Noname manuscript No.
(will be inserted by the editor)

N-Gorputzeko problema grabitazionalaren ebazpenerako zenbakizko metodoen integrazioa.

First Author · Second Author

Received: date / Accepted: date

Contents

F. Author

List of Figures

List of Tables

1 Sarrera.

1.1 Context of the Study

Urte luzez, zientzia arlo ezberdinek N-gorputzeko problema ikertu dute. Arlo nagusien artean aipatu daiteke, astronomoek Eguzki-Sistemaren planeten mugimendua ulertu nahian egidako lana edo kimikariek erreakzio kimikoekin esperimentatzeko molekulen dinamikaren azteketa. Arlo bakoitzak bere ezberditasunak (adibidez lege fisikoak) baditu ere, oinarrian problema berdina lantzen dutenez hauen arteko antzekotasun handiak daude. Azpimarratu ere, N-gorputzen problemaren azterketak garrantzi berezia izan duela matematikako eremu ezberdinen garapenean, esaterako dinamika ez-lineal eta kaos teorian.

Garai batean, N-gorputzen problemen azterteketak teori analitikoen bidez egiten ziren baina konputagailuen sorrerarekin, zenbakizko integrazioak bilakatu ziren tresna nagusia. Urteekin, bai konputazio teknologien aurrerapenari bai algoritmo berrien sorrerari esker, zenbakizko azteketek garapen handia izan dute. Zenbakizko simulazioen laguntzaz, Eguzki-Sistemaren mugimenduaren funtsezko galdera batzuk ezagutu ditugu eta berriki, Karplusen taldeak 2013. kimikako nobel saria jaso du kimika konputazionalean egindako lanarengatik.

Guk lan honetan, N-gorputzen problema grabitazionala aztertuko dugu. Orohar eta gaia kokatzeko asmoarekin, N-gorputzen zenbakizko ohiko integrazioak hiru taldeetan sailkatu ditzakegu:

1. Epe motzeko eta doitasun oso handiko integrazioak. Eguzki-Sistemaren efemeride zehatzak edo espazioko satelite artifizialen kokapenen kalkuluetarako erabili ohi dira.
2. Epe luzeko integrazioak baina doitasun handi gabekoa. Denbora oso luzean planeta-sistemen mugimendu ezagutzeko egindako ikerketak ditugu. Azterketa hauetan, garrantzitsua da gorputzen mugimenduaren argazkia orokorra ezagutzea baina zehaztasun handirik gabe. Normalean gorputzen arteko kolisio gertuko egoerak egoerak ez dira agertzen.
3. N-gorputz kopurua edozein izanik, hauen arteko kolisioak gerta daitezkeen problemak. Integrazio hauetan, konplexutasun handiari aurre egin behar zaio : N-gorputz kopurua miliotako izan dateke; gainera kolisio gertuko egoeren ondorioz, kalkulutan egindako zenbakizko errore txikiek eragin handia izan ditzakete soluzioan.

Gure lana goian sailkatutako integrazio moten nahasketa da, gure helburua Eguzki-Sistemaren epe luzeko eta doitasun handikoa algoritmoak garatzea baita. Aurreko hamarkadetan, Eguzki-Sistemaren planeten epe luzeko zenbakizko integrazioa erronka garrantzitsua izan zen. Adibidez, Sussmanek eta Wisdomek (1993) Eguzki sistemaren 100 miliotako integrazioaren bidez, planeten mugimendua kaotikoa zela baieztatu zuten. Aldi berean, paleoklimatologia zientziak orain milioika urte gertatutako klima zikloak azaltzeko (epel, hotz eta glaziazio artekoa), Lurraren orbitan izandako aldaketaren eraginez gertatu

zela azaltzen duen teoria (Milankovitch 1941) frogatzeko planeten orbiten efemeride zehatzak beharrezkoak dira.

Konputazio-teknologi arrerapenak handiak izan arren, Eguzki sistemaren simulazio hauek konputazionalki oso garestiak dira eta exekuzio denbora luzeak behar dituzte (Laskar La2010a 18 hilabete). Azken urteotako konputagailu berrien arkitekturaren bilakaerak, algoritmo azkarren diseinua aldatu du: simulazioak azkartzeko paralelizazioan oinarritu behar dira eta eragiketa aritmetikoek baino kostu handiagoa du memorien arteko datu komunikazioak. Beraz, oraindik ere algoritmo eraginkorragoak beharrezkoak dira eta gaur egun hauek garatzeko bide berriak ikertu behar dira.

1.2 Problem statement or motivation for the study.

Gaur egun, epe luzeko integrazioetarako hainbat zenbakizko metodo erabiltzen dira bereziki beren izaera Hamiltondarra mantentzen duten metodoak (metodo sinpletikoak). Metodo horien artean, gehien erabiltzen direnak izaera esplizituko algoritmoak dira.

Lehenik, metodo esplizitu eta inplizituei buruz dagoen ikuspegi nagusia aipatu nahi genuke. Metodo esplizituak problema ez-stiffa denean metodo inplizituak bainon eraginkorragoak dira. Metodo inplizituek duten eraginkortasun arazo handiena ekuazio sistema ez-lineala askatzea da, eta honek metodo esplituekiko CPU denbora gainkarga suposatzen du. Horregaitik problema ez-stiffa bada, metodo esplizituak erabili ohi dira eta problema stiff-a denean bakarrik jotzen dugu metodo inplizituengana. Baieztapen hau eztabaidagarria da, eta praktikan metodo inplizituetan gehiago sakondu behar dela iruditzen zaigu.

Euler metodo esplizitua.

$$y_{n+1} = y_n + hf(y_n) \tag{1}$$

Euler metodo inplizitua.

$$y_{n+1} = y_n + hf(y_{n+1}) \tag{2}$$

Zentzu honetan, metodo inplizituen ezaugarri interesgarri batzuk nabarmenduko ditugu. Abantaila nagusienetakoa malgutasuna da. Metodo inplizituek inplementazio malgua onartzen dute eta ondorioz, integratu nahi dugun problemari egokitzeko aukera gehiago eskeintzen dizkigu. Aipatzekoa da ere, metodo esplizituak sistema Hamiltondar banagarrietan bakarrik aplika daitezkeela: Hamiltondarraren egitura hau aprobetxatuz oso eraginkorrak dira baina integratu nahi den problemak bete behar duen muga ere. Metodo inplizituak aldiz, Hamiltondar orokorrei aplika daitezke eta gainera, lehen ordenako ekuazio diferentzialetarako metodo sinpletikoak inplizituak izan behar dira. Azkenik ez dugu ahaztu behar, metodo inplizituen artean orden altuko metodoak existitzen direla eta hauek nahitaezkoak dira doitasun handiko integrazioak behar ditugunean.

Lan honetan, metodo inplizituen artean Gauss zenbakizko integrazio metodoa aukeratu dugu. Hainbat autorek (Hairer eta Sanz Serna) metodo honen potentziala nabarmendu dute eta guk ere, iritzi berekoak gara. Laburki aipatuz, s ataletako metodo hau $2s$ ordenekoa da, sinpletikoa da, estabilite ezaugarri onak ditu eta paralizatzeko gaitasuna ahaztu gabe.

1.3 Aim and scope

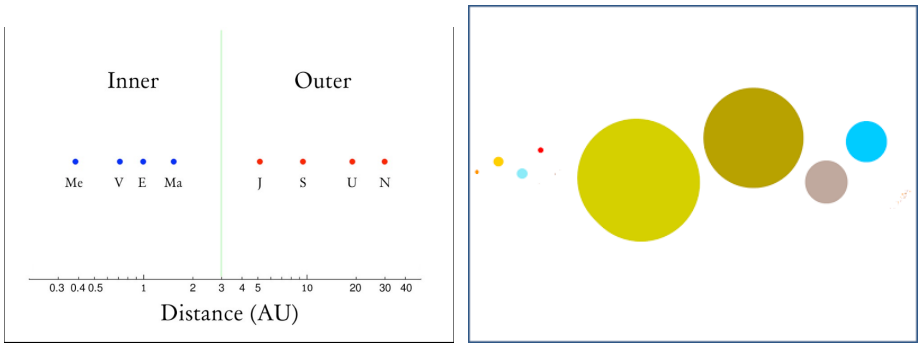
Gure helburua, Eguzki Sistemaren ebazpenerako Gauss inplizituaren inplementazio eraginkorra proposatzea da. Hau lortzeko bereziki honako aspektu hauek kontutan izango ditugu: Eguzki-sistemaren problemaren ezaugarriak, konputagailuen koma-higikorreko aritmetika eta algoritmo paraleloen abantaila.

N gorputzeko problema grabitazionalari dagokionez, Eguzki sistemaren eredu sinplea integratuko dugu. Eguzki sistemaren gorputzak masa puntualak kontsideratuko ditugu eta gure ekuazio diferentzialek, gorputz hauen arteko erakarpen grabitazionalak bakarrik kontutan hartzen dituzte. Beraz, eguzki sistemaren eredu konplexuagoetako erlatibitate efektua, gorputzen formaren eragina, eta beste zenbait indar ez-grabitazionalak ez dira kontutan hartu. Bestalde era honetako integrazioetan, gorputzen hasierako balio eta parametro zehatzak sateliteen bidez jasotako datu errealekin bat datoztela egiaztatze prozesua ez dugu landu.

Zeintzuk dira Eguzki-sistemaren problemaren ezaugarri bereziak? Batetik bi gorputzen problemaren (kepler problema) soluzioa zehatza ezaguna da eta Eguzki-Sistemaren gorputzen mugimenduaren konputazioaren oinarria. Bestetik, badugu gorputz nagusi bat (Eguzkia) eta honen inguruan bueltaka sailkatutako planetak: barne planetak, masa txikikoak eta eguzkitik gertu daudenak ; kanpo planetak, masa handikoak eta eguzkitik urrun daudenak (ikus irudia Fig.1). Eguzki-Sistemaren egitura honi abantaila handien lortzen duen planteamendua bilatuko dugu.

Konputagailuen koma-higikorreko aritmetika ondo ulertzea garrantzitsua da. Zenbaki errealen adierazpen finkoa erabiltzen denez bai zenbakiak memorian gordetzeko, bai hauen arteko kalkulu aritmetikoak egiteko, errore bat egiten dugu. Integrazio luzeetan errore hau propagatzen da eta une batetik aurrera, soluzioen zuzentasuna ezereztatzen da. Ondorioz, integrazioan zehar errore honen monitorizazioa ezagutzea interesgarria da eta integrazio luzeen kasuan, doitasun handian lan egiteko beharra azaltzen zaigu. Gaur egun doitasun altuko aritmetiken erabilera oso garestia da, inplementazioa software bidezkoa delako. Exekuzio denborak onargarriak lortzeko tarteko irtenbidea, inplementazioan doitasun ezberdinak nahastea izango litzateke.

Sarrera honetan paralelizazioari buruzko ohar bat ematea komeni da. Algoritmo baten kode unitateak paraleloan exekutatzeak badu gainkarga bat eta beraz, algoritmoaren exekuzioa paralelizazioaz azkartzea lortzeko, unitate bakoitzaren tamainak esanguratsua izan behar du. Gure Eguzki-sistemaren eredua sinplea da eta logikoa da pentsatzea eredu konplexuagoetan, paralelizazioak



(a) Planeten arteko distantziak. (b) Planeten arteko masa konparaketa.

Fig. 1: Eguzki-sistema.

abantaila handiagoa erakutsiko duela. Bestalde N gorputzen kopurua handia den problemetan, hauen arteko interakzio kopuru $O(N^2)$ handia kalkulatu behar da eta indar hauen hurbilpena modu eraginkorrean kalkulatzeko metodo ezagunak daude: *tree code* eta *fast multipole method*. Baina gure probleman gorputz kopurua txikia denez, ideia hauek gure eremutik kanpo utzi ditugu.

1.4 Overview of the study (structure of the Thesis).

Gure lanaren abiapuntua Hairer-en IRK metodoaren inplementazioa da. Autoreak IRK puntu-finkoaren inplementazio estandarrean biribiltze errorearen okerreko konportamenduaz jabetu zen eta arazo hau konpontzeko soluzioak proposatu zituen. Lehen urratsa honetan, biribiltze errorearen arazoari soluzio berri bat eman diogu eta gure IRK inplementazioaren oinarriak finkatu: formulazio, koefizienteak, geratze erizpidea,atalen hasierketa .... Gure inplementazioak biribiltze errorea propagazioa optimotik gertu dagoela baieztatzeko, *integratzaile idealaren* soluzioarekin konparatu dugu. Aldi berean, integratzailean biribiltze errorearen estimazioa monitorizatzeko aukera garatu dugu.

Bigarren urratsean, ekuazio sistema ez-lineala ebazteko puntu-finkoaren ordez, Newton sinplifikatuaren metodoa aztertu dugu. Gure ekarpena, Newton sinplifikatua modu eraginkorrean aplikatzeko teknika proposatzea izan da. S-ataletako IRK metodoa eta d-dimentsioko EDA baditugu, Newton sinplifikatuaren metodoren iterazio bakoitzean *sdxsd* tamainako sistema linealak askatu behar dira. Gure proposamena da, jatorrizko sistema lineala blokeka diagonala den sistema baliokide gisa beridaztea eta matrizearen egitura hau aprobetxatu sistema modu eraginkorrean askatzeko.

Hirugarren urratsean, Eguzki-sitemaren epe luzeko integrazioan arituko gara. Ekarpen handiena, atalen hasieraketa berri bat aplikatzea da alde kepleriarraren fluxuan oinarrituz. IRK metodoak eskeintzen digun malgutasunari esker eta N gorputzetako problema grabitazionalaren ezaugarrietaz baliatuz in-

plementazio ezberdinak egingo ditugu. Inplementazio hauen eraginkortasuna, egungo integratzaile simplektiko esplizituekin konparatuko ditugu.

Azken urratsean, esperimentalki , eguzki-sistemaren integrazioan birparametrizazio teknikaren aplikazio sinple bat erakutsiko dugu. Integratzaile sinpletikoak luzeera finkoko urratsa eduki behar du eta zentzu honetan,birparametrizazioa eraginkortasuna hobetzeko beste bide bat da.

1.5 laburpena

2 Zenbakizko Integratzaile Sinplektikoak.

2.1 Sarrera.

2.1.1 Hasierako baliodun problemak.

Hau dugu, hasierako baliodun problemaren formulazio estandarra

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{t}, \mathbf{y}(\mathbf{t})), \quad \mathbf{y}(\mathbf{t_0}) = \mathbf{y_0}, \tag{3}$$

non $\mathbf{y} : \mathbb{R} \longrightarrow \mathbb{R}^{\mathbf{d}}$ soluzioa, $\mathbf{y_0} \in \mathbb{R}^{\mathbf{d}}$ hasierako balioa eta $\mathbf{f} : \mathbb{R} \times \mathbb{R}^{\mathbf{d}} \longrightarrow \mathbb{R}^{\mathbf{d}}$ bektore eremua deskribatzen funtzioa dugu.

Goiko ekuazio (3), ekuazio-sistema moduan idatzi daiteke:

$$\begin{aligned}
\dot{y_1}(t) &= f_1(t, (y_1(t), y_2(t), \ldots, y_d(t)), \quad y_1(t_0) = y_{1,0} \\
\dot{y_2}(t) &= f_2(t, (y_1(t), y_2(t), \ldots, y_d(t)), \quad y_2(t_0) = y_{2,0} \\
&\ldots \\
\dot{y_d}(t) &= f_d(t, (y_1(t), y_2(t), \ldots, y_d(t)), \quad y_d(t_0) = y_{d,0}
\end{aligned}$$

Zenbakizko metodo baten bidez, $\mathbf{y(t)}$ soluzioaren $\mathbf{y_n} \approx \mathbf{y(t_n)}$ hurbilpena kalkulatuko dugu $t = t_n = t_0 + nh$, $n = 1, 2, \ldots$ une ezberdinetarako.

Dena den, notazioa sinplifikatzeko gure ekuazio diferentzialak *autonomoak direla suposatuko dugu, hau da, denborarekiko independenteak.*

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(\mathbf{t})), \quad \mathbf{y}(\mathbf{t_0}) = \mathbf{y_0}, \tag{4}$$

Fluxua.

Jarraian, fluxua oinarrizko kontzeptua definituko dugu. Fase-espazioko edozein $\mathbf{y_0}$ *puntuari,* $\mathbf{y}(\mathbf{t_0}) = \mathbf{y_0}$ *hasierako balio duen* $\mathbf{y}(\mathbf{t})$ *soluzioa asignatzen dion mapping-ari deitzen diogu. Izendatzeko* φ_t *notazioa erabiliko dugu,*

$$\varphi_t(\mathbf{y_0}) = \mathbf{y}(\mathbf{t}) \quad \textbf{baldin} \quad \mathbf{y}(\mathbf{t_0}) = \mathbf{y_0}$$

Zenbakizko diskretizazioa.

$\mathbf{y_n}$ *balioa emanda,* $\mathbf{y_{n+1}}$ *soluzioaren hurbilpena kalkulatzeko formulari zenbakizko fluxua deritzogu. Honako notazioa,* $\mathbf{y_{n+1}} = \phi_{\mathbf{h}}(\mathbf{y_n})$ *erabiliko dugu.*

Metodoaren ordena.

h urrats finkoko zenbakizko metodoa p ordenekoa dela esaten da, errore lokalak honakoa betetzen duenean

$$\mathbf{y_{n+1}} - \mathbf{y}(\mathbf{t_n} + \mathbf{h}) = \mathbf{O}(\mathbf{h^{p+1}}) \quad , \quad \mathbf{h} \to \mathbf{0}. \tag{5}$$

Adibidea.

Euler metodoa da, hasierako baliodun problemetarako oinarrizko zenbakizko metodoa. $p = 1$ ordeneko metodoa da eta era honetan definitzen da,

$$\mathbf{y_{n+1}} = \phi_{\mathbf{h}}(\mathbf{y_n}), \quad \mathbf{non} \quad \phi_{\mathbf{h}}(\mathbf{y_n}) = \mathbf{y_n} + \mathbf{hf}(\mathbf{y_n}), \quad \mathbf{h} = \mathbf{t_{n+1}} - \mathbf{t_n} \tag{6}$$

2.1.2 Sistema-Hamiltondarrak.

Ekuazio diferentzial arrunten formulazio Hamiltondarra erabili ohi da errealitateko sistemak matematikoki adierazteko. Azpimarratu metodo simpletikoak sistema Hamiltondar hauen soluzioaren hurbilpena kalkulatzeko zenbakizko metodo bereziki onak ditugula.

$H(p, q)$ funtzio leuna izanik , non $H : \mathbb{R}^{2d} \longrightarrow \mathbb{R}$ eta $(p, q) = (p_1, \ldots, p_d, q_1, \ldots, q_d)$, dagokion ekuazio diferentzialak era honetan definitzen dira

$$\frac{d}{dt}p_j = -\frac{\partial H(p,q)}{\partial q_j}, \quad \frac{d}{dt}q_j = \frac{\partial H(p,q)}{\partial p_j}, \qquad j = 1, \ldots, d. \tag{7}$$

Edo notazio baliokidea erabiliz,

$$\dot{y} = J^{-1} \nabla H(y), \quad y = \begin{pmatrix} p \\ q \end{pmatrix}, \quad J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}, \tag{8}$$

p eta q bektoreen d dimentsioa sistemaren askatasun maila deritzo. $H(p, q)$ funtzioaren balioa integrazioan zehar konstante mantentzen da.

Hamiltondar banagarriak.

Hamiltondar banagarriak egitura bereziko sistema Hamiltondarrak ditugu, $H(p, q) = T(p) + U(q)$. Horien artean, bigarren ordeneko ekuazio diferentzial mota garrantzitsuak aipatu behar ditugu ,

$$H(p, q) = \frac{1}{2}p^T p + U(q),$$

eta beraz, dagokien ekuazio diferentzialak,

$$\dot{p} = -\frac{\partial U(q)}{\partial q}, \quad \dot{q} = p.$$

Adibidea.

Kepler problemari dagokion Hamiltondarra,

$$H(p_1, p_2, q_1, q_2) = \frac{1}{2}(p_1^2 + p_2^2) - \frac{1}{\sqrt{q_1^2 + q_2^2}}, \tag{9}$$

eta dagozkion ekuazio diferentzialak

$$\frac{d}{dt}p_1 = -\frac{q_1}{(q_1^2 + q_2^2)^{\frac{3}{2}}}, \quad \frac{d}{dt}p_2 = -\frac{q_2}{(q_1^2 + q_2^2)^{\frac{3}{2}}} \tag{10}$$

$$\frac{d}{dt}q_i = p_i, \quad i = 1, \dots d \tag{11}$$

Hamiltondar perturbatuak.

Hamiltondar perturbatuak egitura hau duten $H = H_A + \epsilon H_B$ ($|H_B| \ll |H_A|$) sistemak ditugu. Adibidez Eguzki sistemaren probleman, Hamiltondarra modu honetan idatzi daiteke $H = H_k + H_I$, non alde nagusia H_K planeta bakoitzaren eguzki inguruko mugimendu kepleriarra den eta H_I aldiz, planeten arteko interakzioek eragiten duten perturbazio txikia.

Aukera bat Jakobi koordenatuak erabiliz Hamiltondarra $H(p, q) = H_K(p, q) + H_I(q)$ moduan banatzen du, non $H_K(p, q)$ (kepler problema independienteak) eta $H_I(q)$ integratu daiteke. Beste aukera, koordenatu Heliozentrikoak erabiliz $H(p, q) = H_K(p, q) + H_I(p, q)$ moduan banatzen da, non $H_I(p, q)$ ezin daitekeen zuzenean integratu.

2.2 Gauss metodoak.

2.2.1 Runge-Kutta metodoak.

b_i, a_{ij} eta $c_i = \sum_{i=0}^{s} a_{ij}$ $(1 \le i, j \le s)$ koefiziente errealek definitzen dute s-ataleko Runge-Kutta metodoa. Butcher izeneko taulan moduan laburtu ohi dira koefiziente hauek,

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b^T} \end{array}, \qquad \begin{array}{c|ccc} c_1 & a_{11} & a_{12} \dots & a_{1s} \\ c_2 & a_{21} & a_{22} \dots & a_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} \dots & a_{ss} \\ \hline & b_1 & b_2 \dots & b_s \end{array} \tag{12}$$

Runge-kutta metodoak urrats bakarreko integratzaileak dira eta bi mota bereizi ditzakegu: explizituak (ERK) eta inplizitua (IRK). ERK metodoak eraginkorrak kontsideratzen dira problema ez-stiff-etarako eta IRK metodoak ordea, problema stiff-etarako. Hasierako baliodun problema baten $y(t)$ soluzioaren $y_n \approx y(t_n)$ hurbilpena era honetan kalkulatzen da,

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i \ f(Y_{n,i}) \quad , \tag{13}$$

non $Y_{n,i}$ atalak era honetan definitzen diren,

$$Y_{n,i} = y_n + \ h \ \sum_{j=1}^{s} a_{ij} \ f(Y_{n,j}) \qquad i = 1, \ldots, s. \tag{14}$$

Gauss metodoa IRK metodoa bat da. S-ataletako Runge-Kutta metodoen artean $p = 2s$ ordena duen metodo bakarra dugu. Gauss metodoen koefizienteek honako baldintzak betetzen dituzte:

1. *Sinplektizidade baldintza.*

$$b_i a_{ij} + b_j a_{ji} - b_i b_j = 0, \quad 1 \leqslant i, j \leqslant s \tag{15}$$

2. *Koefiziente simetrikoak.*

$$b_i = b_{(s-i+1)}, \quad i = 1, 2, \ldots \lceil \frac{s}{2} \rceil \tag{16}$$

$$c_{(s-i+1)} = 1 - c_i, \quad i = 1, 2, \ldots \lceil \frac{s}{2} \rceil \tag{17}$$

Runge-Kutta Inplizituaren algoritmo orokorra,

for $n \leftarrow 1$ **to** *endstep* **do**
  Hasieratu $Y_{i,n}^{[0]} \quad , \quad i = 1, \ldots, s;$
  while *(konbergentzia lortu)* **do**
    $F_{n,i} = f(Y_{n,i}) \quad , \quad i = 1, \ldots, s;$
    $Y_{n,i} = y_{n-1} + h \ \sum_{j=1}^{s} a_{ij} F_{n,j} \quad , \quad i = 1, \ldots, s;$
  end
  $y_n = y_{n-1} + h \ \sum_{i=1}^{s} b_i F_{n,i};$
end

ALGORITHM 1: Main Algorithm

Algotimo nagusiko agindu bakoitzari ohar moduko egingo diogu, IRK metodoaren hainbat zehaztapen emateko helburuarekin.

1. *Hasieratu* $Y_{i,n}^{[0]}$.
   Atalen hasieraketa egokia definitu behar da. Aukera sinpleena $Y_{i,n}^{[0]} = y_{n-1}$ *hasieratzea da baina aurreko urratseko informazioa erabiliz hurbilketa hobea lortu daiteke. Aurreko urratseko atalen polinomio intepolatzailearen bidezko hasieraketa era honetan adierazi dezakegu* $Y_{i,n}^{[0]} = g(Y_{i,n-1})$, $i = 1, \ldots, s$.
2. $F_{n,i} = f(Y_{n,i})$.
   Atal bakoitzarentzat ekuazio diferentzialaren ebaluaztapena indenpendientea da eta paraleloan exekutatu daiteke.
3. $Y_{n,i} = y_{n-1} + h \sum_{i=1}^{s} a_{ij} F_{n,j}$, $i = 1, \ldots, s$.
   Ekuazio-sistema ez lineala metodo iterateratibo bat erabiliz askatu behar da. Metodo hau, Puntu Finkoaren metodoa edo Newtonen metodo sinplifikatua izan daiteke.
4. $y_n = y_{n-1} + h \sum_{i=1}^{s} b_i F_{n,i}$
   Integrazio luzeak direnean, urrats asko ematen dira eta koma higikorreko aritmetika dela eta, doitsun galera ekiditzeko batura konpensatu teknika erabili ohi da.

2.2.2 Kolokazio metodoak.

Kolokazio metodoak ekuazio diferentzialen zenbakizko soluzioa azaltzeko beste modu bat dira. Gauss metodoak kolokazio metodoak ditugu eta hauen abantaila da, zenbakizko soluzioa diskretizazio puntuetan ez ezik, polinomio interpolatzaile batek modu jarraian emandako soluzioa lortzen dugula. Honako definizioa emango dugu,

Definizioa. $c_1, c_2, \ldots, c_s$ $(0 \leq c_i \leq 1)$ *zenbaki errealak izanik, s-mailako* $u(t)$ *kolokazio polinomioak honakoa betezen du,*

$$u(t_0) = y_0$$

$$\dot{u}(t_0 + c_i h) = f(t_0 + c_i h, u(t_0 + c_i h)), \quad i = 1, \ldots, s,$$

eta soluzioa $y_1 = u(t_0 + h)$.

Theorem 1.4 *(Guillou and Soule 1969, Wright 1970). Kolokazio metodoaren definizioa eta jarraian emandako moduan kalkulatutako koefizienteko s-ataleko Runge-Kutta metodoa baliokideak dira.*

$$a_{ij} = \int_0^{c_i} l_j(\tau)\, d\tau, \quad b_i = \int_0^1 l_i(\tau)\, d\tau \tag{18}$$

non $l_i(\tau)$ *Lagrangiar polinomioa dugu* $l_i(\tau) = \prod_{l \neq i} \frac{(\tau - c_l)}{(c_i - c_l)}$.

Definizioa. *Gauss metodoak c_i $(1 \leq i \leq s)$ koefizienteak "sth shifted Legendre" polinomioaren zeroak aukeratuz,*

$$\frac{d^s}{dx^s}\big(x^s(x-1)^s\big),$$

Nodo hauetan oinarritutako Runge-Kutta metodoa $p = 2s$ ordena du.

Adibidea. *$s = 1$ "Implicit Midpoint Rule" izeneko $p = 2$ ordeneko metodoa eta $s = 2$, $p = 4$ ordeneko metodoa.*

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array} \quad , \qquad \begin{array}{c|cc} \frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array},$$

Irudia

2.3 Konposizio metodoak.

2.3.1 Konposizio metodoak.

Oinarrizko metodo baten konponsizioaren bidez, orden handiagoko metodoak lortzen dira oinarrizko metodoaren propietateak mantenduz.

Definizioa orokorra. *ϕ_h oinarrizko metodoa eta $\gamma_1, \ldots, \gamma_s$ zenbaki errealak emanik, urrats luzeera hauen $\gamma_1 h, \gamma_2 h, \ldots, \gamma_s h$ konposaketari dagokion konposizio metodoa,*

$$\Psi_h = \phi_{\gamma_s h} \circ \cdots \circ \phi_{\gamma_1 h}. \tag{19}$$

Teorema. *Demagun ϕ_h urrats bakarreko eta p ordeneko metodoa. Konposizio metodoa gutxienez $p+1$ ordeneko izango da baldin,*

$$\gamma_1 + \cdots + \gamma_s = 1$$

$$\gamma_1^{p+1} + \cdots + \gamma_s^{p+1} = 1 \tag{20}$$

Metodo simetrikoen konposizio simetrikoa. *ϕ_h metodoa $p = 2$ ordenekoa eta simetrikoa izanik, era honetako konposizioak aurkitu dira,*

$$\Psi_h = \phi_{\gamma_s h} \circ \phi_{\gamma_s - 1h} \circ \cdots \circ \phi_{\gamma_{2h}} \circ \phi_{\gamma_1 h} \tag{21}$$

non $\gamma_s = \gamma_1, \gamma_{s-1} = \gamma_2, \ldots$

Algoritmoa. *Konposizio metodoen algoritmo orokorra honakoa izango litzateke:*

for $n \leftarrow 1$ **to** *endstep* **do**
  $Y_{0,n} = y_{n-1}$;
  for *i=1,2,...,s* **do**
    $Y_{i,n} = \phi_{\gamma_i h}(Y_{i-1,n})$;
  end
  $y_n = Y_{s,n}$;
end

ALGORITHM 2: Konposizio metodoak.

Oharrak. Algoritmoari buruzko hainbat ohar azpimarratuko ditugu:

1. *Esplizitua.*
   Konposizio metodo hauek esplizituak dira. Metodo hauetan ez da ekuazio sistemarik askatu behar, eta kalkuluak azkarrak dira.
2. *Sekuentziala. Urrats bakoitzaren kalkuluak modu sekuntzialean egin behar ditugu.*
3. *Oinarrizko metodoa: Störmer-Verlet.*
   Bigarren ordeneko ekuazio diferentzialak ditugunean, Störmer-Verlet integratzailean oinarritzen diren konposizio metodoekin urrats bakoitzean s ekuazio diferentzialaren ebaluaztapena egin behar ditugu.

2.3.2 Gure inplementazioa.

*Gure erreferentzia, Störmer-Verlet metodoan oinarritzen den konposizio metodoa izango da. Zehazki, Sofroniok eta Spalettak (*2004*) aurkitutako $p = 10$ ordeneko metodo optimoa. Beraz, lehenik Störmer-Verlet metodoa definituko dugu eta jarraian, metodoaren koefizienteak emango ditugu.*

Störmer-Verlet metodoa. *$p = 2$ ordeneko metodo sinplektikoa eta simetrikoa dugu.*

$$
\begin{aligned}
p_{\frac{n+1}{2}} &= p_n - \frac{h}{2}\nabla_q H(p_{\frac{n+1}{2}}, q_n) \\
q_{n+1} &= q_n + \frac{h}{2}\big(\nabla_p H(p_{\frac{n+1}{2}}, q_n) + \nabla_p H(p_{\frac{n+1}{2}}, q_{n+1})\big) \\
p_{n+1} &= p_{\frac{n+1}{2}} - \frac{h}{2}\nabla_q H(p_{\frac{n+1}{2}}, q_{n+1})
\end{aligned}
\tag{22}
$$

edo

$$
q_{\frac{n+1}{2}} = q_n + \frac{h}{2}\nabla_p H(p_n, q_{\frac{n+1}{2}})
$$

$$p_{n+1} = p_n - \frac{h}{2}\big(\nabla_q H(p_n, q_{\frac{n+1}{2}}) + \nabla_q H(p_{n+1}, q_{\frac{n+1}{2}})\big) \tag{23}$$
$$q_{n+1} = q_{\frac{n+1}{2}} + \frac{h}{2}\nabla_p H(p_{n+1}, q_{\frac{n+1}{2}})$$

Bigarren ordeneko ekuazio diferentziala ditugunean metodoa esplizitua da eta modu honetan labur daiteke,

$$\begin{aligned} p_{\frac{n+1}{2}} &= p_n + \frac{h}{2} f(q_n) \\ q_{n+1} &= q_n + h p_{\frac{n+1}{2}} \\ p_{n+1} &= p_{\frac{n+1}{2}} + \frac{h}{2} f(q_{n+1}) \end{aligned} \tag{24}$$

edo

$$\begin{aligned} q_{\frac{n+1}{2}} &= q_n + \frac{h}{2} p_n \\ p_{n+1} &= p_n - h f(q_{\frac{n+1}{2}}) \\ q_{n+1} &= q_{\frac{n+1}{2}} + \frac{h}{2} p_{n+1} \end{aligned} \tag{25}$$

10 ordeneko metodoa konposizio metodoa (CO1035)*. Sofronio eta Spalettaren (*2004*), $s = 35$ eta $p = 10$ ordeneko metodoa, orainarteko orden altuko konposizio metodo eraginkorrena kontsideratu daiteke.*

$$\begin{aligned}
\gamma_1 = \gamma_{35} &= 0.078795722521686419263907 68 \\
\gamma_2 = \gamma_{34} &= 0.313096103415108527764812 47 \\
\gamma_3 = \gamma_{33} &= 0.027918383235078066109520 27 \\
\gamma_4 = \gamma_{32} &= -0.229592841593907094151213 40 \\
\gamma_5 = \gamma_{31} &= 0.130962061077164863174656 86 \\
\gamma_6 = \gamma_{30} &= -0.269733405654510714344609 73 \\
\gamma_7 = \gamma_{29} &= 0.074973343155891435666137 11 \\
\gamma_8 = \gamma_{28} &= 0.111993423999810204889575 08 \\
\gamma_9 = \gamma_{27} &= 0.366133449546226751193148 12 \\
\gamma_{10} = \gamma_{26} &= -0.399105630136035897878629 81 \\
\gamma_{11} = \gamma_{25} &= 0.103087398527471077315802 77 \\
\gamma_{12} = \gamma_{24} &= 0.411430873955890237820704 12 \\
\gamma_{13} = \gamma_{23} &= -0.004866360583135261762195 66 \\
\gamma_{14} = \gamma_{22} &= -0.392033353708639906448081 94 \\
\gamma_{15} = \gamma_{21} &= 0.051942502962449647037182 90 \\
\gamma_{16} = \gamma_{20} &= 0.050665090759924496335874 34 \\
\gamma_{17} = \gamma_{19} &= 0.049674370639729879054568 80 \\
\gamma_{18} &= 0.049317735759594537917680 01
\end{aligned}$$

2.4 Splitting metodoak.

2.4.1 Splitting metodoak.

Demagun jatorrizko $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$ *problema era honetan bana daitekeela,*

$$\dot{\mathbf{y}} = \mathbf{f}^{[\mathbf{1}]}(\mathbf{y}) + \mathbf{f}^{[\mathbf{2}]}(\mathbf{y}) \tag{26}$$

non $\dot{\mathbf{y}} = \mathbf{f}^{[\mathbf{1}]}(\mathbf{y})$ *eta* $\dot{\mathbf{y}} = \mathbf{f}^{[\mathbf{2}]}(\mathbf{y})$ *sistemen fluxu zehatzak,* $\varphi_t^{[1]}$ *eta* $\varphi_t^{[2]}$ *esplizituki kalkula daitezke.*

Lie-Trotter splitting*.* $p = 1$ *ordeneko metodoak,*

$$\phi_h^* = \varphi_h^{[2]} \circ \varphi_h^{[1]}$$

$$\phi_h = \varphi_h^{[1]} \circ \varphi_h^{[2]} \tag{27}$$

Strang splitting*.* $p = 2$ *ordeneko metodo simetrikoa,*

$$\phi_h = \varphi_{\frac{h}{2}}^{[1]} \circ \varphi_h^{[2]} \circ \varphi_{\frac{h}{2}}^{[1]} \tag{28}$$

Adibidea*. Störmer-Verlet metodoa, Strang Splitting metodoa dela ikusiko dugu. Suposatu dezagun Hamiltondar banagarria dugula,* $H(p,q) = T(p) + U(q)$ *. Jatorrizko sistema Hamiltondarra bitan banatuko dugu,*

$$\dot{p} = 0, \qquad \dot{p} = -U_q(q)$$

$$\dot{q} = T_p(p), \qquad \dot{q} = 0 \tag{29}$$

eta integratuz lortuko ditugu bakoitzari dagokion ($\varphi_t^{[T]}$,$\varphi_t^{[U]}$*) fluxu zehatzak,*

$$p(t) = p_0, \qquad p(t) = p_0 - tU_q(q_0)$$

$$q(t) = q_0 + tT_p(p_0), \qquad q(t) = q_0 \tag{30}$$

Beraz, Störmer-Verlet metodoa bat dator Strang Splitting definizioarekin $\varphi_{\frac{h}{2}}^{[U]} \circ \varphi_h^{[T]} \circ \varphi_{\frac{h}{2}}^{[U]}$.

Splittig metodo orokorrak. *Konposizio metodoen modu berean, oinarrizko Splitting metodoak konposatuz orden altuagoko metodoak lortzen dira.* $a_1, b_1, a_2, \ldots, a_m, b_m$ *koefiziente errealak izanik,*

$$\Psi_h = \varphi^{[2]}_{b_m h} \circ \varphi^{[1]}_{a_m h} \circ \varphi^{[2]}_{b_m - 1 h} \circ \cdots \circ \varphi^{[1]}_{a_2 h} \circ \varphi^{[2]}_{b_1 h} \circ \varphi^{[1]}_{a_1 h} \tag{31}$$

for $n \leftarrow 1$ **to** *endstep* **do**
  $Y_{0,n} = y_{n-1}$;
  for *i=1,2,...,m* **do**
    $Y_{i,n} = (\varphi^{[2]}_{b_i h} \circ \varphi^{[1]}_{a_i h})(Y_{i-1,n})$;
  end
  $y_n = Y_{m,n}$;
end

ALGORITHM 3: Splitting metodoak.

2.4.2 Erreferentzia.

N-gorputzeko problema grabitazionalaren Hamiltondarra $H(p,q) = T(p) + U(q)$, *koordenatu sistema egokia erabiliz modu honetan* $H = H_K + H_I$ *(*$|H_I| \ll |H_K|$*) beridatzi daiteke. Azken egitura honetan oinarrituz orden altuko hainbat Spliiting metodo aurkitu dira. Gure errefentziazko metodoak Hamiltondar egitura honi bereziki egokitutako integratzaileak izango dira:*

1. . $SABA_4$ *(Laskar, 2001).*

   Hamiltondarra, $H = H_A + \epsilon H_B$ *izanik eta goiko notazioa erabiliz,*

   $$SABA_4 = \varphi^{[A]}_{c_1 h} \circ \varphi^{[B]}_{d_1 h} \circ \varphi^{[A]}_{c_2 h} \circ \varphi^{[B]}_{d_2 h} \circ \varphi^{[A]}_{c_3 h} \circ \varphi^{[B]}_{d_2 h} \circ \varphi^{[A]}_{c_2 h} \circ \varphi^{[B]}_{d_1 h} \circ \varphi^{[A]}_{c_1 h}$$

   Koefizienteak,

   $$c_1 = \frac{1}{2} - \frac{\sqrt{525 + 70\sqrt{30}}}{70}, \qquad d_1 = \frac{1}{4} - \frac{\sqrt{30}}{72}$$
   $$c_2 = \frac{\left(\sqrt{525 + 70\sqrt{30}} - \sqrt{525 - 70\sqrt{30}}\right)}{70}, \qquad d_2 = \frac{1}{4} + \frac{\sqrt{30}}{72}$$
   $$c_3 = \frac{\sqrt{525 - 70\sqrt{30}}}{35}$$

   Corrected integrator. Urrats bat gehitutako integratzailea $SABAC_4$,

   $$SABAC_4 = \varphi[B]_{\frac{-c}{2}} \circ SABA_4 \circ \varphi[B]_{\frac{-c}{2}}$$

   non $c = 0.00339677504820860133153215778349214 4$.

2. . $ABAH1064$ *(Blanes, 2013).*

Eguzki sistemaren integraziorako koordenatu Heliozentrikoei dagokion Hamiltondarra era honetakoa dugu,

$$H(p,q) = H_K(p,q) + H_I(p,q), \quad H_I(p,q) = T_1(p) + U_1(q)$$

$H_I(p,q)$ *fluxua zehazki kalkulatu ordez honen hurbilpen bat erabiliz,*

$$\varphi_t^I \approx \tilde{\varphi}_t^I = \varphi_{\frac{tb_i}{2}}^{[U_1]} \circ \varphi_t b_i^{[T_1]} \circ \varphi_{\frac{tb_i}{2}}^{[U_1]}$$

garatutako $ABAH1064$ *Splitting metodoa aztertuko dugu,*

$$ABAH1064 = \prod_{i=1}^{5} \varphi_{a_i h}^K \circ \tilde{\varphi}_{b_i h}^I$$

$$a_1 = 0.047319086976533822704043717963208132509 88$$

$$a_2 = 0.265110523574878515953948003618569320107 8$$

$$a_3 = -0.009976522883811240843267468164812380613143$$

$$a_4 = -0.059929199734941551263952479877296760040 16$$

$$a_5 = 0.257476112067340453449228226460331688035 6$$

$$b_1 = 0.119688462458532203531286429748989214385 2$$

$$b_2 = 0.375295585537937425042012853768750319945 1$$

$$b_3 = -0.468459341832599378365082040980538174060 5$$

$$b_4 = 0.335139734275589701039309894294956904927 5$$

$$b_5 = 0.276671119121080097504945726335683469605 5$$

2.5 Kepler fluxua.

Bi gorputzen problema*. Elkar erakartzen diren bi gorputzen mugimendua kalkutzeko, gorputz baten kokapena koordenatu sistemaren jatorria kontsideratuko dugu; bigarren gorputzaren kokapena Newtonen legeak definitutako ekuazio diferentzialaren arabera kalkulatuko dugu,*

$$\ddot{\mathbf{q}} = -\frac{\mu}{\|\mathbf{q}\|^{\mathbf{3}}}\ \mathbf{q}, \tag{32}$$

non $\mu = G(m_0 + m_1)$ *eta* $\mathbf{q} \in \mathbb{R}^{\mathbf{3}}$.

Baliokidea dugu, era honetan definitutako sistema Hamiltondarra,

$$H(\mathbf{p},\mathbf{q}) = \frac{\mathbf{1}}{\mathbf{2}}(\mathbf{v^2}) - \frac{\mu}{\|\mathbf{q}\|}, \quad \dot{\mathbf{q}} = \mu\ \mathbf{v}. \tag{33}$$

Ideia nagusia. *Koordenatu kartesiarretatik koordenatu eliptikoetara* (a, e, i, Ω, E) *itzulpena egingo dugu. Kontutan hartuta* E *(izena??) aldagai ezik beste aldagaiek konstante mantentzen direla,* E_0 *abiapuntua harturik,* $\triangle t$ *denbora tartea aurrera egingo dugu* E_1 *balioa berria kalkulatzeko. Azkenik, koordenatu eliptikoetatik koordenatu kartesiarrak berreskuratuko ditugu kokapen eta abiadura berriekin.*

$$(\mathbf{q_0}, \mathbf{v_0}) \in \mathbb{R}^{\mathbf{6}} \quad \longrightarrow \quad (\mathbf{a}, \mathbf{e}, \mathbf{i}, \mathbf{\Omega}, \mathbf{E_0}) \in \mathbb{R}^{\mathbf{6}}$$

$$\downarrow \triangle t$$

$$(\mathbf{q_1}, \mathbf{v_1}) \in \mathbb{R}^{\mathbf{6}} \quad \longleftarrow \quad (\mathbf{a}, \mathbf{e}, \mathbf{i}, \mathbf{\Omega}, \mathbf{E_1}) \in \mathbb{R}^{\mathbf{6}}$$

Newton metodoa. *Kepler-en ekuazioan oinarrituz* $(E - e \sin E = n(t - t_p))$, $E_1 = \triangle E + E_0$ *balioa kalkulatuko Newtonen metodoa aplikatuz,*

$$f(\triangle E) = \triangle E - ce \sin(\triangle E) - se(\cos(\triangle E) - 1) - n\triangle t = 0$$

$$\triangle E^{[k+1]} = \triangle E^{[k]} - \frac{f(\triangle E^{[k]})}{f'(\triangle E^{[k]})} \tag{34}$$

Ekuazioak. *Gure inplementazioan erabilitako ekuazio guztien azalpenak eta definizoak eranskinean eman ditugu.*

2.6 Laburpena.

Hauek dira Eguzki sistemaren integraziorako konparatuko ditugun metodoak,

Table 1: Integrazio metodoen laburpena

	C1035	ABAH1064	GAUSS-12
	Konposizio met. Sofronio (2004)	Splitting met. Blanes et al. 2013	IRK met.
Hamiltoniarra	Orokorra	Perturbatua	Orokorra
Mota	Esplizitua	Esplizitua	Inplizitua
Ordena	10	10	12
Atalak	35	9	6
Parall.	Ez	Ez	Bai

3 Problemak.

3.1 Sarrera.

3.2 Pendulu bikoitza.

Planoan pendulu bikoitzaren problema era honetan definitzen da: m_1,m_2 masadun bi pendulu eta l_1, l_2 luzeerako makilez (masa gabekoak kontsideratuko dituguak) elkar lotuta. Penduluen aldagai-egoerak bi angelu (Θ_1,Θ_2) eta dagokion momentuak (P_1,P_2) dira.

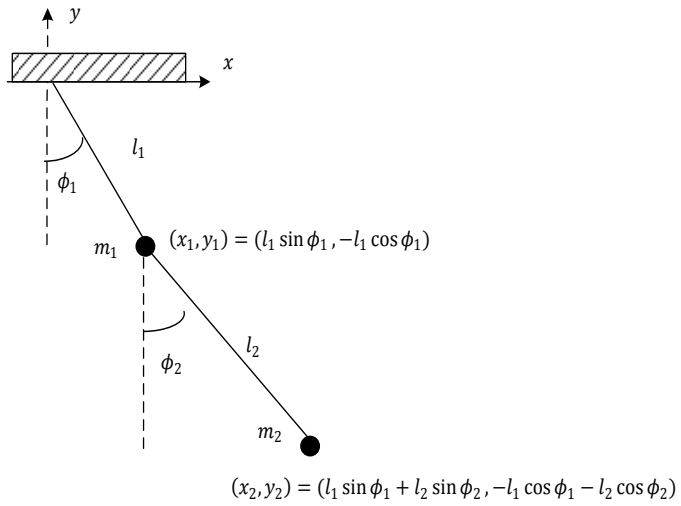


Fig. 2: Pendulu bikoitza.

3.2.1 Ekuazioak.

Hamiltondarra.

$$q = (\Theta_1, \Theta_2) \quad , \quad p = (P_1, P_2) \quad ,$$

$$H(q,p) = \left(\frac{C1\ P_1^2 + C2\ P_2^2 + C3\ P_1\ P_2\ \cos(\Theta_1 - \Theta_2)}{(C4 + C5\ \sin^2(Q_1 - Q_2))} \right) - C6\ \cos(\Theta_1) - C7\ \cos(\Theta_2),$$

$C1 = l_2^2 * m_2,$
$C2 = l_1^2 * (m_1 + m_2),$
$C3 = -2 * l_1 * l_2 * l_2,$
$C4 = 2 * l_1^2 * l_2^2 * m_2 * m_1,$
$C5 = 2 * m_1^2 * l_2^2 * m_2^2,$
$C6 = g * l_1 * (m_1 + m_2),$
$C7 = g * l_2 * m_2.$

Ekuazio diferentzialak.

$$\dot{\Theta}_1 = \frac{2 * C1 * P1 + C3 * \cos(Q1 - Q2) * P2}{aux1},$$

$$\dot{\Theta}_2 = \frac{(2 * C2 * P2 + C3 * \cos(Q1 - Q2) * P1)}{aux1},$$

$$\dot{P1} = -(aux4 + C6 * \sin(Q1)),$$

$$\dot{P1} = (aux4 - C7 * \sin(Q2)),$$

$aux1 = C4 + C5 * \sin(Q1 - Q2) * \sin(Q1 - Q2),$
$aux2 = C3 * \cos(Q1 - Q2),$
$aux3 = 2 * C5 * \sin(Q1 - Q2) * \cos(Q1 - Q2),$
$aux4 = \frac{(-1/aux1^2) * (C1 * P1^2 + C2 * P2^2 + P1 * P2 * aux2) * aux3) - (C3 * P1 * P2 * \sin(Q1 - Q2))}{aux1}.$

Jakobiarra.???

3.2.2 Hasierako balioak.

Sistemaren parametroak. *Gure esperimentuetarako honako parametroak kontsideratuko ditugu,*

$$g = 9.8\ \frac{m}{sec^2} \quad l_1 = 1.0\ m\ ,\ l_2 = 1.0\ m\ ,\ m_1 = 1.0\ kg\ ,\ m_2 = 1.0\ kg$$

Hasierako balioak. *Pendulu bikoitza konportamendu kaotikoa duen sistema ezlineala da. Zentzu honetan bi hasierako balio ezberdin kontsideratu ditugu [?]:*

1. *Hasierako balio ez-kaotikoak:* $q(0) = (1.1, 0)$, $p(0) = (0, 2.7746)$.
2. *Hasierako balio kaotikoak:* $q(0) = (0, 0)$, $p(0) = (0, 3.873)$.

3.2.3 Kodeak.

Mathematican DoublePendulum.m paketean honako funtzioak inplementatu ditugu:

1. *Hamiltondarra: DoublePendulumHam.*
2. *EDA: DoublePendulumODE.*
3. *Jakobiarra: DoublePendulumJAC.*

C-lengoaian GaussUserProblem.c fitxategian honako funtzioak inplementatu ditugu:

1. *Hamiltondarra: HamPendulum().*
2. *EDA: OdePendulum().*
3. *Jakobiarra: JacPendulum().*

3.3 N-Body problema.

N gorputzeko problema grabitazionalari dagokionez, Eguzki sistemaren eredu sinplea integratuko dugu. Eguzki sistemaren gorputzak masa puntualak kontsideratuko ditugu eta gure ekuazio diferentzialek, gorputz hauen arteko erakarpen grabitazionalak bakarrik kontutan hartzen dituzte.Beraz, eguzki sistemaren eredu konplexuagoetako erlatibitate efektua, gorputzen formaren eragina, eta beste zenbait indar ez-grabitazionalak ez dira kontutan hartu.

$(N+1)$ gorputz kopurua izanik, $q_i, p_i \in \mathbb{R}^3, m_i \in \mathbb{R}$ gorputz bakoitzaren kokapena, momentua eta masa dira. Bestalde , momentua era honetan definituko dugu $p_i = m_i * v_i$ non $\frac{dq_i}{dt} = v_i$ den.

3.3.1 Ekuazioak.

Hamiltondarra.

$$H(q,p) = \frac{1}{2} \sum_{i=0}^{N} \frac{\|p_i\|^2}{m_i} - G \sum_{0 \le i < j \le N}^{N} \frac{m_i m_j}{\|q_i - q_j\|} \tag{35}$$

Ekuazio diferentzialak.

$$\dot{q_i} = v_i, \ i = 0, 1, \ldots N \tag{36}$$

$$\dot{v_i} = \sum_{j=0, j \neq i}^{N} \frac{G m_j}{\|q_j - q_i\|^3} (q_j - q_i) \tag{37}$$

Jakobiarra.

$$\dot{y} = f(y)$$

$$Jac = \begin{pmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \cdots & \frac{\partial f_1}{\partial q_n} & \frac{\partial f_1}{\partial v_1} & \frac{\partial f_1}{\partial v_2} & \cdots & \frac{\partial f_1}{\partial v_n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial q_1} & \frac{\partial f_n}{\partial q_2} & \cdots & \frac{\partial f_n}{\partial q_n} & \frac{\partial f_n}{\partial v_1} & \frac{\partial f_n}{\partial v_2} & \cdots & \frac{\partial f_n}{\partial v_n} \end{pmatrix} \tag{38}$$

And 2-body example,

$$J = \begin{pmatrix} 0 & I \\ A & 0 \end{pmatrix}, \quad Jac = \begin{pmatrix} & q_{1x} & q_{1y} & q_{1z} & q_{2x} & q_{2y} & q_{2z} & v_{1x} & v_{1y} & v_{1z} & v_{2x} & v_{2y} & v_{2z} \\ \frac{\partial f_1}{\partial q_{1x}} & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial f_1}{\partial q_{1y}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{\partial f_1}{\partial q_{1z}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{\partial f_1}{\partial q_{2x}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{\partial f_1}{\partial q_{2y}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{\partial f_1}{\partial q_{2z}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{\partial f_1}{\partial v_{1x}} & A & A & A & A & A & A & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial f_1}{\partial v_{1y}} & A & A & A & A & A & A & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial f_1}{\partial v_{1z}} & A & A & A & A & A & A & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial f_1}{\partial v_{2x}} & A & A & A & A & A & A & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial f_1}{\partial v_{2y}} & A & A & A & A & A & A & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial f_1}{\partial v_{2z}} & A & A & A & A & A & A & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{39}$$

Eta A matrizearen deskribapena honako notazioaren laguntzaz,

$$q_j - q_i = (x_{ji}, y_{ji}, z_{ji})$$

$$\sum = \sum_{j=0, j \neq i}^{N}$$

1. $i == j$.

$$A = \begin{pmatrix} \sum \left(\frac{3 * G m_j * x_{ji}^2}{\|q_j - q_i\|^5} \right) - \sum \left(\frac{G m_j}{\|q_j - q_i\|^3} \right) & \sum \left(\frac{3 * G m_j * (x_{ji} * y_{ji})}{\|q_j - q_i\|^5} \right) & \sum \left(\frac{3 * G m_j * (x_{ij} * z_{ji})}{\|q_j - q_i\|^5} \right) \\ \sum \left(\frac{3 * G m_j * (y_{ji} * x_{ji})}{\|q_j - q_i\|^5} \right) & \sum \left(\frac{3 * G m_j * y_{ji}^2}{\|q_j - q_i\|^5} \right) - \sum \left(\frac{G m_j}{\|q_j - q_i\|^3} \right) & \sum \left(\frac{3 * G m_j * (y_{ji} * z_{ji})}{\|q_j - q_i\|^5} \right) \\ \sum \left(\frac{3 * G m_j * (z_{ji} * x_{ji})}{\|q_j - q_i\|^5} \right) & \sum \left(\frac{3 * G m_j * (z_{ji} * y_{ji})}{\|q_j - q_i\|^5} \right) & \sum \left(\frac{3 * G m_j * z_{ji}^2}{\|q_j - q_i\|^5} \right) - \sum \left(\frac{G m_j}{\|q_j - q_i\|^3} \right) \end{pmatrix}$$

2. $i \neq j$.

$$A = \begin{pmatrix} \left(\frac{-3*Gm_j*x_{ji}^2}{\|q_j-q_i\|^5}\right)+\left(\frac{Gm_j}{\|q_j-q_i\|^3}\right) & \left(\frac{-3*Gm_j*(x_{ji}*y_{ji})}{\|q_j-q_i\|^5}\right) & \left(\frac{-3*Gm_j*(x_{ij}*z_{ji})}{\|q_j-q_i\|^5}\right) \\ \left(\frac{-3*Gm_j*(y_{ji}*x_{ji})}{\|q_j-q_i\|^5}\right) & \left(\frac{-3*Gm_j*y_{ji}^2}{\|q_j-q_i\|^5}\right)+\left(\frac{Gm_j}{\|q_j-q_i\|^3}\right) & \left(\frac{-3*Gm_j*(y_{ji}*z_{ji})}{\|q_j-q_i\|^5}\right) \\ \left(\frac{-3*Gm_j*(z_{ji}*x_{ji})}{\|q_j-q_i\|^5}\right) & \left(\frac{-3*Gm_j*(z_{ji}*y_{ji})}{\|q_j-q_i\|^5}\right) & \left(\frac{-3*Gm_j*z_{ji}^2}{\|q_j-q_i\|^5}\right)+\left(\frac{Gm_j}{\|q_j-q_i\|^3}\right) \end{pmatrix}$$

3.3.2 Hasierako balioak.

The initial conditions and parameters have been taken from JPL Solar System ephemerids DE-405. We have modified the velocities to get zero linear momentum.

Hasierako kokapenak (au)

3.3.3 Kodeak.

Mathematicako NBodyProblem.m paketean honako funtzioak garatu ditugu.

1. *Hamiltondarra: NBodyHAM.*
2. *EDA: NBodyODE.*
3. *Jakobiarra: Ez dut garatu.*

-

C-lengoaiako inplementazioa:

1. *Hamiltondarra: HamNBody().*
2. *EDA: OdeNbody().*
3. *Jakobiarra: JacNBody().*

Equations relativistic.

Ekuazio diferentzialak.

$$\dot{q_i} = v_i, \ i = 0, 1, \ldots N \tag{40}$$

$$\dot{v_i} = \sum_{j=0,j\neq i}^{N} \frac{Gm_j}{\|q_j - q_i\|^3}(q_j - q_i)\Bigg(1 - \frac{2(\beta+\gamma)}{c^2}\sum_{k=0,k\neq i}^{N}\frac{Gm_k}{\|q_k - q_i\|} - \frac{2\beta-1}{c^2}\sum_{k=0,k\neq j}^{N}\frac{Gm_k}{\|q_k - q_j\|}$$
$$+ \gamma\big(\frac{v_i}{c}\big)^2 + (1+\gamma)\big(\frac{v_j}{c}\big)^2 - \frac{2(1+\gamma)}{c^2}v_i\ v_j$$
$$- \frac{3}{2c^2}\big(\frac{(q_i - q_j)v_j}{\|q_j - q_i\|}\big)^2 + \frac{1}{2c^2}(q_j - q_i)\dot{v_i}\Bigg)$$
$$+ \frac{1}{c^2}\sum_{j=0,j\neq i}^{N}\frac{Gm_j}{\|q_j - q_i\|^3}((q_i - q_l)((2+2\gamma)v_i - (1+2\gamma)v_j))(v_i - v_j)$$
$$+ \frac{3+4\gamma}{2c^2}\sum_{j=0,j\neq i}^{N}\frac{Gm_j\dot{v_j}}{\|q_j - q_i\|} \quad (41)$$

Table 2: Konstanteak

c	299792.458 km/s	Argiaren abiadura
au	149597870.700 km	Astronomical unit
β	1.0	PPN parametroa
γ	1.0	PPN parametroa

4 Koma Higikorreko Aritmetika.

4.1 Sarrera.

Gaur-egungo konputagailuetan, IEEE-754 estandarraren araberako koma-higikorreko artimetika erabiltzen da. Koma-higikorreko aritmetikaren gaiak ez dira zenbaki errealak, koma-higikorreko zenbakiak baizik. Zenbaki errealak bit kopuru finituen bidez adierazten dira eta adierazpen finitu honek, biribiltze errorea eragiten du. Zenbakizko integrazio luzeetan biribiltze errorearen garapenak garrantzia handia du eta errore honen gaineko ahalegin berezia beharrezkoa da.

4.2 Adierazpena.

Definizioa*.Koma-higikorreko adierazpen zehatza duen zenbaki errealei koma-higikorreko zenbakiak deritzogu. Koma-higikorreko zenbakien multzoa* $\mathbb{F}$ *izendatuko dugu eta* $\phi : \mathbb{F} \to W$ *koma-higikorreko adierazpen funtzioa.*

$$\mathbb{F} \subset \mathbb{R},$$
$$\mathbb{F} = \{x \in \mathbb{R} \ \mid \ \phi(x) \in W\}. \tag{42}$$

$\mathbb{F}$ *zenbaki multzoa finitua da. Bai zenbaki positiboentzat,bai negatiboentzat, adieraz daitekeen zenbaki handienaren eta txikienaren arteko balio bakanez osatuta dago. Multzoaren kanpokaldean zenbaki hauek guztiak ditugu: batetik overflow tartean* $(-\infty, \max_{x\in\mathbb{F}_-} |x|)$ *eta* $(\max_{x\in\mathbb{F}_+} |x|, \infty+)$ *daudenak; bestetik underflow tartean* $(\min_{x\in\mathbb{F}_-} |x|, 0)$ *eta* $(0, \min_{x\in\mathbb{F}_+} |x|)$ *daudenak.*

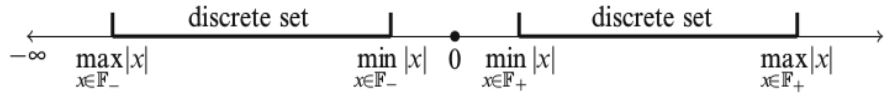


Fig. 3: Floating-point number line.

IEEE-785 estadarraren arabera, n-biteko koma-higikorrezko adierazpenak bi zati ditu,

1. *m bitez osatutako zatia , mantisa deitutakoa eta M zenbaki erreala adierazten duena. Horietako bit bat* (S_M) *zeinua adierazten du. Bestalde M mantisa modu normalizatu honetan emana da,* $\pm 1.F$ *eta zati erreala (F) bakarrik gordetzen da.*
2. *Exponentea (E),* $(n - m)$ *bitez adierazitako zenbaki osoa. Zeinuarentzat ez da bit zehatz bat erabiltzen, baizik bias izeneko balio bat gehituz adierazten dira zenbaki positiboak eta negatiboak.*

Eta beraz, oinarri bitarrean koma-higikorreko zenbaki hauek adierazten dira,

$$M \times b^E,\ b = 2.$$

Fig. 4: 32-biteko koma-higikorreko zenbakiaren adierazpena: exponentearentzat 8 bit eta mantisarentzat 24 bit (bit bat zeinuarentzat eta beste 23 bit, $1.F$ eran normalizatutako mantisarentzat) banatuta.

Bi koma-higikor zenbaki jarraien arteko diferentzia,

$$\triangle x = x_1 - x_2 = 2^{E-m+1}$$

Machine epsilon (ϵ_M), $E = 0$ deneko koma-higikorrezko bi zenbakien arteko diferentzia bezala definitzen da,

$$\epsilon_M = 2^{-m+1}$$

Azkenik, roundoff definituko dugu zenbaki erreal batek koma-higikorrean adierazpenean izan dezakeen errore erlatibo maximoa bezala,

$$u = \frac{\epsilon_M}{2} = 2^{-m}$$

IEEE-785 estandarrean honako formatu bitarrak definitzen dira:

Table 3

Mota	Tamaina	m	e	Tartea	$u = 2^{-m}$
Single	32 bit	24	8	$10^{\pm 38}$	6×10^{-8}
Double	64 bit	53	11	$10^{\pm 308}$	1×10^{-16}
Quadruple	128 bit	113	15	$10^{\pm 11356}$	1×10^{-35}

Gaur egungo konputagailuetan, Single eta Double koma-higikorreko aritmetika Hardwarez inplementatuta dago eta oso azkarrak dira. Single koma-higikorreko aritmetikak Double baino azkarragoa da: batetik garraiatu behar den bit kopuru erdia da eta bestetik, Inteleko txipen SSE modulo bereziei esker eragiketa artihmetikoak azkarragoak dira.

2008. urtean, IEEE-785 estandarrak 128 biteko koma-higikorreko aritmetika onartu zuen. Quadruple aritmetika softwarez inplementatuta dago eta horregaitik exekuzio motela da, gutxigorabehera Double aritmetika bainon 10 aldiz

garestiago. Horretarako hainbat liburutegi daude, guk GCC libquadmath liburutegia aukeratu dugu gure garapeneterako.

Bestalde, badaude doitasun arbitrariotan lan egiteko beste software batzuk ere. Doitasun altuetako kalkulu hauekin, soluzio zehatzak lortzen dira eta horrela, algoritmoen testak egiteko bidea ematen dute. Matlab eta Mathematica bezalako softwareetan doitasun handian lan egiteko aukera ematen dute eta beraz, algoritmo berri baten garapenean oso tresna erabilgarriak izan daitezke.

4.3 Biribiltze errorea.

Bi motako biribiltze errorea dugu, bata adierzapen errorea eta beste eragiketa (aritmetika) errorea.

Adierazpen errorea.

$x \in \mathbb{R}$ eta $fl : \mathbb{R} \to \mathbb{F}$ koma-higikorreko zenbaki gertuen esleitzen duen funtzioa emanik, errore absolutuaren $\triangle x$ definizioa,

$$\triangle x = fl(x) - x = \tilde{x} - x,$$

eta errore erlatiboa,

$$\delta x = \frac{\triangle x}{x} = \frac{\tilde{x} - x}{x}.$$

Aurreko bi definizioen ondorioz honako formula erabilgarria dugu,

$$\tilde{x} = x + \triangle x = x(1 + \delta x),$$

zeinek IEEE-785 estandarrak $|\delta x| < u$ dela bermatzen duen.

Eragiketen errorea.

Zenbaki errealen arteko funtsezko eragiketak $* : \mathbb{R}^2 \to \mathbb{R}$, hauek dira

$$* \in \{+, -, x, /\}.$$

Modu berean, koma-higikorrezko zenbakien arteko funtsezko eragiketak hauek dira $\circledast : \mathbb{F}^2 \to \mathbb{F}$

$$\circledast \in \{\oplus, \ominus, \otimes, \oslash\}.$$

$\tilde{x}, \tilde{y} \in \mathbb{F}$ emanik eta $z = \tilde{x} * \tilde{y}$ emaitza zehatza bada, $\tilde{z} = \tilde{x} \circledast \tilde{y}$ eragiketaren emaitzaren errore absulotua eta erlaziboak definituko ditugu,

$$\triangle z = \tilde{z} - z = (\tilde{x} \circledast \tilde{y}) - (\tilde{x} * \tilde{y})$$

$$\delta z = \frac{\triangle z}{z} == \frac{(\tilde{x} \circledast \tilde{y}) - (\tilde{x} * \tilde{y})}{(\tilde{x} * \tilde{y})}$$

$$\tilde{z} = (\tilde{x} \circledast \tilde{y}) = z + \triangle z = z(1 + \delta z),\ |\delta z| < u.$$

Adibidea. *Errore erlatiboak emitzaren digitu zuzenak neurtzen du:*

$$\delta z \approx 10^{-k} \Rightarrow \approx \ k \ digitu \ zuzen.$$

Ezabapen arazoa.

Algoritmoen kalkuluetan, doitasuna galera azkarra gerta daiteke. Horren adibidea ezabapen arazoa dugu: oso antzekoak diren bi zenbaki kentzen ditugunean gerta daitekeena.

Adibidea.

CatastrophicCancelation.nb adibidea idatzi.

```
>>  InputForm[N[Pi]]
>>  3.1415926535897793

>>  y=N[Pi]*10^(-10);
>>  InputForm[y]
>>  3.1415926535897934*10^(-10)

>>  z=1.+y;
>>  InputForm[z]
>>  1.0000000003141594

>>  InputForm[z-1.]
>>  3.141593651889707*10^(-10)
```

Errore propagazioa.

Konputazioetan eragiketa aritmetiko kopuru handia egin behar dugu emaitza lortzeko eta biribiltze errorea metatu daiteke. Batzuetan, eragiketa bakoitzeko biribiltze errorea elkar ezereztatzen da baina kasu txarrenean, biribiltze errorea metatu eta magnitude handikoa izan daiteke.

Adibidea. *Modu honetako batura batean , non $n > 2$ eta $\tilde{x}_1, \ldots, \tilde{x}_n \in \mathbb{F}$, ezin daiteke bermatu,*

$$\bigoplus_{i=1}^{n}(\tilde{x}_i) = (\sum_{i=1}^{n} \tilde{x}_i)(1+\delta), \ |\delta| < u.$$

Eta $n = 3$ deneko adibidean,

$$((\tilde{x}_1 \oplus \tilde{x}_2) \oplus \tilde{x}_3) = (\tilde{x}_1 + \tilde{x}_2)(1+\delta_1)(1+\delta_2) + \tilde{x}_3(1+\delta_2), \quad \delta_1, \delta_2 < u.$$

4.3.1 Tresnak.

Batura konpensatua.

Batura errekurtsiboetan, biribiltze errorea gutxitzeko metodoa dugu. Ideia da, bi zenbakien baturan egindako biribiltze errorearen estimazioa lortu eta estimazio hau hurrengo baturan erabiltzea.

Estimazioa nola kalkulatu azaltzeko ikus irudia (Fig. 5). Koma-higikorreko bi zenbaki baditugu, $\tilde{x}, \tilde{y} \in \mathbb{F}$ *non* $|\tilde{x}| \geq |\tilde{y}|$*, eta* $\tilde{z} = \tilde{x} \oplus \tilde{y}$ *,*

$$\tilde{e} = -\Big(\big((\tilde{x} \oplus \tilde{y}) \ominus \tilde{x}\big) \ominus \tilde{y}\Big) = (\tilde{x} \ominus \tilde{z}) \oplus \tilde{y}$$

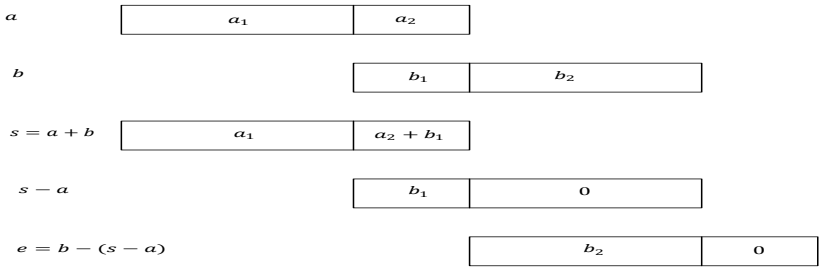


Fig. 5: Biribiltze errorea.

Lortutako errore estimazioa, koma-higikorreko aritmetikan zehazki benetazko biribiltze errorea da (frogapena Kahan),

$$\tilde{x} + \tilde{y} = \tilde{z} + \tilde{e}.$$

Batura konpensatu algoritmoa biribiltze errore honen estimazioan oinarritzen da. Honako batura $z = \sum_{i=1}^{n} \tilde{x}_i$ *kalkulatzeko , urrats bakoitzaren amaieran errore estimazioa (e) kalkulatuko dugu eta hurrengo urratsean, batugaiari gehituko diogu (*$y = \tilde{x}_i + e$*).*

$z = 0; e = 0;$
for $i \leftarrow 1$ **to** *endstep* **do**
  $x = z;$
  $y = \tilde{x}_i + e;$
  $z = x + y;$
  $e = (x - z) + y;$
end

ALGORITHM 4: Batura konpensatua.

Theorem Sterbenz

. Let x and y be floating point numbers with $\frac{y}{2} \leq x \leq 2y$. Then $x - y$ is computed exactly (assuming $x - y$ does not underflow.

FMA.

IEEE-785 (2008 revision).

$$\tilde{x} \otimes \tilde{y} \oplus \tilde{z} = (\tilde{x} \times \tilde{y} + \tilde{z})(1 + \delta),\ \delta < u$$

4.4 Laburpena.

5 Scientific Computing.

5.1 Sarrera.

Gaur-egun, orohar konputagailuak (superkonputagailu, portatila,...) paraleloak dira. 1986-2002 urteen artean, prozesadore bakarreko konputagailuen eraginkortasuna hobetuz joan zen, txipean transistore dentsitatea handitzen zen heinean baina teknologi-garapena muga fisikoetara iritsita, bide honetatik konputagailuen abiadura hobetzea ezinezkoa bilakatu zen. Horrela, 2005.urtetik aurrera fabrikatzaileek konputagailuen gaitasuna hobetzeko, txipan prozesadore bat baino gehiago erabiltzea erabaki zuten.

Moore's Law (1965). Processor speed doubles every 18 months.

Moore's Law Reinterpreter. Number of cores per chip can double every two years.

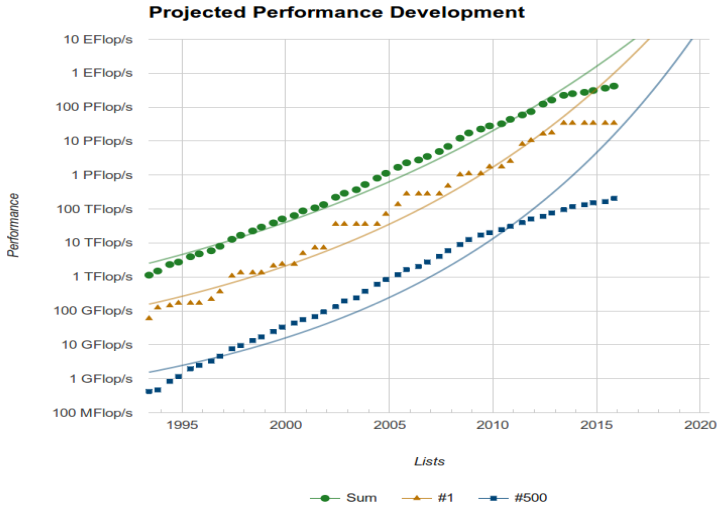


Fig. 6: www.top500.org, Top: total computing power of top 500 computers. Middle: 1 computer. Bottom: 500 computer.

Konputagailuen eredu aldaketa honen ondorioz, algoritmo azkarrak garatzeko kodearen paralelizazio gaitasunari heldu behar zaio. Beraz programazio paralelo teknikak inplementatzeko, beharrezko da prozesadore berrien hardware arkitekturak nahiz software ingurune berriak ulertzea. Gaia nahiko konplexua izanik,

ikuspegi orokorra eman ondoren, gure inplementazioan erabilitako hardware arkitektura eta software teknika zehatzak azalduko ditugu. Memoria-konpartitutako sistemak eta OpenMP programazio eredua deskribatuko dugu.

Bi dira, algoritmo azkarrak disenatzeko erronkak:

1. *Paralelizatzeko pisuko lana identifikatzea.*
2. *Memoria eta prozesadorearen arteko datu mugimendua gutxitzea.*

Bestalde, inplementazio berrien garapenean optimizatutako liburutegiak erabiltzea komeni da. Horien artean, LAPACK eta BLAS algebra linealeko liburutegiak erabilgarriak izan zaizkigu. Liburutegi hauen gaineko azalpenak emango ditugu.

5.2 Parallel Hardware.

Zein azkarrak dira konputagailuak?

Gaur egungo prozesadoreen abiadura Gigahertzioetan neurtzen da.

- *Kilo = mila* (10^3).
- *Mega = milioi* (10^6).
- *Giga = bilioi* (10^9).
- *Tera = trilioi* (10^{12}).
- *Peta* $= 10^{15}$.
- *Exa* $= 10^{18}$.

Hertzioak "makina zikloak segunduko" esan nahi du. Koma-higikorrezko eragiketa bat egiteko ($\oplus, \ominus, \otimes, \oslash$) *ziklo gutxi batzuk behar dira. Honek esan nahi du,* $1GHz$*-ko prozesagailu batek,* $> 100.000.000$ *koma-higikorrezko eragiketa segunduko egiten dituela* (> 100 *Megaflops).*

Adibidea. $C = AB$ *matrize-matrize biderketa.*

Demagun A, B *eta* C $(n \times n)$ *dimentsioko matrizeak.*

$$c_{ij} = \sum_{i,j=1}^{n} a_{ij} * b_{ji}$$

c_{ij} *gai bakoitza kalkulatzeko* n *biderketa eta* $(n-1)$ *batura egin behar ditugu.*
C *matrizeak* n^2 *osagaia ditu* $\Rightarrow O(n^3)$ *koma-higikorrezko ariketak.*
$n = 1000 \;\Rightarrow\; n^3 = 10^{12}$
> 1000 *segundu* $1GHz$ *prozesagailuan.*

Zientzia konputazioaren eraginkortasuna neurtzeko, koma-higikorrezko eragiketa kopurua (flops) erabiltzen zen. Problema handia denean, datuen mugimendua koma-higikorrezko eragiketak baino garestiagoa da, eta beraz eraginkortasuna aztertzeko koma-higikorrezko eragiketa kopurua neurtzea okerra izan daiteke. Kodearen exekuzioa azkartzeko derrigorrezkoa da konputagailuan datuen mugimendua minimizatzea.

Memoria Hierarkia.

Lehenik, konputagailuan dauden memoria mota ezberdinen hierarkia azalduko dugu.

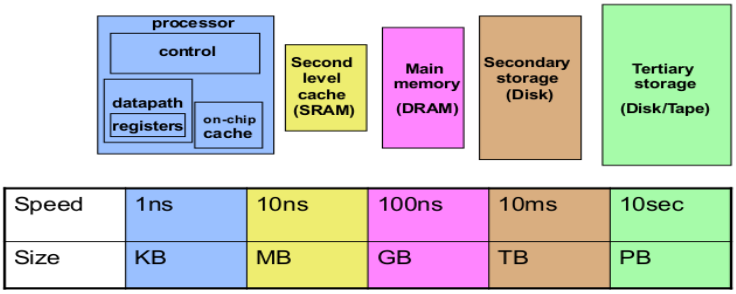


Fig. 7: Memoria hierarkia.

CPU-k koma-higikorrezko eragiketak egiten ditu: datuak erregistroetatik irakurri, eragiketa egin eta emaitza erregistroetan idazten ditu. Memoria nagusia eta erregistroen artean, 2 edo 3 mailako Cache memoria dugu: lehen Cache memoria (L1) txikiena eta azkarrena da, eta beste mailak (L2,L3,...), handiagoak eta motelagoak. Memoria nagusian, exekutatzen diren programak eta datuak gordetzen dira $(1-4$ *GB artekoa). Azkenik, disko gogorrean konputagailuko datu (argazki, bideo,...) eta erabilgarri ditugun programa guztiak gordetzen dira.*

Cache memorian, programak hurrengo unean behar dituen datuak gertu dauden printzipioaren arabera gordetzen da informazioa. Cache memoria blokeka (line) egituratuta dago eta bloke bakoitza 64 *edo* 128 *bytez (*8 *edo* 16 *double zenbaki) osatuta dago.*

Adibidea*. Badakigunez, C-lengoaian matrizeak lerroka gordetzen dira. Beheko adibidean, matrizearen lehen osagaia* $a(1,1)$ *behar dugunean, memoria nagusitik Cachera osagai honetaz gain jarraiko 16 osagaiak ekarriko dira (*$a(1,1), a(1,2), \ldots, a(1,16)$*). Honela, hurrengo* 15 *batura egiteko behar ditugun datuak Cachean eskuara izango ditugu memoria irakurketa berririk egin gabe.*

```
int n;
double a[n][n];
sum = 0;
for i ← 1 to n do
    for j ← 1 to m do
        sum+ = a(i, j);
    end
end
```

ALGORITHM 5: Main Algorithm

$$a = \begin{pmatrix} 1 & 2 & 3 & \dots & 1000 \\ 1001 & 1002 & 1003 & \dots & 2000 \\ 2001 & 2002 & 2003 & \dots & 2000 \\ \dots & \dots & \dots & \dots & \dots \\ 9001 & 9002 & 9003 & \dots & 10000 \end{pmatrix}.$$

CPUk datu bat behar duenean, memoria hierarkian zehar bilatuko du: lehenik L1 cachean, ondoren L2 cachean,...eta hauetan ez badago, memoria nagusira joko du. Memoria nagusi eta cache memoria arteko irakurketa eta idazketa guzti hauetan, informazio konsistentzia mantentzeko hainbat arau aurrera ematen dira.

Hardware.

MIMD (Multiple instruction, multiple data) sistemak, guztiz independienteak diren prozesadore multzoak osatzen dituzte. Bi dira MIMD sistema nagusiak: memoria konpartitutako eta memoria distribuitutako sistemak. Memoria konpartitutako sistemetan, prozesadore guztiek memoria osoa konpartitzen dute eta inplizituki konpartitutako datuen atzipenaren bidez komunikatzen dira. Memoria distribuitutako sistemetan aldiz, prozesadore bakoitzak bere memoria pribatua du eta explizituki bidalitako mezuen bidez komunikatzen dira.

Hirugarren hardware arkitektura ere aipatuko dugu, general purpose GPU computing (Graphical Processor Unit). Jokuen eta animazio industriak, grafiko oso azkarrak beharrak biltzatuta sortutako teknologia da. Oinarrian, imaginak oantailaratzeko prozesagailu asko paraleloan lan egiten dute. Azken hamarkadan, GPU unitate hauek zientzia konputaziora zabaldu dira.

***Shared-memory systems**. Multicore bat edo gehiagoz osatutako sistema dugu. Multicore prozesadore bakoitzak txipean CPU bat baino gehiago ditu. Normalean CPU bakoitzak L1 bere cache memoria du. Aipatzeko da, era honetako sistemetan prozesadore kopurua ezin dela nahi adina handitu eta mugatua dela (normalean* ≤ 32 *).*

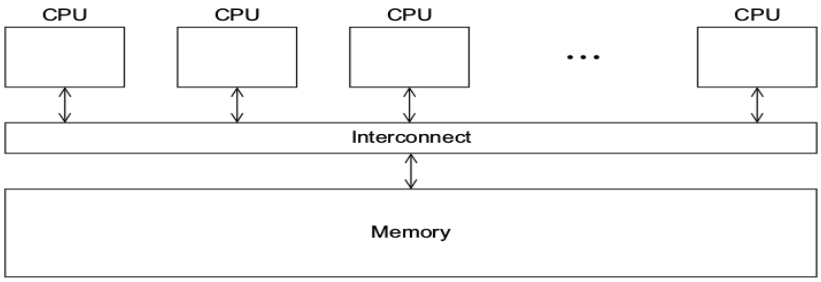


Fig. 8: Shared Memory System.

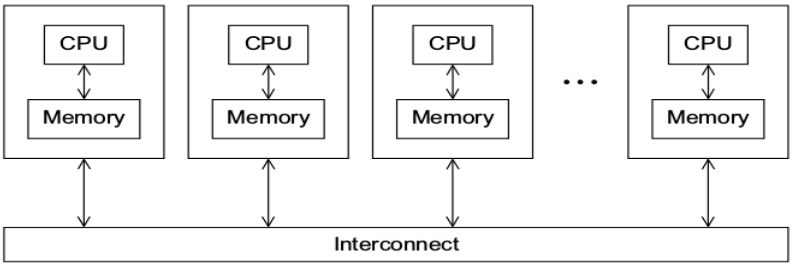


Fig. 9: Distribuited Memory System.

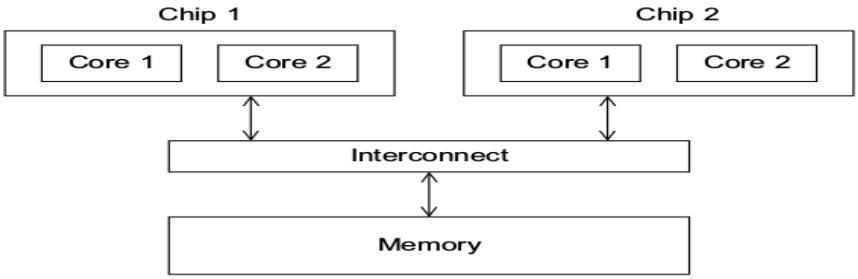


Fig. 10: Shared Memory System (UMA).

Sofwarea.

C-lengoaia programazio paraleloan erabiltzeko, lengoiaren bi extensio dira nagusienak: bata memori-distribuitutako sistemetarako diseinatuta MPI (Message-Passing Inteface) eta bestea, memoria-konpartitutako sistemetarako diseinutakoa OpenMP (Open Specifications for MultiProcessing). MPI datu moten definizio, funtzio eta makroen liburegia da. OpenMP liburutegia bat eta C kon-

piladorearen aldaketa batzuk. OpenMP erabili dugu gure inplementaziorako eta jarraian honi buruzko idei nagusienak emango ditugu.

***OpenMP**. Memoria konpartitutako programazio paraleloaren estandarra dugu. Programazioan paralelizazio kontrola, "fork-join" modeloa jarraituz egiten da.*

1. *OpenMP programen hasieran prozesu bakarra dago, hari (thread) nagusia.*
2. *FORK: hari nagusiak hari talde paraleloa sortzen du.*
3. *JOIN: hariak kode paraleloa bukatzen dutenean, behin sinkronizatuta amaitzen dute eta hari nagusiak bakarrik jarraitzen du.*

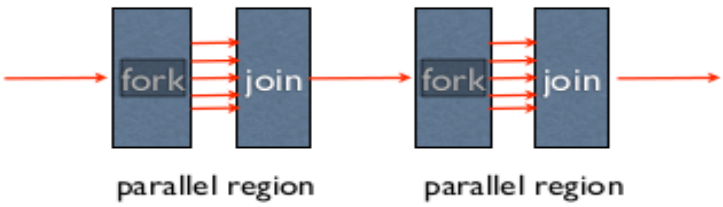


Fig. 11: Fork-Join.

Aldagai batean (threadcount) paralelizazioan zenbat hari erabili adierazten da eta ohikoa izaten da hari bat prozesadore bakoitzeko sortzea. Konpilazio direktiben bidez, paralelizazioa nola exekutatu behar den zehazten zaio.

***Adibidea**.*

```
#     pragma omp parallel for num_threads(thread_count)
      for (i = 0; i<n; i++)
      {
        ! Aginduak
      }
```

5.3 Software liburutegiak.

Matematika bi software errekurtso nagusienak aipatuko ditugu; BLAS (Basic Linear Algebra Subroutines) eta LAPACK (Linear Algebra Package). Kalitate handiko software orokorrak dira eta hauek erabiltzea abantaila asko ditu:

1. *Garapen berriak egiteko denbora aurrezten du.*
2. *Problema askotan ondo probatutako softwareak dira.*
3. *Konplexutasun handikoak dira, modu seguruan eta azkarrean exekutatzeko disenatu direlako.*

Konputagailu hardware bakoitzerako optimizatutako bertsioak daude. Inplementazioa Fortranen egina dago eta datu-motei dagokionez:

1. *S: float (32 bit).*
2. *D: double (64 bit).*
3. *C: complex.*
4. *Z: complex double.*

BLAS

.

BLAS liburutegian, bektore eta matrizeen arteko funtzio estandarrak inplementatuta daude. Hiru mailetan banatuta dago:

1. *BLAS-1: bektore-bektore eragiketak.*
   Adibidez: $y = \alpha * x + y$ *,* $2n$ *flop eta* $3n$ *irakurketa/idazketa.*
   Konputazio intentsitatea: $\frac{2n}{3n} = \frac{2}{3}$.
2. *BLAS-2: matrize-bektore eragiketak.*
   Adibidez: $y = \alpha * A * x + \beta * x$*,* $O(n^2)$ *flop eta* $O(n^2)$ *irakurketa/idazketa.*
   Konputazio intentsitatea: $\approx \frac{2n^2}{n^2} = 2$.
3. *BLAS-3: matrize-matrize eragiketak.*
   Adibidez: $C = \alpha * A * B + \beta * C$*,* $O(n^3)$ *flop eta* $O(n^2)$ *irakurketa/idazketa.*
   Konputazio intentsitatea: $\approx \frac{2n^3}{4n^2} = \frac{n}{2}$.

Azpimarratu, BLAS-1 eta BLAS-2 funtzioen konputazio intetsitatea txikia dela eta beraz, datuen komunikazioa nagusia dela. BLAS-3 aldiz, konputazio intentsitatea handiagoa da eta eazugarri honi esker, konputagailuaren konputazio gaitasuna ondo aprobetxatu ahal izango da.

Fabrikatzaile bakoitzak optimizatutako BLAS liburutegiak (AMD ACML,Intel MKL) dituzte eta beraz, multi-threaded dira. Beste aukera bat, optimizatutako BLAS instalazioa ATLAS (Automatically Tuned Linear Algebra Software) bidez egitea.

LAPACK

.

Zenbakizko algebra linealaren liburutegia da.

1. *Sistema linealak:* $AX = b$.
2. *Least Square: choose* x *to minimize* $\|Ax - b\|$.
3. *Eigenvalues.*
4. *Balio singularren deskonposaketa (SVD).*

Posible den guztietan, BLAS-3 funtzioetan oinarritzen da.

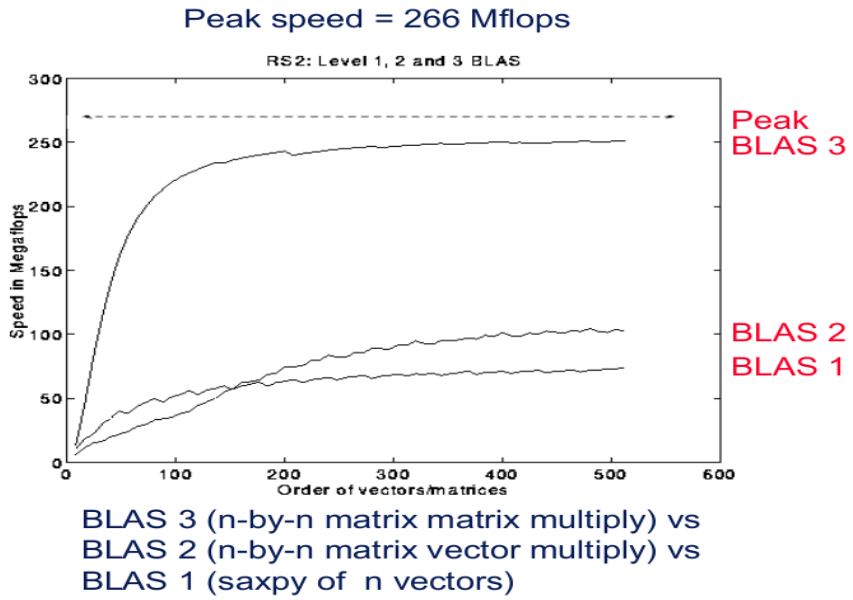


Fig. 12: BLAS speeds.

5.4 Laburpena.

Algoritmo bat inplementatzen dugunean kontutan hartu beharrekoa:

1. *Lerro edo zutabe araberako iterazioak exekuzio denboran eragin handia du.*
2. *Kodea garbia eta ulergarria mantendu behar da.*
3. *Badaude kodearen exekuzio denboraren analisia egiteko tresnak (adibidez gprof). Algoritmoaren funtzio bakoitzaren exekuzio denborari buruzko informazio erabilgarria lortuko dugu. Zenbait gauza modu sinplean azkartu daitezke baina zenbait beste gauza azkartzeko esfuertzu handia eskatu dezake.*
4. *Optimizatutako beste hainbat kode erabiltzea komenigarria da. LAPACK aljebra lineal paketea Fortran eta C-lengoaitetatik deitu daiteke. Eraberean, LAPACKek BLAS subrutinak erabiltzen ditu. Subrutinak hauek matrizen arteko biderketak, "inner product", ... BLAS konputagailu arkitektura ezberdinetarako optimizatutako bertsioak daude.*

6 IRK: Puntu-Finkoa.

6.1 Sarrera.

*Gure helburua, biribiltze errore txikia duen IRK metodoaren inplementazioa proposatzea da. Integrazioaren exekuzio denborak onargarriak izan daitezen behartuta, honako aurrebaldintza finkatu dugu: ekuazio diferentzialaren eskubi aldeko funtzioaren sarrera eta irteera argumentuak makina zenbakiak izatea, hau da, konputagailuan Hardware bidezko exekuzioa (azkarra) duen koma-higikorrezko aritmetika erabiltzea. Gaur-egun, zientzia-konputazioan double (*64 *bit) koma-higigorrezkoa aritmetikarekin lan egiten da eta beraz, praktikan erabiltzaileak ekuazio diferentziala double datu-mota honetan zehaztuko duela suposatuko dugu.*

Lehenengo Hairer-en inplementazioa aztertuko dugu. Ondoren, IRK inplementazioa hobetzeko gure proposamenak azalduko ditugu. Azkenik, zenbakizko esperimentoetan erakutsiko dugu gure inplementazioaren emaitzak.

6.2 Hairer-en inplementazioa.

Gure abiapuntua, Haier et al. [?] proposatutako inplementazio hartu dugu. Lan honetan, IRK metodo sinplektikoaren puntu-finkoaren inplementazio estandarraren biribiltze errorearen garapen okerraz jabetu ziren eta gainera, metodo sinplektiko esplizituetan agertzen ez zena. Hauen ustez, bi ziren errore honen jatorriak:

1. *Integrazioan $a_{ij}, b_i \in \mathbb{R}$ koefiziente zehatzak erabili ordez, biribildutako $\tilde{a}_{ij}, \tilde{b}_i \in \mathbb{F}$ erabiltzeak, aplikatutako IRK metodoa zehazki sinpletikoa ez izatea eragiten du.*
2. *Puntu-finkoaren geratze erizpide estandarra dela eta, urrats bakoitzean errore sistematikoa gertatzen da.*

Arrazoi hauek aztertu ondoren, honako konpobideak proposatu zituzten:

1. *Doitasun handiagoko koefizienteak erabili, hauetako bakoitza bi koma-higikorreko koefizienteen batura kontsideratuz $a_{ij} = a^*_{ij} + \tilde{a}_{ij},\ b_i = b^*_i + \tilde{b}_i$.*
2. *Iterazioak geratu, definitutako norma txikitzeari uzten dionean.*

$$\triangle^{[k]} = \max_{i=1,\ldots,s} \|Y_i^{[k]} - Y_i^{[k-1]}\|_\infty$$

$$\triangle^{[k]} = 0 \quad or \ \triangle^{[k]} \geqslant \triangle^{[k-1]}$$

Jarraian Hairer-en algoritmoa laburtuko dugu (notazioa sinplifikatze aldera $Y_{n,i}$ gaiaren ordez, Y_i adierazpenak erabiliko ditugu).

$e = 0$;
for $n \leftarrow 1$ **to** *endstep* **do**
  $k = 0$;
  Hasieratu $Y_i^{[0]}$;
  while $(\triangle^{[k]} \; != 0 \;\; and \; \triangle^{[k]} < \triangle^{[k-1]})$ **do**
    $k = k + 1$;
    $F_i^{[k]} = f(Y_i^{[k-1]})$;
    $Y_i^{[k]} = y_{n-1} + h \; (\sum_{j=1}^{s} a_{ij}^{*} F_j^{[k]}) + h \; (\sum_{j=1}^{s} \tilde{a}_{ij} F_j^{[k]})$;
    $\triangle^{[k]} = \max_{i=1,\ldots,s} \|Y_i^{[k]} - Y_i^{[k-1]}\|_{\infty}$;
  end
  $\delta_n = \left(h \; (\sum_{i=1}^{s} b_i^{*} F_i^{[k]}) + h \; (\sum_{i=1}^{s} \tilde{b}_i F_i^{[k]}) \right) + e$;
  $y_n = y_{n-1} + \delta_n$;
  $e = (y_{n-1} - y_n) + \delta_n$;
end

ALGORITHM 6: Main Algorithm

6.3 Gure inplementazioa.

IRK metodoaren puntu-finkoaren inplementazioan lau proposamen berri egin ditugu. Lehen bi proposamenak Hairer-ek bere lanean proposatutako konponbideen hobekuntzak dira. Batetik, IRK-ren birformulazio bat erabiliz, IRK metodoaren koma-higikorrezko koefizienteak sinplektizidade baldintza zehazki betetzea lortuko dugu. Bestetik, geratze erizpidean arazo batzu topatu ditugu eta arazo hauek gainditzen dituen geratze erizpide sendoagoa garatu dugu. Beste bi proposamenak dagokionez, bata batura-konpensatuari erlazionatuta dago eta bestea biribiltze errorea monitorizatzeko proposamena da.

Bestalde kapitulu honen bukaeran, batetik interpolazio bidezko atalen hasieraketa eta bestetik, Gauss-Seidel moduko puntu-finkoaren iterazioak azaldu ditugu. Bukatzeko, gure algoritmoa azalduko dugu.

6.3.1 Koefizienteak (1.proposamena).

IRK metodoa definitzen duten a_{ij}, b_i koefizienteak, biribildutako $\tilde{a}_{ij}, \tilde{b}_i \in \mathbb{F}$ ordezkatzerakoan, sinpletizide baldintza ez da beteko,

$$b_i a_{ij} + b_j a_{ji} - b_i b_j = 0, \quad 1 \leqslant i, j \leqslant s. \tag{43}$$

Arazo hau gainditzeko asmoarekin, IRK metodoa era honetan birformulatuko dugu,

$$Y_{n,i} = y_n + \sum_{j=1}^{s} \mu_{ij} L_{n,j}, \quad L_{n,i} = h b_i f(Y_{n,i}) \tag{44}$$

$$y_{n+1} = y_n + \sum_{i=1}^{s} L_{n,i} \tag{45}$$

non

$$\mu_{ij} = \frac{a_{ij}}{b_j}, \quad 1 \leq i, j \leq s.$$

eta sinplekzidade baldintza,

$$\mu_{ij} + \mu_{ji} - 1 = 0, \quad 1 \leq i, j \leq s. \tag{46}$$

Birformulazio honek formulazio estandarrarekiko duen abantaila handiena , sinplektizidade baldintzan biderketarik agertzen ez denez, baldintza hau betetzen duten $\tilde{\mu}_{ij} \in \mathbb{F}$ koefizienteak aurkitzeko bidea errezten zaigu. Zehazki era honetan finkatuko ditugu gure koefizienteak:

1. *μ_{ij} koefizienteak.*

   Batetik s-ataleko Gauss metodoetan, $\tilde{\mu}_{ii} := \frac{1}{2}$, $i = 1, \ldots, s$. Bigarrenik $\tilde{\mu}_{ij} := fl(\mu_{ij})$, $1 \leq j < i \leq s$ finkatuko dugu. Azkenik $\frac{1}{2} < |\mu_{ij}| < 2$ denez, eta Sterbenz-en Teoremaren arabera $\tilde{\mu}_{ji} := 1 - \tilde{\mu_{ij}}$ koma-higikorrezko adierazpen zehatza du. Ondorioz, simplektizitate baldintza zehazki betetzen duten koma-higikorrezko $\tilde{\mu}_{ij}$ koefizienteak lortu ditugu.

$$\begin{pmatrix} \frac{1}{2} & 1 - fl(\mu_{21}) & \ldots & 1 - fl(\mu_{s1}) \\ fl(\mu_{21}) & \frac{1}{2} & \ldots & 1 - fl(\mu_{s2}) \\ \vdots & \ddots & & \vdots \\ fl(\mu_{s1}) & fl(\mu_{s2}) & \ldots & \frac{1}{2} \end{pmatrix} \tag{47}$$

2. *b_i koefizienteak.*

   Gure inplementazioan, hbi koefizienteak erabiliko ditugu. Batetik, koefiziente hauek simetrikoak direla eta bestetik, $\sum_{i=1}^{s} hb_i = h$ berdintza bete behar dela kontutan hartuz,

$$hb_1 = hb_s := h - \sum_{i=2}^{s-1} hb_i \tag{48}$$

6.3.2 Geratze erizpidea (2.proposamena).

Ekuazio inplizituaren (44) soluzioaren hurbilpena lortzeko puntu-finkoko iterazioa era honetan definituko dugu. Iterazioaren abiapuntua $Y_i^{[0]}$ *finkatu eta* $k = 1, 2, \ldots$ *iterazioetarako* $Y_i^{[k]}$ *hurbilpenak lortu dagokigun geratze erizpidea bete arte.*

$$L_i^{[k]} = hb_i f(Y_i^{[k-1]}), \quad Y_i^{[k]} = y_n + \sum_{j=1}^{s} \mu_{ij} L_j^{[k]} \tag{49}$$

IRK metodoaren inplementazio estandarrean geratze erizpidea honakoa da,

$$\triangle^{[k]} = (Y_1^{[k]} - Y_1^{[k-1]}, \ldots, Y_s^{[k]} - Y_s^{[k-1]}) \in \mathbb{R}^{sd},$$

$$\|\triangle^{[k]}\| \leq tol \tag{50}$$

non $\|.\|$ *aurre-finkatutako bektore norma eta tol tolerantzia errorea den . Tolerantzia txikiegia aukeratzen bada, gerta daiteke tolerantzi hori ez lortzea eta infinituki iterazioak exekutatzea. Baina tolerantzia ez bada behar adina txikia aukeratzen, iterazioak puntu-finkora iritsi aurretik geratuko dira eta lortutako* $Y_i^{[k]}$ *hurbilpenaren errorea biribiltze errorea baino handiago izango da.*

Gogoratuz Hairer-ek proposatu zuen geratze erizpidea : $\triangle^{[k]} = 0$ *(puntu-finkora iritsi delako) ; edo* $\triangle^{[k]} \geqslant \triangle^{[k-1]}$ *(biribiltze errorea nagusi delako). Orokorrean, geratze erizpide honek ondo funzionatzen du baina batzuetan, iterazioak goizegi geratu direla konprobatu dugu. Gure iritziz, honen arrazoia da* $\triangle^{[k]} \geqslant \triangle^{[k-1]}$ *biribiltze errorea naguia dela adierazten duen arren, badago* $j \in \{1, \ldots, sd\}$ *osagairik,* $|\triangle_j^{[k]}| < |\triangle_j^{[k-1]}|$ *hobetzeko tartea duena.*

Gure proposamena azaldutako arazoari soluzioa emateko asmoarekin, iterazioak jarraitzea honako baldintza betetzen ez den bitartean,

$$\exists j \in \{1, \ldots, sd\} \ , \ |\triangle_j^{[1]}| > |\triangle_j^{[2]|} > \cdots > |\triangle_j^{[k]}| > 0. \tag{51}$$

6.3.3 Batura konpensatua (3.proposamena).

Integrazioaren zenbakizko soluzioa $y_n \approx y(t_n)$ $(n = 1, 2, \ldots)$ *lortzeko, urrats bakoitzean honako batura dugu,*

$$y_n = y_{n-1} + \phi(y_{n-1,h}).$$

IRK metodoetan, $\phi : \mathbb{R}^{[d+1]} \to \mathbb{R}^d$ *gehikuntza,*

$$\phi(y_{n,h}) = \sum_{i=1}^{s} L_{n,i},$$

titlenon $L_{n,i}, (i = 1, \ldots, s)$ inplizituki definitzen diren.

Urrats askotako integrazioetan, batura honetan gertatutako biribiltze erroreak doitasun galera garrantzitsua sortzen du. Beraz, zenbakizko integrazioetan oso erabilgarria zaigu batura konpensatu teknika aplikatzea biribiltze errorea gutxitzeko.

$\tilde{e}_0 = 0;$
for $n \leftarrow 1$ **to** *endstep* **do**
  $\ldots;$
  $\tilde{\delta}_n = (\sum_{i=1}^{s} L_i^{[k]}) \oplus \tilde{e}_{n-1};$
  $\tilde{y}_n = \tilde{y}_{n-1} \oplus \tilde{\delta}_n;$
  $\tilde{e}_n = (\tilde{y}_{n-1} \ominus \tilde{y}_n) \oplus \tilde{\delta}_n;$
end

ALGORITHM 7: Batura konpensatua

Badakigunez, $y_n \in \mathbb{R}^d$, $y_n = \tilde{y}_{n-1} + \tilde{\delta}_n$ soluzioa zehatza bada eta $\tilde{y}_n \in \mathbb{F}^d$, $\tilde{y}_n = \tilde{y}_{n-1} \oplus \tilde{\delta}_n$ koma-higikorreko hurbilpena izanik, lortutako errore estimazioa $\tilde{e_n}$, zehazki benetazko biribiltze errorea da,

$$y_n = \tilde{y}_n + \tilde{e}_n. \tag{52}$$

Horregaitik, IRK metodoaren inplementazioan, inplizituki $Y_{n,i}$ atalak askatzeko ekuazioetan, $\tilde{y}_n$ ordez $\tilde{y}_n \oplus \tilde{e}_n$ erabiltzea proposatzen dugu,

$$L_i^{[k]} = hb_i f(Y_i^{[k-1]}), \quad Y_i^{[k]} = \tilde{y}_n \oplus \big(\tilde{e}_n \oplus \sum_{j=1}^{s} \mu_{ij} L_j^{[k]}\big). \tag{53}$$

Aldaketa honekin, lortutako zenbakizko soluzioaren doitasuna pixka bat hobetzea espero dugu.

6.3.4 Biribiltze errorearen estimazioa (4.proposamena).

Zenbakizko integrazioaren biribiltze errorearen estimazioa, bigarren zenbakizko integrazio baten soluzioaren diferentzi gisa kalkulatuko dugu. Bigarren integrazio honetan, $Y_i^{[k]}$ atalak mantisa txikiagoko zenbakitara biribiltzen ditugu eta horrela doitasun gutxiagoko soluzioa lortzen dugu.

$r \geq 0$ zenbaki osoa, eta $x \in \mathbb{F}$ (m doitasunezko koma-higikorrezko zenbakia) izanik, honako funtzioa definituko dugu,

Function `floatR(`x`,`r`)`
  $res = (2^r x \oplus x) \ominus 2^r x$
  return *res*

ALGORITHM 8: floatR

Funtzio honek itzultzen duen balioa, $(m - r)$ doitasunezko koma-higikorrezko zenbakia da. Beste modu batera esanda, m biteko koma-higikorrezko x zenbakiaren azken r bitak zeroan jartzen dituen funtzioa.

$r < m$ zenbaki osoa finkatuta, bigarren integrazioaren puntu-finkoaren iterazioa honela kalkulatuko dugu,

$$L_i^{[k]} = hb_i f(Y_i^{[k-1]}), \quad Y_i^{[k]} = floatR\Big(\tilde{y}_n \oplus \big(\tilde{e}_n \oplus \sum_{j=1}^{s} \mu_{ij} L_j^{[k]}\big), r\Big). \tag{54}$$

Biribiltze errorearen estimazioa, zenbakizko soluzio nagusiaren $(y_n^{[main]} + e_n^{[main]})$ eta r balio txiki baterako (adibidez $r = 3$) kalkulatutako bigarren zenbakizko soluzioaren $(y_n^{[sub]} + e_n^{[sub]})$ arteko diferentzia bezala kalkulatuko dugu.

$$estimazioa_n = (y_n^{[main]} + e_n^{[main]}) - (y_n^{[sub]} + e_n^{[sub]}) \tag{55}$$

Gure algoritmoan estimazioa zuzenean lortzeko, bi integrazioak modu eraginkorrean kalkulatzen dira. Urrats bakoitzean, bi integrazioen Y_i $(i = 1, \ldots, s)$ ataletako balioak, biribiltze errorea estimazio handiegia ez den artean, antzekoak mantentzen dira. Beraz, bigarren integrazioaren iterazio kopuru txikia beharko dugu, lehen integrazioaren bukaerako Y_i $(i = 1, \ldots, s)$ atalen balioak, bigarren integrazioaren $Y_i^{[0]}$ (i = 1, . . . , s) atalen hasieratzeko erabiliz (algoritmoa zehaztu ???).

for $n \leftarrow 1$ **to** *endstep* **do**
  $Y_n^{[0]} = G(Y_{n-1}, h)$;
  $\ldots$;
  $y_{n+1} = y_n + \delta_n$;
  if $(initwithfirst)$ **then**
    $\hat{Y}_n^{[0]} = Y_n + (\hat{y}_n - y_n)$;
  else
    $\hat{Y}_n^{[0]} = G(\hat{Y}_{n-1},, h)$;
  end
  $\ldots$;
  $\hat{y}_{n+1} = \hat{y}_n + \hat{\delta}_n$;
  $estimation_n = (y_n + e_n) - (\hat{y}_n - \hat{e}_n)$;
end

ALGORITHM 9: RKG2: errore estimazioa

6.3.5 Atalen hasieraketa.

Ideia da, aurreko urratseko uneetako, $(t_{n-1} + hc_i, Y_{n-1,i})$, $i = 1, \ldots, s$ *eta* $(t_{n-1}+h, y_n)$, *balioei dagokien polinomio interpolatzailea erabiliz, urrats berriaren atalen hasieraketa* $(t_n + hc_i, Y_{n,i}^{[0]})$, $i = 1, \ldots, s$ *kalkulatzea.*

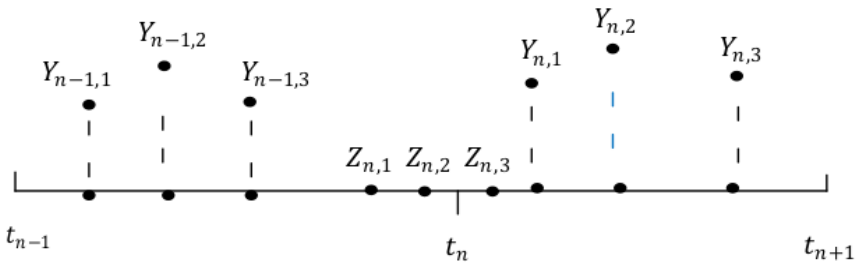


Fig. 13: Interpolazioa.

$(n-1)$ *urratseko informazioa erabiliz,*

$$Y_{n-1,i} = y_{n-1} + h\sum_{j=1}^{s} a_{ij} f(Y_{n-1,j})$$

$$y_n = y_{n-1} + h\sum_{j=1}^{s} b_j f(Y_{n-1,j})$$

$$Y_{n-1,i} = y_n + h\sum_{j=1}^{s} (a_{ij} - b_j) f(Y_{n-1,j}) \tag{56}$$

Dagokion polinomio interpolatzailea,

$$P(t) = l_1(t)Y_{n-1,1} + \cdots + l_s(t)Y_{n-1,s} + l_{s+1}(t)y_n$$

non $l_i(t)$ *Lagrangiar polinomioa dugu,*

$$l_i(t) = \prod_{l\neq i, l=1}^{s+1} \frac{(t - (t_{n-1} + hc_l))}{(c_i - c_l)}, \quad c_{s+1} = 1.$$

Eta beraz,

$$Y_{n,i} \approx Y_{n,i}^{[0]} = P(t_n + hc_i) = y_n + h\sum_{j=1}^{s} \lambda_{ij} f(Y_{n-1,j}) \tag{57}$$

Modu honetan s-ataletako IRK metodo bakoitzari dagokion λ_{ij} koefiziente interpolatzaileak lortu daitezke. Polinomio interpolatzailearen bidezko hasieraketa ona izango da, emandako urratsa ez bada oso handia eta problema stiff ez denean. Era berean aipatu nahi genuke, atal askotako metodoetan (adibidez $s = 16$) interpolaziozko koefizienteen kalkuluan kantzelazio arazoak, doitasun handian lan egitea behartzen gaituela interpolaziozko hasieraketa ona izateko.

6.3.6 Gauss-Seidel.

6.3.7 Algoritmoa.

6.4 Esperimentuak.

Biribiltze erroreari dagokionez gure inplementazioa optimotik gertu dagoela erakutsi nahi dugu. Esperimentuetan lau integrazio mota egingo ditugu:

1. *Quadruple doitasuna. Zenbakizko integrazio hau soluzio zehatzat hartuko dugu eta errore globala kalkultzeko errefentzizko soluzioa izango da.*
2. *Integrazio optimoa (ideala). Ekuazio diferentzialaren eskubi aldeko funtzioaren ebaluazioa ezik, konputazioa doitasun quadruplean egiten duen inplementazioa.*
3. *Double doitasuna.*
4. *Double doitasuna (klasikoa.)*

6.4.1 Doitasun azterketa.

Integrazio bakarra egin ordez, perturbatutako $P = 100$ hasierako balioekin zenbakizko integrazioak exekutatu ditugu eta emaitza guzti hauen batazbestekoan oinarritu gara, biribiltze errorearen azterketa egokia egiteko.

k. $(1, \ldots, P)$ integrazio bakoitzean N urrats eman baditugu, $t_i = t_0 + i * h,\ i = 1, \ldots, N$ uneetarako lortuko dugu zenbakizko soluzioa,

$$(q_i^{[k]}, p_i^{[k]}) \approx (q(t_i)^{[k]}, p(t_i)^{[k]}).$$

Sistema Hamiltondarretan energia kontserbatzen da eta definizioa hau izanik $H(q(t), p(t)) = E(t)$,

$$E_i^{[k]} = H(q_i^{[k]}, p_i^{[k]}).$$

1. *Energia errorea.*

$$\triangle E_i^{[k]} = \frac{(E_i^{[k]} - E_0^{[k]})}{E_0^{[k]}}, \quad i = 1, \ldots, N \text{ eta } k = 1, \ldots, P.$$

$$\bar{\triangle E}_i = \frac{1}{P} \sum_{k=1}^{P} \triangle E_i^{[k]}, \quad i = 1, \ldots, N.$$

$$\bar{Max} E = \max_{i=1,\ldots,N} |\bar{\triangle E}_i|$$

2. *Energia errore lokala.*
   $P = 100$ *integrazio guztietarako, bi urratsen arteko energia lokalaren batazbestekoa* (μ) *eta desbiazio estarrada* (σ).

$$\blacktriangle E_i^{[k]} = \frac{(E_i^{[k]} - E_{i-1}^{[k]})}{E_0^{[k]}}, \quad i = 1, \ldots, N \; eta \; k = 1, \ldots, P.$$

$$\bar{\mu} = \frac{1}{N \cdot P}\left(\sum_{k=1}^{P}\sum_{i=1}^{N} \blacktriangle E_i^{[k]}\right), \quad \bar{\sigma} = \sqrt{\frac{1}{N \cdot P}\left(\sum_{k=1}^{P}\sum_{i=1}^{N} (\blacktriangle E_i^{[k]} - \bar{\mu})^2\right)}$$

3. *Errore Globala* ($\bar{Ge}$).
   Doitasun laukoitzean lortutako soluzioari soluzio zehatza deituko dugu,

$$yexact_i^{[k]} = \tilde{y}_i^{[k]} = (\tilde{q}_i^{[k]}, \tilde{p}_i^{[k]})$$

   eta k. soluzioari dagokion errorea,

$$Ge_i^{[k]} = \|\tilde{q}_i^{[k]} - q_i^{[k]}\|$$

$$\bar{Ge}_i = (\frac{1}{P}\sum_{k=1}^{P} Ge_i^{[k]}) \; , \; \bar{MaxGe} = \max_{i=1,\ldots,N}(\bar{Ge}_i)$$

4. *Puntu-finkoa lortutako urratsen portzentaia* ($\bar{\triangle}0$).

   $\triangle 0^{[k]}$, *k. integrazioan puntu-finkoa lortutako urratsen portzentaia izanik,*

$$\bar{\triangle}0 = \frac{1}{p}\sum_{k=1}^{p} \triangle 0^{[k]}$$

5. *Errore estimazioa* ($\mu\bar{Q}_i$, $\sigma\bar{Q}_i$).

   Lehenengo estimazioa honela definituko dugu,

$$Est_i^{[k]} = \|q_{main_i}^{[k]} - q_{sub_i}^{[k]}\|.$$

   Errore estimazioaren kalitatea neurtzeko,

$$Q_i^{[k]} = \log_{10}\left(\frac{Est_i^{[k]}}{Ge_i^{[k]}}\right) \tag{58}$$

$$\mu\bar{Q}_i = \frac{1}{P}\sum_{k=1}^{p} Q_i^{[k]} \; , \; \sigma\bar{Q}_i = \sqrt{\frac{1}{P}\sum_{k=1}^{P}(Q_i^{[k]} - \mu\bar{Q}_i)^2}$$

6.4.2 Brouwer legea.

In order to test the randomness of numerical error (not systematic) in the integration, some researches verify that the method achieves Brouwer's law [?]. Denote by ϵ_n the error contribution over one step in the Hamiltonian H(y),

Zenbakizko integrazioaren errorea hausazkoa dela ziurtatzeko, metodoak Broweren legea [?] duela konprobatu ohi izan da. Urrats batean $H(y)$ Hamiltondarrean egindako errorea ϵ_n deituko diogu,

$$H(y_{n+1}) - H(y_n) = \epsilon_n. \tag{59}$$

Batazbestekoa zero eta bariantza, biribiltze errorearen karratuaren ($u^2 = (2^{-m})^2$) proportzionala duen hausazko aldagaia dela kontsideratuz, Brouwer legearen arabera, energia-errorea ...

and assuming it is a random variable with mean zero and variance proportional to the square of the round-off unit, Brouwer's law says that error of first integrals conservations due to round-off will grow like the square-root of time. See also Hairer [?][VIII.5]. Figure 15 plots the histogram of the Local energy error against the normal distribution $N(\mu, \delta)$.

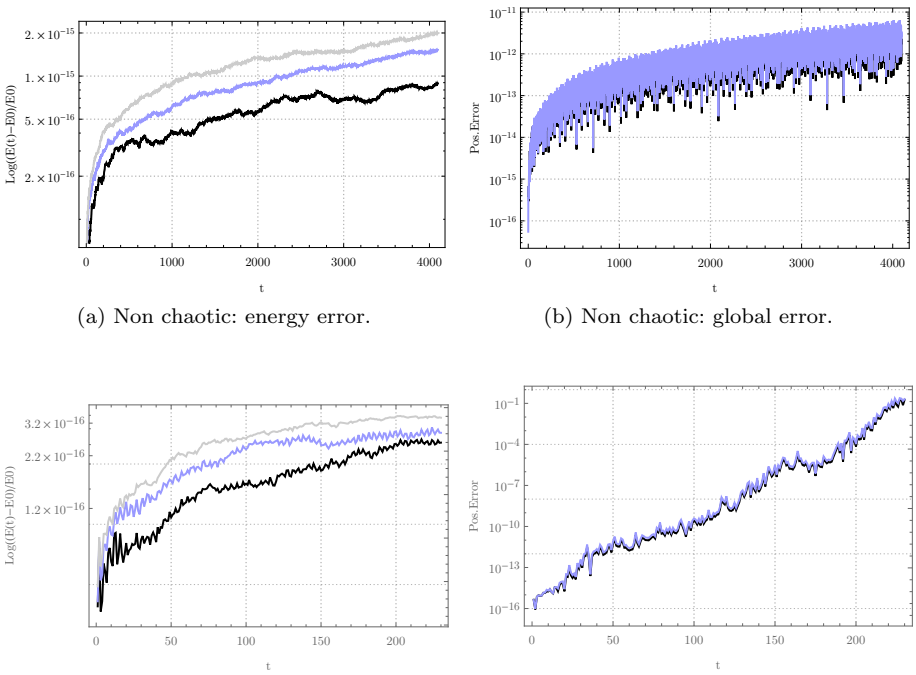
6.4.3 Pendulu bikoitza.

Table 4: Summary of Non-Chaotic case.

Arithmetic	$\bar{\triangle}0$ %	$M\bar{a}xE$	$\bar{\mu}$	$\bar{\sigma}$	$M\bar{a}xGe$
Quadruple prec	93.6	$3e10^{-19}$	$3e10^{-29}$	$2e10^{-20}$	
Ideal Integrator	98.3	$9e10^{-16}$	$2e10^{-19}$	$8e10^{-18}$	$4e10^{-12}$
Double prec	94.8	$2e10^{-15}$	$4e10^{-19}$	$8e10^{-18}$	$6e10^{-12}$

Table 5: Summary of Chaotic case.

Arithmetic	$\bar{\triangle}0$ %	$M\bar{a}xE$	$\bar{\mu}$	$\bar{\sigma}$	$M\bar{a}xGe$
Quadruple prec	93.6	$2e10^{-19}$	$7e10^{-22}$	$1e10^{-20}$	
Ideal Integrator	98.3	$3e10^{-16}$	$1e10^{-18}$	$9e10^{-18}$	0.18
Double prec	94.7	$3e10^{-16}$	$1e10^{-18}$	$1e10^{-17}$	0.23



(a) Non chaotic: energy error.

(b) Non chaotic: global error.

(c) Chaotic: energy error.

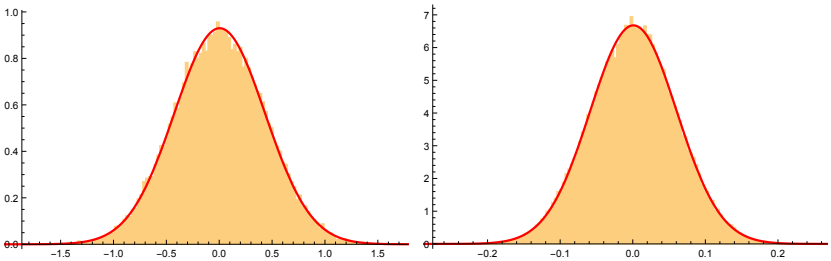
(d) Chaotic: global error.

Fig. 14: We show Non-Chaotic case (a,b) and Chaotic case (c,d). Left figure mean energy error evolution $\triangle\bar{E}_i$ and right figure mean Global error evolution $\bar{Ge}_i$ of the 100 integrations for *Ideal Integrator* (black) , *Double prec* (blue) and *Classic Implementation* (gray).

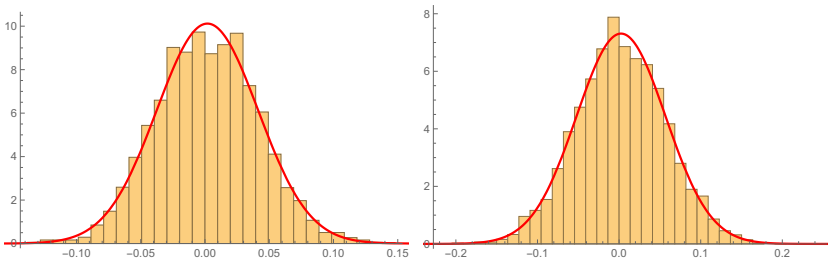
Brouwer-legea.

Biribiltze erroreaen estimazioa.

6.4.4 N-Body problema.

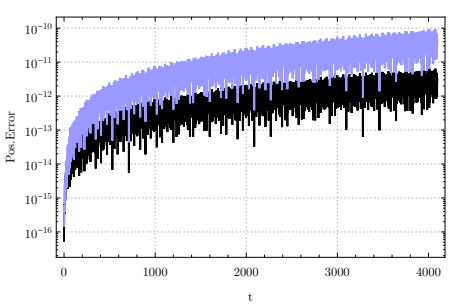


(a) Non-Chaotic: Ideal. (b) Non-Chaotic: rdigits=0.

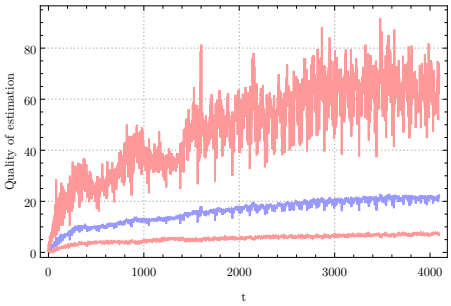


(c) Chaotic: Ideal. (d) Chaotic: rdigits=0.

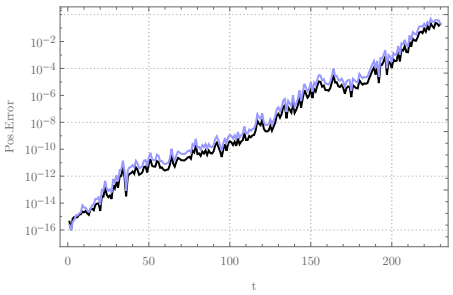
Fig. 15: Histogram of energy errors for Non-Chaotic case (a,b) and for Chaotic case (c,d).



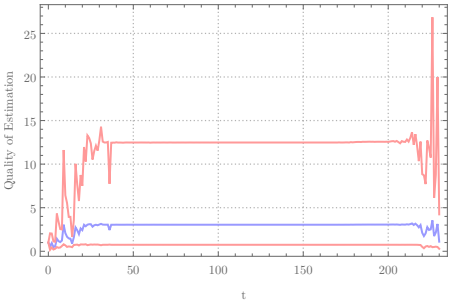
(a) Non Chaotic: estimation



(b) Non Chaotic: quality of estimation

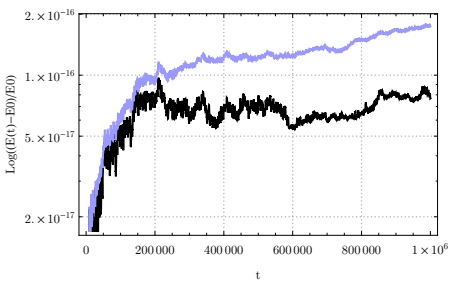


(c) Chaotic: estimation

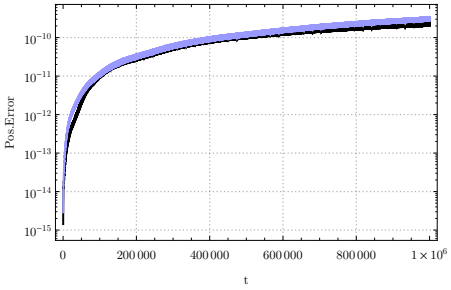


(d) Chaotic: quality of estimation

Fig. 16: Estimation round-off error. We compare evolution of our estimation error (blue) with evolution of global error (black). Estimation Quality. We show mean (blue) and standard deviation (red) of the quality according our definition of (58).

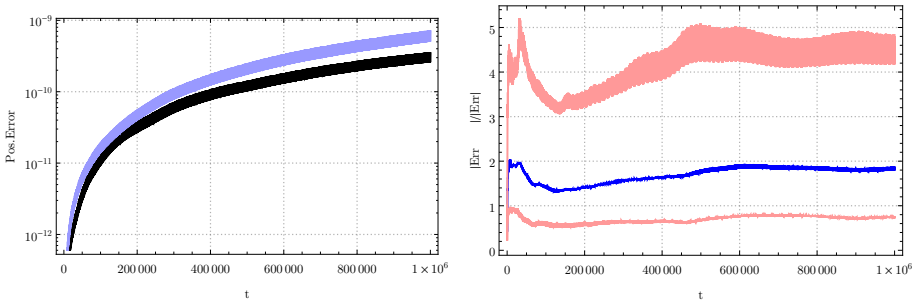


(a) Energy error.



(b) Global error.

Fig. 17: N-body: left figure mean energy error evolution $\triangle \bar{E}_i$ and right figure mean Global error evolution $\bar{Ge}_i$ of the 100 integrations for Ideal Integrator (black) and Double prec(blue).



(a) Estimation. (b) Quatlity.

Fig. 18: Left estimation round-off error, we compare evolution of our estimation error (blue) with evolution of global error (black). Right estimation Quality ,we show mean (blue) and standard deviation (red) of the quality according our definition of (58). We use rdigits1=0 and rdigits2=3.

7 Eranskinak

.

7.1 Kepler ekuazioak eta definizioak.

Kepler ekuazioak (definizioa),

$$E_0 - e\sin E_0 = n(t_0 - t_p)$$

$$E_1 - e\sin E_1 = n(t_1 - t_p)$$

non $n = \frac{2\pi}{P}$, P *periodoa den.*

Honako garapen hau egingo dugu,

$$E_1 - E_0 - e(\sin(E_1) - \sin(E_0)) = n\triangle t \quad \longrightarrow \quad \triangle E - e(\sin(E_0 + \triangle E) - \sin(E_0)) = n\triangle t$$

$$E_1 = E_0 + \triangle E$$

$$\triangle E - ce\ \sin(\triangle E - se\ (\cos(\triangle E) - 1) = n\triangle t$$

non, $ce = e\ \cos(E_0)$ *eta* $se = e\ \sin(E_0)$

Newton metodoa,

$$f(\triangle E) = \triangle E - ce\sin(\triangle E) - se(\cos(\triangle E) - 1) - n\triangle t = 0$$

$$f'(\triangle E) = 1 - ce\cos(\triangle E) + se\sin(\triangle E)$$

$$\triangle E^{[k+1]} = \triangle E^{[k]} - \frac{f(\triangle E^{[k]})}{f'(\triangle E^{[k]})} \tag{60}$$

$\triangle E^{[0]}$ ***hasierako balioa***, *finkatzea da dugun zailtasun handiena. Horretarako honako garapena egingo dugu,*

$$\triangle E - ce\ \sin(\triangle E - se\ (\cos(\triangle E) - 1) = n\triangle t$$

$$x = \triangle E - n\triangle t$$

eta beraz,

$$x - ce\sin(n\triangle t + x) - se(cos(n\triangle t + x) - 1) = 0$$

Honako baliokidetasun trigonometrikoak aplikatuz,

$$\cos(A + B) = \cos(A)\cos(B) - \sin(A)\cos(B)$$

$$\sin(A + b) = \cos(A)\sin(B) + \sin(A)\cos(B)$$

berdintza hau lortzen dugu,

$$x-(se\ \cos(n\triangle t)+ce\ \sin(n\triangle t))\cos(x)+(se\ \sin(n\triangle t)-ce\ \cos(n\triangle t))\sin(x)+se=0$$

x txikia denean honako hurbilpenak ordezkatuz,

$$x \approx \sin(x),\ \cos(x) \approx 1-\frac{x^2}{2}$$

$$(se\ \cos(n\triangle t)+ce\ \sin(n\triangle t))\frac{x^2}{2}+(1+se\ \sin(n\triangle t)-ce\cos(n\triangle t))x-(se)=0 \tag{61}$$

Goiko ekuazio hau askatuz ($Ax^2+Bx+C=0,\ \rightarrow x=\frac{-B\pm\sqrt{B^2-4AC}}{2A}$) *lortuko dugu* $\triangle E^{[0]}=x+n\triangle t$.

Koordenatu kartesiarren*, kalkulua modu ekuazio hauen bidez egingo dugu,*

$$(q_1,v_1)=(q_0,v_0)+(q_0,v_0)\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22}\end{pmatrix}$$

$$b_{11}=(C-1)\frac{a}{\|q\|}$$

$$b_{21}=\triangle t+(S-\triangle E)\frac{a^{\frac{3}{2}}}{\mu^{\frac{1}{2}}}$$

$$b_{12}=\frac{s}{\|q\|\sqrt{a}(1-ce\ C+se\ S)}$$

$$b_{22}=\frac{C-1}{1-ce\ C+se\ S}$$

Eta osagai bakoitzaren definizioa,

$$C=\cos(\triangle E),\ S=\sin(\triangle E)$$

$$ce=e\ \cos(E_0)=\|q\|\|v\|^2-1$$

$$se=e\ \sin(E_0)=\frac{(q\cdot v)}{\sqrt{\mu\ a}}$$

$$a=\frac{\mu\|q\|}{2\mu-\|q\|\|v\|^2}$$

$$n=\frac{\mu^{\frac{1}{2}}}{a^{\frac{3}{2}}}$$

Biribiltze errorea. *$\triangle E$ txikia denean,* $\cos(\triangle E) - 1$ *espresioaaren kalkuluak biribiltze errore handia eragin dezake. Hori konpontzeko baliokidetasun hau erabiliko dugu,*

$$\cos(\triangle E) - 1 = -\frac{(\sin^2(\triangle E)}{1 + \cos(\triangle E)}$$

Eta beraz, kepler-en ekuazioak hauek izango dira,

$$f(\triangle E) = \triangle E - ce\sin(\triangle E) + se\left(\frac{(\sin^2(\triangle E)}{1 + \cos(\triangle E)}\right) - n\triangle t = 0$$

Eta (q_1, v_1) *balioak kalkulatzeko,*

$$b_{11} = (C-1)\frac{a}{\|q\|}, \longrightarrow b_{11} = -\frac{(\sin^2(\triangle E)}{1 + \cos(\triangle E)}\frac{a}{\|q\|}$$