N-GORPUTZEKO PROBLEMA GRABITAZIONALAREN EBAZPENERAKO ZENBAKIZKO METODOEN AZTERKETA.

by

Mikel Antonana Otano

Informatika Fakultatea

Euskal-Herriko Unibertsitatea
Donostia

Gaien Aurkibidea

Irudien Zerrenda

Taulen Zerrenda

Atala I

Introduction

Kapitulua 1

Sarrera.

1.1 Context of the Study

Urte luzez, zientzia arlo ezberdinek N-gorputzeko problema ikertu dute. Arlo nagusien artean , astronomoek eguzki-sistemaren planeten mugimendua ulertu nahian egindako lana edo kimikariek erreakzio kimikoekin esperimentatzeko molekulen dinamikaren azterketa aipatu daitezke. Arlo bakoitzak bere ezberdintasunak (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 azterketak 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 azterketek garapen handia izan dute. Zenbakizko simulazioen laguntzaz, eguzki-sistemaren mugimenduaren funtsezko galdera batzuk ezagutu ditugu eta berriki, Karplusen taldeak 2013. kimikako Nobel saria [20] jaso du kimika konputazionalean egindako lanarengatik.

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

1. Epe motzeko eta doitasun 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 da. Adibidez, Sussmanek eta Wisdomek (1993) [24] eguzki sistemaren 100 milioiko integrazioaren bidez, planeten mugimendua kaotikoa zela baieztatu zuten. Aldi berean, paleoklimatologi-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) [2] frogatzeko planeten orbiten efemeride zehatzak beharrezkoak dira.

Konputazio-teknologi aurrerapenak handiak izan arren, eguzki-sistemaren simulazio hauek konputazionalki oso garestiak dira eta exekuzio denbora luzeak behar dituzte (Laskar [15, 2010] 18 hilabete). Azken urteotako konputagailu berrien arkitekturaren bilakaerak, algoritmo azkarren diseinua aldatu du: simulazioak azkartzeko algoritmoak 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 baino eraginkorragoak dira. Metodo inplizituek duten eraginkortasun arazo handiena ekuazio sistema ez-lineala askatzea da, eta honek metodo esplizituekiko CPU denbora gainkarga suposatzen du. Horregatik 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.

$$y_{n+1} = y_n + hf(y_n) \tag{1.1}$$

Euler metodo inplizitua.

$$y_{n+1} = y_n + hf(y_{n+1}) \tag{1.2}$$

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 eskaintzen 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.

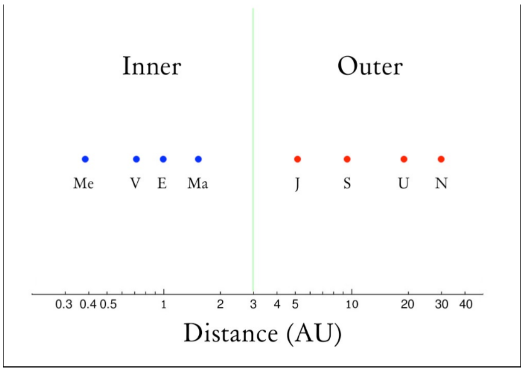
Lan honetan, metodo inplizituen artean Gauss zenbakizko integrazio metodoa aukeratu dugu. Hainbat autorek (Hairer [9][10] eta Sanz Serna[14]) metodo honen potentziala nabarmendu dute eta guk ere, iritzi berekoak gara. Laburki aipatuz, s ataletako metodo hau $2s$ ordenekoa da, sinpletikoa da, egonkortasun 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 abantailak.

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 datozela egiaztatze prozesua ez dugu landu.

Zeintzuk dira eguzki-sistemaren problemaren ezaugarri bereziak? Batetik bi gorputzen problemaren (Keplerren problema) soluzioa zehatza ezaguna da eta eguzki-sistemaren gorputzen mugimenduaren konputazioaren oinarria. Bestetik, badugu gorputz nagusi bat (eguzkia) eta honen inguruan bueltaka planetak: barne planetak, masa txikikoak eta eguzkitik gertu daudenak ; kanpo planetak, masa handikoak eta eguzkitik urrun daudenak (ikus irudia Fig.1.1). Eguzki-sistemaren egitura honi abantaila handien lortzen duen planteamendua bilatuko dugu.



(a) Planeten arteko distantziak.

(b) Planeten arteko masa konparaketa.

Irudia 1.1: Eguzki-sistema.

Konputagailuen koma-higikorreko aritmetika ondo ulertzea garrantzitsua da. Zenbaki errealen adierazpen finkoa erabiltzen denez bai zenbakiak memorian gordetzerakoan, bai hauen arteko kalkulu aritmetikoak egiterakoan, 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 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 co-*

de[1] eta *fast multipole method*[4]. 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 [10]. Autoreak IRK puntu-finkoaren inplementazio estandarrean biribiltze errorearen okerreko portaeraz 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 irizpidea,atalen hasieraketa .... 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 berridaztea eta matrizearen egitura hau aprobetxatu sistema modu eraginkorrean askatzeko.

Hirugarren urratsean, eguzki-sistemaren epe luzeko integrazioan arituko gara. Ekarpen handiena, atalen hasieraketa berri bat aplikatzea da alde Kepleriarraren fluxuan oinarrituz. IRK metodoak eskaintzen digun malgutasunari esker eta N gorputzetako problema grabitazionalaren ezaugarriez baliatuz inplementazio 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 luzera finkoko urratsa eduki behar du eta zentzu honetan, birparametrizazioa eraginkortasuna hobetzeko beste bide bat da.

1.5 laburpena

Atala II

Background

Kapitulua 2

Zenbakizko Integratzaile Sinplektikoak.

2.1 Sarrera.

2.1.1 Zenbakizko metodoak.

Hau dugu, hasierako baliodun problemaren formulazio estandarra,

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{t}, \mathbf{y}(\mathbf{t})), \quad \mathbf{y}(\mathbf{t_0}) = \mathbf{y_0}, \tag{2.1}$$

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 dugun.

Goiko ekuazio (2.1), ekuazio-sistema moduan idatzi daiteke:

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

Metodo analitikoak (funtzio ezagunen araberako soluzio zehatza) eta erdi-analitikoak, ez dira problema askoren soluzioa bilatzeko teknika egokiak. Zenbakizko metodoak, aldiz, modu errazean aplika daitezke eta horregatik, kontsideratzen da soluzio metodo nagusiena. 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. Zenbakizko soluzioa, integrazio tarte baterako kalkulatzen dugu.

Problema kaotikoak. Hasierako balio edo parametroen perturbazioekiko, diskretizazio-erroreekiko (trunkatze) edo birbitze erroreekiko oso sentikorrak diren problemei esaten zaie.

Problema stiff.

Notazioa sinplifikatzeko gure ekuazio diferentzialak *autonomoak* kontsideratuko dugu, hau da, denborarekiko independenteak.

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(\mathbf{t})), \quad \mathbf{y}(\mathbf{t_0}) = \mathbf{y_0}, \tag{2.2}$$

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,*

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

Zenbakizko diskretizazioa.

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

Orokorrean $\mathbf{y_{n+1}}$ *hurbilketa, aurreko hurbilketen* $\mathbf{y_n}, \mathbf{y_{n-1}}, \ldots, \mathbf{y_0}$ *arabera kalkulatzen da,*

$$\mathbf{y_{n+1}} = \phi(\mathbf{y_{n-1}}, \ldots, \mathbf{y_0}; h; f).$$

ϕ *metodoa,*$\mathbf{y_{n+1}}$ *balioaren menpe ez dagoenean,* $\mathbf{y_{n+1}}$ *zuzenean kalkula daiteke eta metodoa esplizitua dela esaten zaio. Aldiz,* ϕ *metodoari* $\mathbf{y_{n+1}}$ *menpe dagoenean,* $\mathbf{y_{n+1}}$ *askatzeko zeharkako bidea erabili behar da (adibidez Newton sinplifikatua edo puntu finkoaren metodoa) eta metodoari inplizitua dela esaten zaio.*

Metodoaren ordena.

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

$$\mathbf{y}_{n+1} - \mathbf{y}(t_n + h) = O(h^{p+1}) \ , \ \ h \to 0. \tag{2.3}$$

Adibidea.

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

$$\mathbf{y_{n+1}} = \phi_h(\mathbf{y_n}), \ \ non \ \ \phi_h(\mathbf{y_n}) = \mathbf{y_n} + h\mathbf{f}(\mathbf{y_n}), \ \ h = t_{n+1} - t_n. \qquad (2.4)$$

2.1.2 Sistema-Hamiltondarrak.

Ekuazio diferentzial arrunten (EDA) formulazio Hamiltondarra erabili ohi da errealitateko sistemak matematikoki adierazteko. Azpimarratu metodo sinpletikoak 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*

$$\frac{d}{dt}p_j = -\frac{\partial H(p,q)}{\partial q_j}, \ \ \frac{d}{dt}q_j = \frac{\partial H(p,q)}{\partial p_j}, \quad j = 1,\ldots,d. \qquad (2.5)$$

Edo notazio laburtua erabiliz,

$$\dot{y} = J^{-1}\nabla H(y), \ \ y = \begin{pmatrix} p \\ q \end{pmatrix}, \ \ J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}, \qquad (2.6)$$

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, Sistema-mekanikoak era honetako Hamiltondarra dute $H(p,q) = T(p) + U(q)$ *eta 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,

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

Adibidea.

Kepler problemari dagokion Hamiltondarra,

$$H(p_1,p_2,q_1,q_2) = \frac{1}{2}(p_1^2+p_2^2) - \frac{1}{\sqrt{q_1^2+q_2^2}}, \tag{2.7}$$

eta dagozkion ekuazio diferentzialak,

$$\frac{d}{dt}p_1 = -\frac{q_1}{(q_1^2+q_2^2)^{\frac{3}{2}}}, \quad \frac{d}{dt}p_2 = -\frac{q_2}{(q_1^2+q_2^2)^{\frac{3}{2}}}. \tag{2.8}$$

$$\frac{d}{dt}q_i = p_i, \quad i = 1, \dots d. \tag{2.9}$$

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.*

2.1.3 Metodo sinplektikoak.

Urrats luzera. Metodo gehienak eraginkorrak izateko, errore estimazio baten arabera integrazioan zehar urrats luzera egokitzen dute. Integratzaile sinplektikoetan, urrats luzeera finkoa erabili behar da metodoaren propietateak ez galtzeko.

2.2 Gauss metodoak.

2.2.1 Runge-Kutta metodoak.

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

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b^T} \end{array}, \qquad \begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \dots & a_{1s} \\ c_2 & a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \ddots & & \vdots \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss} \\ \hline & b_1 & b_2 & \dots & b_s \end{array} \tag{2.10}$$

Runge-kutta metodoak urrats bakarreko integratzaileak dira. Hasierako baliodun problema baten $y(t)$ soluzioaren $y_n \approx y(t_n)$ hurbilpena era honetan kalkulatzen da,

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i \, f(Y_{n,i}) \ , \tag{2.11}$$

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

$$Y_{n,i} = y_n + \ h \ \sum_{j=1}^{s} a_{ij} \, f(Y_{n,j}) \quad i = 1, \dots, s. \tag{2.12}$$

Metodo esplizituak (ERK), inplizituak (IRK) eta semi-inplizituak (DIRK).

Bi mota nagusi bereizi ditzakegu: esplizituak (ERK) eta inplizitua (IRK). ERK metodoak eraginkorrak kontsideratzen dira problema ez-stiff-etarako eta IRK metodoak ordea, problema stiff-etarako. IRK metodoen artean, semi-inplizitu metodoak (edo Runge-Kutta inplizitu diagonalak) ditugu: koefiziente matrize behe triangeluarra eta gutxienez diagonalean zero ez den koefiziente bat duena . DIRK metodoak doitasun txikia nahikoa denerako oso erabiliak dira, baina doitasun handiago behar denerako IRK metodoetan oinarritu behar dugu.

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.*

$$b_i a_{ij} + b_j a_{ji} - b_i b_j = 0, \ \ 1 \leqslant i, j \leqslant s \tag{2.13}$$

2. *Koefiziente simetrikoak.*

$$b_i = b_{(s-i+1)}, \ \ i = 1, 2, \dots \lceil s/2 \rceil \tag{2.14}$$

$$c_{(s-i+1)} = 1 - c_i, \ \ i = 1, 2, \dots \lceil s/2 \rceil \tag{2.15}$$

Runge-Kutta Inplizituaren algoritmo orokorra,

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

ALGORITHM 1: Main Algorithm

Algoritmo 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 interpolatzailearen 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 balioztapena independentea da eta modu 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 iteratibo 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, doitasun galera ekiditeko 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,

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

$$u(t_0) = y_0$$

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

eta soluzioa $y_1 = u(t_0 + h)$. 1 *zenbakia* bat *deitzen da.*

2.1 Teorema 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.*

$$a_{ij} = \int_0^{c_i} l_j(\tau)\, d\tau, \;\; b_i = \int_0^1 l_i(\tau)\, d\tau \tag{2.16}$$

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,*

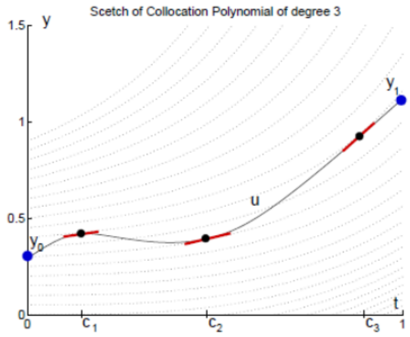
$$\frac{d^s}{dx^s}\big(x^s(x-1)^s\big),$$

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.*

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

Irudia



(a) kolokazio metodoak.

Irudia 2.1: kolokazio metodoak.

2.3 Konposizio metodoak.

2.3.1 Konposizio metodoak.

Oinarrizko metodo baten konposizioaren 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 luzera hauen* $\gamma_1 h, \gamma_2 h, \ldots, \gamma_s h$ *konposaketari dagokion konposizio metodoa,*

$$\Psi_h = \phi_{\gamma_s h} \circ \cdots \circ \phi_{\gamma_{1h}}. \tag{2.17}$$

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

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

$$\gamma_1^{p+1} + \cdots + \gamma_s^{p+1} = 1 \tag{2.18}$$

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

$$\Psi_h = \phi_{\gamma_s h} \circ \phi_{\gamma_s - 1h} \circ \cdots \circ \phi_{\gamma_{2h}} \circ \phi_{\gamma_{1h}} \tag{2.19}$$

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 sekuentzialean 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 balioztapena egin behar ditugu.

2.3.2 Gure inplementazioa.

*Gure erreferentzia, Stömer-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.*

$$
\begin{aligned}
p_{\frac{n+1}{2}} &= p_n - \frac{h}{2}\nabla_q H(p_{\frac{n+1}{2}}, q_n) \\
q_{n+1} &= q_n + \frac{h}{2}\big(\nabla_p H(p_{\frac{n+1}{2}}, q_n) + \nabla_p H(p_{\frac{n+1}{2}}, q_{n+1})\big) \qquad (2.20)
\end{aligned}
$$

$$p_{n+1} = p_{\frac{n+1}{2}} - \frac{h}{2}\nabla_q H(p_{\frac{n+1}{2}}, q_{n+1})$$

edo

$$\begin{aligned} q_{\frac{n+1}{2}} &= q_n + \frac{h}{2}\nabla_p H(p_n, q_{\frac{n+1}{2}}) \\ p_{n+1} &= p_n - \frac{h}{2}\big(\nabla_q H(p_n, q_{\frac{n+1}{2}}) + \nabla_q H(p_{n+1}, q_{\frac{n+1}{2}})\big) \\ q_{n+1} &= q_{\frac{n+1}{2}} + \frac{h}{2}\nabla_p H(p_{n+1}, q_{\frac{n+1}{2}}) \end{aligned} \tag{2.21}$$

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

$$\begin{aligned} p_{\frac{n+1}{2}} &= p_n + \frac{h}{2}f(q_n) \\ q_{n+1} &= q_n + hp_{\frac{n+1}{2}} \\ p_{n+1} &= p_{\frac{n+1}{2}} + \frac{h}{2}f(q_{n+1}) \end{aligned} \tag{2.22}$$

edo

$$\begin{aligned} q_{\frac{n+1}{2}} &= q_n + \frac{h}{2}p_n \\ p_{n+1} &= p_n - hf(q_{\frac{n+1}{2}}) \\ q_{n+1} &= q_{\frac{n+1}{2}} + \frac{h}{2}p_{n+1} \end{aligned} \tag{2.23}$$

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

$$\begin{aligned} \gamma_1 = \gamma_{35} &= 0.078795722521686419263907680 \end{aligned}$$

$$\begin{aligned} \gamma_1 = \gamma_{35} &= 0.07879572252168641926390768 \\ \gamma_2 = \gamma_{34} &= 0.31309610341510852776481247 \\ \gamma_3 = \gamma_{33} &= 0.02791838323507806610952027 \\ \gamma_4 = \gamma_{32} &= -0.22959284159390709415121340 \\ \gamma_5 = \gamma_{31} &= 0.13096206107716486317465686 \\ \gamma_6 = \gamma_{30} &= -0.26973340565451071434460973 \end{aligned}$$

$$\gamma_7 = \gamma_{29} = 0.0749733431558914356661371 1$$

$$\gamma_8 = \gamma_{28} = 0.1119934239998102048895750 8$$

$$\gamma_9 = \gamma_{27} = 0.3661334495462267511931481 2$$

$$\gamma_{10} = \gamma_{26} = -0.3991056301360358978786298 1$$

$$\gamma_{11} = \gamma_{25} = 0.1030873985274710773158027 7$$

$$\gamma_{12} = \gamma_{24} = 0.4114308739558902378207041 2$$

$$\gamma_{13} = \gamma_{23} = -0.0048663605831352617621956 6$$

$$\gamma_{14} = \gamma_{22} = -0.3920333537086399064480819 4$$

$$\gamma_{15} = \gamma_{21} = 0.0519425029624496470371829 0$$

$$\gamma_{16} = \gamma_{20} = 0.0506650907599244963358743 4$$

$$\gamma_{17} = \gamma_{19} = 0.0496743706397298790545688 0$$

$$\gamma_{18} = 0.0493177357595945379176800 1$$

2.4 Splitting metodoak.

2.4.1 Splitting metodoak.

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

$$\dot{\mathbf{y}} = \mathbf{f}^{[\mathbf{1}]}(\mathbf{y}) + \mathbf{f}^{[\mathbf{2}]}(\mathbf{y}) \tag{2.24}$$

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,*

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

$$\phi_h = \varphi_h^{[1]} \circ \varphi_h^{[2]} \tag{2.25}$$

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

$$\phi_h = \varphi_{\frac{h}{2}}^{[1]} \circ \varphi_h^{[2]} \circ \varphi_{\frac{h}{2}}^{[1]} \tag{2.26}$$

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,*

$$\begin{aligned} \dot{p} &= 0, & \dot{p} &= -U_q(q) \\ \dot{q} &= T_p(p), & \dot{q} &= 0 \end{aligned} \tag{2.27}$$

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

$$\begin{aligned} p(t) &= p_0, & p(t) &= p_0 - tU_q(q_0) \\ q(t) &= q_0 + tT_p(p_0), & q(t) &= q_0 \end{aligned} \tag{2.28}$$

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,*

$$\Psi_h = \varphi_{b_m h}^{[2]} \circ \varphi_{a_m h}^{[1]} \circ \varphi_{b_m - 1h}^{[2]} \circ \cdots \circ \varphi_{a_2 h}^{[1]} \circ \varphi_{b_1 h}^{[2]} \circ \varphi_{a_1 h}^{[1]} \tag{2.29}$$

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

ALGORITHM 3: Splitting metodoak.

2.4.2 Erreferentztia.

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 $c = 0.003396775048208601331532157783492144.$

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,

$$\varphi^I_t \approx \tilde{\varphi}^I_t = \varphi^{[U_1]}_{\frac{tb_i}{2}} \circ \varphi_t b_i^{[T_1]} \circ \varphi^{[U_1]}_{\frac{tb_i}{2}}$$

garatutako $ABAH1064$ Splitting metodoa aztertuko dugu,

$$ABAH1064 = \prod_{i=1}^{5} \varphi^K_{a_i h} \circ \tilde{\varphi}^I_{b_i h}$$

$$a_1 = 0.04731908697653382270404371796320813250988$$

$$a_2 = 0.2651105235748785159539480036185693201078$$

$$a_3 = -0.0099765228838112408432674681648123806131\,43$$
$$a_4 = -0.05992919973494155126395247987729676004016$$
$$a_5 = 0.2574761120673404534492282264603316880356$$

$$b_1 = 0.1196884624585322035312864297489892143852$$
$$b_2 = 0.3752955855379374250420128537687503199451$$
$$b_3 = -0.4684593418325993783650820409805381740605$$
$$b_4 = 0.3351397342755897010393098942949569049275$$
$$b_5 = 0.2766711191210800975049457263356834696055$$

2.5 Kepler fluxua.

Bi gorputzen problema . *Kepler problemari dagokion Hamiltondarra,*

$$H(\mathbf{p}, \mathbf{q}) = \frac{\mathbf{p^2}}{\mathbf{2m}} - \frac{\mu}{\|\mathbf{q}\|}. \tag{2.30}$$

Elkar erakartzen diren bi gorputzen mugimendua kalkulatzeko, gorputz baten kokapena koordenatu sistemaren jatorria kontsideratuko dugu. Honen arabera, $m = (1/m_1 + 1/m_2)^{-1}$ *eta* $\mu = Gm_1m2$ *definituko ditugu.*

Hamiltondarrari dagozkion bigarren ordeneko ekuazio diferentzialak,

$$\ddot{\mathbf{q}} = -\frac{k\mathbf{q}}{\|\mathbf{q}\|^\mathbf{3}}, \tag{2.31}$$

non $k = \mu/m$ *eta* $\mathbf{q} \in \mathbb{R}^3$.

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.*

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

$$\downarrow \triangle t$$

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

htb

Taula 2.1: 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

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,*

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

$$\triangle E^{[k+1]} = \triangle E^{[k]} - \frac{f(\triangle E^{[k]})}{f'(\triangle E^{[k]})} \tag{2.32}$$

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

2.6 Laburpena.

Hauek dira Eguzki sistemaren integraziorako konparatuko ditugun metodoak,

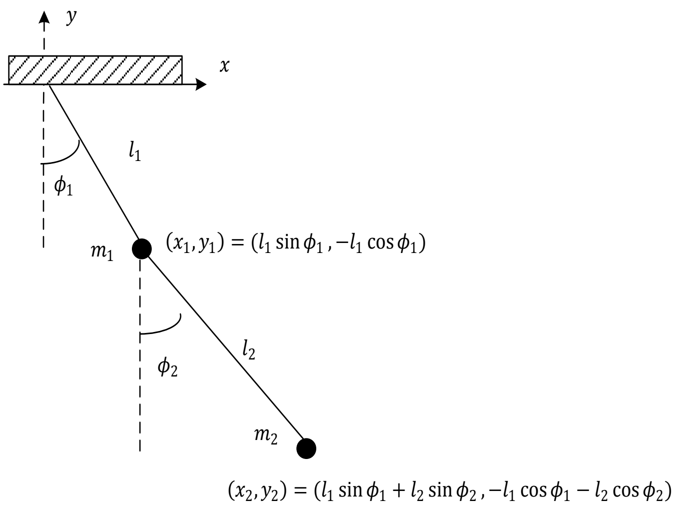
Kapitulua 3

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 *luzerako makilez (masa gabekoak kontsideratuko ditugunak) elkar lotuta. Penduluen aldagai-egoerak bi angelu (*Θ_1,Θ_2*) eta dagokion momentuak (*P_1,P_2*) dira.*



Irudia 3.1: Pendulu bikoitza.

3.2.1 Ekuazioak.

Hamiltondarra.

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

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

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

Ekuazio diferentzialak.

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

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

$$\dot{P}1 = -(aux4 + C6 * \sin(Q1)),$$

$$\dot{P}1 = (aux4 - C7 * \sin(Q2)),$$

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

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 izaera kaotikoa duen sistema ez-lineala da. Zentzu honetan bi hasierako balio ezberdin kontsideratu ditugu [22]:*

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* $\dot{q}_i = v_i$ *den.*

3.3.1 Ekuazioak.

Hamiltondarra.

$$H(q,p) = \frac{1}{2}\sum_{i=0}^{N}\frac{\|p_i\|^2}{m_i} - G\sum_{0\leq i<j\leq N}\frac{m_i m_j}{\|q_i - q_j\|} \tag{3.1}$$

Ekuazio diferentzialak.

$$\dot{q}_i = v_i, \quad i = 0,1,\ldots,N \tag{3.2}$$

$$\dot{v}_i = \sum_{j=0,j\neq i}^{N}\frac{Gm_j}{\|q_j - q_i\|^3}(q_j - q_i), \quad i = 0,1,\ldots,N \tag{3.3}$$

Ekuazio diferentzialak. *Eguzkiaren erlatibitate efektua kontutan hartzen duten ekuazio diferentzialak azalduko ditugu.*

$$\dot{q}_i = v_i, \ i = 0,1,\ldots N \tag{3.4}$$

$$\begin{aligned}
\dot{v}_i = \sum_{j=0,j\neq i}^{N}\frac{Gm_j}{\|q_j - q_i\|^3}(q_j - q_i)\Bigg(1 - \frac{2(\beta+\gamma)}{c^2}\sum_{k=0,k\neq i}^{N}\frac{Gm_k}{\|q_k - q_i\|} - \frac{2\beta-1}{c^2}\sum_{k=0,k\neq j}^{N}\frac{Gm_k}{\|q_k - q_j\|} \\
+ \gamma\big(\frac{v_i}{c}\big)^2 + (1+\gamma)\big(\frac{v_j}{c}\big)^2 - \frac{2(1+\gamma)}{c^2}v_i\, v_j \\
- \frac{3}{2c^2}\big(\frac{(q_i - q_j)v_j}{\|q_j - q_i\|}\big)^2 + \frac{1}{2c^2}(q_j - q_i)\dot{v}_i\Bigg) \\
+ \frac{1}{c^2}\sum_{j=0,j\neq i}^{N}\frac{Gm_j}{\|q_j - q_i\|^3}((q_i - q_l)((2+2\gamma)v_i - (1+2\gamma)v_j))(v_i - v_j) \\
+ \frac{3+4\gamma}{2c^2}\sum_{j=0,j\neq i}^{N}\frac{Gm_j\dot{v}_j}{\|q_j - q_i\|}
\end{aligned} \tag{3.5}$$

3.3.2 Hasierako balioak.

Eguzki eta planeten hasierako kokapenak (au) eta abiadurak (au/day), Julian data (TDB) 2440400.5 *(Ekainaren* 28,1969*) eta ICRFR2 (International Celestial Reference Frame) koordenatu sisteman [8],* Taula 3.1: Konstanteak

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

Taula 3.2: Eguzki eta planeten hasierako balioak integrazio jatorriarekiko.

Eguzkia	x, y, z	0.00450250878464055477	0.00076707642709100705	0.00026605791776697764
	v_x, v_y, v_z	-0.00000035174953607552	0.00000517762640983341	0.00000222910217891203
Mercury	x, y, z	0.36176271656028195477	-0.09078197215676599295	-0.08571497256275117236
	v_x, v_y, v_z	0.00336749397200575848	0.02489452055768343341	0.01294630040970409203
Venus	x, y, z	0.61275194083507215477	-0.34836536903362219295	-0.19527828667594382236
	v_x, v_y, v_z	0.01095206842352823448	0.01561768426786768341	0.00633110570297786403
EMB	x, y, z	0.12051741410138465477	-0.92583847476914859295	-0.40154022645315222236
	v_x, v_y, v_z	0.01681126830978379448	0.00174830923073434441	0.00075820289738312913
Mars	x, y, z	-0.11018607714879824523	-1.32759945030298299295	-0.60588914048429142236
	v_x, v_y, v_z	0.01448165305704756448	0.00024246307683646861	-0.00028152072792433877
Jupiter	x, y, z	-5.37970676855393644523	-0.83048132656339789295	-0.22482887442656542236
	v_x, v_y, v_z	0.00109201259423733748	-0.00651811661280738459	-0.00282078276229867897
Saturn	x, y, z	7.89439068290953155477	4.59647805517127300705	1.55869584283189997764
	v_x, v_y, v_z	-0.00321755651650091552	0.00433581034174662541	0.00192864631686015503
Uranus	x, y, z	-18.26540225387235944523	-1.16195541867586999295	-0.25010605772133802236
	v_x, v_y, v_z	0.00022119039101561468	-0.00376247500810884459	-0.00165101502742994997
Neptune	x, y, z	-16.05503578023336944523	-23.94219155985470899295	-9.40015796880239402236
	v_x, v_y, v_z	0.00264276984798005548	-0.00149831255054097759	-0.00067904196080291327
Pluto	x, y, z	-30.48331376718383944523	-0.87240555684104999295	8.91157617249954997764
	v_x, v_y, v_z	0.00032220737349778078	-0.00314357639364532859	-0.00107794975959731297

Masa zentrua. *Integrazio hasieran, gorputzen kokapen eta abiadurak masa zentruaren kokapen eta abiadurak zero izateko aldatzen dira.*

$$M = \sum_{i=0}^{N} Gm_i$$

Masa zentruaren kokapena (Q) eta abiadura (V),

$$Q = \frac{(\sum_{i=0}^{N} Gm_i * q_i)}{M}, \; V = \frac{(\sum_{i=0}^{N} Gm_i * v_i)}{M}$$

Eta integrazio hasierako balioak,

$$qnew_i = q_i - R, \;\; vnew_i = v_i - V, \;\; i = 0, \ldots, N.$$

3.3.3 Kodeak.

Mathematicako NBodyProblem.m paketean honako funtzioak garatu ditugu.

Taula 3.3: Ilargiaren Lurrarekiko hasierako balioak.

Ilargia	x, y, z	-0.00080817735147818490	-0.00199462998549701300	-0.00108726268307068900
	v_x, v_y, v_z	0.00060108481561422370	-0.00016744546915764980	-0.00008556214140094871

Taula 3.4: Planeten masa parametroak.

Gorputza	GM (au^3/day^3)
Eguzkia	$0.295912208285591100E-03$
Mercury	$0.491248045036476000E-10$
Venus	$0.724345233264412000E-09$
Earth	$0.888769244512563400E-09$
Mars	$0.954954869555077000E-10$
Jupiter	$0.282534584083387000E-06$
Saturn	$0.845970607324503000E-07$
Uranus	$0.129202482578296000E-07$
Neptune	$0.152435734788511000E-07$
Pluto	$0.217844105197418000E-11$
Moon	$0.109318945074237400E-10$

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

-

C-lengoaiako inplementazioa:

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

Kapitulua 4

Koma Higikorreko Aritmetika.

4.1 Sarrera.

Gaur-egungo konputagailuetan, IEEE-754 estandarraren araberako koma-higikorreko aritmetika 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.

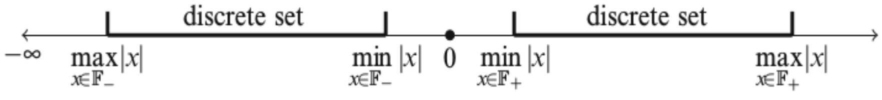
4.2 Adierazpena.

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.*

$$\mathbb{F} \subset \mathbb{R},$$
$$\mathbb{F} = \{x \in \mathbb{R} \ | \ \phi(x) \in W\}. \tag{4.1}$$

$\mathbb{F}$ zenbaki multzoa finitua da. Bai zenbaki positiboentzat,bai negatiboentzat, adieraz daitekeen zenbaki handienaren eta txikienaren arteko balio bakanez osatuta dago. Multzoaren kanpoaldean 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.

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



Irudia 4.1: Floating-point number line.

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. *Esponentea (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,

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

Irudia 4.2: 32-biteko koma-higikorreko zenbakiaren adierazpena: esponentearentzat 8 bit eta mantisarentzat 24 bit (bit bat zeinuarentzat eta beste 23 bit, $1.F$ eran normalizatutako mantisarentzat) banatuta.

Bi koma-higikor zenbaki jarraituen 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,

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

Taula 4.1

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}

IEEE-785 estandarrean honako formatu bitarrak definitzen dira:

Gaur egungo konputagailuetan, Single eta Double koma-higikorreko aritmetika Hardwarez inplementatuta dago eta oso azkarra da. 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 aritmetikoak azkarragoak dira.

2008. urtean, IEEE-785 estandarrak 128 biteko koma-higikorreko aritmetika onartu zuen. Quadruple aritmetika softwarez inplementatuta dago eta horregatik exekuzio motela da, gutxi gorabehera Double aritmetika baino 10 aldiz garestiago. Horretarako hainbat liburutegi daude, guk GCC libquadmath liburutegia aukeratu dugu gure garapeneterako.

Bestalde, badaude doitasun arbitrariotan lan egiteko beste lan-ingurune (Problem Solving Enviroment [13]) 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 adierazpen errorea eta beste eragiketa (aritmetika) errorea.

Adierazpen errorea.

$x \in \mathbb{R}$ *eta* $fl : \mathbb{R} \to \mathbb{F}$ *koma-higikorreko gertuen dagoen zenbakia esleitzen duen funtzioa emanik, errore absolutua,*

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

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 absolutua eta errore erlatiboa definituko ditugu,*

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

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

Modu berean honako erlazio dugu,

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

eta IEEE-785 estandarrak $|\delta z| < u$ dela bermatzen du.

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

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

Ezabapen arazoa.

Algoritmoen kalkuluetan, doitasuna galera azkarra gerta daiteke. Horren adibidea ezabapen arazoa dugu: oso antzekoak diren bi zenbaki arteko kendura egiten dugunean gerta daitekeena.

```
>>  InputForm[N[Pi]]
>> 3.141592653589793

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

>> z=1.+y;
>> InputForm[z]
>> 1.0000000003141594                # 16-digitu hamartar zuzenak.

>> InputForm[z-1.]
>> 3.141593651889707*10^(-10)        # 6-digitu hamartar zuzenak.
```

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 $\tilde{x}_1, \dots, \tilde{x}_n \in \mathbb{F}$, 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,

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

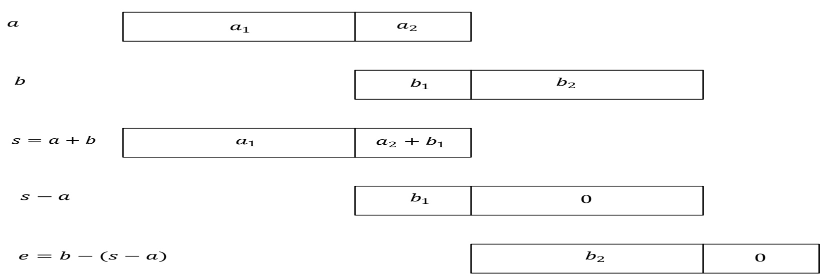
4.3.1 Tresnak.

Batura konpensatua.

Batura errekurtsiboetan, biribiltze errorea gutxitzeko metodoa dugu [12]. Ideia da, bi zenbakien baturan egindako biribiltze errorearen estimazioa lortu eta estimazio hau hurrengo baturan erabiltzea.

Estimazioa nola kalkulatu azaltzeko ikus irudia (Irudia 4.3). Koma-higikorreko bi zenbaki baditugu, $\tilde{x}, \tilde{y} \in \mathbb{F}$ *non* $|\tilde{x}| \geq |\tilde{y}|$*, eta* $\tilde{z} = \tilde{x} \oplus \tilde{y}$ *,*

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



Irudia 4.3: Biribiltze errorea.

Lortutako errore estimazioa, koma-higikorreko aritmetikan zehazki benetako biribiltze errorea da,

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

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.

Biderketaren errore estimazioa.

Konputagailu berrietan , hardware unitate bereziak bermatzen du biribiltze errore bakarra era honetako eragiketean,

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

Orduan konputagailuak FMA (fused multiply-add) eragiketak exekutatzen dituela esan ohi da. Hau honela den kasuetan, modu errezan estimatu daiteke biderketa baten biribiltze errorea,

$$\tilde{z} = \tilde{x} \otimes \tilde{y},\ \ \tilde{e} = (\tilde{x} \otimes \tilde{y}) \ominus \tilde{z}$$

Sterbenz Teorema.

Sterbenz teoremaren arabera [23], bi zenbaki elkarrekiko gertu daudenean, horien arteko kendura zehatza da.

$$x, y \in \mathbb{F},\ \frac{y}{2} \leq x \leq 2y \quad \Rightarrow \quad x - y \in \mathbb{F} \tag{4.2}$$

4.4 Laburpena.

Koma-higikorreko aritmetikan sakontzeko honako biografia azpimarratuko dugu, Overton [21], Muller [19] eta Corless [6].

Kapitulua 5

Scientific Computing.

5.1 Sarrera.

Gaur-egun, orohar konputagailuak (superkonputagailu, portatila,...) paraleloak dira. 1986-2002 urteen artean, prozesadore bakarreko konputagailuen eragin-kortasuna 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.

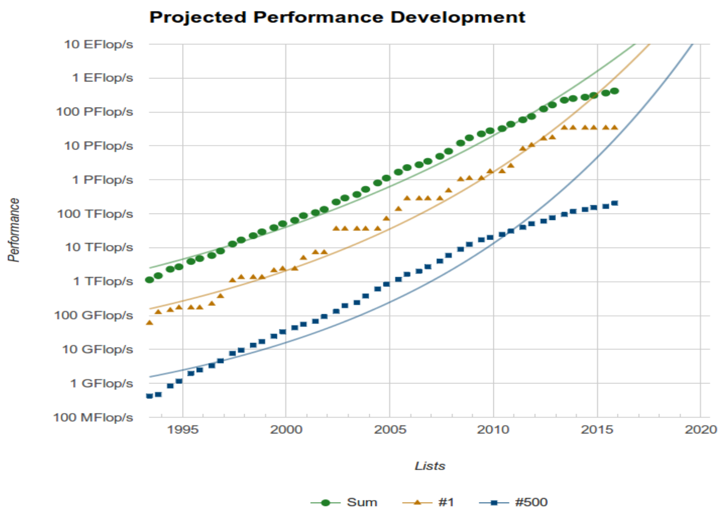
Moore's Law (1965) . *Processor speed doubles every 18 months.*

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

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.*



Irudia 5.1: www.top500.org, Top: total computing power of top 500 computers. Middle: 1 computer. Bottom: 500 computer.

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.

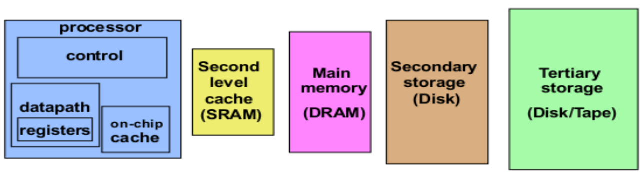
$$c_{ij} = \sum_{i,j=1}^{n} a_{ij} * b_{ji}$$

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.



Speed	1ns	10ns	100ns	10ms	10sec
Size	KB	MB	GB	TB	PB

Irudia 5.2: 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 $i \leftarrow 1$ **to** n **do**
  for $j \leftarrow 1$ **to** m **do**
    $sum+ = a(i,j)$;
  end
end

ALGORITHM 5: Main Algorithm

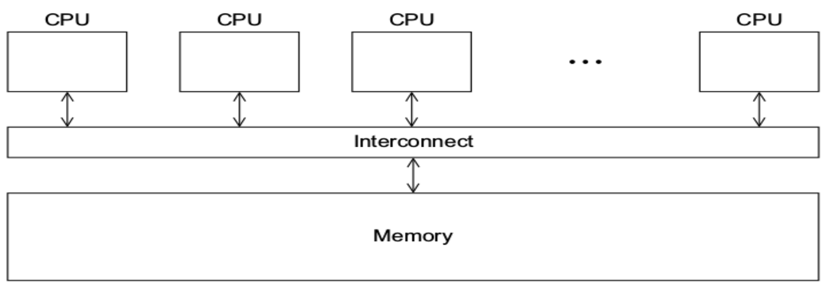
$$a = \begin{pmatrix} 1 & 2 & 3 & \ldots & 1000 \\ 1001 & 1002 & 1003 & \ldots & 2000 \\ 2001 & 2002 & 2003 & \ldots & 2000 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 9001 & 9002 & 9003 & \ldots & 10000 \end{pmatrix}.$$

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.*

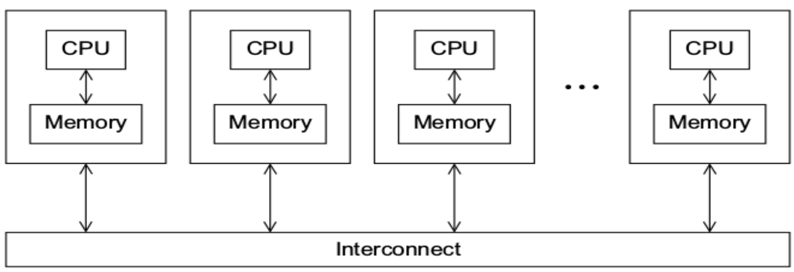
Hardware.

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.

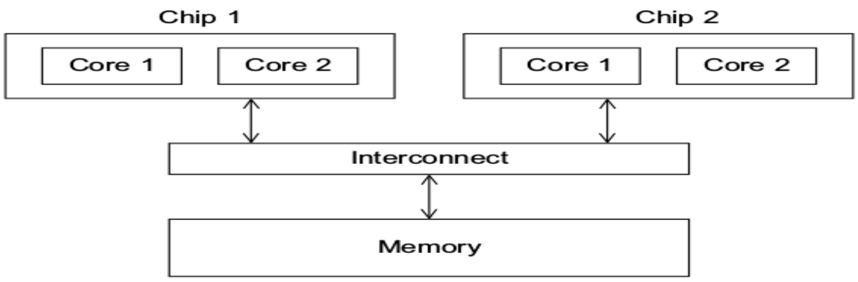


Irudia 5.3: Shared Memory System.



Irudia 5.4: Distribuited Memory System.

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).*



Irudia 5.5: Shared Memory System (UMA).

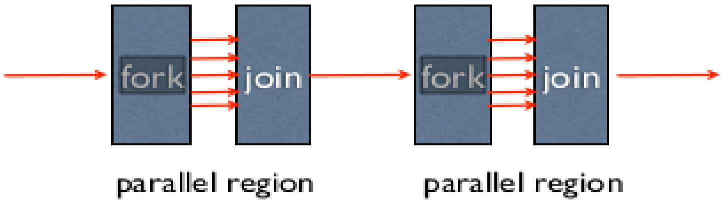
Sofwarea.

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 konpiladorearen 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.*

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.



Irudia 5.6: Fork-Join.

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.

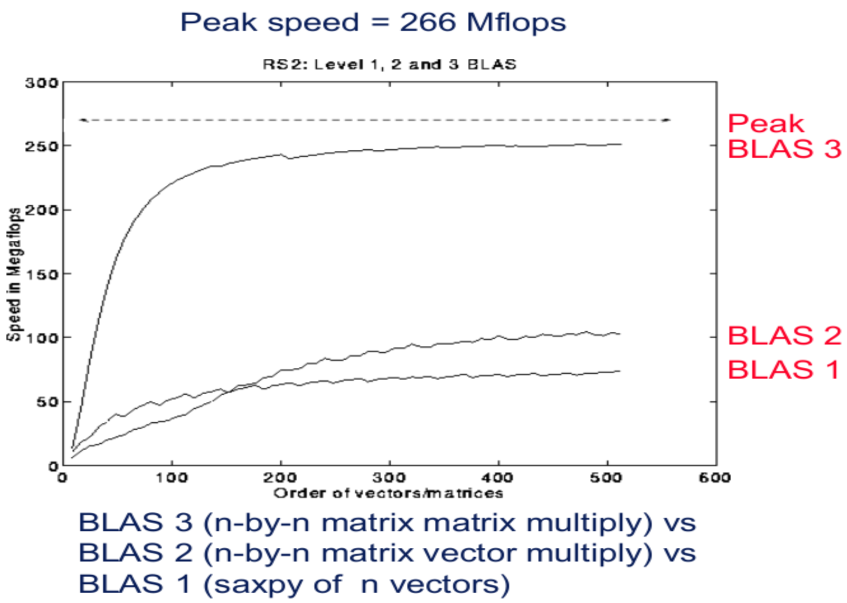
1. *Sistema linealak:* $AX = b$.

2. *Least Square: choose* x *to minimize* $\|Ax - b\|$.

3. *Eigenvalues.*

4. *Balio singularren deskonposaketa (SVD).*

Posible den guztietan, BLAS-3 funtzioetan oinarritzen da.



Irudia 5.7: BLAS speeds.

5.4 Laburpena.

Best practices for code vectorization and parallelization, and additional tips and tricks.

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. Enberean, LAPACKek BLAS subrutinak erabiltzen ditu. Subrutinak hauek matrizen arteko biderketak, "inner product", ... BLAS konputagailu arkitektura ezberdinetarako optimizatutako bertsioak daude.*

Kapitulua 6

Review.

6.1 Sarrera.

6.2 Efemerideak.

Hiru dira planetak efemerideak,

1. *Jet Propulsion Laboratory, DE (Development Ephemerides).*

   Integrazio tartea: $1550 - 2650$.

   Zenbkizko metodoa: "DIVA"(Krogh,1997).A variable order Adams method.

   Doitasuna: "QIVA"a quaruple precision of "DIVA": the equations of motion , the newtonian part is computed in quadruple precision; all of the rest are computed in double precision.

2. *Paris Observatory, INPOP (Intégrateur Númerique Planétaire de l'Observatoire de Paris).*

   Integrazio tartea: .

   Zenbakizko metodoa: The integrator is an Adams-Cowell method with fixed step-size.

   Doitasuna: the programming is done in C language, thus allowing to use the extended precision (80 bits). Integrating in quadruple precision would of course reduce the round off error in a very large amount, but the CPU time is about 15 time larger than for double precision arithmetic (or extended arithmetic) on our machine (Itanium II with Intel C++ compiler). Nevertheless, it was possible to obtain an additional order of magnitude im- provement by using a single addition in simulated quadruple precision in the corrector step with a very small over- head.

Hardware: Intel Itanium II processors.

3. *St. Petersburg, EPM (Ephemerides Planets-Moon).*

   Integrazio tartea: .

   Zenbakizko metodoa: Everhart. Implicit RK method (Gauss-Radau). (An efficient integrator that uses Gauss-Radau Spacings)

   Doitasuna: double precision. The change of ERA system integrator (19 decimal digits instead of 15 ones) with the aim to reduce the round-off error. (Extended precision).

Influence of the methods of constructing ephemerides... *It is obvious that such ephemerides in themselves must have 128 bits, that is, comprise the coefficients calculated with quadruple precision.*

6.3 Eguzki-sistemaren integrazio luzeak.

Wisdomek eta Holmanenek bere lanean [24, 1991], eguzki-sistemaren epe luzeko simulazioetarako integratzaile sinplektikoen erabilerak arrakasta izan zuen. N-planeta eta masa nagusiko gorputza bat dugula kontsideratuta, problemaren Hamiltondarra bitan banatu zuten: Hamiltondar Kleperiarra eta interakzioen Hamiltondarra. Metodo honetan, Hamiltondar bakoitzaren soluzioa tartekatuz, problema osoaren ebazpena kalkulatuko da.

Wisdom eta Holmanen inplementazioak ez ditu kolisio gertuko egoerak onartzen. Arazo hau gainditzeko, urteetan zehar algoritmo honen hainbat aldaera proposatu dira: Levinson eta Duncan-ek [17, 1994] SWIFT *softwarea garatu zuten; Duncan, Levinson eta Lee-k [7, 1998]* SYMBA *softwarea garatu zuten; Chambers-ek [5]* MERCURY *softwarea garatu zuen. Berriki, Hernandez eta Bertschinger-ek [11, 2015] garapen berri bat proposatu dute.*

Koordenatu sistema aukera ezberdinak erabili dira Hamiltondarraren banaketa lortzeko. Jacobi koordenatuak eta koordenatu Heliozentrikoak erabili ohi dira bakoitzak bere abantaila eta desbaintailekin.

Problema integratzeko oinarrizko metodoa leapfrog *metodoa dugu. Metodo hau 2 ordeneko da. Orden altuagoko splitting eskemak : McLachlan [18, 1995], Laskar eta Robutel [16, 2001], Blanes [3].*

Atala III

Core

Kapitulua 7

IRK: Puntu-Finkoa.

7.1 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 eskuin 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-higikorrezko 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 gure inplementazioaren emaitzak erakutsiko dugu.

7.2 Hairer-en inplementazioa.

Gure abiapuntua, Haierek [10] 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 irizpide estandarra dela eta, urrats bakoitzean errore sistematikoa gertatzen da.*

Arrazoi hauek aztertu ondoren, honako konponbideak 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.*

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

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

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

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

ALGORITHM 6: Main Algorithm

7.3 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 irizpidean arazo batzuk topatu ditugu eta arazo hauek gainditzen dituen geratze irizpide 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.

7.3.1 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,*

$$b_i a_{ij} + b_j a_{ji} - b_i b_j = 0, \;\; 1 \leqslant i, j \leqslant s. \tag{7.1}$$

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

$$Y_{n,i} = y_n + \sum_{j=1}^{s} \mu_{ij} L_{n,j}, \;\; L_{n,i} = h b_i f(Y_{n,i}) \tag{7.2}$$

$$y_{n+1} = y_n + \sum_{i=1}^{s} L_{n,i} \tag{7.3}$$

non

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

Eta sinplekzidade baldintza modu honetan berridatziko dugu,

$$\mu_{ij} + \mu_{ji} - 1 = 0, \quad 1 \le i, j \le s. \tag{7.4}$$

Formulazio honek estandarrarekiko duen abantaila handiena , sinplektizidade baldintzan biderketarik agertzen ez denez, baldintza hau betetzen duten $\tilde{\mu}_{ij} \in \mathbb{F}$ *koefizienteak aurkitzeko bidea errazten 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 (ikus. 4.2) 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.

$$\begin{pmatrix} \frac{1}{2} & 1 - fl(\mu_{21}) & \ldots & 1 - fl(\mu_{s1}) \\ fl(\mu_{21}) & \frac{1}{2} & \ldots & 1 - fl(\mu_{s2}) \\ \vdots & \ddots & & \vdots \\ fl(\mu_{s1}) & fl(\mu_{s2}) & \ldots & \frac{1}{2} \end{pmatrix} \tag{7.5}$$

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,

$$hb_1 = hb_s := h - \sum_{i=2}^{s-1} hb_i \tag{7.6}$$

7.3.2 Geratze irizpidea (2.proposamena).

Ekuazio inplizituaren (9.2) 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 irizpidea bete arte.

$$L_i^{[k]} = hb_i f(Y_i^{[k-1]}), \;\; Y_i^{[k]} = y_n + \sum_{j=1}^{s} \mu_{ij} L_j^{[k]} \tag{7.7}$$

IRK metodoaren inplementazio estandarrean geratze erizpidea honakoa da,

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

$$\|\triangle^{[k]}\| \leq tol \tag{7.8}$$

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

Gogoratuz Hairer-ek proposatu zuen geratze irizpidea : $\triangle^{[k]} = 0$ *(puntu-finkora iritsi delako) ; edo* $\triangle^{[k]} \geqslant \triangle^{[k-1]}$ *(biribiltze errorea nagusi delako). Orokorrean, geratze irizpide honek ondo funtzionatzen du baina batzuetan, iterazioak goizegi geratu direla konprobatu dugu. Gure iritziz, honen arrazoia da* $\triangle^{[k]} \geqslant \triangle^{[k-1]}$ *biribiltze errorea nagusia 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,

$$\exists j \in \{1, \ldots, sd\} \; , \; |\triangle_j^{[1]}| > |\triangle_j^{[2]|} > \cdots > |\triangle_j^{[k]}| > 0. \tag{7.9}$$

7.3.3 Batura konpentsatua (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,*

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

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

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

$\tilde{e}_0 = 0;$

for $n \leftarrow 1$ **to** $endstep$ **do**

$\quad \ldots;$

$\quad \tilde{\delta}_n = (\sum_{i=1}^{s} L_i^{[k]}) \oplus \tilde{e}_{n-1};$

$\quad \tilde{y}_n = \tilde{y}_{n-1} \oplus \tilde{\delta}_n;$

$\quad \tilde{e}_n = (\tilde{y}_{n-1} \ominus \tilde{y}_n) \oplus \tilde{\delta}_n;$

end

ALGORITHM 7: Batura konpensatua

$y_n \in \mathbb{R}^d$, $y_n = \tilde{y}_{n-1} + \tilde{\delta}_n$ soluzioa zehatza izanik 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 benetako biribiltze errorea da,

$$y_n = \tilde{y}_n + \tilde{e}_n. \tag{7.10}$$

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

$$L_i^{[k]} = hb_i f(Y_i^{[k-1]}), \;\; Y_i^{[k]} = \tilde{y}_n \oplus \left(\tilde{e}_n \oplus \sum_{j=1}^{s} \mu_{ij} L_j^{[k]}\right). \tag{7.11}$$

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

7.3.4 Biribiltze errorearen estimazioa (4.proposamena).

Zenbakizko integrazioaren biribiltze errorearen estimazioa, bigarren zenbakizko integrazio baten soluzioaren diferentzia 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*)

$\quad res = (2^r x \oplus x) \ominus 2^r x$

$\quad$ **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,

$$L_i^{[k]} = hb_i f(Y_i^{[k-1]}), \;\; Y_i^{[k]} = floatR\Big(\tilde{y}_n \oplus \big(\tilde{e}_n \oplus \sum_{j=1}^{s} \mu_{ij} L_j^{[k]}\big), r\Big). \qquad (7.12)$$

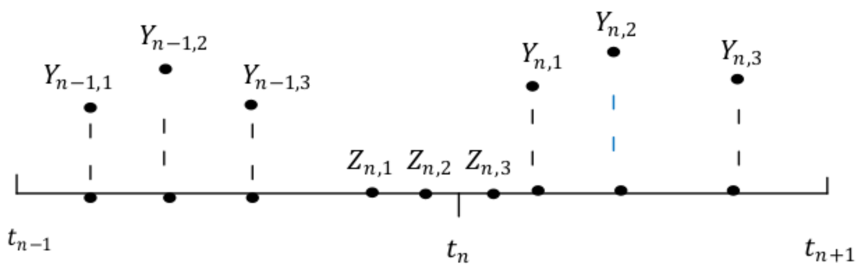
Biribiltze errorearen estimazioa, zenbakizko soluzio nagusiaren $(y_n^{[main]}+e_n^{[main]})$ eta r balio txiki baterako (adibidez $r=3$) kalkulatutako bigarren zenbakizko soluzioaren $(y_n^{[sub]}+e_n^{[sub]})$ arteko diferentzia bezala kalkulatuko dugu.

$$estimazioa_n = (y_n^{[main]}+e_n^{[main]}) - (y_n^{[sub]}+e_n^{[sub]}) \qquad (7.13)$$

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 ???).

7.3.5 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.



Irudia 7.1: Interpolazioa.

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

$(n-1)$ urratseko informazioa erabiliz,

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

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

$$Y_{n-1,i} = y_n + h \sum_{j=1}^{s} (a_{ij} - b_j) f(Y_{n-1,j}) \tag{7.14}$$

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,

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

Eta beraz,

$$Y_{n,i} \approx Y_{n,i}^{[0]} = P(t_n + hc_i) = y_n + h \sum_{j=1}^{s} \lambda_{ij} f(Y_{n-1,j}) \tag{7.15}$$

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 ezabapen arazoak, doitasun handian lan egitea behartzen gaituela interpolaziozko hasieraketa ona izateko.

7.3.6 Gauss-Seidel.

7.3.7 Algoritmoa.

Formulazio berriari dagokion algoritmo orokorra,

$e = 0$;
for $n \leftarrow 1$ **to** $endstep$ **do**
  Hasieratu $Y_{i,n}^{[0]}$, $i = 1, \ldots, s$;
  while *(konbergentzia lortu)* **do**
    $L_{n,i} = hb_i f(Y_{n,i})$, $i = 1, \ldots, s$;
    $Y_{n,i} = y_{n-1} + \sum_{j=1}^{s} \mu_{ij} L_{n,j}$, $i = 1, \ldots, s$;
  end
  $\delta_n = \sum_{i=1}^{s} L_{n,i} + e$;
  $y_n = y_{n-1} + \delta_n$;
  $e = (y_{n-1} - y_n) + \delta_n$;
end

ALGORITHM 10: Main Algorithm

Eta puntu-finkoa erabiliz,

$e = 0$;
for $n \leftarrow 1$ **to** $endstep$ **do**
  $k = 0$;
  Hasieratu $Y_i^{[0]}$;
  while *(konbergentzia lortu)* **do**
    $k = k + 1$;
    $L_i^{[k]} = hb_i f(Y_i^{[k-1]})$;
    $Y_i^{[k]} = y_{n-1} + \ \left(e + \sum_{j=1}^{s} \mu_{ij} L_j^{[k]}\right)$;
  end
  $\delta_n = \sum_{i=1}^{s} L_i^{[k]} + e$;
  $y_n = y_{n-1} + \delta_n$;
  $e = (y_{n-1} - y_n) + \delta_n$;
end

ALGORITHM 11: Main Algorithm

7.4 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 zehatza kontsideratuko dugu eta errore globala kalkulatzeko erreferentziazko soluzioa izango da.*
2. *Integrazio optimoa (ideala). Ekuazio diferentzialaren eskuin aldeko funtzioaren ebaluazioa ezik, konputazioa doitasun quadruplean egiten duen inplementazioa.*
3. *Double doitasuna.*
4. *Double doitasuna (klasikoa.)*

7.4.1 Doitasun azterketa.

Integrazio bakarra egin ordez, perturbatutako $P = 100$ *hasierako balioekin zenbakizko integrazioak exekutatu ditugu eta emaitza guzti hauen batezbestekoan oi-*

narritu 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.*

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

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

$$\sigma_i = \sqrt{\frac{1}{P}\sum_{k=1}^{P}(\triangle E_i^{[k]})^2 - (\bar{\triangle E_i})^2},\ \ i = 1,\ldots,N.$$

$$M\bar{a}xE = \max_{i=1,\ldots,N} |\bar{\triangle E_i}|$$

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

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

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

3. *Errore Globala ($\bar{G}e$).*
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,

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

$$\bar{G}e_i = (\frac{1}{P}\sum_{k=1}^{P} Ge_i^{[k]})\ ,\ Ma\bar{x}Ge = \max_{i=1,\ldots,N}(\bar{G}e_i)$$

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,

$$Q_i^{[k]} = \log_{10}\left(\frac{Est_i^{[k]}}{Ge_i^{[k]}}\right) \qquad (7.16)$$

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

7.4.2 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,

$$H(y_{n+1}) - H(y_n) = \epsilon_n. \tag{7.17}$$

Batezbestekoa 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 [9][VIII.5]. Figure 7.3 plots the histogram of the Local energy error against the normal distribution $N(\mu, \delta)$.

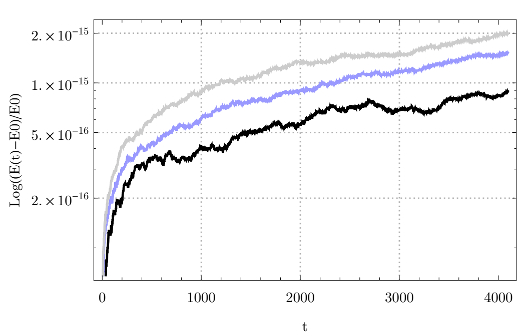
7.4.3 Pendulu bikoitza.

Taula 7.1: Summary of Non-Chaotic case.

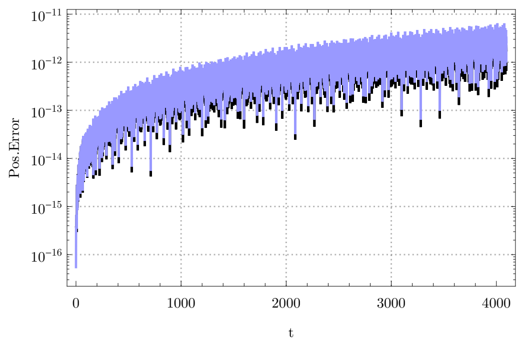
Arithmetic	$\bar{\triangle}0$ %	$M\bar{a}xE$	$\bar{\mu}$	$\bar{\sigma}$	$M\bar{a}xGe$
Quadruple prec	93.6	$3e10^{-19}$	$3e10^{-29}$	$2e10^{-20}$	
Ideal Integrator	98.3	$9e10^{-16}$	$2e10^{-19}$	$8e10^{-18}$	$4e10^{-12}$
Double prec	94.8	$2e10^{-15}$	$4e10^{-19}$	$8e10^{-18}$	$6e10^{-12}$

Taula 7.2: Summary of Chaotic case.

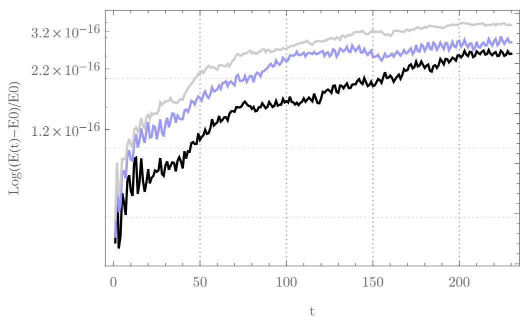
Arithmetic	$\bar{\triangle}0$ %	$M\bar{a}xE$	$\bar{\mu}$	$\bar{\sigma}$	$M\bar{a}xGe$
Quadruple prec	93.6	$2e10^{-19}$	$7e10^{-22}$	$1e10^{-20}$	
Ideal Integrator	98.3	$3e10^{-16}$	$1e10^{-18}$	$9e10^{-18}$	0.18
Double prec	94.7	$3e10^{-16}$	$1e10^{-18}$	$1e10^{-17}$	0.23



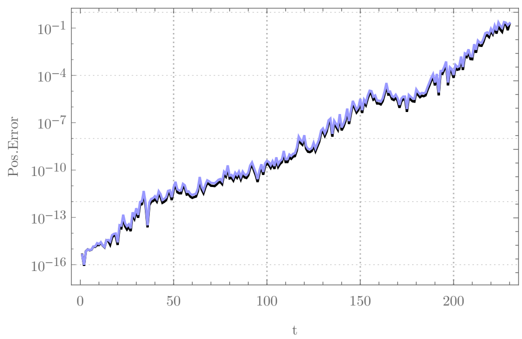
(a) Non chaotic: energy error.



(b) Non chaotic: global error.



(c) Chaotic: energy error.



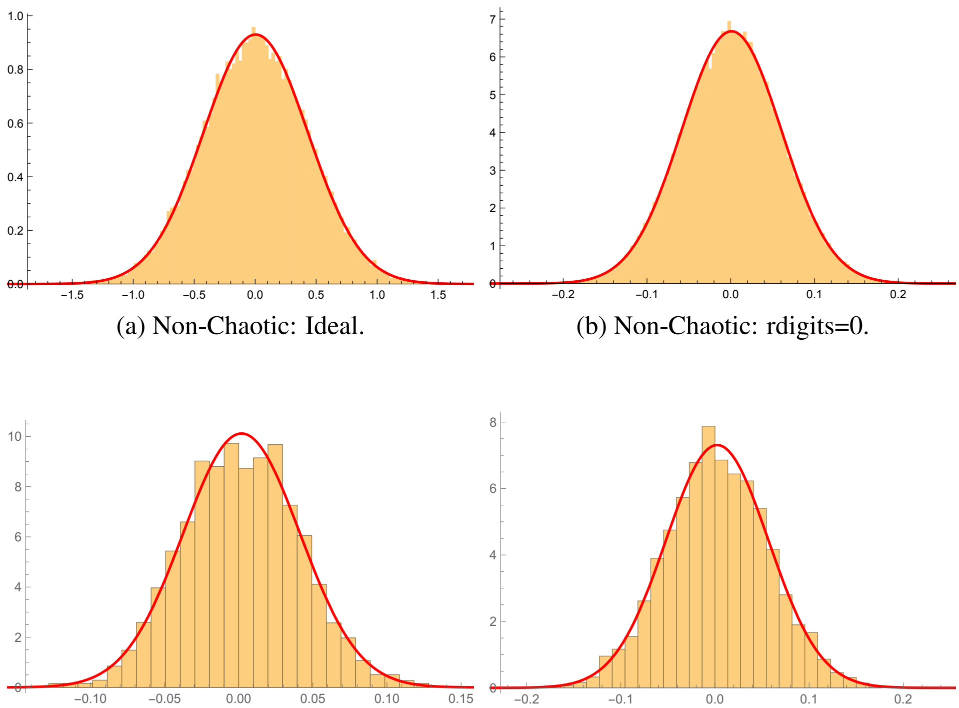
(d) Chaotic: global error.

Irudia 7.2: 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.

7.4.4 N-Body problema.



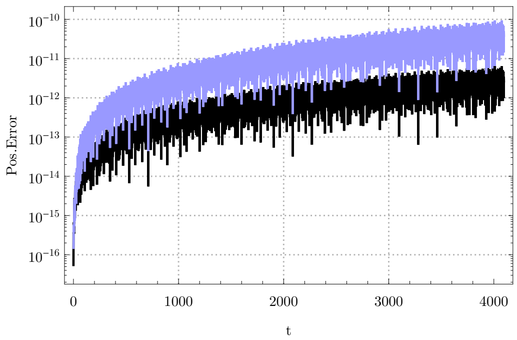
(a) Non-Chaotic: Ideal.

(b) Non-Chaotic: rdigits=0.

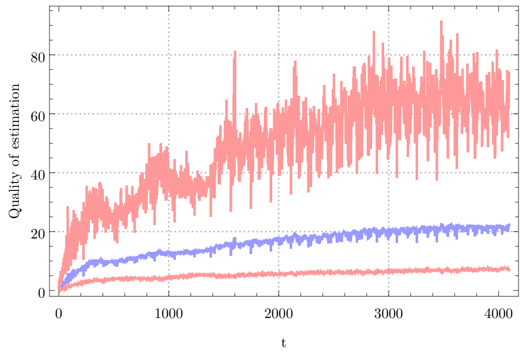
(c) Chaotic: Ideal.

(d) Chaotic: rdigits=0.

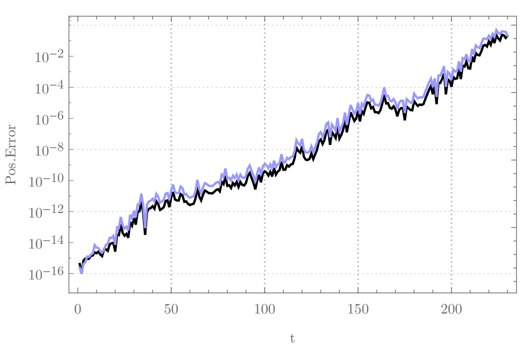
Irudia 7.3: Histogram of energy errors for Non-Chaotic case (a,b) and for Chaotic case (c,d).



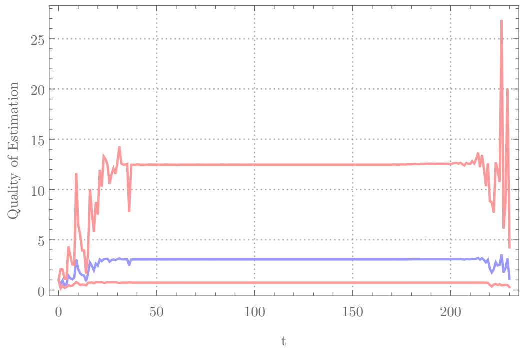
(a) Non Chaotic: estimation



(b) Non Chaotic: quality of estimation

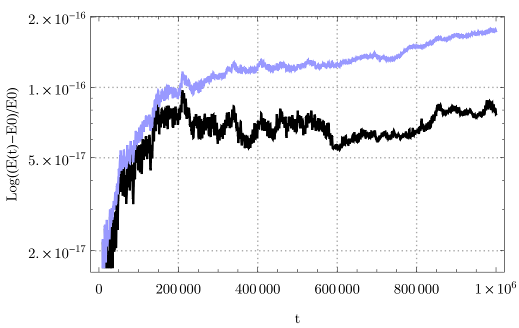


(c) Chaotic: estimation

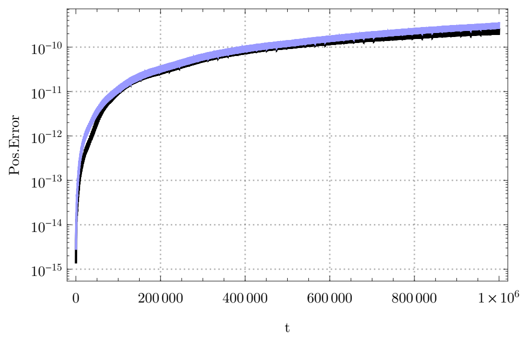


(d) Chaotic: quality of estimation

Irudia 7.4: 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 (7.16).

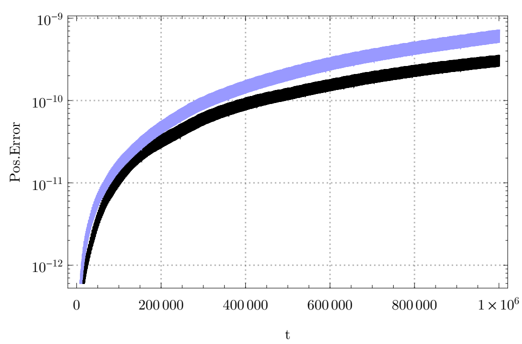


(a) Energy error.

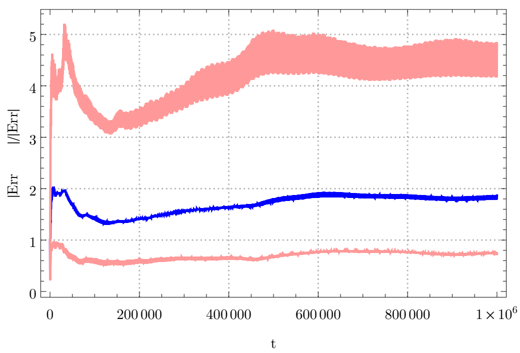


(b) Global error.

Irudia 7.5: 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.

Irudia 7.6: 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 (7.16). We use rdigits1=0 and rdigits2=3.

Kapitulua 8

IRK: Newton.

8.1 Sarrera.

Kapitulua 9

IRK: Eguzki-sistema.

9.1 Sarrera.

9.2 Meta-Algoritmoa.

Demagun Hamiltondar banagarria,

$$H(y) = H_A(y) + H_B(y),$$

non $H_A \gg H_B$.

Hau izanik dagokion hasierako problema orokorra,

$$\dot{y} = J^{-1} \nabla H(y) = f(y), \;\; y(t_0) = y_0.$$

$f(y)$ *eredua, eredu sinple* $k(y) = J^{-1} \nabla H_A(y)$ *eta eredu konplexu* $g(y) = J^{-1} \nabla H_B(y)$ *baten arteko batura gisa deskonposatu daiteke,*

$$\dot{y} = f(y) = k(y) + g(y).$$

Adibidea*.*

N-gorputzen problemaren Hamiltondarra, alde Kepleriarra eta planeten interakzioen batura gisa bana daiteke,

$$H(q,p) = H_k + H_I, \;\; H_k \gg H_I.$$

$$H_k(q,p) = \frac{1}{2} \sum_{i=0}^{N} \frac{p_i^2}{m_i} - G m_0 \sum_{i=1}^{N} \frac{m_i}{\|q_i - q_0\|} \tag{9.1}$$

$$H_I(q) = \sum_{1 \le i < j \le N}^{N} \frac{G\, m_i m_j}{\|q_j - q_i\|} \tag{9.2}$$

Banaketa honi dagokion ekuazio diferentzialak, $f(y) = k(y) + g(y)$ *modu honetan laburtuko ditugu. Honako notazioa erabiliz,*

$$\dot{y} = f(y) = \begin{pmatrix} \dot{q} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} f_q(y) \\ f_v(y) \end{pmatrix} = \begin{pmatrix} k_q(y) \\ k_v(y) \end{pmatrix} + \begin{pmatrix} g_q(y) \\ g_v(y) \end{pmatrix}$$

Batetik, $f_q(y)$ *ekuazio diferentzialen deskonposaketa honakoa da,*

$$\dot{q} = f_q(y) = v_i,\ i = 0, \ldots, N.$$

$$\dot{q} = f_q(y) \;\Rightarrow\; \begin{cases} k_q(y) = v_i, & i = 0, \ldots, N. \\ g_q(y) = 0, & i = 0, \ldots, N. \end{cases} \tag{9.3}$$

Bestetik, $f_v(y)$ *ekuazio diferentzialen deskonposaketa honakoa da,*

$$\dot{v} = f_v(y) = \sum_{j=0, j\ne i}^{N} \frac{G m_j}{\|q_j - q_i\|^3}(q_j - q_i) \quad i = 0, \ldots, N.$$

$$\dot{v} = f_v(y) \;\Rightarrow\; \begin{cases} k_v(y) = \begin{cases} \dot{v}_0 = \sum_{j=1, j\ne i}^{N} \frac{G m_j}{\|q_j - q_0\|^3}(q_j - q_0). \\ \dot{v}_i = \frac{G m_0}{\|q_0 - q_i\|^3}(q_0 - q_i), \ \ i = 1, \ldots, N. \end{cases} \\ g_v(y) = \begin{cases} \dot{v}_0 = 0. \\ \dot{v}_i = \sum_{j=1, j\ne i}^{N} \frac{G m_j}{\|q_j - q_i\|^3}(q_j - q_i), \ \ i = 1, \ldots, N. \end{cases} \end{cases} \tag{9.4}$$

Meta-algoritmoa.

IRK metodoaren formulazioa gogoratuz,

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

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

s-ataleko IRK metodoaren iterazio bakoitzean, $(s{*}d)$ ezezagunetako (Y_i) ekuazio-sistema askatu behar dugu:

$$Y_i - y_n - \sum_{j=1}^{s} \mu_{ij}\, hb_j \Big(k(Y_j) + g(Y_j) \Big) = 0, \;\; i = 1, \ldots, s.$$

Gure planteamenduan ekuazio-sistema Newton-sinplifikatuaren bidez askatuko dugu, baina jakobiarraren kalkulurik gabe. Lortzen dugun metodoa, jakobiarraren hurbilpena alde kepleriarra kontsideratzen duen ($J = k'(Y_i)$) Newton-sinplifikatuaren baliokidea da.

Garapena.

Ekuazio-sisteman Newton metodoa aplikatuz. Soluziotik gertu dagoen balio batetik abiatuta, ($Y_i^{[0]}$) eta $k = 1, 2, \ldots,$

$$\triangle Y^{[k]} = -\frac{F(Y^{[k]})}{F'(Y^{[k]})},$$

$$Y^{[k+1]} = Y^{[k]} + \triangle Y^{[k]}.$$

IRK metodoaren ekuazio-sistemari aplikatuz,

$$\triangle Y_i^{[k]} = -\frac{\big(Y_i^{[k]} - y_n - \sum_{j=1}^{s} \mu_{ij}\, hb_j \big(k(Y_j^{[k]}) + g(Y_j^{[k]})\big)\big)}{\big(1 - \sum_{j=1}^{s} \mu_{ij}\, hb_j \big(k'(Y_j^{[k]}) + g'(Y_j^{[k]})\big)\big)}$$

Ekuazio laburtzeko $\delta_i^{[k]}$ aldagai laguntzailea erabiliz hau da askatu beharreko ekuazio,

$$\triangle Y_i^{[k]} = \sum_{j=1}^{s} \mu_{ij}\, hb_j \big(k'(Y_j^{[k]}) + g'(Y_j^{[k]})\big) \triangle Y_j^{[k]} + \delta_i^{[k]}, \tag{9.5}$$

non,

$$\delta_i^{[k]} = -Y_i^{[k]} + y_n + \sum_{j=1}^{s} \mu_{ij}\, hb_j \big(k(Y_j^{[k]}) + g(Y_j^{[k]})\big).$$

Lortutako espresioa garatuko dugu.

1. ***Lehen hurbilpena.***

   $g'(Y_j^{[k]}) << k'(Y_j^{[k]})$ *eta* $g'(Y_j^{[k]}) < \triangle Y_j^{[k]}$ *denez,*

$$\triangle Y_i^{[k]} \approx \sum_{j=1}^{s} \mu_{ij}\ hb_j\ k'(Y_j^{[k]})\ \triangle Y_j^{[k]} + \delta_i^{[k]} \tag{9.6}$$

2. ***Bigarren hurbilpena (Linealizazioa).***

$$k(Y_j^{[k+1]}) = k'(Y_j^{[k]})(Y_j^{[k+1]} - Y_j^{[k]}) + k(Y_j^{[k]}) + O(\|\triangle Y_j^{[k]}\|^2)$$

$$\triangle Y_j^{[k]} = Y_j^{[k+1]} - Y_j^{[k]}$$

   Honako hurbilpena,

$$k'(Y_j^{[k]})\triangle Y_j[k] \approx k(Y_j^{[k]} + \triangle Y_j^{[k]}) - k(Y_j^{[k]})$$

   ordezkatuz eta garatuz,

$$\triangle Y_i^{[k]} = \sum_{j=1}^{s} \mu_{ij}\ hb_j\ k(Y_j^{[k]} + \triangle Y_j^{[k]})\ - \sum_{j=1}^{s} \mu_{ij}\ hb_j\ k(Y_j^{[k]}) + \delta_i^{[k]},$$

$$\triangle Y_i^{[k]} = -Y_i^{[k]} + y_n + \sum_{j=1}^{s} \mu_{ij}\ hb_j\ k(Y_j^{[k]} + \triangle Y_j^{[k]})\ + \sum_{j=1}^{s} \mu_{ij}\ hb_j\ g(Y_j^{[k]}). \tag{9.7}$$

3. ***Ekuazioak berridatiziz.***

   $\triangle Y_i^{[k]} = Y_i^{[k+1]} - Y_i^{[k]}$ *definizioa erabiliaz,*

$$Y_i^{[k+1]} = y_n + \sum_{j=1}^{s} \mu_{ij}\ hb_j\ k(Y_j^{[k+1]})\ + \sum_{j=1}^{s} \mu_{ij}\ hb_j\ g(Y_j^{[k]}). \tag{9.8}$$

Meta-Algoritmoa.

IRK metodoa aplikatzeko meta algoritmoa planteatuko dugu.

$e = 0$;
for $n \leftarrow 1$ **to** $endstep$ **do**
  Hasieratu $Y_i^{[0]}$, $i = 1, \ldots, s$;
  $k = 1$;
  $g_i^{[0]} = \sum_{j=1}^{s} \mu_{ij}\ hb_j\ g(Y_j^{[0]})$;
  Askatu $Y_i^{[k]} = y_{n-1} + \sum_{j=1}^{s} \mu_{ij}\ hb_j\ k(Y_j^{[k]})\ + g_i^{[k-1]}$;
  while *(konbergentzia lortu)* **do**
    $k = k + 1$;
    $g_i^{[k-1]} = \sum_{j=1}^{s} \mu_{ij}\ hb_j\ g(Y_j^{[k-1]})$;
    Askatu $Y_i^{[k]} = y_{n-1} + \sum_{j=1}^{s} \mu_{ij}\ hb_j\ k(Y_j^{[k]})\ + g_i^{[k-1]}$;
  end
  $L_i = hb_i f(Y_i^{[k-1]})$;
  $\delta_n = \sum_{i=1}^{s} L_i + e$;
  $y_n = y_{n-1} + \delta_n$;
  $e = (y_{n-1} - y_n) + \delta_n$;
end

ALGORITHM 12: Main Algorithm

Meta-algoritmoari buruzko hainbat ohar:

1. *Barne iterazioak.*

   Kanpo iterazioa Newton metodoa aplikatuz zehaztu dugu. Barne iterazio aldiz, aukera ezberdinak ditugu.

***Barne-interazioa** puntu finkoaren bidez:*

$l = 0$;
$Y_i^{[k,0]} = Y_i^{[k-1]}$;
while *(konbergentzia lortu)* **do**
  $l = l + 1$;
  $K_i^{[k,l]} = k(Y_j^{[k,l-1]})$;
  $Y_i^{[k,l]} = y_{n-1} + \sum_{j=1}^{s} \mu_{ij}\, hb_j\, K_j^{[k,l]} + g_i^{[k-1]}$;
end

ALGORITHM 13: Main Algorithm

2. *Problema independenteak.*

   Era honetako deskonposaketa bat dugunean,

$$f\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} f_1(y_1) \\ f_2(y_2) \end{pmatrix} + \begin{pmatrix} g_1(y_1, y_2) \\ g_2(y_1, y_2) \end{pmatrix},$$

   eredu sinplifikatua problema independenteak osatzen dituzte eta barne iterazioak modu independentean kalkula daitezke. N-gorputzen eguzki-sistemaren adibidean, eredu sinplifikatua $k(y)$ (eguzkiarekiko interakzioa) planeta bakoitzarentzat problema independentea dugu. Kasu honetan urrats bat finkatuta, kanpo planeten $k(y)$ problema barruko planeta baino azkarrago konbergituko du.

Orokorpena.

Aurreko atalean, maila bakarreko ereduen deskonposaketa aztertu dugu. Ideia orokortuz, eredu deskonposaketa maila ezberdinetan egin daiteke. Problema bat emanda $\dot{y} = f(y)$,

$$\text{1. maila} \begin{cases} \text{Eredu osoa. } f(y) \\ \text{Eredu sinplea. } \tilde{f}(y) \end{cases} \Rightarrow \quad f = \tilde{f} + (f - \tilde{f}) \tag{9.9}$$

$$\text{2. maila} \begin{cases} \text{Eredu osoa. } \tilde{f}(y) \\ \text{Eredu sinplea. } \tilde{\tilde{f}}(y) \end{cases} \Rightarrow \quad \tilde{f} = \tilde{\tilde{f}} + (\tilde{f} - \tilde{\tilde{f}}) \tag{9.10}$$

Adibidea.

Demagun ekuazio diferentzialak, m perturbazio funtzio dituela,

$$\dot{y} = f(y) = k(y) + g1(y) + g2(y) + \cdots + gm(y)$$

non $k(y) << gl(y)$, $l = 1, \ldots, m$.

$m = 2$ deneko kasu partikulara aztertuko dugu,

$$\dot{y} = f(y) = k(y) + g1(y) + g2(y).$$

Askatu $Y_i = y_{n-1} + \sum_{j=1}^{s} \mu_{ij}\, hb_j\, (k(Y_j) + g1(Y_j) + g2(Y_j))$*;*

while *(konbergentzia lortu)* **do**

  $k = k + 1$*;*

  $g2_i^{[k-1]} = \sum_{j=1}^{s} \mu_{ij}\, hb_j\, g2(Y_j^{[k-1]})$*;*

  Askatu $Y_i^{[k]} = y_{n-1} + \sum_{j=1}^{s} \mu_{ij}\, hb_j\, (k(Y_j^{[k]}) + g1(Y_j^{[k]}))\; + g2_i^{[k-1]}$*;*

  while *(konbergentzia lortu)* **do**

    $l = l + 1$*;*

    $g1_i^{[k,l-1]} = \sum_{j=1}^{s} \mu_{ij}\, hb_j\, g1(Y_j^{[k,l-1]})$*;*

    Askatu $Y_i^{[k,l]} = y_{n-1} + \sum_{j=1}^{s} \mu_{ij}\, hb_j\, k(Y_j^{[k,l]}) + g1_i^{[k,l-1]}\; + g2_i^{[k-1]}$*;*

  end

end

ALGORITHM 14: Main Algorithm

Erlatibitate efektua gehitzerakoan, N-gorputzen problemari dagokion ekuazio diferentziala,

$$\dot{y} = f(y),\; f(y) = k(y) + g(y) + rs(y) + rn(y),$$

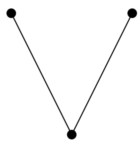
k(y): kepleriarra.
g(y): planeten arteko grabitazio interakzioak.
rs(y): eguzkiarekiko erlatibitate efektua.
rn(y): planeten arteko erlatibitate efektuak.

Adierazpena.

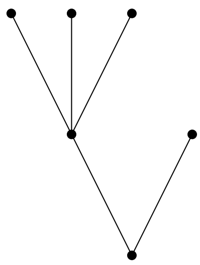
Ekuazio diferentzialen deskonposaketak, zuhaitz moduan adieraz daitezke.

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

$$f \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} f_1(y_1) \\ f_2(y_2) \end{pmatrix} + \begin{pmatrix} g(y) \end{pmatrix},$$



$$f_1 \begin{pmatrix} y_1 \end{pmatrix} = \begin{pmatrix} f_{11}(y_{11}) \\ f_{12}(y_{11}, y_{12}) \\ f_{13}(y_{11}, y_{12}, y_{13}) \end{pmatrix} + \begin{pmatrix} g_1(y_1) \end{pmatrix},$$



Atala IV

synthesys

Kapitulua 10

Eranskinak

.

10.1 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 ,

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

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

$$\triangle E^{[k+1]} = \triangle E^{[k]} - \frac{f(\triangle E^{[k]})}{f'(\triangle E^{[k]})} \tag{10.1}$$

$\triangle E^{[0]}$ **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,

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

$$(se\ \cos(n\triangle t)+ce\ \sin(n\triangle t))\frac{x^2}{2}+(1+se\ \sin(n\triangle t)-ce\cos(n\triangle t))x-(se) = 0 \tag{10.2}$$

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, $\cos(\triangle E) - 1$ espresioaren kalkuluaren ezabapen arazoak 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,

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

Bibliografia

[1] *Josh Barnes and Piet Hut. A hierarchical o (n log n) force-calculation algorithm.* nature*, 324(6096):446–449, 1986.*

[2] *André Berger.* A brief history of the astronomical theories of paleoclimates. *Springer, 2012.*

[3] *Sergio Blanes, Fernando Casas, Ariadna Farres, Jacques Laskar, Joseba Makazaga, and Ander Murua. New families of symplectic splitting methods for numerical integration in dynamical astronomy.* Applied Numerical Mathematics*, 68:58–72, 2013.*

[4] *J Carrier, Leslie Greengard, and Vladimir Rokhlin. A fast adaptive multipole algorithm for particle simulations.* SIAM journal on scientific and statistical computing*, 9(4):669–686, 1988.*

[5] *John E Chambers. A hybrid symplectic integrator that permits close encounters between massive bodies.* Monthly Notices of the Royal Astronomical Society*, 304(4):793–799, 1999.*

[6] *Robert M Corless and Nicolas Fillion. A graduate introduction to numerical methods.* AMC*, 10:12, 2013.*

[7] *Martin J Duncan, Harold F Levison, and Man Hoi Lee. A multiple time step symplectic algorithm for integrating close encounters.* The Astronomical Journal*, 116(4):2067, 1998.*

[8] *William M Folkner, James G Williams, Dale H Boggs, Ryan S Park, and Petr Kuchynka. The planetary and lunar ephemerides de430 and de431.* Interplanet. Netw. Prog. Rep*, 196:1–81, 2014.*

[9] *Ernst Hairer, Christian Lubich, and Gerhard Wanner.* Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006.*

[10] *Ernst Hairer, Robert I McLachlan, and Alain Razakarivony. Achieving brouwer's law with implicit runge–kutta methods.* BIT Numerical Mathematics, *48(2):231–243, 2008.*

[11] *David M Hernandez and Edmund Bertschinger. Symplectic integration for the collisional gravitational n-body problem.* Monthly Notices of the Royal Astronomical Society, *452(2):1934–1944, 2015.*

[12] *Nicholas J Higham.* Accuracy and stability of numerical algorithms. *Siam, 2002.*

[13] *Nicholas J Higham. Programming languages: An applied mathematics view. 2015.*

[14] *MP Calvo JM Sanz-Serna.* Numerical Hamiltonian problems. *Chapman and Hall, 1994.*

[15] *Jacques Laskar, Agnes Fienga, Mickael Gastineau, and Herve Manche. La2010: a new orbital solution for the long-term motion of the earth.* Astronomy & Astrophysics, *532:A89, 2011.*

[16] *Jacques Laskar and Philippe Robutel. High order symplectic integrators for perturbed hamiltonian systems.* Celestial Mechanics and Dynamical Astronomy, *80(1):39–62, 2001.*

[17] *Harold F Levison and Martin J Duncan. The long-term dynamical behavior of short-period comets.* Icarus, *108(1):18–36, 1994.*

[18] *Robert I McLachlan. Composition methods in the presence of small parameters.* BIT Numerical Mathematics, *35(2):258–268, 1995.*

[19] *Jean-Michel Muller, Nicolas Brisebarre, Florent De Dinechin, Claude-Pierre Jeannerod, Vincent Lefevre, Guillaume Melquiond, Nathalie Revol, Damien Stehlé, and Serge Torres.* Handbook of floating-point arithmetic. *Springer Science & Business Media, 2009.*

[20] *Nobelprize.org. The nobel prize in chemistry 2013, May 2014.*

[21] *Michael L Overton.* Numerical computing with IEEE floating point arithmetic. *Siam, 2001.*

[22] *M Papadrakakis, M Kojic, and I Tuncer. Fast detection of chaotic or regular behavior of double pendulum system: application of the fast norm vector indicator method.*

[23] Pat H Sterbenz. Floating-point computation*. Prentice Hall, 1973.*

[24] Gerald J Sussman and Jack Wisdom. Chaotic evolution of the solar system. Technical report, DTIC Document, 1992.

[25] Jack Wisdom and Matthew Holman. Symplectic maps for the n-body problem. The Astronomical Journal*, 102:1528–1538, 1991.*