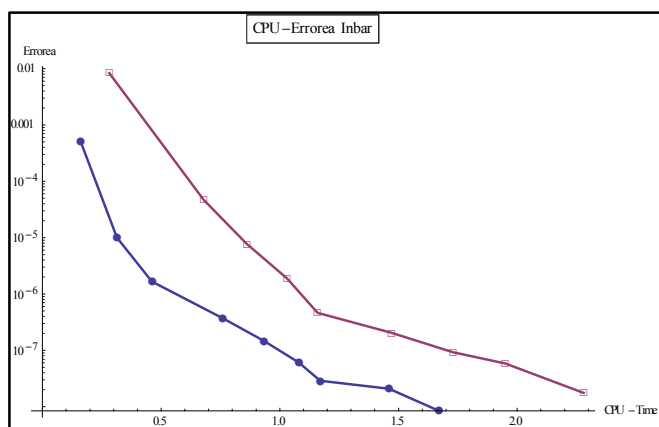
Hurrengo irudiek erregularizazioa bertsioak exekuzio denbora gutxiagoa behar izan duela erakusten du: exekuzio denbora berdinean errore txikiagoa lortu da.



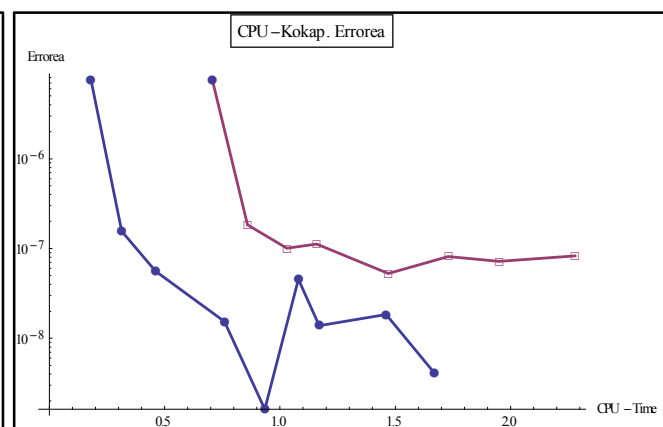
Irdia 7.11 Denborak



Irdia 7.12 Denbora-Errore1



Irdia 7.13 Denbora-Errore2



Irdia 7.14 Denbora-Errore3.

Kodea.

Azterketaren Mathematica-ko fitxategiak eskuragarri daude
/kodeak/7-AUTO/1-ESPERIMENTUA/

erregularizazioa.

- balioakdat1.nb: AUTOren r3b.dat fitxategia sortzeko scripta.
- azterketa1.nb: erregularizazio bertsioa

erregularizazioGabe

- balioakdat2.nb: AUTOren r3b.dat fitxategia sortzeko scripta
- azterketa2.nb: erregularizazio gabekoa.

laburpena.nb: bi soluzioen konparaketen laburpena.

7.6.4- Esperimentua II.

Esperimentu honen helburua gure inplementazio (Urabe) eta AUTO Jarraipen algoritmoak konparatzea izan da. Horretarako bi programetan kasu berdina probatu da: erregularizatutako bertsioa eta periodoaren arabera jarraipena.

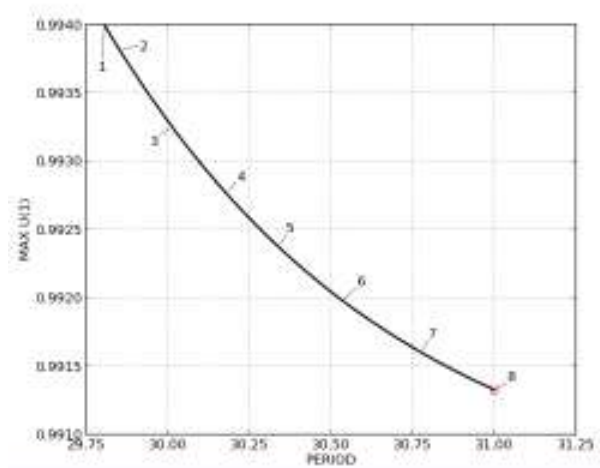
- Periodoa hasieran: **29.80472130126875080466217100557**
- Periodoa bukaeran: **31.**

Auto programa diskretizazio ezberdinak erabiliz exekutatu da: NTST=50,100,150,200,250,300,350,400,500. Urabe programan, 6.4.2.2. ataleko esperimentuen emaitzak berreskuratu ditugu: M=100,150,200,250,300,350,400.

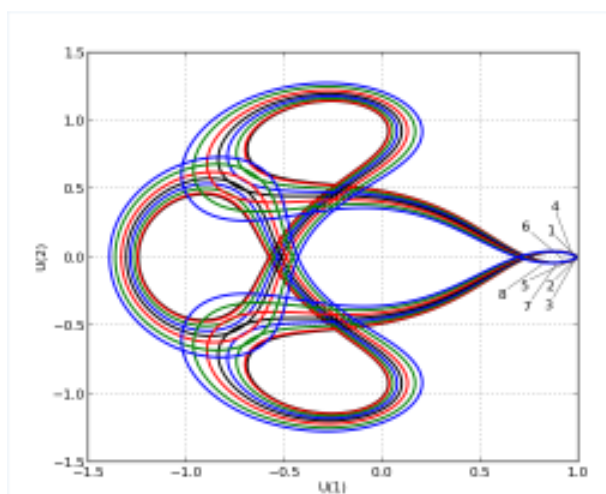
Irudiak. Honako araua erabili da grafikoak hobeto uler daitezzen.

AUTO: Kolore urdinez eta "●" marka.

Urabe: Kolore moreez eta "□" marka.



Irudia 7.15 AUTOren soluzio adarraren grafikoa



Irudia 7.16 AUTOren soluzioen grafikoa.

Exekuzioen laburpenak

NTST	Diskret.	Errore1	Errore2	Errore3	Denbora
50	201	8×10^{-5}	5×10^{-4}	2×10^{-5}	0.37s
100	401	3×10^{-7}	9×10^{-6}	1×10^{-7}	0.76s
150	601	1×10^{-8}	1×10^{-6}	2×10^{-8}	1.15s
200	801	1×10^{-9}	4×10^{-7}	3×10^{-8}	1.35s
250	1001	2×10^{-10}	1×10^{-7}	5×10^{-8}	1.60s
300	1201	4×10^{-11}	5×10^{-8}	1×10^{-8}	1.86s
350	1401	1×10^{-11}	3×10^{-8}	6×10^{-8}	2.13s
400	1601	4×10^{-12}	2×10^{-8}	2×10^{-8}	2.42s
500	2001	7×10^{-13}	1×10^{-8}	4×10^{-8}	2.98s

Taula-1: AUTOren emaitzak

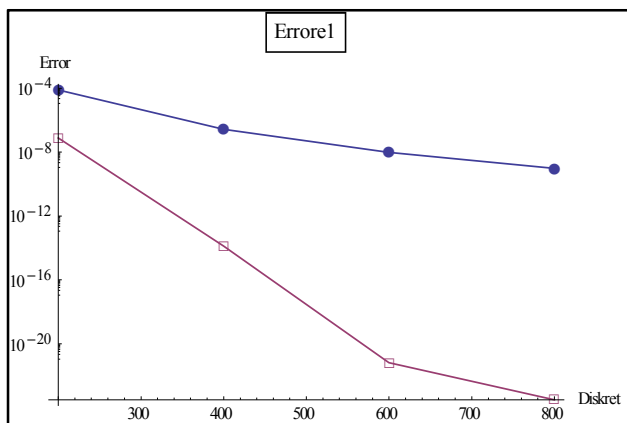
M	Diskret.	Errore1	Errore2	Errore3	Denbora
100	201	7×10^{-8}	6×10^{-4}	3×10^{-3}	12.67s
150	301	2×10^{-11}	8×10^{-6}	5×10^{-5}	39.54s
200	401	1×10^{-14}	2×10^{-7}	1×10^{-6}	113.92s
250	501	1×10^{-18}	2×10^{-9}	4×10^{-9}	169.60s
300	601	6×10^{-22}	1×10^{-11}	1×10^{-8}	283.28s
350	701	1×10^{-24}	2×10^{-12}	3×10^{-8}	447.48s
400	801	3×10^{-24}	1×10^{-12}	8×10^{-9}	879.12s

Taula-2: Uraberen emaitzak

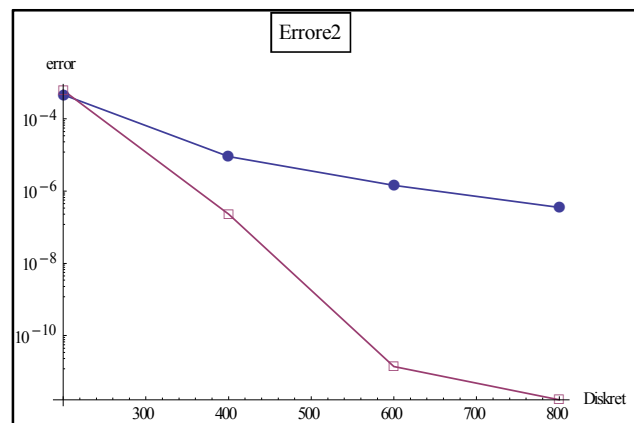
Erroreak.

Bi metodoak konparatu ahal izateko zutabe berri bat gehitu diogu taulari, esperimentuan erabilitako diskretizazio kopuruaren informazioarekin. AUTOn eta Uraberen diskretizazio berdina dituzten kasuak konparatuko ditugu (diskretizazioa=201,401,601,801).

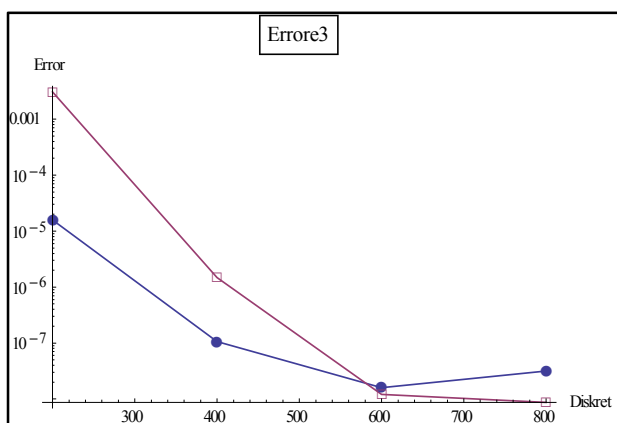
Konparazio hauetan Urabe programak errore txikiagoa lortzen du.



Irudia 7.17 Errore1: EDA ordezkapena.



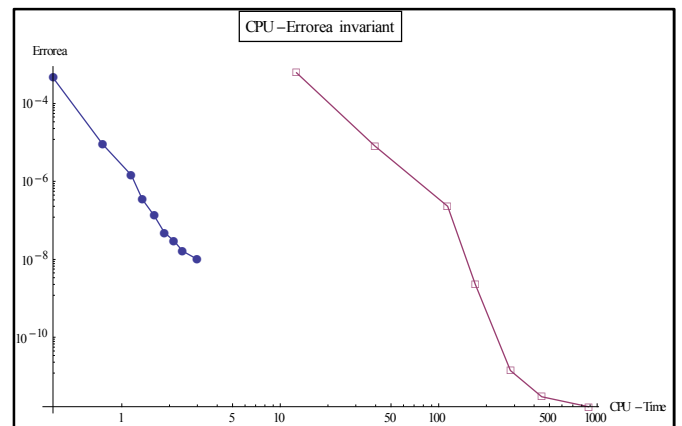
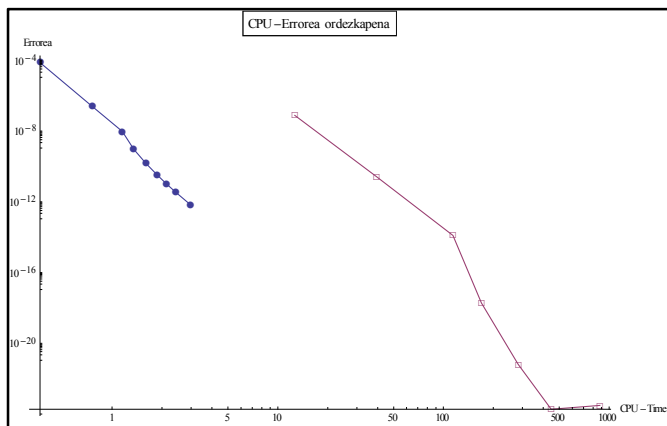
Irudia 7.18 Errore2: Inbariantea.



Irudia 7.19 Errore3: kokapen errorea.

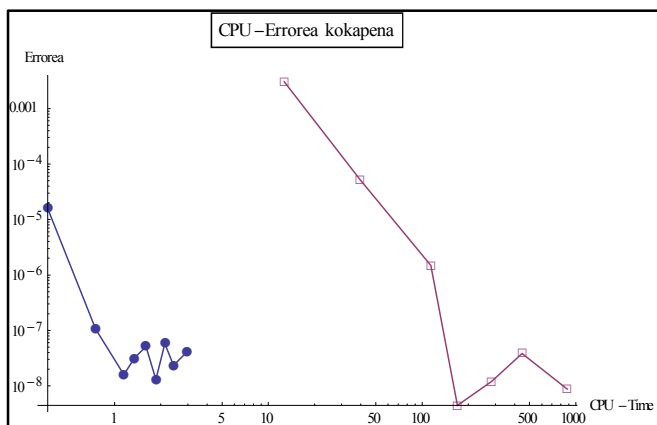
Denborak.

AUTO , Urabe metodoarekin konparatuz oso azkarra da. Diferentzia handia hau logikoa dirudi: Uraberen metodoan , oraindik eraginkortasuna hobetzeko metodoak aztertu gabe daude eta etorkizuneko lan bat da. AUTOren datu hauek erakusten dute, eraginkortasunaren gaian ondo diseinatutako metodotan oinarritzen dela.



Irudia 7.20 Errore1-Denbora

Irudia 7.21 Errore2-Denbora.



Irudia 7.22 Errore3-Denbora.

Kodea.

Azterketaren Mathematica-ko fitxategiak eskuragarri daude
/kodeak/7-AUTO/2-ESPERIMENTUA/

azterketa.nb:

laburpena.nb:

7.6.5 -AUTO eranskina.

AUTO erabiltzeko orduan hainbat berezitasun eta arazo topatu ditugu. Jarraian berezitasun nagusienak aipatuko ditugu.

1-r3bp.dat fitxategia.

AUTOri hasierako soluzio zehatza eman behar diogu (r3bp.dat) ; AUTO dokumentazioan ez da honi buruzko azalpen handirik ematen. Probak eginez honela finkatu ditugu:

- Erregularizazio bertsioan , 800 puntu.
- Erregularizazio gabe, 22000 puntu.

2-c.r3bp fitxategia: parametroak.

Erregularizazio probleman dimentsioa=7 dugu. AUTOk ez du konbergitzen eta probak NDIM=6 balioarekin egin dira.

Jarraipena egiteko honela finkatu ditugu problemaren parametroak:

PAR (1) : Disipazio terminoa.

PAR (11): Periodoa.

PAR (3) : Energia.

Periodoaren arabera jarraipena.

ICP = [1,11,3]

THL = {}

UZSTOP = {11: 31.}

Energiaren arabera jarraipena.

ICP = [1,11,3]

THL = {}

UZSTOP = {3: -1.34}

DS parametroan jarraipenaren lehen urratsa adierazten da eta zeinuak, jarraipena handituz (+) edo txikituz (-) egin nahi dugun adierazten du. Gure azken planteamenduan funtzionamendu egokia lortu dugu baina beste planteamendu batzuk egin ditugunean ez da horrela izan.

3-Soluzioaren doitasuna AUTOn.

AUTOk Soluzioa 10 digituzko doitasunarekin itzultzen ditu. Ez dakigu nola lortu daitekeen 16 digituzko doitasunezko soluzioa.

4-Zenbakizko integrazioa AUTOn.

[4] Artikuluan gure problema kolokazio metodoa integratuz aztertu da. Horretarako AUTO programa erabili da. AUTOn manuallean ez da argibiderik ematen honi buruz eta proba ezberdinak egin arren, ez dugu asmatu nola kalkulatu zenbakizko soluzioa hasieradun EDA problema batentzat AUTO erabiliz.

8- Ondorioak.

Erregularizazioa.

Lehenik, soluzioen jarraipenean R3BP problemarentzat proposatutako erregularizazioa aztertuko dugu.

URABE metodoan azpimarratzekoa da erregularizaziorik gabe ezin izango genukeela soluzioen jarraipenarik egin; erregularizazio gabeko ekuazio diferentzialak erabiltzen baditugu, soluzio hurbiltzeko behar dugu koefiziente kopurua oso handia da eta ondorioz, ezinezkoa da soluzioen jarraipena exekutatzea. Erregularizazioa erabiliz, ordea, koefiziente kopuru txikiagoarekin hurbilpen on bat dugu eta soluzioen jarraipena egin ahal izango dugu exekuzio denbora egokitan.

AUTOren erregularizazio bertsioarekin emaitza hobekak lortu ditugu; errore txikiagoa lortu dugu CPU denbora gutxiagorekin.

URABE programa.

Bigarrenik, URABE programa aztertuko dugu.

URABE soluzio periodikoen jarraipena egiteko algoritmoa interesgarria gertatu da. Horrela baieztatu dute egindako esperimendu ezberdinek. Behin algoritmoa egokia dela ikusi dugunean, exekuzioa azkartzeko aukerak aztertu behar genituzke.

AUTOrekin konparaketa egiterakoan URABEren exekuzio denborak altuagoak izan dira. Erregularizazioa erabili arren jakobiarra handia da, dentsua eta honen ondorioz konputazio denborak altuagoak izan ditugu. Eraginkortasuna hobetzeko hainbat aukera azpimarratu nahi genituzke:

1- Problemaren ezezagun kopurua.

A-Simetriak.

Soluzioa x -ardatzarekiko eta y -artzarekiko simetrikoa da,

$$x(t) = x(-t)$$

$$y(t) = -y(-t)$$

Hau honela izanik, ezezagun kopurua erdia izango da.

$$\mathbf{u}_j, j = 0, \dots, N/2$$

B-Problemaren dimentsioa.

Abiadura kokapenaren deribatua dela kontsideratuz, 2 dimentsioko sistema dinamikoa izango dugu

$$\mathbf{u}_j \in \mathbb{R}^2$$

A + B kontsideratuz, ezezagun kopurua /4 gutxitzea lortuko da.

2- Jakobiar sinplifikatua.

Ikusi dugunez URABE metodoaren Jakobiarra dentsua da. Jatorrizko problema jarraituari dagokion Jakobiar diskretua erabili daiteke; jakobiar hau blokeka diagonal izango da. Newton metodoan, Jakobiar diagonalentzako LU-deskonposaketa algoritmoa erabiliko genuke kalkulu denborak hobetuz.

3-Jarraipena M txikiagoarekin

Jarraipena M balioa txikiagoarekin egin daiteke. Azken urratsa, M handi baterako exekutatu.

4-FTT Algoritmoa.

Algoritmoaren konplexutasuna: $N^2 + N \text{ konp}(f)$

FTT Konplexutasuna : $N (\text{Log}_2 N) + N \text{ konp}(f)$

8- Ondorioak.

Erregularizazioa.

Lehenik, soluzioen jarraipenean R3BP problemarentzat proposatutako erregularizazioa aztertuko dugu.

URABE metodoan azpimarratzekoa da erregularizaziorik gabe ezin izango genukeela soluzioen jarraipenarik egin; erregularizazio gabeko ekuazio diferentzialak erabiltzen baditugu, soluzio hurbiltzeko behar dugu koefiziente kopurua oso handia da eta ondorioz, ezinezkoa da soluzioen jarraipena exekutatzeko. Erregularizazioa erabiliz, ordea, koefiziente kopuru txikiagoarekin hurbilpen on bat dugu eta soluzioen jarraipena egin ahal izango dugu exekuzio denbora egokitan.

AUTOn erregularizazio bertsioarekin emaitza hobekak lortu ditugu; errore txikiagoa lortu dugu CPU denbora gutxiagorekin.

URABE programa.

Bigarrenik, URABE programa aztertuko dugu.

URABE soluzio periodikoen jarraipena egiteko algoritmoa interesgarria gertatu da. Horrela baieztatu dute egindako esperimentu ezberdinek. Behin algoritmoa egokia dela ikusi dugunean, exekuzioa azkartzeko aukerak aztertu behar genituzke.

AUTOrekin konparaketa egiterakoan URABEren exekuzio denborak altuagoak izan dira. Erregularizazioa erabili arren jakobiarra handia da, dentsoa eta honen ondorioz konputazio denborak altuagoak izan ditugu. Eraginkortasuna hobetzeko hainbat aukera azpimarratu nahi genituzke:

1- Problemaren ezezagun kopurua.

A-Simetriak.

Soluzioa x -ardatzarekiko eta y -ardatzarekiko simetrikoa da,

$$x(t) = x(-t)$$

$$y(t) = -y(-t)$$

Hau honela izanik, ezezagun kopurua erdia izango da.

$$u_j, j = 0, \dots, N/2$$

B-Problemaren dimentsioa.

Abiadura kokapenaren deribatua dela kontsideratuz, 2 dimentsioko sistema dinamikoa izango dugu

$$u_j \in \mathbb{R}^2$$

A + B kontsideratuz, ezezagun kopurua /4 gutxitzea lortuko da.

2- Jakobiar sinplifikatua.

Ikusi dugunez URABE metodoaren Jakobiarra dentsoa da. Jatorrizko problema jarraituari dagokion Jakobiar diskretua erabili daiteke; jakobiar hau blokeka diagonal izango da. Newton metodoan, Jakobiar diagonalentzako LU-deskonposaketa algoritmoa erabiliko genuke kalkulu denborak hobetuz.

3-Jarraipena M txikiagoarekin

Jarraipena M balioa txikiagoarekin egin daiteke. Azken urratsa, M handi baterako exekutatu.

4-FTT Algoritmoa.

Algoritmoaren konplexutasuna: $N^2 + N \text{ konp}(f)$

FTT Konplexutasuna : $N (\text{Log}_2 N) + N \text{ konp}(f)$

Bibliografia.

[A1] AUTO. <http://cmvl.cs.concondia.ca/>

[D1] D.G. Bettis and V.Szebehely. Treatment of close approaches in the numerical integration of the gravitational problem of N-bodies.

[E1] Eusebius J.Doedel . Lectures Notes on Numerical Analysis of Nonlinear Equations.
Numerical Continuation Methods for Dynamical Systems. Chapter-1.

[H1] Hairer , E; Norsett, S.P; Wanner, G. Solving Ordinary Differential Equations I. Nonstiff Problems.
Springer, 1993, orrialdeak 129-131.

[H2] Kinoshita, H., Nakai, H. Numerical integration methods in dynamical astronomy. 1988.

[J1] Jorge Galán-Vioque and André Vanderbauwhede.Continuation of Periodic Orbits in Symmetric Hamiltonian Systems.
Numerical Continuation Methods for Dynamical Systems. Chapter-9.

[M1] Muñoz-Almaraz F.J, Freire E., Galán-Vioque, J., Vanderbauwhede, A.. Continuation of normal doubly symmetric orbits in conservative reversible systems. Celest.Mech. Dyn. Astron 97. 17-47 (2007).

Murua,A., Makazaga J. "Time-scale preprocessing in numerical integration of ordinary differential equations " (Numerische Mathematik)

[R1] René Van Dooren.Orbit Computation in Celestial Mechanics by Urabe's Method.

[T1] A software package for numerical integration of ODEs by means of high-order Taylor methods.

[V1] V.Szebehely. Theory of orbits. The Restricted Problem of three bodies.

A- Eranskina.

A.1-Ordenagailuaren ezaugarriak.

Intel Core i5-2520M CPU 2.50 GHz x 4.

Ubuntu 12.04 LTS (64 bit).

RAM: 3.8 GB.

Wolfram Mathematica 8.0.4.0 bertsioa.

A.2- R3BP ekuazioak.

[5, 129.orr] liburutik R3BP dagozkion ekuazioak hartu ditugu.

$$x'' = x + 2 y' - umu \frac{x + mu}{D_1} - mu \frac{x - (umu)}{D_2}$$

$$y'' = y - 2 x' - umu \frac{y}{D_1} - mu \frac{y}{D_2}$$

$$D_1 = ((x + mu)^2 + y^2)^{3/2}, \quad D_2 = ((x - umu)^2 + y^2)^{3/2}$$

$$mu = 0.012277471, umu = 1 - mu$$

$$E = \frac{1}{2}(\dot{x}^2 + \dot{y}^2) - U(x, y) - \frac{1}{2} \mu (1 - \mu), \quad U(x, y) = \frac{1}{2} (x^2 + y^2) + \frac{1 - \mu}{r_1} + \frac{\mu}{r_2}$$

Soluzio periodiko ezagunaren lehen balioa,

$$x(0) = 0.994, \quad x'(0) = 0$$

$$y(0) = 0, \quad y'(0) = -2.00158510637908252240537862224,$$

$$t_{end} = 17.0652165601579625588917206249$$

Formulazio Hamiltondarraren egoera aldagaiak kokapena eta momentua dira. Liburuko ekuazioetan abiaduraren arabera definituta ditugu ekuazioak eta horregatik, momentuaren arabera ekuazioak kalkulatu ditugu.

1-Lehen urratsa: lehen ordenako ekuazioak idatzi.

$$x' = vx$$

$$y' = vy$$

$$vx' = x + 2 vy - umu \frac{x + mu}{D_1} - mu \frac{x - (umu)}{D_2}$$

$$vy' = y - 2 vx - umu \frac{y}{D_1} - mu \frac{y}{D_2}$$

2- Bigarren urratsa: momentuak. aldagai aldaketa.

Abiadura eta momentua baliokideak dira. Bien arteko erlazio honako da:

$$vx = p_x + y \Rightarrow p_x = vx - y$$

$$vy = p_y - x \Rightarrow p_y = vy + x$$

$$vx = p_x + y \Rightarrow vx' = p_x' + y' = p_x' + vy = p_x' + p_y - x$$

$$vy = p_y - x \Rightarrow vy' = p_y' - x' = p_y' - vx = p_y' - (p_x + y)$$

Liburuko jatorrizko ekuazioei aldagai aldaketa eginez,

$$x' = vx \Rightarrow \boxed{x' = p_x + y}$$

$$y' = vy \Rightarrow \boxed{y' = p_y - x}$$

$$vx' = x + 2vy - umu \frac{x + mu}{D_1} - mu \frac{x - (umu)}{D_2} \Rightarrow$$

$$p_x' + p_y - x = x + 2(p_y - x) - umu \frac{x + mu}{D_1} - mu \frac{x - (umu)}{D_2} \Rightarrow$$

$$\boxed{p_x' = p_y - umu \frac{x + mu}{D_1} - mu \frac{x - (umu)}{D_2}}$$

$$vy' = y - 2vx - umu \frac{y}{D_1} - mu \frac{y}{D_2} \Rightarrow p_y' - (p_x + y) = y - 2(p_x + y) - umu \frac{y}{D_1} - mu \frac{y}{D_2} \Rightarrow$$

$$\boxed{p_y' = -p_x - umu \frac{y}{D_1} - mu \frac{y}{D_2}}$$

3- Hirugarren urratsa: energiaren ekuazioa

$$E = \frac{1}{2}(\dot{x}^2 + \dot{y}^2) - U(x, y) - \frac{1}{2} \mu (1 - \mu), U(x, y) = \frac{1}{2} (x^2 + y^2) + \frac{1 - \mu}{r_1} + \frac{\mu}{r_2}$$

Aldagai aldaketa eginez

$$vx = p_x + y$$

$$vy = p_y - x$$

$$E = \frac{1}{2}(\dot{x}^2 + \dot{y}^2) - \left(\frac{1}{2} (x^2 + y^2) + \frac{1 - \mu}{r_1} + \frac{\mu}{r_2} \right) - \frac{1}{2} \mu (1 - \mu) \Rightarrow$$

$$E = \frac{1}{2}((p_x + y)^2 + (p_y - x)^2) - \left(\frac{1}{2} (x^2 + y^2) + \frac{1 - \mu}{r_1} + \frac{\mu}{r_2} \right) - \frac{1}{2} \mu (1 - \mu) \Rightarrow$$

$$E = \frac{1}{2}(p_x^2 + 2p_x y + y^2 + p_y^2 - 2p_y x + x^2) - \left(\frac{1}{2} (x^2 + y^2) + \frac{1 - \mu}{r_1} + \frac{\mu}{r_2} \right) - \frac{1}{2} \mu (1 - \mu) \Rightarrow$$

$$E = \frac{1}{2}(p_x^2 + p_y^2) + p_x y - p_y x - \left(\frac{1 - \mu}{r_1} \right) - \frac{\mu}{r_2} - \frac{1}{2} \mu (1 - \mu)$$

$-\frac{1}{2} \mu (1 - \mu)$ konstantea denez.

$$E = \frac{1}{2} (p_x^2 + p_y^2) + p_x y - p_y x - \left(\frac{1 - \mu}{r_1} \right) - \frac{\mu}{r_2}$$

4- Laugarren urratsa: hasierako balioak.

$$x' = p_x + y \Rightarrow p_x = x' - y \Rightarrow p_x(0) = x'(0) - y(0) = 0$$

$$y' = p_y - x \Rightarrow p_y = y' + x \Rightarrow p_y(0) = y'(0) + x(0) \Rightarrow$$

$$\Rightarrow p_y(0) = -2.00158510637908252240537862224 + 0.994 \Rightarrow$$

$$\Rightarrow p_y(0) = -1.0075851063790825277344491404407$$

A.3- Fourier.

Sarrera.

Zientzilariek eta Ingeniariek, konportamendu periodikoa duten gertaerak (argia, soinua, ...) ikertzen dituzte. Gertaera hauek funtzio $g(x)$ periodikoen bidez deskribatzen dira [10, 297.orr],

$$g(x + P) = g(x), \quad x \text{ guztietarako.}$$

P zenbakiari funtzioaren periodoa deitzen zaio.

Nahikoa da 2π periodun funtzioak kontsideratzea. Hau da, $g(x)$ funtzioak P periodoa badu, $f(x) = g(Px/2\pi)$ funtzioak 2π periodoa du.

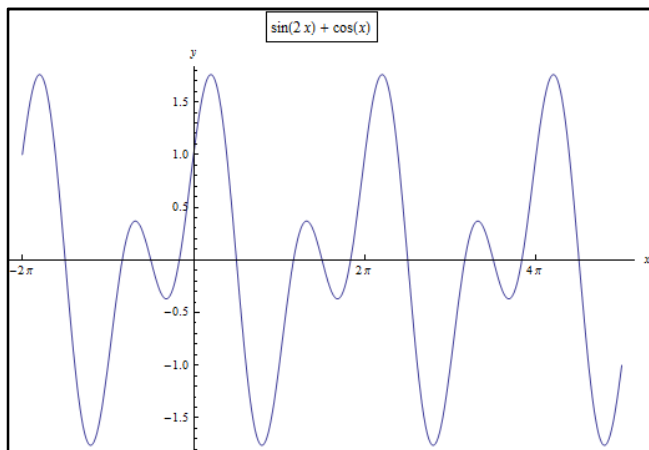
Froga:

$$f(x + 2\pi) = g\left(\frac{Px}{2\pi} + P\right) = g\left(\frac{Px}{2\pi}\right) = f(x)$$

Beraz, hurrengo azalpenetan suposatuko dugu $f(x)$ funtzioak 2π periodoa duela,

$$f(x + 2\pi) = f(x), \quad x \text{ guztietarako}$$

Funtzio 2π periodun adibideak, $\sin(jx)$ eta $\cos(jx)$ non j zenbaki osoa den.



Irudia. A.1 funtzio periodiko baten adibidea.

Fourier seriea.

Orain, edozein funtzio periodiko $a_j \cos(jx)$ eta $b_j \sin(jx)$ terminoen konbinazio bezala adieraz daitekeela azalduko dugu.

Definizioa.

$f(x)$ 2π Funtzio periodikoari dagokion $S(x)$ Fourier seriea,

$$(1) \quad S(x) = \frac{a_0}{2} + \sum_{j=1}^{\infty} (a_j \cos(jx) + b_j \sin(jx))$$

non a_j eta b_j koefizienteak Eulerren formulak erabiliz kalkula daitezken,

$$(2) \quad a_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(jx) dx, \quad j = 0, 1, \dots$$

$$b_j = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(jx) dx, \quad j = 1, 2, \dots$$

Cosenu seriea.

$f(x)$ funtzioa bikoitia bada, hau da, $f(-x) = f(x)$ x guztietarako, orduan dagokion $S(x)$ Fourier seriea coseno osagaiez soilik osatua dago,

$$(3) \quad S(x) = \frac{a_0}{2} + \sum_{j=1}^{\infty} a_j \cos(jx)$$

Sinu seriea.

$f(x)$ funtzioa bakoitia bada, hau da, $f(-x) = -f(x)$ x guztietarako, orduan dagokion $S(x)$ Fourier seriea sinu osagaiez soilik osatua dago,

$$(4) \quad S(x) = \sum_{j=1}^{\infty} b_j \sin(jx)$$

Polinomio Trigonometriko hurbilketa.

M ordeneko polinomio trigonometrikoa deitzen zaio,

$$(5) \quad T_M(x) = \frac{a_0}{2} + \sum_{j=1}^M (a_j \cos(jx) + b_j \sin(jx))$$

Fourier Serie Diskretua.

Suposatuz $\{x_j, y_j\}_{j=0}^N$ uniformeki banatutako $N + 1$ puntu:

$$(6) \quad y_j = f(x_j)$$

$$x_j = -\pi + \frac{2j\pi}{N} \quad j = 0, 1, \dots, N$$

$f(x)$ 2π funtzioa periodikoa bada eta $N > 2M$, (25) minimizatzen duen $T_M(x)$ polinomio trigonometrikoa existitzen da,

$$(7) \quad \sum_{k=1}^N (f(x_k) - T_M(x_k))^2$$

a_j eta b_j koefizienteak honako formulekin kalkulatu ditugularik,

$$(8) \quad a_j = \frac{2}{N} \sum_{k=1}^N f(x_k) \cos(jx_k) \quad j = 0, 1, \dots, M$$

$$b_j = \frac{2}{N} \sum_{k=1}^N f(x_k) \sin(jx_k) \quad j = 1, \dots, M$$

(2) Eulerren formulen integralen zenbakizko hurbilketa eginez, (8) ekuazio berdinak lortzen ditugu.

Adibidea.

$f(x) = x/2$ funtzioaren $M = 5$ ordeneko hurbilketa polinomio trigonometrikoa kalkulatu dugu.

Uniformeki banatutako 12 puntu erabiliko dituen, $x_k = -\pi + k\pi/6$, $k = 1, 2, \dots, 12$

$$T_5(x) = 0.9770486 \sin(x) - 0.4534498 \sin(2x) + 0.26179938 \sin(3x) - 0.1511499 \sin(4x) \\ + 0.0701489 \sin(5x)$$

Kalkulatarako puntu kopurua 60 eta 360 puntuetarako handituz, $T_5(x)$ koefizienteak aldatzen dira eta Fourier Seriearen koefizientei hurbiltzen zaie.

	Polinomio Trigonometriko Koefizienteak			Fourier Serie Koef.
	12 puntu	60 puntu	360 puntu	
b_1	0.9770486	0.999088598	0.99997462	1.0
b_2	-0.4534498	-0.49817096	-0.49994923	-0.5
b_3	0.26179938	0.33058726	0.33325718	0.333333333
b_4	-0.1511499	-0.24633386	-0.24989845	-0.25
b_5	0.0701489	0.19540972	0.19987306	0.2

FFT: Fast Fourier Transform.

$f(x)$ funtzioa T periododuna, hau da, $f(x + T) = f(x)$, x guztietarako.

$$f(x) = \frac{A_0}{2} + \sum_{k=1}^{\infty} A_k \cos(\omega kx) + \sum_{k=1}^{\infty} B_k \sin(\omega kx), \quad \omega = \frac{2\pi}{T}$$

$$A_k = \frac{1}{T} \int_{-T/2}^{T/2} f(x) \cos(\omega kx) dx, \quad B_k = \frac{1}{T} \int_{-T/2}^{T/2} f(x) \sin(\omega kx) dx$$

Polinomio trunkatua,

$$T_M(t) = a_0 + \sum_{k=1}^M A_k \cos\left(\frac{2\pi}{T} kt\right) + \sum_{k=1}^M B_k \sin\left(\frac{2\pi}{T} kt\right)$$

$N = 2m$, $B_M = 0$ Ezezagun kopurua bikoitia izan dadin.

1-ZATIA.

$$i^2 = -1$$

$$e^{i\theta} = \cos(\theta) + i \sin(\theta)$$

$$\cos(\theta) = \frac{1}{2}(e^{i\theta} + e^{-i\theta}), \quad \sin(\theta) = \frac{1}{2i}(e^{i\theta} - e^{-i\theta})$$

$$T_M(t) = a_0 + \sum_{k=1}^M \left(\frac{A_k - i B_k}{2} e^{\frac{i2\pi kt}{T}} + \frac{A_k + i B_k}{2} e^{-\frac{i2\pi kt}{T}} \right)$$

$$T_M(t) = \sum_{k=-m}^m a_k e^{\frac{i2\pi kt}{T}}$$

$$a_k = \frac{A_{|k|} - i \operatorname{zeinu}(k) B_{|k|}}{2}, \quad k = \pm 1, \pm 2, \dots, \pm m$$

$$a_m = a_{-m} = \frac{A_m}{2}$$

$a_k \leftrightarrow A_k, B_k$ Baliokideak dira.

2- ZATIA.

Polinomio trigonometrikoari, diskretizazio puntuetan balio berdina izatea eskatuko diogu.

$$t_j = \frac{jT}{N}, \quad j = 0, 1, 2, \dots, N-1$$

$$f_j = P(t_j) = \sum_{k=-m}^m a_k e^{\frac{i2\pi kt_j}{T}} \xrightarrow{t_j = \frac{jT}{N}} P(t_j) = \sum_{k=0}^{N-1} \hat{f}_k e^{\frac{i2\pi kj}{N}}$$

$$\text{non } \hat{f}_0, \hat{f}_1, \dots, \hat{f}_{m-1}, \hat{f}_m, \hat{f}_{m+1}, \hat{f}_{m+2}, \dots, \hat{f}_{2m-1}$$

$$a_0, a_1, \dots, a_{m-1}, a_m, a_{-m}, a_{-(m-1)}, \dots, a_{-1}$$

Oharra: k negatiboa den kasuetarako honako espresioak baliokideak dira.

$$e^{\frac{i2\pi(-3)j}{N}} \rightarrow e^{i\left(\frac{2\pi(-3)j}{N} + 2\pi j\right)} = e^{i\left(\frac{2\pi j(-3+N)}{N}\right)}$$

$$* e^{i\theta} = \cos(\theta) + i \sin(\theta) = e^{i\theta+2\pi j}; \quad 2\pi j \text{ gehituta balioa berdina da.}$$

Adibidea,

$$m = 3, N = 2m = 6$$

$$\sum_{k=-m}^m a_k e^{\frac{i2\pi kj}{6}} = a_0 + a_1 e^{\frac{i2\pi j}{6}} + a_2 e^{\frac{i2\pi 2j}{6}} + a_3 e^{\frac{i2\pi 3j}{6}} + a_{-1} e^{\frac{i2\pi 5j}{6}} + a_{-2} e^{\frac{i2\pi 4j}{6}} + a_{-3} e^{\frac{i2\pi 3j}{6}} =$$

$$= a_0 + a_1 e^{\frac{i2\pi j}{6}} + a_2 e^{\frac{i4\pi j}{6}} + (a_3 + a_{-3}) e^{\frac{i6\pi j}{6}} + a_{-2} e^{\frac{i8\pi j}{6}} + a_{-1} e^{\frac{i10\pi j}{6}}$$

$$\hat{f}_0, \hat{f}_1, \hat{f}_2, \hat{f}_3, \hat{f}_4, \hat{f}_5$$

$$a_0, a_1, a_2, a_3 + a_{-3}, a_{-2}, a_{-1}$$

$$a_m + a_{-m} = 2 a_m = A_m$$

$$f_j = P(t_j) = \sum_{k=0}^{N-1} \hat{f}_k e^{\frac{i2\pi kj}{N}} \Leftrightarrow \hat{f}_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j e^{\frac{-i2\pi kj}{N}}$$

A.4- Implementazioaren oharrak.

C lengoian inplementatutako soluzio jarraipenaren programa bi modutara erabil daiteke:

1- Terminaleko bertsioa.

Exekutatzeko urrats hauek egin behar ditugu:

1.1- Hasierako soluzioa sortu.

“HasierakoBalioakSortu.nb” fitxategia exekutatu, “fbalioak.dat” ASCII fitxategia sortzen du, zehaztutako hasierako balioei eta M parametroari dagokion hasierako soluzioarekin.

1.2- “Erabiltzaileak.c”

Fitxategi honetan R3BP problemari dagozkion funtzioak definitzen dira,

- $odefun \equiv f_l = f(u_l) + \alpha \nabla I(u_l)$
- $Hfun \equiv I(u_l)$
- $fodegrad \equiv \frac{df_l}{du_l}$
- $Hfgrad \equiv \nabla I(u_l)$

1.3- “gure-homotopia.c”

MathHomotopia terminaletik erabiltzeko adibide bat da.

Emaiza: homotopia.log fitxategian gorde da.

1.4- Konpilazioa eta exekuzioa.

make gure-homotopia.

./gure-homotopia

2- Mathematica-tik deitzeko programa.

Mathematica interfazea erabiliz homotopiaren exekuzioa: “Homotopia.nb” Erabiltzaileak honakoa zehaztu behar du:

- Odefun (problemaren ekuazio diferentzialak) eta Hfun (problemaren Hamiltondarra) funtzioak.
- Hasierako soluzioa

Exekuzioa.

{x1, init1, end1, normafx} = mathHomotopia[x0, init, end, parreal, parint];]

Parametroak.

- x0: hasierako soluzioaren diskretizazioa.
- init: jarraipen parametroaren hasierako balioa.
- end: jarraipen parametroaren bukaerako balioa.
- Parreal: problemaren parametro errealak.
- Parint: problemaren parametro osoak.

Emitza.

- $x1$: end1 parametro balioari dagokion soluzio diskretua.
- $init1$: jarraipen parametroaren hasierako balioa.
- $end1$: jarraipen parametroaren bukaerako balioa.
- $normafx$: $Norma(F(x1, \mu_{end}))$