

---

N-GORPUTZEKO PROBLEMA GRABITAZIONALAREN  
EBAZPENERAKO ZENBAKIZKO METODOEN  
AZTERKETA.

---

by

Mikel Antonana Otano

Informatika Fakultatea  
Euskal-Herriko Unibertsitatea  
Donostia

©Mikel Antonana 2017



# Gaien Aurkibidea

<b>I</b>	<b>Introduction</b>	<b>1</b>
<b>1</b>	<b>Sarrera.</b>	<b>3</b>
1.1	Context of the Study . . . . .	3
1.2	Problem statement or motivation for the study. . . . .	4
1.3	Aim and scope . . . . .	5
1.4	Overview of the study (structure of the Thesis). . . . .	7
1.5	laburpena . . . . .	7
<b>II</b>	<b>Background</b>	<b>9</b>
<b>2</b>	<b>Zenbakizko Integratzaile Sinplektikoak.</b>	<b>11</b>
2.1	Sarrera. . . . .	11
2.1.1	Zenbakizko metodoak. . . . .	11
2.1.2	Problema motak. . . . .	14
2.1.3	Sistema-Hamiltondarrak. . . . .	14
2.2	Metodo sinplektikoak. . . . .	16
2.3	Gauss metodoak. . . . .	18
2.3.1	Runge-Kutta metodoak. . . . .	18
2.3.2	Kolokazio metodoak. . . . .	23
2.4	Konposizio metodoak. . . . .	25
2.4.1	Sarrera. . . . .	25
2.4.2	CO1035: 10 ordeneko konposizio metodoa. . . . .	26
2.5	Splitting metodoak. . . . .	28
2.5.1	Sarrera. . . . .	28
2.5.2	Eguzki-sistemari egokitutako splitting metodoak. . . . .	30
2.6	Kepler fluxua. . . . .	31
2.7	Laburpena. . . . .	33

<b>3</b>	<b>Problemak.</b>	<b>35</b>
3.1	Sarrera.	35
3.2	Pendulu bikoitza.	35
3.2.1	Ekuazioak.	36
3.2.2	Hasierako balioak.	36
3.2.3	Kodeak.	37
3.3	N-Body problema.	37
3.3.1	Zailtasunak.	38
3.3.2	Ekuazio barizentrikoak.	38
3.3.3	Ekuazio Heliozentrikoak.	39
3.3.4	Hasierako balioak.	40
3.3.5	Kodeak.	41
<b>4</b>	<b>Koma Higikorreko Aritmetika.</b>	<b>43</b>
4.1	Sarrera.	43
4.2	Adierazpena.	43
4.3	Biribiltze errorea.	45
4.3.1	Tresnak.	48
4.4	Laburpena.	49
<b>5</b>	<b>Zientzia konputazioa.</b>	<b>51</b>
5.1	Sarrera.	51
5.2	Parallel Hardware.	52
5.3	Software liburutegiak.	58
5.4	Kode Optimizazioak.	60
5.4.1	Scalar Tuning and General Optimization.	60
5.4.2	Optimization of vectorization.	61
5.4.3	Multi-threading.	61
5.4.4	Memory access.	62
5.5	Laburpena.	62
<b>6</b>	<b>Review.</b>	<b>63</b>
6.1	Sarrera.	63
6.2	Efemerideak.	63
6.3	Eguzki-sistemaren integrazio luzeak.	64
<b>III</b>	<b>Core</b>	<b>67</b>
<b>7</b>	<b>IRK: Puntu-Finkoa.</b>	<b>69</b>
7.1	Sarrera.	69

7.2	Hairer-en inplementazioa. . . . .	69
7.3	Gure inplementazioa. . . . .	70
7.3.1	Koefizienteak (1.proposamena). . . . .	71
7.3.2	Geratze irizpidea (2.proposamena). . . . .	72
7.3.3	Batura konpensatua (3.proposamena). . . . .	73
7.3.4	Biribiltze errorearen estimazioa (4.proposamena). . . . .	74
7.3.5	Atalen hasieraketa. . . . .	75
7.3.6	Gauss-Seidel. . . . .	77
7.3.7	Algoritmoa. . . . .	77
7.4	Esperimentuak. . . . .	78
7.4.1	Doitasun azterketa. . . . .	78
7.4.2	Brouwer legea. . . . .	80
7.4.3	Pendulu bikoitza. . . . .	81
7.4.4	N-Body problema. . . . .	81
<b>8</b>	<b>IRK: Newton.</b>	<b>87</b>
8.1	Sarrera. . . . .	87
<b>9</b>	<b>IRK: Eguzki-sistema.</b>	<b>89</b>
9.1	Sarrera. . . . .	89
9.2	Meta-Algoritmoa. . . . .	89
9.3	Denbora birparametrizazioa. . . . .	96
9.3.1	Denbora birparametrizazioa. . . . .	96
9.3.2	Adibidea. . . . .	97
<b>IV</b>	<b>Synthesys</b>	<b>99</b>
<b>V</b>	<b>Eranskinak</b>	<b>101</b>
<b>10</b>	<b>A-Eranskinak</b>	<b>103</b>
10.1	Kepler ekuazioak eta definizioak. . . . .	103
10.2	Ekuazio Heliozentrikoak. . . . .	105



# Irudien Zerrenda

1.1	Eguzki-sistema . . . . .	6
2.1	Pendulua. . . . .	15
2.2	Kolokazio metodoen simetria. . . . .	20
2.3	kolokazio metodoak. . . . .	24
3.1	Pendulu bikoitza. . . . .	35
3.2	Eguzki-sistema. . . . .	39
4.1	Floating-point number line. . . . .	44
4.2	32-biteko koma-higikorra . . . . .	44
4.3	Biribiltze errorea. . . . .	48
5.1	Processor clock rate. . . . .	52
5.2	Konputagailuen eraginkortasuna. . . . .	53
5.3	Memoria hierarkia. . . . .	54
5.4	Shared Memory System. . . . .	56
5.5	Distributed Memory System. . . . .	56
5.6	Shared Memory System (UMA). . . . .	57
5.7	OpenMp programming model. . . . .	57
5.8	BLAS speeds. . . . .	59
5.9	Optimization. . . . .	61
5.10	Optimization. . . . .	62
7.1	Interpolazioa. . . . .	75
7.2	Pendulu-bikoitza: energiaren erroreak eta errore globalaren eboluzioa. . . . .	82
7.3	Histogram of energy errors for Non-Chaotic case (a,b) and for Chaotic case (c,d). . . . .	83
7.4	Pendulu-bikoitza: biribiltze erroreak estimazio. . . . .	84
7.5	N-Body: energiaren erroreak eta errore globalaren eboluzioa. . . . .	84
7.6	N-Body: biribiltze erroreak estimazioa. . . . .	85





# Taulen Zerrenda

2.1	Runge-Kutta metodo inplizituak. . . . .	19
2.2	C01035 konposizio metodoa. . . . .	27
2.3	$SABA_4$ splitting metodoa. . . . .	30
2.4	$ABAH1064$ splitting metodoa. . . . .	32
2.5	Integrazio metodoen laburpena . . . . .	33
3.1	. . . . .	38
3.2	Konstanteak . . . . .	40
3.3	Eguzki-sistemaren hasierako balioak . . . . .	41
3.4	Ilargiaren Lurrarekiko hasierako balioak. . . . .	41
3.5	Planeten masa parametroak. . . . .	42
4.1	. . . . .	45
7.1	Summary of Non-Chaotic case. . . . .	81
7.2	Summary of Chaotic case. . . . .	81



# **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 problemaren azterketak teori analitikoaren 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 dinamikaren funtsezko galdera batzuk ezagutu ditugu eta berriki, Karplusen taldeak 2013. kimikako Nobel saria [25] jaso du kimika konputazionalan 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 txikiko integrazioak. 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 ez dira izaten.

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-distemaren planeten epe luzeko zenbakizko integrazioa erronka garrantzitsua izan da. Adibidez, Sussmanek eta Wisdomek (1993) [30] 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 [19, 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 eraginkorrak 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 eraginkorrak 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 implizituengana. Baieztapen hau eztabaidagarria da, eta praktikan metodo implizituetan gehiago sakondu behar dela iruditzen zaigu.

Zentzu honetan, metodo implizituen ezaugarri interesgarri batzuk nabarmendu dugu. Abantaila nagusienetakoa malgutasuna da. Metodo implizituek inplementazio malgua onartzen dute eta ondorioz, integratu nahi dugun problemari egokitzeko aukera gehiago eskaintzen dizkigu. Aipatzekoa da ere, metodo esplizituak sistema Hamiltondar banagarrietan bakarrik aplikatu daitezkeela: Hamiltondarraren egitura hau aprobetxatuz oso eraginkorrak dira baina badira integratu nahi den problemak bete behar duen muga ere. Metodo implizituak aldiz, Hamiltondar orokorrekin aplikatu daitezke eta gainera, lehen ordenako ekuazio diferentzialetarako metodo sinpletikoak implizituak izan behar dira. Azkenik ez dugu ahaztu behar, metodo implizituen artean orden altuko metodoak existitzen direla eta hauek nahi-taezkoak dira doitasun handiko integrazioak behar ditugunean.

Lan honetan, metodo implizituen artean Gauss zenbakizko integrazio metodoa aukeratu dugu. Hainbat autorek (Hairer [13][14] eta Sanz Serna[18]) 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.

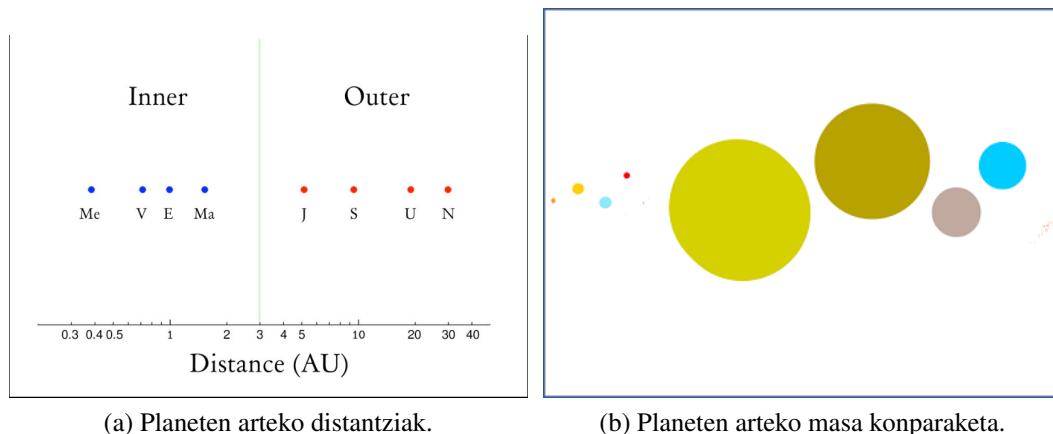
## 1.3 Aim and scope

Gure helburua, eguzki-sistemaren ebazpenerako Gauss implizituaren inplementazio eraginkorra proposatzea da. Hau lortzeko bereziki honako aspektu hauek kontutan izango ditugu: eguzki-sistemaren problemaren ezaugarriak, konputagailuen koma-higikorrek 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 datozeela 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 dauden gorputzak: 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.



Irudia 1.1: Eguzki-sistema.

Konputagailuen koma-higikorrek aritmetika ondo ulertzea garrantzitsua da. Zenbaki errealeen adierazpen finitua 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. 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 code*[1] eta *fast multipole method*[6]. 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 [14]. Autoreak IRK puntu-finkoaren inplementazio estandarrean biribiltze errorearen okerreko portaeraz jabetu zen eta arazo hau konpontzeko soluzioa proposatu zuen. Lehen urratsa honetan, biribiltze errorearen arazoari soluzio berri bat eman dio-gu 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 orde-z, Newton sinplifikatuaren metodoa aztertu dugu. Gure ekarpena, Newton sinplifikatua modu eraginkorrean aplikatzeko teknika proposatzea izan da. S-ataletako IRK metodoa eta d-dimentsioko EDA badugu, Newton sinplifikatuaren metodoren iterazio bakoitzean (*sdxsd*) tamainako sistema lineala askatu behar da. Gure proposamena da, jatorrizko sistema lineala blokeka diagonal 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 denbora-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.

Ekuazio diferentzial arruntetarako (*ODE*) hasierako baliodun problemen (*IVP*) formulazioa,

$$\dot{\mathbf{y}}(t) = \mathbf{f}(t, \mathbf{y}(t)), \quad \mathbf{y}(t_0) = \mathbf{y}_0, \quad (2.1)$$

non  $\mathbf{y} : \mathbb{R} \rightarrow \mathbb{R}^d$  soluzioa,  $\mathbf{y}_0 \in \mathbb{R}^d$  hasierako balioa eta  $\mathbf{f} : \mathbb{R} \times \mathbb{R}^d \rightarrow \mathbb{R}^d$  bektore eremua deskribatzen funtzioa dugun ( $\dot{\mathbf{y}}$  notazioa erabiliko dugu  $d\mathbf{y}/dt$  adierazteko).

$t_0$  unetik  $t_0 + h$  integrazioa,

$$\mathbf{y}(t_0 + h) = \mathbf{y}_0 + \int_{t_0}^{t_0+h} \mathbf{f}(t, \mathbf{y}(t)) dt.$$

Goiko ekuazio-sistema bektoriala (2.1), ekuazio-sistema modu eskalarrean idatz daiteke:

$$\begin{aligned} \dot{y}_1(t) &= f_1(t, (y_1(t), y_2(t), \dots, y_d(t))), & y_1(t_0) &= y_{1,0} \\ \dot{y}_2(t) &= f_2(t, (y_1(t), y_2(t), \dots, y_d(t))), & y_2(t_0) &= y_{2,0} \\ &\dots & & \\ \dot{y}_d(t) &= f_d(t, (y_1(t), y_2(t), \dots, y_d(t))), & y_d(t_0) &= y_{d,0} \end{aligned}$$

Metodo analitikoak (funtzio ezagunen arabera soluzio zehatza) eta erdi-analitikoak, ez dira problema askoren soluzioa bilatzeko teknika egokiak. Zenbakizko metodoak, aldiz, modu errazean aplikatu daitezke eta horregatik, soluzio metodo nagusia kontsideratzen da.

Zenbakizko metodo baten bidez,  $\mathbf{y}(t)$  soluzioaren  $\mathbf{y}_n \approx \mathbf{y}(t_n)$  hurbilpena lortu dugu  $t_n = t_{n-1} + h_n$  ( $n = 1, 2, \dots$ ) une diskretu ezberdinetarako. Zenbakizko soluzioa urratsez-urrats sekuentzialki eta zehaztutako tarte baterako ( $t_0 \leq t \leq t_f$ ) kalkulatu dugu. Beraz, lortutako balio multzoak  $(t_0, \mathbf{y}_0), (t_1, \mathbf{y}_1), \dots, (t_f, \mathbf{y}_f)$  zenbakizko soluzioa definitzen du.

Nola jakin zenbakizko soluzioa matematika modeloarekiko zuzena dela? Zenbakizko soluzioaren errorea neurtzeko teknika ezberdinak ditugu.

Azkenik argitu beharra dago ekuazio-sistema beti *sistema autonomo* moduan, hau da, denborarekiko independentea idatz daitekeela. Hori horrela izanik, notazioa sinplifikatzeko era honetako sistemak kontsideratuko ditugu,

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t)), \quad \mathbf{y}(t_0) = \mathbf{y}_0. \quad (2.2)$$

Jarraian zenbakizko metodoen oinarriko kontzeptuak eta notazioa finkatuko dugu.

### 1. Fluxua.

Fase-espazioko edozein  $\mathbf{y}_0$  puntuari,  $\mathbf{y}(t_0) = \mathbf{y}_0$  hasierako balio duen  $\mathbf{y}(t)$  soluzioa asignatzen dion mapping-ari deitzen diogu. Izendatzeko  $\varphi_t$  notazioa erabiliko dugu,

$$\varphi_t(\mathbf{y}_0) = \mathbf{y}(t) \quad \text{baldin} \quad \mathbf{y}(t_0) = \mathbf{y}_0$$

### 2. Zenbakizko diskretizazioa.

$\mathbf{y}_n, \mathbf{y}_{n-1}, \dots, \mathbf{y}_0$  balioak emanda,  $\mathbf{y}_{n+1} \approx \mathbf{y}(t_{n+1})$  soluzioaren hurbilpena kalkulatzeko formulari *zenbakizko fluxua* deritzogu. Honako notazioa erabiliko dugu,

$$\mathbf{y}_{n+1} = \phi(\mathbf{y}_{n+1}, \mathbf{y}_n, \dots, \mathbf{y}_0; h; f).$$

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

Adibideak.

Eurler metodo esplizitua.

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \mathbf{f}(\mathbf{y}_n)$$

Eurler metodo implizitua.

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \mathbf{f}(\mathbf{y}_{n+1})$$

### 3. Metodoaren ordena.

Definizioa. **Errore globala.** Zenbakizko soluzioaren  $t_0$  hasierako unetik  $t_k$  une arteko errore globala  $ge(t)$ ,

$$ge(t_k) = y_k - y(t_k).$$

Definizioa.  $\phi$  **metodoaren ordena.**  $h$  urrats luzera finkoko  $\phi$  metodoak  $p$  ordenekoa dela esaten da, errore globala  $ge(t) O(h^p)$  ordenekoa bada  $h \rightarrow 0$ ,

$$y_k - y(t_k) = O(h^p), \quad h \rightarrow 0.$$

Definizioa. **Errore lokala.** Zenbakizko soluzioaren urrats bakarreko  $[t_k, t_{k+1}]$  errore lokala  $le(t)$ ,

$$le(t_{k+1}) = y_{k+1} - y(t_{k+1})$$

non  $y_k(t), y(t_k) = y_k$  hasierako balio lokaleko soluzio zehatza den.

Metodoaren ordena  $O(h^p)$  bada, errore lokala  $O(h^{p+1})$  da.

### 4. Metodo simetrikoak.

**Adjoint method.** Jatorrizko metodoaren alderantzizko eta kontrako denbora  $-h$  esleipenari (map),  $\phi_h$  metodoaren  $\phi_h^*$  *adjoint metodoa* esaten zaio.

$$\phi_h^* = \phi_{-h}^{-1}$$

$\phi_h^* = \phi_h$  -betetzen denean, metodoa simetrikoa da.

### 2.1.2 Problema motak.

#### 1. Problema kaotikoak.

Hasierako balio edo parametroen perturbazioekiko, diskretizazio-erroreekiko (trunkatze) edo birbitze erroreekiko esponenzialki sentikorrak diren problemaei esaten zaie.

#### 2. Problema stiff.

Irakurri 12.1 Solution of stiff problems (555 – 559).

Irakurri 13.2.3 Convergence and consistency and 13.3 Stiffness and Implicitness (600).

### 2.1.3 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.

$H(\mathbf{p}, \mathbf{q})$  funtzio leuna izanik, non  $H : \mathbb{R}^d \rightarrow \mathbb{R}$  eta  $(\mathbf{p}, \mathbf{q}) = (p_1, \dots, p_n, q_1, \dots, q_n)$  ( $d = 2n$ ), dagokion ekuazio diferentzialak era honetan definitzen dira,

$$\frac{d}{dt}p_j = -\frac{\partial H(\mathbf{p}, \mathbf{q})}{\partial q_j}, \quad \frac{d}{dt}q_j = \frac{\partial H(\mathbf{p}, \mathbf{q})}{\partial p_j}, \quad j = 1, \dots, n. \quad (2.3)$$

Edo notazio laburtua erabiliz,

$$\dot{\mathbf{y}} = J^{-1} \nabla H(\mathbf{y}), \quad \mathbf{y} = \begin{pmatrix} p \\ q \end{pmatrix}, \quad J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}, \quad (2.4)$$

$\mathbf{p}$  eta  $\mathbf{q}$  bektoreen  $n$  dimentsioa sistemaren *askatasun maila* deritzo.  $H(\mathbf{p}, \mathbf{q})$  funtzioaren balioa integrazioan zehar konstante mantentzen da.

#### Adibidea.

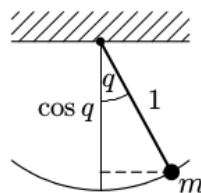
Penduluaren problemaren (masa  $m = 1$ ,  $l = 1$  luzerako makila eta  $g = 1$  grabitazioa) askatasun bakarreko sistema Hamiltondarra,

$$H(p, q) = \frac{1}{2}p^2 - \cos q.$$

Ekuazio diferentzialak,

$$\dot{p} = -\sin q, \quad \dot{q} = p.$$





(a) Pendulua.

Irudia 2.1: Pendulua.

### Hamiltondar banagarriak.

Hamiltondar banagarriak egitura bereziko sistema Hamiltondarrak ditugu. Sistema-mekanikoak era honetako Hamiltondarra dute  $H(\mathbf{p}, \mathbf{q}) = T(\mathbf{p}) + U(\mathbf{q})$ .

Horien artean, *bigarren ordeneko* ekuazio diferentzialak aipatu behar ditugu, zeintzuk Hamiltondar banagarri kasu partikularra bat diren,

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

Beraz, dagokien ekuazio diferentzialak,

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

### Adibidea.

*Bi-gorputzen problema* edo *Kepler problema*. Planoan elkar erakartzen diren bi gorputzen (planeta eta eguzkia) mugimendua kalkulatzeko, horietako gorputz baten kokapena koordinatu sistemaren jatorria kontsideratuko dugu eta beste gorputzaren kokapenaren koordinatuak  $\mathbf{q} = (q_1, q_2)$  aukeratuko ditugu. Normalizatutako Newton legearen arabera ekuazio diferentzialak,

$$\dot{p}_1 = -\frac{q_1}{(q_1^2 + q_2^2)^{3/2}}, \quad \dot{p}_2 = -\frac{q_2}{(q_1^2 + q_2^2)^{3/2}}. \quad (2.5)$$

$$\dot{q}_i = p_i, \quad i = 1, 2. \quad (2.6)$$

Baliokide den sistema Hamiltondarra,

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

Planetaren mugimendua orbita eliptiko bat da. Honako hasierako balioei dagokien soluzioa,

$$q_1(0) = 1 - e, \quad q_2(0) = 0, \quad p_1(0) = 0, \quad p_2(0) = \sqrt{\frac{1+e}{1-e}},$$

$e$  ezentrizidade ( $0 \leq e < 1$ ) duen elipsea da, eta  $P = 2\pi$  periododuna.

### Hamiltondar perturbatuak.

Hamiltondar perturbatuak, honako egitura duten,

$$H = H_A + \epsilon H_B \quad (|H_B| \ll |H_A|),$$

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.

Eguzki-sistemaren kasuan, *Jakobi* koordenatuak erabiliz Hamiltondarra  $H(\mathbf{p}, \mathbf{q}) = H_K(\mathbf{p}, \mathbf{q}) + H_I(\mathbf{q})$  moduan banatzen da, non  $H_K(\mathbf{p}, \mathbf{q})$  Kepler problema independentea eta  $H_I(\mathbf{q})$  integratu daitezkeen. Beste aukera, koordenatu *Heliozentrikoak* erabiliz  $H(\mathbf{p}, \mathbf{q}) = H_K(\mathbf{p}, \mathbf{q}) + H_I(\mathbf{p}, \mathbf{q})$  moduan banatzen da, non  $H_I(\mathbf{p}, \mathbf{q})$  ezin daitezkeen zuzenean integratu.

## 2.2 Metodo sinplektikoak.

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

### Implicit midpoint.

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

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

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

$$\begin{aligned} p_{\frac{n+1}{2}} &= p_n - \frac{h}{2} \nabla_q H(p_{\frac{n+1}{2}}, q_n) \\ q_{n+1} &= q_n + \frac{h}{2} (\nabla_p H(p_{\frac{n+1}{2}}, q_n) + \nabla_p H(p_{\frac{n+1}{2}}, q_{n+1})) \end{aligned} \quad (2.8)$$

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

edo

$$\begin{aligned} q_{\frac{n+1}{2}} &= q_n + \frac{h}{2} \nabla_p H(p_n, q_{\frac{n+1}{2}}) \\ p_{n+1} &= p_n - \frac{h}{2} (\nabla_q H(p_n, q_{\frac{n+1}{2}}) + \nabla_q H(p_{n+1}, q_{\frac{n+1}{2}})) \end{aligned} \quad (2.9)$$

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

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

$$\begin{aligned} p_{n+1/2} &= p_n + \frac{h}{2} f(q_n) \\ q_{n+1} &= q_n + h p_{n+1/2} \end{aligned} \quad (2.10)$$

$$p_{n+1} = p_{n+1/2} + \frac{h}{2} f(q_{n+1})$$

edo

$$\begin{aligned} q_{n+1/2} &= q_n + \frac{h}{2} p_n \\ p_{n+1} &= p_n + h f(q_{n+1/2}) \end{aligned} \quad (2.11)$$

$$q_{n+1} = q_{n+1/2} + \frac{h}{2} p_{n+1}$$

## 2.3 Gauss metodoak.

### 2.3.1 Runge-Kutta metodoak.

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

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \end{array}, \quad \begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \dots & a_{1s} \\ c_2 & a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \ddots & & \vdots \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss} \\ \hline & b_1 & b_2 & \dots & b_s \end{array} \quad (2.12)$$

Hasierako baliodun problema (2.1) baten  $\mathbf{y}(t)$  soluzioaren  $\mathbf{y}_n \approx \mathbf{y}(t_n)$  hurbilpena era honetan kalkulatzen da,

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h \sum_{i=1}^s b_i \mathbf{f}(\mathbf{Y}_{n,i}), \quad (2.13)$$

non  $\mathbf{Y}_{n,i}$  atalak era honetan definitzen diren,

$$\mathbf{Y}_{n,i} = \mathbf{y}_n + h \sum_{j=1}^s a_{ij} \mathbf{f}(\mathbf{Y}_{n,j}) \quad i = 1, \dots, s. \quad (2.14)$$

### Metodo esplizituak (ERK) eta implizituak (IRK).

Bi mota nagusi bereizi ditzakegu: esplizituak (*ERK*) non  $\forall i \geq j, a_{ij} = 0$  eta implizituak (*IRK*) non  $\exists i \geq j, a_{ij} \neq 0$ . *ERK* metodoak problema Nonstiff-etarako eraginkorrak kontsideratzen dira eta *IRK* metodoak ordea, problema Stiff-etarako.

Adibideak: ERK lau-ataletako metodo klasikoa (ezkerrekoa) and Implicit Mid-point method (eskuindekoa).

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 2/6 & 2/6 & 1/6 \end{array}, \quad \begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array} \quad (2.15)$$

htb

Taula 2.1: Runge-Kutta metodo inplizituak.

Metodoa	Baldintzak			Ordena
Gauss	$B(2s)$	$C(s)$	$D(s)$	$2s$
Radau IA	$B(2s-1)$	$C(s-1)$	$D(s)$	$2s-1$
Radau IB	$B(2s-1)$	$C(s)$	$D(s-1)$	$2s-1$
Lobatto IIIA	$B(2s-2)$	$C(s)$	$D(s-2)$	$2s-2$
Lobatto IIIB	$B(2s-2)$	$C(s-2)$	$D(s)$	$2s-2$
Lobatto IIIC	$B(2s-2)$	$C(s-1)$	$D(s-1)$	$2s-2$

*ERK* lau-ataletako metodo klasikoa  $p = 4$  ordeneko dugu. Orden altuko *ERK* metodoak aurkitzea konplexua da, betearazi behar diren baldintza kopuru esponentzialki hazten baitira. Orden altuko (*ERK*) metodo hauek aurkitu dira:  $p = 8$  ordeneko metodoa  $s = 11$  atalekin,  $p = 10$  ordeneko metodoa  $s = 17$  atalekin eta  $p = 12$  ordeneko metodoa  $s = 25$  atalekin.

*IRK* metodoak *ERK* metodoak baino modu errazagoan eraiki daitezke. *Butcher sinplifikazio baldintzen* [5] arabera definitzen dira,

$$B(p) : \sum_{i=1}^s b_i c_i^{q-1} = \frac{1}{q}, \quad q = 1, \dots, p.$$

$$C(\eta) : \sum_{j=1}^s a_{ij} c_j^{q-1} = \frac{c_i^q}{q}, \quad i = 1, \dots, s, \quad q = 1, \dots, \eta.$$

$$D(\zeta) : \sum_{i=1}^s b_i c_i^{q-1} a_{ij} = \frac{b_j}{q} (1 - c_j^q), \quad j = 1, \dots, s, \quad q = 1, \dots, \zeta.$$

Aurreko taulan (Taula 2.1) *IRK* metodo ezagunenak laburtu ditugu. Gauss metodoa  $s$ -ataletako *IRK* metodoen artean  $p = 2s$  ordena duen metodo bakarra dugu. Gauss metodoa sinplektikoa eta simetrikoa da.

1. Sinplektizidade baldintza [18].

$$b_i a_{ij} + b_j a_{ji} - b_i b_j = 0, \quad 1 \leq i, j \leq s \quad (2.16)$$

2. Koefiziente simetrikoak.

$$b_i = b_{(s-i+1)}, \quad i = 1, 2, \dots, \lceil s/2 \rceil \quad (2.17)$$

$$c_{(s-i+1)} = 1 - c_i, \quad i = 1, 2, \dots, \lceil s/2 \rceil \quad (2.18)$$

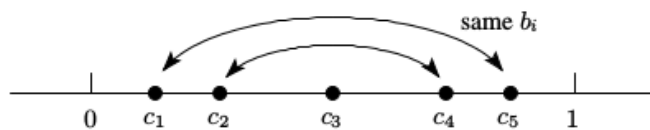


Fig. 2.1. Symmetry of collocation methods

(a) Kolokazio metodoen simetria.

Irudia 2.2: Kolokazio metodoen simetria.

**Adibidea.**  $s = 1$ ,  $s = 2$  eta  $s = 3$  ataletako Gauss metodoak.

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

$$\begin{array}{c|ccc} \frac{1}{2} - \frac{\sqrt{15}}{10} & \frac{5}{36} & \frac{2}{9} - \frac{\sqrt{15}}{15} & \frac{5}{36} - \frac{\sqrt{15}}{30} \\ & \frac{5}{36} + \frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36} - \frac{\sqrt{15}}{24} \\ \frac{1}{2} + \frac{\sqrt{15}}{10} & \frac{5}{36} + \frac{\sqrt{15}}{30} & \frac{2}{9} + \frac{\sqrt{15}}{15} & \frac{5}{36} \\ \hline & \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \end{array}$$

*IRK* metodoetan, erroka handiena ekuazio-sistema ez-linealaren zenbakizko soluzio eraginkorra exekutatzea da. Nonstiff problemetarako, atalen hasieraketa ( $Y_i^{[0]}$ ) egoki bat duen puntu-finkoko iterazioa erabil daiteke. Stiff problemetarako, puntu-finkoa iterazioak urrats tamaina txikiegia izatea behartuko luke eta ondorioz, Newton sinplifikatua erabili ohi da.

**IRK algoritmo orokorra.**

```

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

2.  $F_{n,i} = f(Y_{n,i})$ .

Atal bakoitzarentzat ekuazio diferentzialaren balioztapena independentea da eta modu paraleloan exekutatu daiteke.

3.  $Y_{n,i} = y_{n-1} + h \sum_{j=1}^s a_{ij} F_{n,j}$  ,  $i = 1, \dots, s$ .

Ekuazio-sistema ez lineala metodo iteratibo bat erabiliz askatu behar da. Metodo hau, Puntu-finkoaren metodoa edo Newton sinplifikatuaren metodoa izan daiteke.

(a) Puntu-finkoko iterazioa.

```

for ( $k=1,2,\dots$ ) do
     $Y_i^{[k]} = y_{n-1} + h \sum_{j=1}^s a_{ij} f(Y_i^{[k-1]});$ 
end

```

**ALGORITHM 2:** Main Algorithm

(b) Newton sinplifikatua.

Ekuazio-linealaren matrizea  $M = I_s \otimes I_d - hA \otimes J$ ,  $J = f'(y_n)$  definitzen dugu. Newton sinplifikatua puntu-finkoko iterazioa baino garestiagoa da, jakobiarra eta  $M$  matrizearen  $LU$ -deskonposaketa kalkulatu behar baititugu.

```

 $J = f'(y_n);$ 
 $M = I_s \otimes I_d - hA \otimes J;$ 
for ( $k=1,2,\dots$ ) do
     $M \Delta Y^{[k-1]} = -F(Y^{k-1});$ 
     $Y^{[k]} = Y^{[k-1]} + \Delta Y^{[k-1]};$ 
end

```

**ALGORITHM 3:** Main Algorithm

$$4. \quad y_n = y_{n-1} + h \sum_{i=1}^s b_i F_{n,i}$$

Integrazio luzeak direnean, urrats asko ematen dira eta koma higikorrek aritmetika dela eta, doitasun galera ekiditeko batura konpensatu teknika erabili ohi da.

**Bigarren ordeneko ekuazio diferentzialak.**

Era honetako ekuazio diferentzialak ,

$$\dot{u} = f(v), \quad \dot{v} = g(u),$$

garrantzitsuak dira. Esaterako, bigarren ordeneko ekuazio diferentziala  $\ddot{y} = f(y)$  eta Hamiltondar banagarriak  $H(q, p) = T(p) + U(q)$  hauen kasu partikularra ditugu.

Runge-Kutta metodoaren ekuazioak era honetan bilakatzen dira,

$$U_{i,n} = u_n + \sum_{i=1}^s a_{ij} f(V_{i,n})$$

$$V_{i,n} = v_n + \sum_{i=1}^s a_{ij} g(U_{i,n})$$



Kasu honetan Gauss-Seidel iterazioa, puntu-finkoaren iterazio estandarra baino hobea da,

```

for ( $k=1,2,\dots$ ) do
     $U_i^{[k+1]} = u_n + h \sum_{j=1}^s a_{ij} f(V_i^{[k]});$ 
     $V_i^{[k]} = v_n + h \sum_{j=1}^s a_{ij} g(U_i^{[k+1]});$ 
end

```

**ALGORITHM 4:** Main Algorithm

Bigarren ordeneko ekuazio diferentzialak ditugunean,

$$\dot{u} = f(v), \quad \dot{v} = u,$$

Gauss-Seidel iterazioa honakoa dugu,

```

for ( $k=1,2,\dots$ ) do
     $V_i^{[k+1]} = v_n + hc_i v_n + h^2 \sum_{j=1}^s \tilde{a}_{ij} f(V_i^{[k]});$ 
end

```

**ALGORITHM 5:** Main Algorithm

### 2.3.2 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 interpolatzailer batek modu jarraian emandako soluzioa adierazten duela. Honako definizioa emango dugu,

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

$$u(t_0) = y_0,$$

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

Orduan soluzioa  $y_1 = u(t_0 + h) \approx y(t_0 + h)$  da..

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

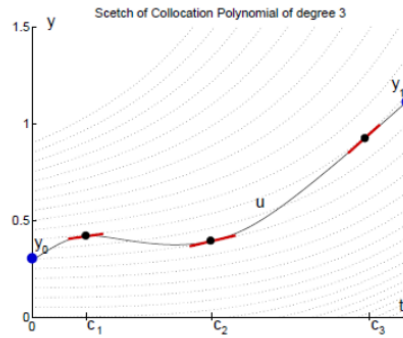
$$a_{ij} = \int_0^{c_i} l_j(\tau) d\tau, \quad b_i = \int_0^1 l_i(\tau) d\tau \quad (2.19)$$

non  $l_i(\tau)$  Lagrangiarr 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,

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

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



(a) kolokazio metodoak.

Irudia 2.3: kolokazio metodoak.

Gauss metodoaren koefizienteak kalkulatzeko bi aukera ditugu. Lehen aukera, *Mathematica*-k NDSolve paketearen inplementatuta dagoen koefizienteak kalkulatzeko funtzioa: `ImplicitRungeKuttaGaussCoefficients[]`. Bigarren aukera, guk koefizienteak kalkulatzeko garatutako funtzioa erabiltzea.

```
GaussCoefficients[s_Integer, doi_, a_Symbol, b_Symbol, c_Symbol]
:=
Module[{f, g, ff, glist, B, A},
  Do[c[i] = N[(Root[LegendreP[s, #] &, i] + 1)/2, doi] //
  Simplify, {i, s}];
  ff = Collect[InterpolatingPolynomial[
    Table[{c[i], f[i]}, {i, s}], x], f[_]];
  glist = Table[g[i] = Collect[ff, f[_],
```

```

Simplify[\\(\\(\\*
SubsuperscriptBox[\\(\\[Integral]\\), \\(0\\), \\(c[
i\\)\\]\\(\\(\\(\\[DifferentialD]x\\)\\)\\)\\) &], {i, 1, s}];
yy = Collect[ff, f[_], Simplify[\\(\\(\\*
\\*SubsuperscriptBox[\\(\\[Integral]\\), \\(0\\), \\(1\\)\\]\\(\\(\\(\\[DifferentialD]x\\)\\)\\)\\) &];
B = Table[b[i] = \\(\\(\\*
\\*SubscriptBox[\\(\\[PartialD]\\), \\(f[i\\)\\)\\)\\)\\)\\), {i, 1, s}];
A = Table[a[i, j] = D[g[i], f[j]], {i, 1, s}, {j, 1, s}];
{Array[c, s], B, A}
]

```

## 2.4 Konposizio metodoak.

### 2.4.1 Sarrera.

Konposizio metodoak, oinarritzko metodo bat edo gehiago konposatuz eraikitako zenbakizko integrazio metodoak dira. Oinarritzko metodoekin segidan exekututako azpi-urrats kopuru batek, konposizio metodoaren integrazioaren urrats bat osatzen du. Helburua orden baxuko metodo batetik abiatuta, orden altuko metodoa eraikitzea da; konposizio metodoak automatikoki konposatutako metodoaren propietateak (simetri, sinplektikoa,...) jasotzen ditu.

**Definizioa orokorra.**  $\phi_h$  oinarritzko metodoa eta  $\gamma_1, \dots, \gamma_s$  zenbaki errealak emanik, urrats luzera hauen  $\gamma_1 h, \gamma_2 h, \dots, \gamma_s h$  konposaketari dagokion konposizio metodoa,

$$\Psi_h = \phi_{\gamma_s h} \circ \dots \circ \phi_{\gamma_1 h}. \quad (2.20)$$

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

$$\gamma_1 + \dots + \gamma_s = 1,$$

$$\gamma_1^{p+1} + \dots + \gamma_s^{p+1} = 0. \quad (2.21)$$

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

```

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

```

**ALGORITHM 6:** Konposizio metodoak.

**Oharrak.** Konposizio metodoei buruzko hainbat ohar azpimarratuko ditugu:

1. Esplizitua.

Konposizio metodo hauek esplizituak dira. Metodo hauetan ez da ekuazio sistematik askatu behar, eta kalkuluak azkarrak dira.

2. Sekuentziala.

Urrats bakoitzaren kalkuluak modu sekuentzialean egin behar ditugu.

3. Memoria gutxi.

Ez da tarteko baliorik gorde behar eta datu-egitura berezirik memorian gorde behar.

4. 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.

Konposizio metodo eraginkorra  $s$ -atal gutxirekin,  $p$  orden altuko metodoa dugu. Sofronio eta Spalettaren (2004),  $s = 35$  eta  $p = 10$  ordeneko metodoa [28], orain arteko lortutako orden altueneko konposizio metodo eraginkorrena kontsideratu daiteke. *IRK* metodoa konposizio metodo honekin alderatuko dugu eta beraz, jarraian modu laburrean konposizio metodo hau azalduko dugu.

## 2.4.2 CO1035: 10 ordeneko konposizio metodoa.

Konposizio metodo simetrikoa da. Oinarrizko metodoa simetrikoa eta  $p = 2$  dela baliatuz eraikitako metodoa da.

Taula 2.2: 10 ordeneko metodoa konposizio metodoa (CO1035).

Koefiziente	Balioa
$\gamma_1 = \gamma_{35}$	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
$\gamma_7 = \gamma_{29}$	0.07497334315589143566613711
$\gamma_8 = \gamma_{28}$	0.11199342399981020488957508
$\gamma_9 = \gamma_{27}$	0.36613344954622675119314812
$\gamma_{10} = \gamma_{26}$	-0.39910563013603589787862981
$\gamma_{11} = \gamma_{25}$	0.10308739852747107731580277
$\gamma_{12} = \gamma_{24}$	0.41143087395589023782070412
$\gamma_{13} = \gamma_{23}$	-0.00486636058313526176219566
$\gamma_{14} = \gamma_{22}$	-0.39203335370863990644808194
$\gamma_{15} = \gamma_{21}$	0.05194250296244964703718290
$\gamma_{16} = \gamma_{20}$	0.05066509075992449633587434
$\gamma_{17} = \gamma_{19}$	0.04967437063972987905456880
$\gamma_{18}$	0.04931773575959453791768001

**Gure implementazioa.** Gure abiapuntua Haireren konposizio metodoaren Fortran kodea izan da. Konposizio metodoaren azalpenak liburuko [13] II.4 eta V.3 ataletan ematen ditu. *GNI-Comp* izeneko kodea eskuragarri dago [?] helbidean. Kodearen C-lengoiako bertsioa garatu dugu.

$\phi_h$  metodoa  $p = 2$  ordenekoa eta simetrikoa izanik, era honetako konposizioak aurkitu dira,

$$\Psi_h = \phi_{\gamma_s h} \circ \phi_{\gamma_{s-1} h} \circ \cdots \circ \phi_{\gamma_2 h} \circ \phi_{\gamma_1 h} \quad (2.22)$$

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

**Koefizienteak.** Konposizio metodoaren oinarritzko metodoa *Stömer-Verlet*  $\phi_h = \varphi_{h/2}^1 \circ \varphi_h^2 \circ \varphi_{h/2}^1$  dugula kontutan hartuta,

$$\Psi_h = (\varphi_{h\gamma_s/2}^1 \circ \varphi_{h\gamma_s}^2 \circ \varphi_{h\gamma_s/2}^1) \circ \cdots \circ (\varphi_{h\gamma_2/2}^1 \circ \varphi_{h\gamma_2}^2 \circ \varphi_{h\gamma_2/2}^1) \circ (\varphi_{h\gamma_1/2}^1 \circ \varphi_{h\gamma_1}^2 \circ \varphi_{h\gamma_1/2}^1)$$

Beraz jarraian dauden  $\varphi^1$  fluxuak elkartuz,

$$\Psi_h = \varphi_{ha_{s+1}}^1 \circ \varphi_{hb_s}^2 \circ \cdots \circ \varphi_{ha_3} \circ \varphi_{hb_2}^2 \circ \varphi_{ha_2}^1 \circ \varphi_{hb_1}^2 \circ \varphi_{ha_1}^1$$

non  $a_1 = a_{s+1} = \gamma_1/2$ ,  $b_i = \gamma_i$ ,  $a_k = (\gamma_k + \gamma_{k-1})/2$ ,  $i = 1, \dots, s$  eta  $k = 2, \dots, s$ .

Tarteko urratsetan, lehen atala  $\varphi_{ha_1}^1$  eta azkena  $\varphi_{ha_1}^1$  bakar batean elkar daitezke,

$$\Psi_h = \varphi_{h2a_{s+1}}^1 \circ \varphi_{hb_s}^2 \circ \cdots \circ \varphi_{ha_3} \circ \varphi_{hb_2}^2 \circ \varphi_{ha_2}^1 \circ \varphi_{hb_1}^2.$$

## 2.5 Splitting metodoak.

### 2.5.1 Sarrera.

*Splitting metodoak*, bektore eremua  $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$  sistema osoa integratzeko baino errazagoa diren azpiproblemetan,  $f^{[i]}$  ( $f = \sum_{i=1}^m f^{[i]}$ ), deskonposatu daitezkeen ekuazio diferentzialetarako zenbakizko integrazioak dira.

Maiz, jatorrizko  $\dot{y} = f(y)$  problema era honetan bana daiteke,

$$\dot{y} = f^{[1]}(y) + f^{[2]}(y), \quad (2.23)$$

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

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

$$\phi_h = \varphi_h^{[1]} \circ \varphi_h^{[2]} \quad \text{edo} \quad \phi_h^* = \varphi_h^{[2]} \circ \varphi_h^{[1]}. \quad (2.24)$$

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

$$\phi_h = \varphi_{\frac{h}{2}}^{[1]} \circ \varphi_h^{[2]} \circ \varphi_{\frac{h}{2}}^{[1]} \quad (2.25)$$

**Splitting metodo orokorrak.** Konposizio metodoen modu berean, oinarritzko Splitting metodoak konposatuz orden altuagoko metodoak lortzen dira,

$$\Psi_h = \varphi_{a_{s+1}h}^{[1]} \circ \varphi_{b_sh}^{[2]} \circ \varphi_{a_sh}^{[1]} \circ \cdots \circ \varphi_{a_2h}^{[1]} \circ \varphi_{b_1h}^{[2]} \circ \varphi_{a_1h}^{[1]}. \quad (2.26)$$

$a_i, b_i$  koefizienteek metodoaren ordena definitzen dute. Metodoa simetrikoa bada  $\Psi_h = \Psi_h^*$ ,

$$a_1 = a_{s+1}, \quad b_1 = b_s, \quad a_2 = a_s, \quad b_2 = b_{s-1}, \dots$$

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

```

for  $n \leftarrow 1$  to endstep do
     $Y_{0,n} = y_{n-1}$ ;
    for  $i=1,2,\dots,s$  do
         $Y_{i,n} = (\varphi_{b_ih}^{[2]} \circ \varphi_{a_ih}^{[1]})(Y_{i-1,n})$ ;
    end
     $y_n = Y_{m,n}$ ;
end

```

**ALGORITHM 7:** Splitting metodoak.

**Oharrak.** Splitting metodoen algoritmoan, konposizio metodoen algoritmoen buruzko ezaugarri berdinak errepikatu beharko genituzke.

**Fluxu zehatza eta zenbakizko fluxua konbinatuz.** Demagun sistema  $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$  era honetan banatzen dugula,

$$\dot{\mathbf{y}} = \mathbf{f}^{[1]}(\mathbf{y}) + \mathbf{f}^{[2]}(\mathbf{y}), \quad (2.27)$$

Taula 2.3:  $SABA_4$  splitting metodoa.

Koefiziente	Balioa	Koefiziente	Balioa
$c_1$	$\frac{1}{2} - \frac{\sqrt{525+70\sqrt{30}}}{70}$	$d_1$	$\frac{1}{4} - \frac{\sqrt{30}}{72}$
$c_2$	$\frac{(\sqrt{525+70\sqrt{30}} - \sqrt{525-70\sqrt{30}})}{70}$	$d_2$	$\frac{1}{4} + \frac{\sqrt{30}}{72}$
$c_3$	$\frac{\sqrt{525-70\sqrt{30}}}{35}$		

Suposatu bakarrik  $\dot{\mathbf{y}} = \mathbf{f}^{[1]}(\mathbf{y})$  sistemaren fluxua zehatza  $\varphi_t^{[1]}$  kalkulatu daitekeela eta  $\phi_t^{[2]}$ ,  $\dot{\mathbf{y}} = \mathbf{f}^{[2]}(\mathbf{y})$  sistemari aplikatutako zenbakizko integrazio dugula. Orduan konposizio metodoaren oinarriko metodoa honakoa kontsideratu daiteke,

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

Splitting metodoak konposizio metodoen interpretazioa emanez,

$$\Psi_h = \varphi_{a_s h}^{[1]} \circ \phi_{a_s h}^{[2]} \circ \phi_{b_s h}^{[2]*} \circ \varphi_{(b_s+a_s-1)h}^{[1]} \circ \phi_{a_s h}^{[2]} \circ \dots \circ \phi_{b_1 h}^{[2]*} \circ \varphi_{b_1 h}^{[1]}. \quad (2.28)$$

## 2.5.2 Eguzki-sistemari egokitutako splitting metodoak.

Honakoa dugu N-gorputzeko problema grabitazionalaren Hamiltondarra,

$$H(p, q) = T(p) + U(q).$$

Bi koordenatu sistema (*Jacobi* edo koordenatu Heliozentrikoak) erabiliaz, Hamiltondarra beste modu honetan berridatzi daiteke,

$$H = H_K + H_I, \quad |H_I| \ll |H_K|,$$

non alde nagusia  $H_K$  planeta bakoitzaren eguzki inguruko mugimendu kepleriarra den eta  $H_I$  aldiz, planeten arteko interakzioek eragiten duten perturbazio txikia.

Eguzki-sistemaren N-gorputzeko problema grabitazionalari egokitutako zenbakizko bi integratzaile sinplektiko azalduko ditugu. Lehen, J.Laskarrek eta P.Robutelek [20] definitutako  $SABAC_4$  integratzailea eta bigarrena, Blanes-ek [3] [10] definitutako  $ABAH1064$  integratzailea.

1. Laskarren (2001)  $SABAC_4$  zenbakizko integratzailea [20].  $SABA_4$  integratzailea definitzen duten koefizienteak (taula 2.3).



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

$$SABA_4 = \varphi_{c_1h}^{[A]} \circ \varphi_{d_1h}^{[B]} \circ \varphi_{c_2h}^{[A]} \circ \varphi_{d_2h}^{[B]} \circ \varphi_{c_3h}^{[A]} \circ \varphi_{d_2h}^{[B]} \circ \varphi_{c_2h}^{[A]} \circ \varphi_{d_1h}^{[B]} \circ \varphi_{c_1h}^{[A]}.$$

**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 ordeztu honen hurbilpen bat erabiliko dugu,

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

$ABAH1064$ ,  $p = 10$  eta  $s = 9$  splitting metodoa aztertuko dugu,

$$ABAH1064 = \prod_{i=1}^s \varphi_{a_ih}^K \circ \tilde{\varphi}_{b_ih}^I$$

non  $a_i, b_i$  koefizienteak beheko taulan (taula 2.4) definitzen diren.

## 2.6 Kepler fluxua.

**Bi gorputzen problema.** Kepler problemari dagokion Hamiltondarra,

$$H(\mathbf{p}, \mathbf{q}) = \frac{\mathbf{p}^2}{2m} - \frac{\mu}{\|\mathbf{q}\|}. \quad (2.29)$$

Taula 2.4: *ABAH1064* splitting metodoa.

Koefiziente	Balioa
$a_1 = a_9$	0.04731908697653382270404371796320813250988
$a_2 = a_8$	0.2651105235748785159539480036185693201078
$a_3 = a_7$	-0.009976522883811240843267468164812380613143
$a_4 = a_6$	-0.05992919973494155126395247987729676004016
$a_5$	0.2574761120673404534492282264603316880356
$b_1 = b_9$	0.1196884624585322035312864297489892143852
$b_2 = b_8$	0.3752955855379374250420128537687503199451
$b_3 = b_7$	-0.4684593418325993783650820409805381740605
$b_4 = b_6$	0.3351397342755897010393098942949569049275
$b_5$	0.2766711191210800975049457263356834696055

Elkar erakartzen diren bi gorputzen mugimendua kalkulatzeko, gorputz baten kokapena koordinatu sistemaren jatorria kontsideratuko dugu. Honen arabera,  $m = (1/m_1 + 1/m_2)^{-1}$  eta  $\mu = Gm_1m_2$  definituko ditugu.

Hamiltondarrari dagokion bigarren ordeneko ekuazio diferentzialak,

$$\ddot{\mathbf{q}} = -\frac{k\mathbf{q}}{\|\mathbf{q}\|^3}, \quad (2.30)$$

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

**Idea nagusia.** Koordinatu kartesiarretatik koordinatu eliptikoetara  $(a, e, i, \Omega, E)$  itzulpena egingo dugu. Kontutan hartuta  $E$  (izena??) aldagai ezik beste aldagaiek konstante mantentzen direla,  $E_0$  abiapuntua harturik,  $\Delta t$  denbora tarte aurrera egingo dugu  $E_1$  balioa berria kalkulatzeko. Azkenik, koordinatu eliptikoetatik koordinatu kartesiarrak berreskuratuko ditugu kokapen eta abiadura berriekin.

$$(\mathbf{q}_0, \mathbf{v}_0) \in \mathbb{R}^6 \longrightarrow (\mathbf{a}, \mathbf{e}, \mathbf{i}, \Omega, E_0) \in \mathbb{R}^6$$

$$\downarrow \Delta t$$

$$(\mathbf{q}_1, \mathbf{v}_1) \in \mathbb{R}^6 \longleftarrow (\mathbf{a}, \mathbf{e}, \mathbf{i}, \Omega, E_1) \in \mathbb{R}^6$$

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

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

htb

Taula 2.5: 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

$$\Delta E^{[k+1]} = \Delta E^{[k]} - \frac{f(\Delta E^{[k]})}{f'(\Delta E^{[k]})} \quad (2.31)$$

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

## 2.7 Laburpena.

Hauek dira Eguzki sistemaren integratzaile konparatuko ditugun metodoak,



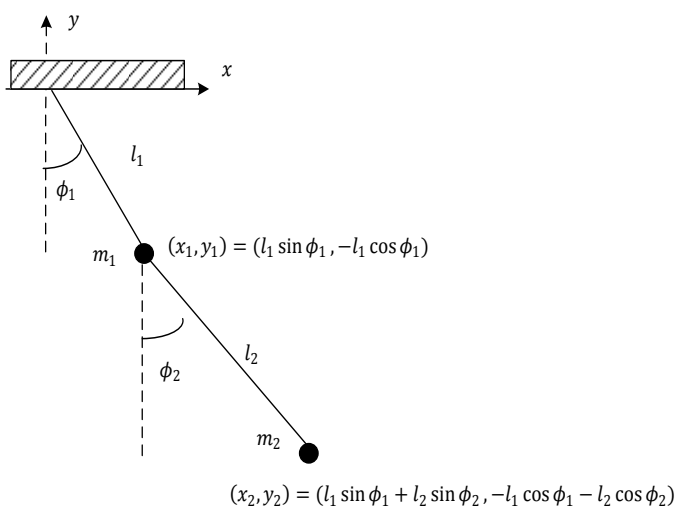
# Kapitulua 3

## 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 ( $\Theta_1, \Theta_2$ ) eta dagokion momentuak ( $P_1, P_2$ ) dira.



Irudia 3.1: Pendulu bikoitza.

### 3.2.1 Ekuazioak.

**Hamiltondarra.**

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

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

$$\begin{aligned} C1 &= l_2^2 * m_2, \\ C2 &= l_1^2 * (m_1 + m_2), \\ C3 &= -2 * l_1 * l_2 * m_2, \\ C4 &= 2 * l_1^2 * l_2^2 * m_2 * m_1, \\ C5 &= 2 * m_1^2 * l_2^2 * m_2^2, \\ C6 &= g * l_1 * (m_1 + m_2), \\ C7 &= g * l_2 * m_2. \end{aligned}$$

**Ekuazio diferentzialak.**

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

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

$$\dot{P}_1 = -(aux4 + C6 * \sin(Q1)),$$

$$\dot{P}_2 = (aux4 - C7 * \sin(Q2)),$$

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

### 3.2.2 Hasierako balioak.

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

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

**Hasierako balioak.** Pendulu bikoitza izaera kaotikoa duen sistema ez-lineala da. Zentzu honetan bi hasierako balio ezberdin kontsideratu ditugu [27]:

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 paketea 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,  $\mathbf{q}_i, \mathbf{p}_i \in \mathbb{R}^3, m_i \in \mathbb{R}, i = 0, \dots, N$  gorputz bakoitzaren kokapena, momentua eta masa dira. Bestalde, momentua era honetan definituko dugu  $\mathbf{p}_i = m_i * \mathbf{v}_i$  non  $\mathbf{q}_i = \mathbf{v}_i$  den.

### 3.3.1 Zailtasunak.

Kontutan hartzekoa da eguzki-sistemaren integrazioaren urrats kopuru handia. Eguzki-sistemaren bizi iraupena  $5 \times 10^9$  urtetakoa izanik eta integrazioetako ohiko urratsa  $h = 0.0025$  urteko bada (orbita txikiaren periodoaren %1), eman beharreko urrats kopurua  $2 \times 10^{12}$  izango da. Era berean, kanpo planeten integrazioa  $5 \times 10^{10}$  urrats kopuru ingurukoa da.

Eguzki-sistemaren gorputzen denbora eskala oso ezberdinak dira. Gainera ilar-gia kontsideratuko bagenu, lurrarekiko distantzia  $D_M = 3.844 \times 10^8$  eta periodoa  $P_M = 27.32$  egunekoa.

Taula 3.1

Planeta	Axis a AU	Periodoa years
Merkurio	0.39	0.24
Venus	0.72	0.007
Earth	1.00	1.007
Mars	1.52	1.88
Jupiter	5.20	11.86
Saturn	9.54	29.42
Uranus	19.19	83.75
Neptune	30.06	163.72
Pluto	39.53	248.02

### 3.3.2 Ekuazio barizentrikoak.

**Hamiltondarra.**

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

**Ekuazio diferentzialak.**

$$\dot{q}_i = v_i, \quad i = 0, 1, \dots, N \quad (3.2)$$

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



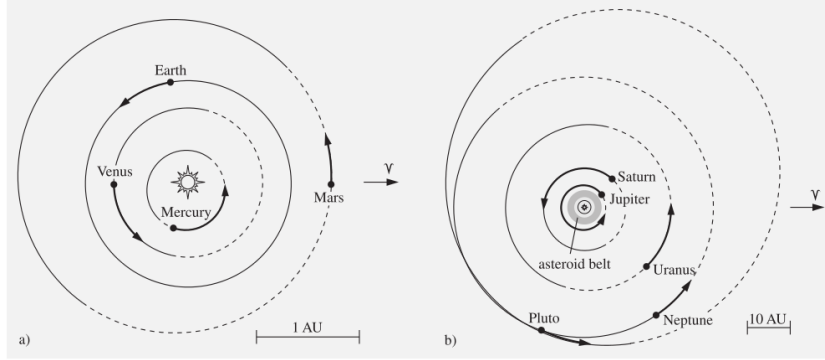


Fig. 7.1. (a) Planetary orbits from Mercury to Mars. The dashed line represents the part of the orbit below the ecliptic; the arrows show the distances travelled by the planets during one month (January 2000). (b) Planets from Jupiter to Neptune and the dwarf planet Pluto. The arrows indicate the distances travelled by the planets during the 10 year interval 2000–2010.

(a) Eguzki-sistema.

Irudia 3.2: Eguzki-sistema.

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

$$\dot{q}_i = v_i, \quad i = 0, 1, \dots, N \quad (3.4)$$

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

### 3.3.3 Ekuazio Heliozentrikoak.

Ohikoa da ekuazio diferentzialak koordenatu heliozentrikoen (eguzkiaren zentroa-  
rekiko) arabera definitzea. Argitu koordenatu heliozentrikoak ere modu ezberdi-

Taula 3.2: Konstanteak

c	299792.458 km/s	Argiaren abiadura
au	149597870.700 km	Astronomical unit
$\beta$	1.0	PPN parametroa
$\gamma$	1.0	PPN parametroa

nean eman daitezkeela eta guk koordenatu heliozentriko kanonikoak deiturikoak erabili ditugula.

$$\dot{\mathbf{Q}}_i = \mathbf{V}_i + \sum_{j \neq i, j=1}^N \frac{\mathbf{V}_j m_j}{(m_0 + m_j)}, \quad i = 1, \dots, N.$$

$$\dot{\mathbf{V}}_i = -G \frac{(m_0 + m_i)}{\|\mathbf{Q}_i\|^3} \mathbf{Q}_i - G \frac{(m_0 + m_i)}{m_0} \sum_{j \neq i, j=1}^N \left( \frac{m_j}{\|\mathbf{Q}_i - \mathbf{Q}_j\|^3} (\mathbf{Q}_i - \mathbf{Q}_j) \right), \quad i = 1, \dots, N.$$

Ekuazioen garapen osoa eranskinean eman dugu.

### 3.3.4 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 [11],

**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 ( $V$ ),

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

Eta integrazio hasierako balioak,

$$q_{new_i} = q_i - R, \quad v_{new_i} = v_i - V, \quad i = 0, \dots, N.$$

Taula 3.3: 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

Taula 3.4: 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

### 3.3.5 Kodeak.

Mathematicako NBodyProblem.m paketea honako funtzioak garatu ditugu.

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

-

C-lengoiako inplementazioa:

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

Taula 3.5: 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	$0.109318945074237400E - 10$

## Kapitulua 4

# Koma Higikorreko Aritmetika.

### 4.1 Sarrera.

Gaur-egungo konputagailuetan, IEEE-754 estandarraren arabera 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 erroreak 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  $\phi : \mathbb{F} \rightarrow W$  koma-higikorreko adierazpen funtzioa.

$$\begin{aligned}\mathbb{F} &\subset \mathbb{R}, \\ \mathbb{F} &= \{x \in \mathbb{R} \mid \phi(x) \in W\}.\end{aligned}\tag{4.1}$$

$\mathbb{F}$  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-754 estandarraren arabera,  $n$ -biteko koma-higikorrezko adierazpenak bi zati ditu,



Taula 4.1

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

*IEEE-785 estandarrean honako formatu bitarrak definitzen dira:*

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

*2008. urtean, IEEE-785 estandarrak 128 biteko koma-higikorreko aritmetika onartu zuen. Quadruple aritmetika softwarez implementatuta 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 garapenerako.*

*Bestalde, badaude doitasun arbitrariotan lan egiteko beste lan-ingurune (Problem Solving Environment [17]) 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.

*Zenbakizko integrazioen errorea, trunakatze eta biribiltze erroreez osatuta dago. Urrats tamaina adina txikia aukeratuz, trunakatze errorea biribiltze errorea baino txikiago izango da eta beraz, zenbakizko integrazioaren errorea biribiltze erroreen eragina nagusituko da. Biribiltze errorea gutxitzea funtsezkoa izango da epe luzeko integrazioetan.*

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

#### Adierazpen errorea.

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

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

eta errore erlatiboa,

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

Aurreko bi definizioen ondorioz honako formula erabilgarria dugu,

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

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

### Eragiketen errorea.

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

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

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

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

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

$$\Delta z = \tilde{z} - z = (\tilde{x} \otimes \tilde{y}) - (\tilde{x} * \tilde{y})$$

$$\delta z = \frac{\Delta z}{z} = \frac{(\tilde{x} \otimes \tilde{y}) - (\tilde{x} * \tilde{y})}{(\tilde{x} * \tilde{y})}$$

Modu berean honako erlazio dugu,

$$\tilde{z} = (\tilde{x} \otimes \tilde{y}) = z + \Delta z = z(1 + \delta z),$$

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

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

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

### 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; edo magnitude oso ezberdinak diren bi zenbaki gehitzen ditugunean.



```

>> 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.

```

## Biderketa.

Orokorrean,  $m$  digituzko bi mantisen biderketaren emaitza zehatza adierazteko,  $2m$  digituzko mantisa behar dugu ( $m$  digituzko galera) [12]. Biderkagaietako bat  $2$ ren berredura bat bada, orduan biderketa zehatza da baina oro har,  $m$  digitu galduko ditugu.

*Adibidea.*

$$1,343 \times 2,103 = 2,824229$$

$$1,343 \times 2,103 \approx 2.824$$

## 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 txarrean, 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,

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

Eta  $n = 3$  deneko adibidean,

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

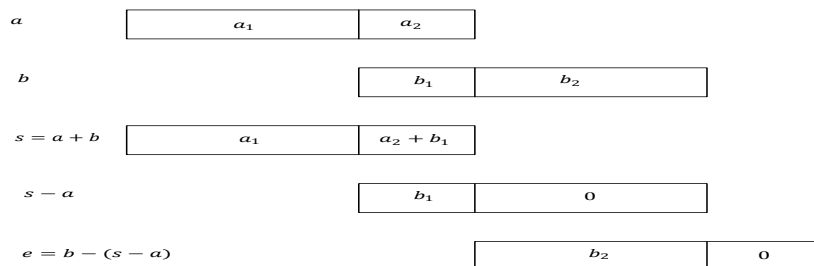
### 4.3.1 Tresnak.

#### Batura konpensatua.

Batura errekursiboetan, biribiltze errorea gutxitzeko metodoa dugu [16]. 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  $\tilde{z} = \tilde{x} \oplus \tilde{y}$ ,

$$\tilde{e} = -\left( ((\tilde{x} \oplus \tilde{y}) \ominus \tilde{x}) \ominus \tilde{y} \right) = (\tilde{x} \ominus \tilde{z}) \oplus \tilde{y}$$



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 ( $e$ ) kalkulatu 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 8:** Batura konpensatua.

#### Biderketaren errore estimazioa.

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

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

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

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

#### Sterbenz Teorema.

*Sterbenz teoremaren arabera [29], bi zenbaki elkarrekiko gertu daudenean, horien arteko kendura zehatza da.*

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

## 4.4 Laburpena.

*Koma-higikorreko aritmetikan sakontzeko honako biografia azpimarratuko dugu, Overton [26], Muller [24] eta Corless [8].*



# Kapitulua 5

## Zientzia konputazioa.

### 5.1 Sarrera.

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

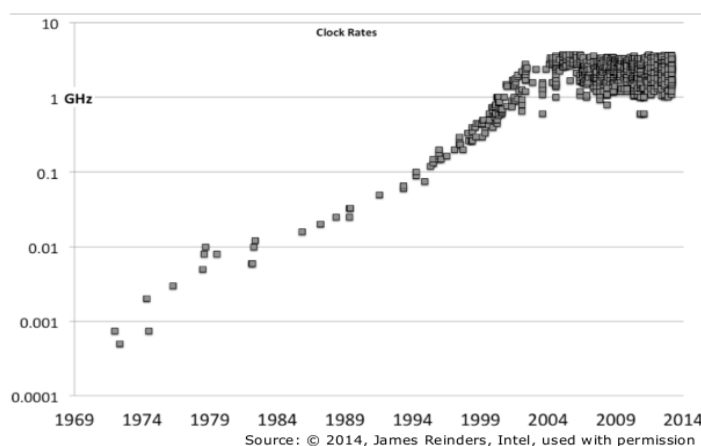
**Moore's Law (1965)** . *Processor speed doubles every 18 months.*

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

*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 diseinatzeko erronkak:*

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



Irudia 5.1: Processor clock rate growth halted around 2005.

*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.*

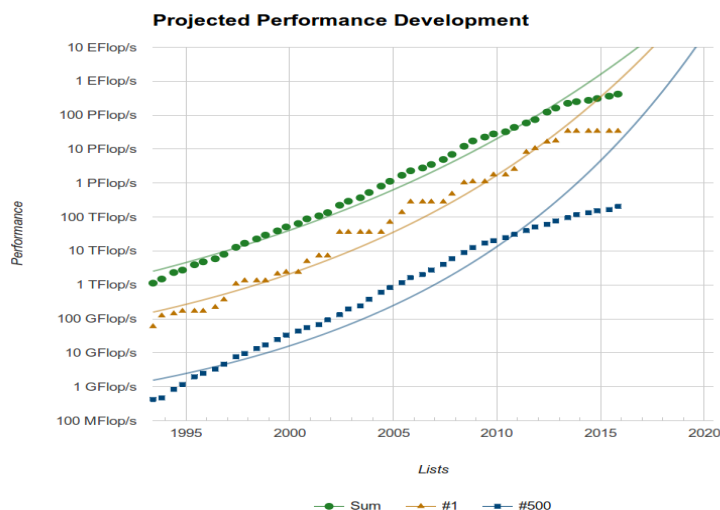
## 5.2 Parallel Hardware.

### 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).*



Irudia 5.2: [www.top500.org](http://www.top500.org), Top: total computing power of top 500 computers. Middle: 1 computer. Bottom: 500 computer.

**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.

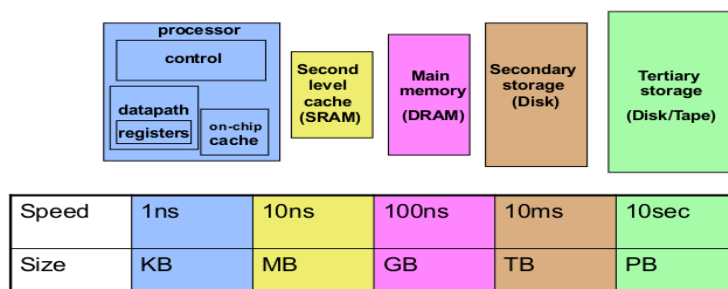
$$n = 1000 \Rightarrow n^3 = 10^{12}$$

$> 1000$  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.*



Irudia 5.3: 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 ( $a(1, 1), a(1, 2), \dots, a(1, 16)$ ). Honela, hurrengo 15 batura egiteko behar ditugun datuak Cachean eskuara izango ditugu memoria irakurketa berririk egin gabe.

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



```

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 9:** Main Algorithm

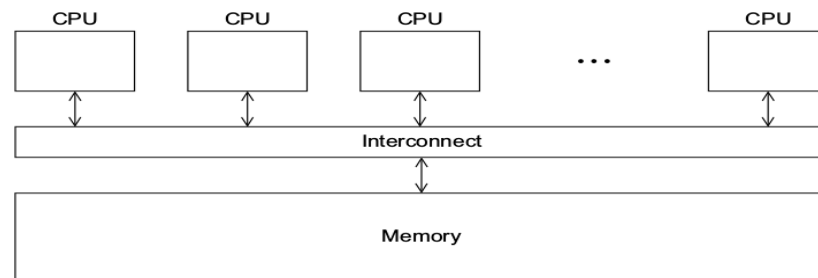
*CPUk datu bat behar duenean, memoria hierarkian zehar bilatuko du: lehenik L1 cachean, ondoren L2 cachean,...eta hauetan ez badago, memoria nagusira joko du. Memoria nagusi eta cache memoria arteko irakurketa eta idazketa guzti hauetan, informazio konsistentzia mantentzeko hainbat arau aurrera ematen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

**Hardware.**

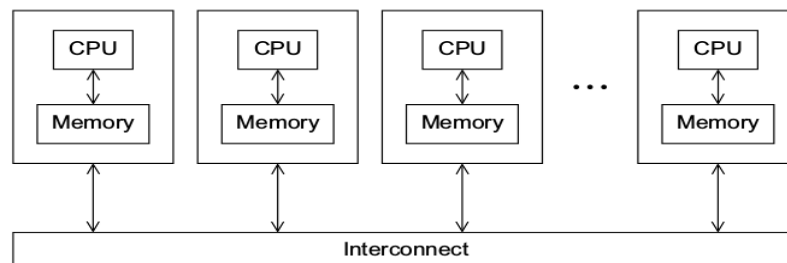
*MIMD (Multiple instruction, multiple data) sistemak, guztiz independenteak diren prozesadore multzoak osatzen dituzte. Bi dira MIMD sistema nagusiak: memoria konpartitutako eta memoria distribuitutako sistemak. Memoria konpartitutako sistemetan, prozesadore guztiek memoria osoa konpartitzen dute eta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

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

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



Irudia 5.4: Shared Memory System.



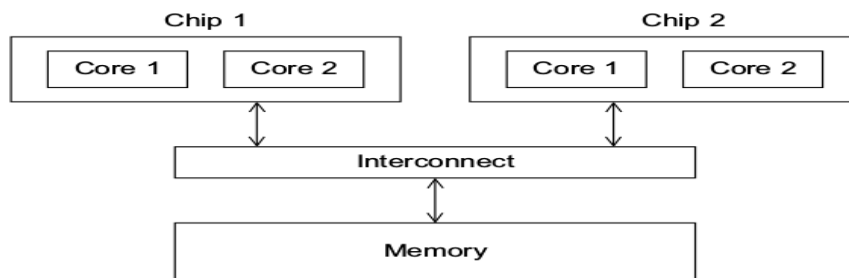
Irudia 5.5: Distributed Memory System.

## Sofwarea.

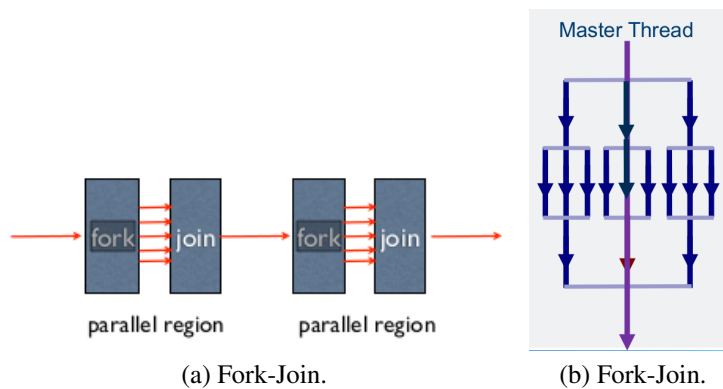
*C-lengoaia programazio paraleloan erabiltzeko, lengoiaren bi extensio dira nagusienak: bata memori-distribuitutako sistemetarako diseinatuta MPI (Message-Passing Inteface) eta bestea, memoria-konpartitutako sistemetarako diseinutakoa OpenMP (Open Specifications for MultiProcessing). MPI datu moten definizio, funtzio eta makroen liburegia da. OpenMP liburutegia bat eta C konpiladorearen aldaketa batzuk. OpenMP erabili dugu gure inplementaziorako eta jarraian honi buruzko ideiak 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.



Irudia 5.6: Shared Memory System (UMA).



Irudia 5.7: OpenMp programming model..

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

### Adibidea .

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

*OpenMP Version 4.5.*

*gcc -v (gcc version 4.8.4 (Ubuntu 4.8.4-2ubuntu1 14.04.3))*

### 5.3 Software liburutegiak.

*Matematika bi software errekurtsio 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.*

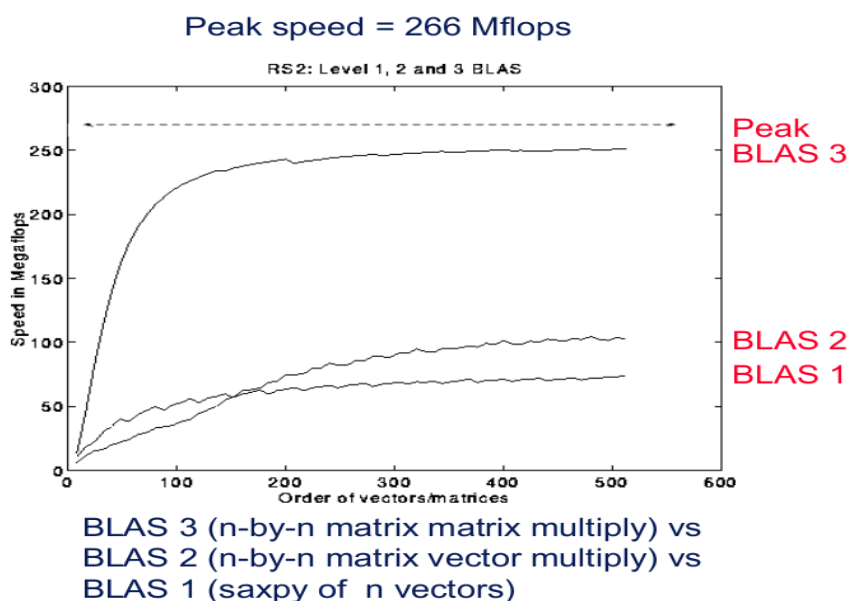
*Konputagailu hardware bakoitzerako optimizatutako bertsioak daude. Inplementazioa Fortranen egina dago eta datu-motei dagokionez:*

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:  $\approx \frac{2n^3}{4n^2} = \frac{n}{2}$ .*



Irudia 5.8: BLAS speeds.

Azpimarratu, *BLAS-1* eta *BLAS-2* funtzioen konputazio intentsitatea 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.

## 5.4 Kode Optimizazioak.

*Applications have two general challenges:*

1. *Numerical Method.*

*Performance required computation the shortest account of time.*

2. *Computer Program.*

*Express the algorithm as fast computer problem, you have realized computer hardware efficiently.*

*Optimization areas are:*

1. *Vectorization.*
2. *Parallelization.*
3. *Memory traffic control.*

*Optimization areas:*

1. *scalar optimization (compiler friendly practices).*
2. *vectorization (must use 16 or 8 wide vectors).*
3. *multi-threading (must scale to 100+ threads).*
4. *memory access (streaming access).*
5. *communication (offload, MPI traffic control).*

### 5.4.1 Scalar Tuning and General Optimization.

*Optimization of scalar arithmetics.*

*One of the most important scalar optimization techniques is **strength reduction** : replace expensive operation for less expensive operations.*

*Precision control.*

1. *Precision Control for transcendental functions.*
2. *Floating-point semantics.*
3. *Consistency of precision: constants and constants.*

*Using incorrect function names in the single precision is a common mistake*

**Strength Reduction**

**Common Subexpression Elimination.**

<pre> 1 for (int i = 0; i &lt; n; i++) { 2   A[i] /= B; 3 } </pre>	<pre> 1 const float Br = 1.0f/B; 2 for (int i = 0; i &lt; n; i++) 3   A[i] *= Br; </pre>
--	--

**Replace division with multiplication.**

<pre> 1 for (int i = 0; i &lt; n; i++) { 2   P[i] = (Q[i]/R[i])/S[i]; 3 } </pre>	<pre> 1 for (int i = 0; i &lt; n; i++) { 2   P[i] = Q[i]/(R[i]*S[i]); 3 } </pre>
--	--

**Use functions with Hardware support.**

<pre> 1 double r = pow(x2, -0.5); 2 double v = exp(x); 3 double y = y0*exp(log(x/x0)* 4               log(y1/y0)/log(x1/x0)); </pre>	<pre> 1 double r = 1.0/sqrt(x2); 2 double v = exp2(x*1.44269504089); 3 double y = y0*exp2(log2(x/x0)* 4               log2(y1/y0)/log2(x1/x0)); </pre>
--	--

Irudia 5.9: Optimization.

### 5.4.2 Optimization of vectorization.

1. *Preferably unid-stride access to data. Very important step in the optimization.*

Artikulua "Auto-Vectorization with the Intel Compilers".

Most CPU architectures today include Single Instruction Multiple Data (SIMD) parallelism in the form of a vector instruction set. Serial codes (i.e., running with a single thread), as well as instruction-parallel calculations (running with several threads) can take advantage of SIMD instructions and significantly increase the performance of some computations. Each CPU core performs SIMD operations on several numbers (integers, single or double precision floating-point numbers) simultaneously, when these variables are loaded into the processor's vector registers, and a vector instruction is applied to them. SIMD instructions include common arithmetic operations (addition, subtraction, multiplication and division), as well as comparisons, reduction and bit-masked operations (see, e.g., the list of SSE 2 intrinsics). Libraries such as the Intel Math Library provide SIMD implementations of common transcendental functions, and other libraries provide vectorized higher-level operations for linear algebra, signal analysis, statistics, etc.

2. *Data Alignment and Padding.*

An important consideration for efficient vectorization is data alignment.

### 5.4.3 Multi-threading.

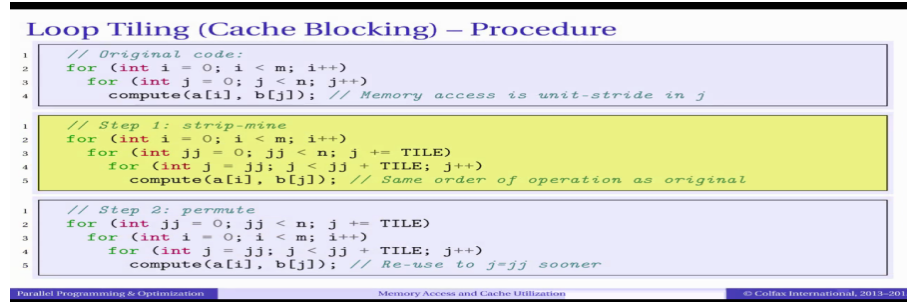
Do you have enough parallelism in your code?

Expanding iteration space, if it is not enough iterations in parallel loop.

Three layers of parallelism: MPI processes, OpenMP threads, vectorization.

### 5.4.4 Memory access.

*Memory access and Cache utilization.  
loop tiling technics*



Irudia 5.10: Optimization.

## 5.5 Laburpena.

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

*Algoritmo bat inplementatzen dugunean kontutan hartu beharrekoa:*

1. *Lerro edo zutabe arabera 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. Erabelean, 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 tarteak: 1550 – 2650.*

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

*Doitasuna: "QIVA" a quadruple 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 Numérique Planétaire de l'Observatoire de Paris).*

*Integrazio tarteak: .*

*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 times 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 improvement by using a single addition in simulated quadruple precision in the corrector step with a very small overhead.*

*Hardware: Intel Itanium II processors.*

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

*Integrazio tarteak: .*

*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.

*A.Morbidellik [23] eguzki-sistemaren zenbakizko integrazioen algoritmoen garapenaren azterketan, gaia hauek sailkatzen ditu:*

1. *The classical period.*  
90. *hamarkada hasiera arte,*
2. *The symplectic period.*
3. *The statistic period.*
4. *The planetary accretion period.*

*Wisdom eta Holmanen bere lanean [30, 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 kalkulatu 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 [21, 1994] SWIFT softwarea garatu zuten; Duncan, Levinson eta Lee-k [9, 1998] SYMBA softwarea garatu zuten; Chambers-ek [7] MERCURY softwarea garatu zuen. Berriki, Hernandez eta Bertschinger-ek [15, 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 oinarritzko metodoa leapfrog metodoa dugu. Metodo hau 2 ordeneko da. Orden altuagoko splitting eskemak : McLachlan [22, 1995], Laskar eta Robutel [20, 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 [14] 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 sinplektikoa ez izatea eragiten du.*
- 2. Puntu-finkoaren geratze irizpide estandarra dela eta, urrats bakoitzean erre 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.

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

$$\Delta^{[k]} = 0 \text{ or } \Delta^{[k]} \geq \Delta^{[k-1]}$$

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

```

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

```

**ALGORITHM 10:** 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 lortu-ko dugu. Bestetik, geratze irizpidean arazo batzuk topatu ditugu eta arazo hauek gainditzeko geratze irizpide sendoagoa garatu dugu. Beste bi proposamenak dagokionez, bata batura-kompensatuari erlazionatuta dago eta bestea biribiltze errorea monitorizatzeko proposamena da.*

*Bestalde kapitulu honen bukaeran, batetik interpolazio bidezko atalen hasiera-keta 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  $\tilde{a}_{ij}, \tilde{b}_i \in \mathbb{F}$  ordezkatzeraokan, sinpletizide baldintza ez da beteko,*

$$b_i a_{ij} + b_j a_{ji} - b_i b_j = 0, \quad 1 \leq i, j \leq s. \quad (7.1)$$

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

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

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

*non*

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

*Eta sinplektizidade baldintza modu honetan berriidatziko dugu,*

$$\mu_{ij} + \mu_{ji} - 1 = 0, \quad 1 \leq i, j \leq s. \quad (7.4)$$

*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.  $\mu_{ij}$  koefizienteak.

Batetik  $s$ -ataleko Gauss metodoetan,  $\tilde{\mu}_{ii} := \frac{1}{2}$ ,  $i = 1, \dots, s$ . 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.

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

2.  $b_i$  koefizienteak.

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

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

**7.3.2 Geratze irizpidea (2.proposamena).**

Ekuazio implizituaren (9.2) soluzioaren hurbilpena lortzeko puntu-finkoko iterazioa era honetan definituko dugu. Iterazioaren abiapuntua  $Y_i^{[0]}$  finkatu eta  $k = 1, 2, \dots$  iterazioetarako  $Y_i^{[k]}$  hurbilpenak lortu dagokigun geratze irizpidea bete arte.

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

IRK metodoaren inplementazio estandarrean geratze irizpidea honakoa da,

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

$$\|\Delta^{[k]}\| \leq tol \quad (7.8)$$

non  $\|\cdot\|$  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.

Gogoratu Hairer-ek proposatu zuen geratze irizpidea :  $\Delta^{[k]} = 0$  (puntu-finkora iritsi delako) ; edo  $\Delta^{[k]} \geq \Delta^{[k-1]}$  (biribiltze errorea nagusi delako). Orokorrean, geratze irizpide honek ondo funtzionatzen du baina batzuetan, iterazioak goizegi geratu direla konprobatu dugu. Gure iritziz, honen arrazoia da  $\Delta^{[k]} \geq \Delta^{[k-1]}$  biribiltze errorea nagusia dela adierazten duen arren, badago  $j \in \{1, \dots, sd\}$  osagairik,  $|\Delta_j^{[k]}| < |\Delta_j^{[k-1]}|$  hobetzeko tartea duena.

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

$$\exists j \in \{1, \dots, sd\}, |\Delta_j^{[1]}| > |\Delta_j^{[2]}| > \dots > |\Delta_j^{[k]}| > 0. \quad (7.9)$$

### 7.3.3 Batura konpensatua (3.proposamena).

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

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

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

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

non  $L_{n,i}$  ( $i = 1, \dots, s$ ) 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 konpensatu teknika aplikatzea.

```

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

```

**ALGORITHM 11:** 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-higikorrek hurbilpena izanik, lortutako errore estimazioa  $\tilde{e}_n$ , zehazki benetako biribiltze errorea da,

$$y_n = \tilde{y}_n + \tilde{e}_n. \quad (7.10)$$

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,

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

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 kalkulatu 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-higikorreko zenbakia) izanik, honako funtzioa definituko dugu,

```

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

```

**ALGORITHM 12:** floatR

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

$r < m$  zenbaki osoa finkatuta, bigarren integrazioaren puntu-finkoaren itera-  
zioa honela kalkulatu dugu,

$$L_i^{[k]} = hbif(Y_i^{[k-1]}), \quad Y_i^{[k]} = floatR\left(\tilde{y}_n \oplus (\tilde{e}_n \oplus \sum_{j=1}^s \mu_{ij} L_j^{[k]}), r\right). \quad (7.12)$$

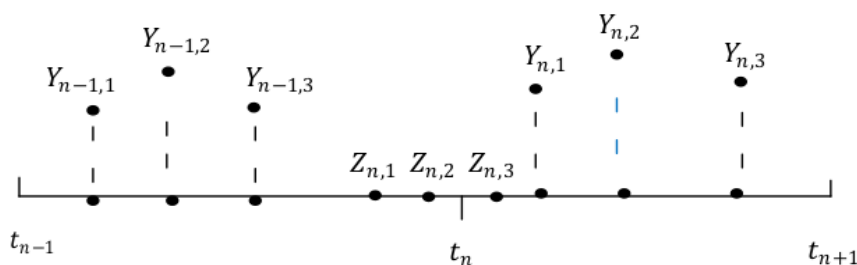
Biribiltze errorearen estimazioa, zenbakizko soluzio nagusiaren  $(y_n^{[main]} + e_n^{[main]})$   
eta  $r$  balio txiki baterako (adibidez  $r = 3$ ) kalkulatuak bigarren zenbakizko so-  
luzioaren  $(y_n^{[sub]} + e_n^{[sub]})$  arteko diferentzia bezala kalkulatu dugu.

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

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

### 7.3.5 Atalen hasieraketa.

Idea da, aurreko urratseko uneetako,  $(t_{n-1} + hc_i, Y_{n-1,i})$ ,  $i = 1, \dots, s$  eta  $(t_{n-1} + h, y_n)$ , balioei dagokien polinomio interpolatzailea erabiliz, urrats berriaren ata-  
len hasieraketa  $(t_n + hc_i, Y_{n,i}^{[0]})$ ,  $i = 1, \dots, s$  kalkulatzeko.



Irudia 7.1: Interpolazioa.

```

for  $n \leftarrow 1$  to  $endstep$  do
     $Y_n^{[0]} = G(Y_{n-1}, h);$ 
     $\dots;$ 
     $y_{n+1} = y_n + \delta_n;$ 

    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
     $\dots;$ 
     $\hat{y}_{n+1} = \hat{y}_n + \hat{\delta}_n;$ 

     $estimation_n = (y_n + e_n) - (\hat{y}_n - \hat{e}_n);$ 
end

```

**ALGORITHM 13:** RKG2: errore estimazioa

$(n - 1)$  urratseko informazioa erabiliz,

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

Dagokion polinomio interpolatzailea,

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

non  $l_i(t)$  Lagrangiar polinomioa dugu,

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

Eta beraz,

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

*Modu honetan s-ataletako IRK metodo bakoitzari dagokion  $\lambda_{ij}$  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 ← 1 to endstep do
    Hasieratu  $Y_{i,n}^{[0]}$  ,  $i = 1, \dots, s$ ;
    while (konbergentzia lortu) do
         $L_{n,i} = hb_i f(Y_{n,i})$  ,  $i = 1, \dots, s$ ;
         $Y_{n,i} = y_{n-1} + \sum_{j=1}^s \mu_{ij} L_{n,j}$  ,  $i = 1, \dots, s$ ;
    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 14:** Main Algorithm

*Eta puntu-finkoa erabiliz,*

```

e = 0;
for n ← 1 to endstep do
    k = 0;
    Hasieratu  $Y_i^{[0]}$ ;
    while (konbergentzia lortu) do
        k = k + 1;
         $L_i^{[k]} = hbif(Y_i^{[k-1]});$ 
         $Y_i^{[k]} = y_{n-1} + (e + \sum_{j=1}^s \mu_{ij} L_j^{[k]});$ 
    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 15:** 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.  $(1, \dots, P)$  integrazio bakoitzean  $N$  urrats eman baditugu,  $t_i = t_0 + i \cdot h$ ,  $i = 1, \dots, N$  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.

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

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

$$\sigma_i = \sqrt{\frac{1}{P} \sum_{k=1}^P (\Delta E_i^{[k]})^2 - (\Delta \bar{E}_i)^2}, \quad i = 1, \dots, N.$$

$$Max E = \max_{i=1, \dots, N} |\Delta \bar{E}_i|$$

### 2. Energia errore lokala.

*$P = 100$  integrazio guztietarako, bi urratsen arteko energia lokalaren batzbestekoa ( $\mu$ ) eta desbiazio estarrada ( $\sigma$ ).*

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

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

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

*Doitasun laukoitzean lortutako soluzioari soluzio zehatza deituko dugu,*

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

*eta k. soluzioari dagokion errorea,*

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

$$\bar{G}e_i = \left(\frac{1}{P} \sum_{k=1}^P Ge_i^{[k]}\right), \quad \text{Max} \bar{G}e = \max_{i=1, \dots, N} (\bar{G}e_i)$$

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

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

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

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,*

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

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

### 7.4.2 Brouwer legea.

Zenbakizko integrazioaren biribiltze errorea hausazkoa dela ziurtatzeko, metodoak Brouweren legea [4] betetzen duela konprobatu ohi izan da. Brouwerren legeak dioenez, t urrats finkoko integrazioaren ( $t_n = t_{n-1} + h$ ) denbora bada, energia errorea  $t^{1/2}$  proportzionalki hasiko da. See also Hairer [13][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	$\bar{\Delta}0$ %	$\bar{Max}E$	$\bar{\mu}$	$\bar{\sigma}$	$\bar{Max}Ge$
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	$2e10^{-15}$	$4e10^{-19}$	$8e10^{-18}$	$6e10^{-12}$

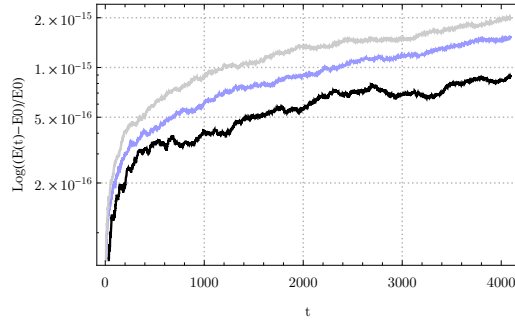
Taula 7.2: Summary of Chaotic case.

Arithmetic	$\bar{\Delta}0$ %	$\bar{Max}E$	$\bar{\mu}$	$\bar{\sigma}$	$\bar{Max}Ge$
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

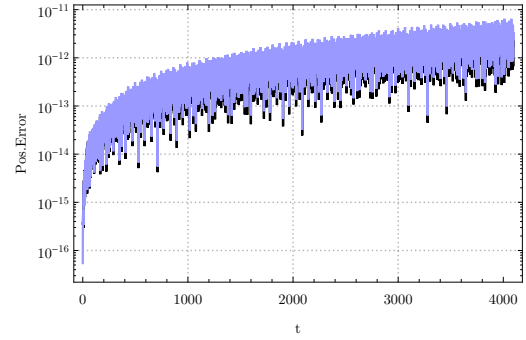
Brouwer-legea.

Biribiltze erroreaen estimazioa.

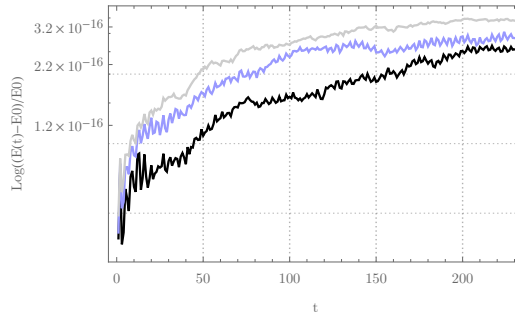
7.4.4 N-Body problema.



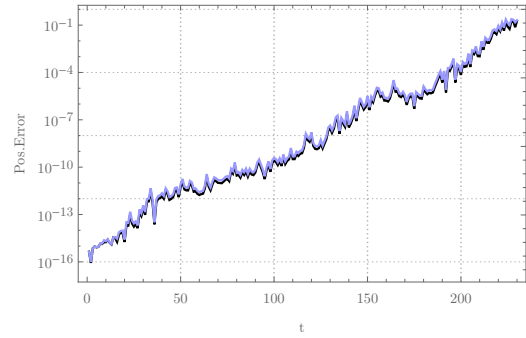
(a) Non chaotic: energy error.



(b) Non chaotic: global error.

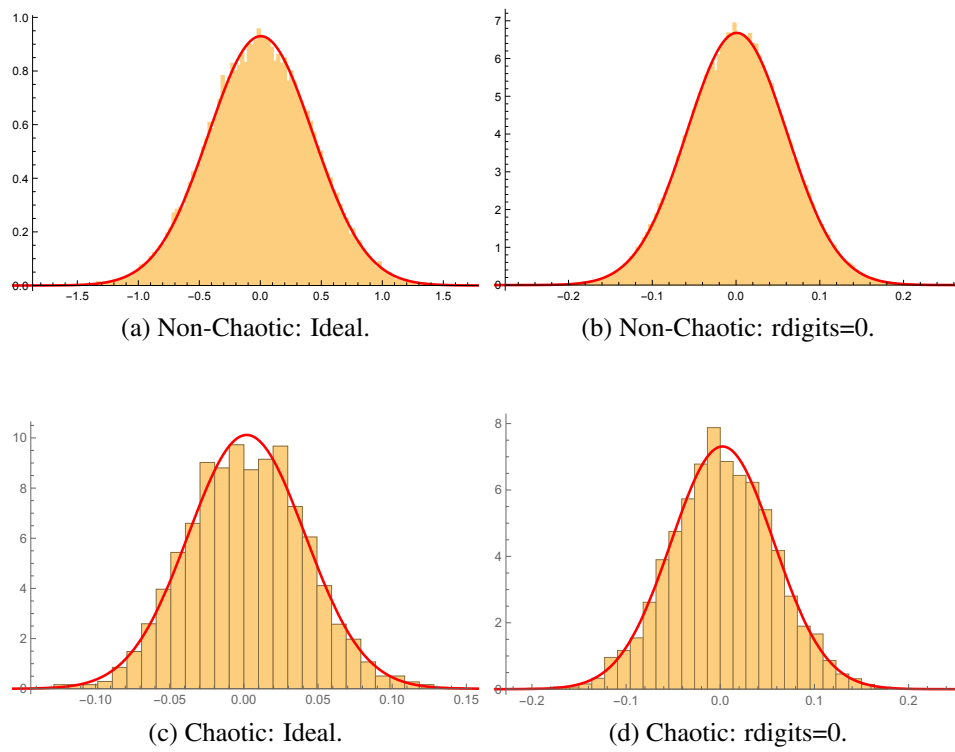


(c) Chaotic: energy error.

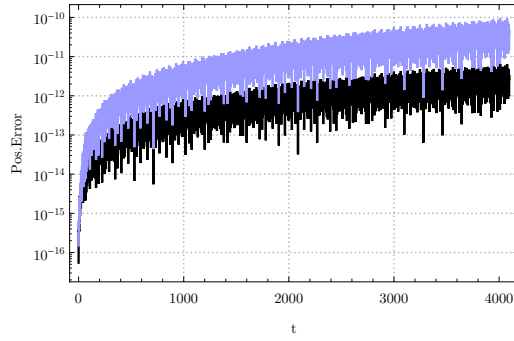


(d) Chaotic: global error.

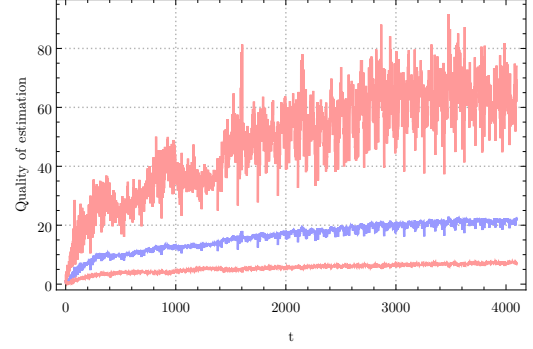
Irudia 7.2: We show Non-Chaotic case (a,b) and Chaotic case (c,d). Left figure mean energy error evolution  $\Delta \bar{E}_i$  and right figure mean Global error evolution  $\bar{G}e_i$  of the 100 integrations for *Ideal Integrator* (black) , *Double prec* (blue) and *Classic Implementation* (gray).



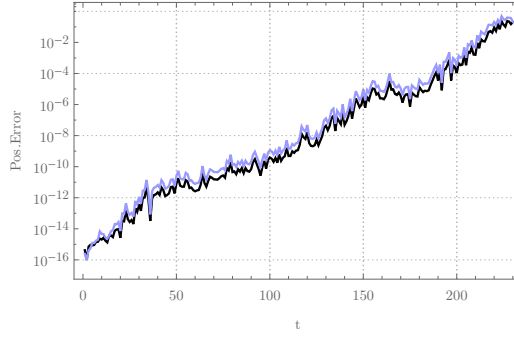
Irudia 7.3: Histogram of energy errors for Non-Chaotic case (a,b) and for Chaotic case (c,d).



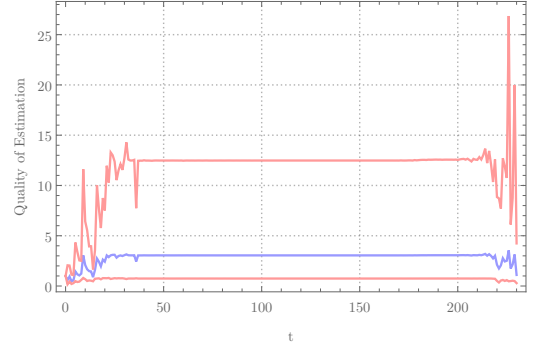
(a) Non Chaotic: estimation



(b) Non Chaotic: quality of estimation

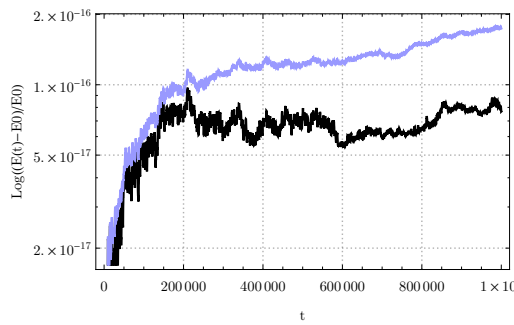


(c) Chaotic: estimation

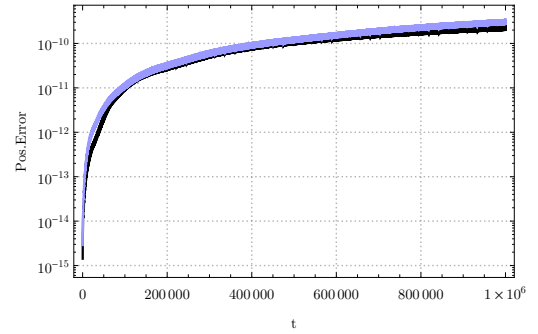


(d) Chaotic: quality of estimation

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).

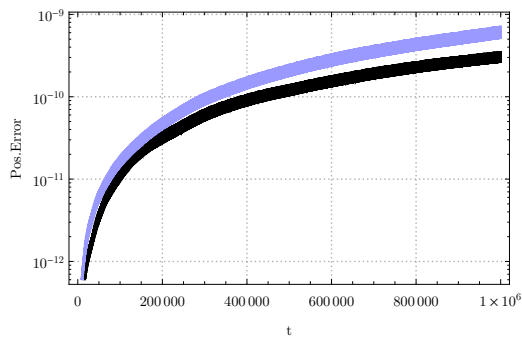


(a) Energy error.

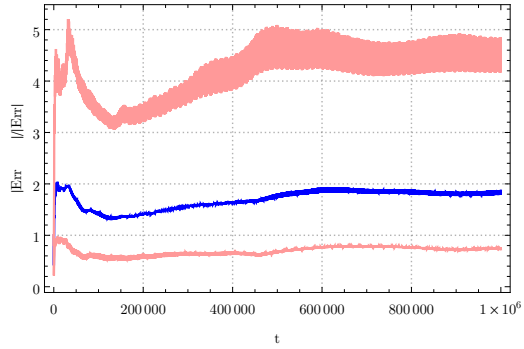


(b) Global error.

Irudia 7.5: N-body: left figure mean energy error evolution  $\bar{\Delta E}_i$  and right figure mean Global error evolution  $\bar{G}e_i$  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.

*Koordenatu kartesiarrak erabiltzearen abantaila.*

### 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,*

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

*$f(y)$  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 (eguzkiarekiko interakzioa) eta planeten interakzioen batura gisa bana daiteke,*

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

*Eguzkiari dagokion azpindizea  $i = 0$  kontsideratzen badugu,*

$$H_k(q, p) = \frac{1}{2} \sum_{i=0}^N \frac{p_i^2}{m_i} - Gm_0 \sum_{i=1}^N \frac{m_i}{\|q_i - q_0\|} \quad (9.1)$$

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

*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,*

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

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

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

$$\dot{v} = f_v(y) = \sum_{j=0, j \neq i}^N \frac{Gm_j}{\|q_j - q_i\|^3} (q_j - q_i) \quad i = 0, \dots, N.$$

$$\dot{v} = f_v(y) \Rightarrow \begin{cases} k_v(y) = \begin{cases} \dot{v}_0 = \sum_{j=1, j \neq 0}^N \frac{Gm_j}{\|q_j - q_0\|^3} (q_j - q_0). \\ \dot{v}_i = \frac{Gm_0}{\|q_0 - q_i\|^3} (q_0 - q_i), & i = 1, \dots, N. \end{cases} \\ g_v(y) = \begin{cases} \dot{v}_0 = 0. \\ \dot{v}_i = \sum_{j=1, j \neq i}^N \frac{Gm_j}{\|q_j - q_i\|^3} (q_j - q_i), & i = 1, \dots, N. \end{cases} \end{cases} \quad (9.4)$$

**Meta-algoritmoa.**

*IRK metodoaren formulazioa gogoratu,*

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

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

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

$$Y_i - y_n - \sum_{j=1}^s \mu_{ij} hb_j \left( k(Y_j) + g(Y_j) \right) = 0, \quad i = 1, \dots, s.$$

*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  $k = 1, 2, \dots$ ,*

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

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

*IRK metodoaren ekuazio-sistemari aplikatuz,*

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

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

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

non,

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

Lortutako espresioa garatuko dugu.

### 1. Lehen hurbilpena.

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

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

### 2. Bigarren hurbilpena (Linealizazioa).

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

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

Honako hurbilpena,

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

ordezkatzuz eta garatuz,

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

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

### 3. Ekuazioak berridatiziz.

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

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

**Meta-Algoritmoa.**

*IRK metodoa aplikatzeko meta algoritmoa planteatuko dugu.*

```

e = 0;
for n ← 1 to endstep do
  k = 0;
  Hasieratu  $Y_i^{[0]}$  ,  $i = 1, \dots, s$ ;
  while (konbergentzia lortu) do
    k = k + 1;
    if k == 1 then
      Hasieratu  $G_i^{[0]}$ ;
    else
       $G_i^{[k-1]} = \sum_{j=1}^s \mu_{ij} \text{hb}_j g(Y_j^{[k-1]});$ 
    end
    Askatu  $Y_i^{[k]} = y_{n-1} + \sum_{j=1}^s \mu_{ij} \text{hb}_j k(Y_j^{[k]}) + G_i^{[k-1]}$ ;
  end
   $L_i = \text{hb}_i f(Y_i^{[k]});$ 
   $\delta_n = \sum_{i=1}^s L_i + e;$ 
   $y_n = y_{n-1} + \delta_n;$ 
   $e = (y_{n-1} - y_n) + \delta_n;$ 
end

```

**ALGORITHM 16:** 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;
Yi[k,0] = Yi[k-1];
while (konbergentzia lortu) do
    l = l + 1;
    Ki[k,l] = k(Yj[k,l-1]);
    Yi[k,l] = yn-1 + ∑j=1s μij h bj Kj[k,l] + gi[k-1];
end

```

#### ALGORITHM 17: 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 du. Kasu honetan urrats bat finakatuta, kanpo planeten  $k(y)$  problema barruko planeta baino azkarrago konbergituko du.

#### Orokorpena.

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

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

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



**Adibidea.**

*Demagun ekuazio diferentzialak,  $m$  perturbazio funtzio dituela,*

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

*non  $k(y) \ll gl(y)$ ,  $l = 1, \dots, m$ .*

*$m = 2$  deneko kasu partikulara aztertuko dugu,*

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

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

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

$$k = k + 1;$$

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

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

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

$$l = l + 1;$$

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

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

**end**

**end**

**ALGORITHM 18:** Main Algorithm

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

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

$k(y)$ : kepleriarra.

$g(y)$ : planeten arteko grabitazio interakzioak.

$rs(y)$ : eguzkiarekiko erlatibitate efektua.

$rn(y)$ : planeten arteko erlatibitate efektuak.

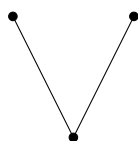
**Adierazpena.**

*Ekuazio diferentzialen deskonposaketak, zuhaitz moduan adieraz daitezke.*

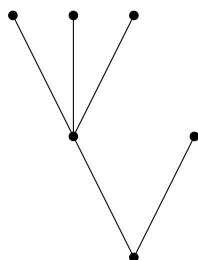
$$\dot{y} = f(y).$$

•

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



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

**9.3 Denbora birparametrizazioa.****9.3.1 Denbora birparametrizazioa.**

*Demagun jatorrizko ekuazio diferentziala,*

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

*non  $y$  menpeko aldagaia eta  $t$  aldagai askea den.*

*Aldagai askeari aldaketa bat aplikatuz ( $s()$  izeneko funtzio), ekuazio diferentziala leuntzea lortuko dugu.*

$$y = z,$$

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

*Honako garapena egingo dugu aldagai berriarekiko ( $\tau$ ) ekuazioak lortzeko.*

$$y = z \Rightarrow \frac{dy}{dt} = \frac{dz}{d\tau} \frac{d\tau}{dt} \Rightarrow \frac{dz}{d\tau} = s(z) f(y(t))$$

*Sistema berrian, mugimendua  $z(\tau)$  funtzioak deskribatzen du:  $z$  aldagai berria  $\tau$  aldagai askearen menpekoa da.*

### 9.3.2 Adibidea.

*Esperimentu honetan, IRK metodoan denbora birparametrizazioa modu errezean aplika daitekeela erakutsi nahi dugu. N9-Body problemen, merkurio ezentritizitate handiena duen planeta da: merkurio araberako denbora birparametrizazioa planteatuko dugu.*

$$s(q) = r_{10}^{3/2} \quad (9.11)$$

$$r_{10} = \|q_1 - q_0\|_2 \quad (9.12)$$

*non  $q_1 = (q_{1x}, q_{1y}, q_{1z})$  merkurio plantearen kokapena eta  $q_0 = (q_{0x}, q_{0y}, q_{0z})$  eguzkiaren kokapena den.*



# **Atala IV**

## **Synthesys**



**Atala V**  
**Eranskinak**





# Kapitulua 10

## A-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,*

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

$$E_1 = E_0 + \Delta E$$

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

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

**Newton metodoa** ,

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

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

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

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

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

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

*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 ordezkatzuz,*

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

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

*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,*

$$C = \cos(\Delta E), \quad S = \sin(\Delta E)$$

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

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

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

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

**Biribiltze errorea** .  $\Delta E$  txikia denean,  $\cos(\Delta E) - 1$  espresioaren kalkulua-  
ren ezabapen arazoak biribiltze errore handia eragin dezake. Hori konpontzeko  
baliokidetasun hau erabiliko dugu,

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

*Eta beraz, kepler-en ekuazioak hauek izango dira,*

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

*Eta  $(q_1, v_1)$  balioak kalkulatzeko,*

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

## 10.2 Ekuazio Heliozentrikoak.

### Sarrera.

*Lehenik koordenatu barizentrikoei,*

$$\mathbf{q}_i, \mathbf{p}_i \in \mathbb{R}^3, \quad i = 0, \dots, N$$

dagokien *Hamiltondarra gogoratuko dugu*,

$$H(\mathbf{q}, \mathbf{p}) = \frac{1}{2} \sum_{i=0}^N \frac{\|\mathbf{p}_i\|^2}{m_i} - G \sum_{0 \leq i < j \leq N}^N \frac{m_i m_j}{\|\mathbf{q}_i - \mathbf{q}_j\|}. \quad (10.3)$$

*Koordenatu barizentrikoetatik abiatuta eta aldagai aldaketa bat aplikatuz ekuazio koordenatu heliozentrikoen*

$$\mathbf{Q}_i, \mathbf{P}_i \in \mathbb{R}^3, \quad i = 0, \dots, N$$

*arabera berridatziko ditugu.*

### **Aldagai aldaketa.**

*Lehenik honako aldagai aldaketa aplikatuko dugu,*

$$\mathbf{Q}_0 = \mathbf{q}_0, \quad \mathbf{Q}_i = \mathbf{q}_i - \mathbf{q}_0, \quad i = 1, \dots, N$$

$$\mathbf{P}_0 = \sum_{i=0}^N \mathbf{p}_i, \quad \mathbf{P}_i = \mathbf{p}_i, \quad i = 1, \dots, N.$$

*Hamiltondarra era honetan deskonposatu daiteke  $H = H_k + T_1 + U_1$ ,*

$$\begin{aligned} H_k &= \sum_{i=1}^N \left( \frac{\|\mathbf{P}_i\|^2}{2} \left( \frac{m_0 + m_i}{m_0 m_i} \right) - G \frac{m_0 m_i}{\|\mathbf{Q}_i\|} \right) \\ T_1 &= \sum_{0 < i < j \leq N}^N \frac{\mathbf{P}_i \mathbf{P}_j}{m_0}. \\ U_1 &= -G \sum_{0 \leq i < j \leq N}^N \frac{m_i m_j}{\|\mathbf{Q}_i - \mathbf{Q}_j\|} \end{aligned}$$

*Hamiltondarraren alde bakoitza bere aldetik deribatuz,*

1.  $H_k$ .

$$\dot{\mathbf{Q}}_i = \nabla_{\mathbf{p}} H_k \Rightarrow \dot{\mathbf{Q}}_i = \mathbf{P}_i \left( \frac{m_0 + m_i}{m_0 m_i} \right), \quad i = 1, \dots, N.$$

$$\dot{\mathbf{P}}_i = -\nabla_{\mathbf{q}} H_k \Rightarrow \dot{\mathbf{P}}_i = -G \frac{m_0 m_i}{\|\mathbf{Q}_i\|^3} \mathbf{Q}_i, \quad i = 1, \dots, N.$$

2.  $T_1$ .

$$\dot{\mathbf{Q}}_i = \nabla_p T_1 \Rightarrow \dot{\mathbf{Q}}_i = \sum_{j \neq i, j=1}^N \frac{\mathbf{P}_j}{m_0}, \quad i = 1, \dots, N.$$

$$\dot{\mathbf{P}}_i = -\nabla_q T_1 \Rightarrow \dot{\mathbf{P}}_i = 0, \quad i = 1, \dots, N.$$

3.  $U_1$ .

$$\dot{\mathbf{Q}}_i = \nabla_p U_1 \Rightarrow \dot{\mathbf{Q}}_i = 0, \quad i = 1, \dots, N.$$

$$\dot{\mathbf{P}}_i = -\nabla_q U_1 \Rightarrow \dot{\mathbf{P}}_i = -G \sum_{j \neq i, j=1}^N \left( \frac{m_i m_j}{\|\mathbf{Q}_i - \mathbf{Q}_j\|^3} (\mathbf{Q}_i - \mathbf{Q}_j) \right), \quad i = 1, \dots, N.$$

### **Ekuazio diferentzialak ( $Q_i, V_i$ ).**

*Ekuazio diferentzialak ( $\mathbf{Q}_i, \mathbf{P}_i$ ) aldagai berriekiko idatziko ditugu, honako definizioa kontutan harturik,*

$$\mathbf{V}_i = \frac{\mathbf{P}_i}{(m_0 + m_i)}, \quad \mu_i = \frac{m_0 m_i}{(m_0 + m_i)}.$$

1.  $H_k$ .

$$\dot{\mathbf{Q}}_i = \mathbf{V}_i \quad i = 1, \dots, N.$$

$$\dot{\mathbf{V}}_i = -G \frac{(m_0 + m_i)}{\|\mathbf{Q}_i\|^3} \mathbf{Q}_i, \quad i = 1, \dots, N.$$

2.  $T_1$ .

$$\dot{\mathbf{Q}}_i = \sum_{j \neq i, j=1}^N \frac{\mathbf{V}_j m_j}{(m_0 + m_j)}, \quad i = 1, \dots, N.$$

$$\dot{\mathbf{V}}_i = 0, \quad i = 1, \dots, N.$$

3.  $U_1$ .

$$\dot{\mathbf{Q}}_i = 0, \quad i = 1, \dots, N.$$

$$\dot{\mathbf{V}}_i = -G \frac{(m_0 + m_i)}{m_0} \sum_{j \neq i, j=1}^N \left( \frac{m_j}{\|\mathbf{Q}_i - \mathbf{Q}_j\|^3} (\mathbf{Q}_i - \mathbf{Q}_j) \right), \quad i = 1, \dots, N.$$

### Energia kalkulua.

*Koordenatu heliozentrikoetan integrazioak egiten ditugunean, sistemaren energia kalkulatzeko koordenatu barizentrikoeetara bihurtuko dugu soluzioa.*

*Hauek dira koordenatu heliozentrikoetatik abiatuta  $(\mathbf{Q}_i, \mathbf{V}_i)$ , koordenatu bari-zentrikoak  $(\mathbf{q}_i, \mathbf{v}_i)$  kalkulatzeko ekuazioak,*

1.  $\mathbf{q}_i, \quad i = 0, \dots, N.$

$$\mathbf{q}_0 = - \sum_{i=1}^M \frac{m_i \mathbf{Q}_i}{M}, \quad M = \sum_{i=0}^N m_i.$$

$$\mathbf{q}_i = \mathbf{q}_0 + \mathbf{Q}_i, \quad i = 1, \dots, N.$$

2.  $\mathbf{v}_i, \quad i = 0, \dots, N.$

$$\mathbf{v}_i = \frac{m_0}{m_0 + m_i} \mathbf{V}_i, \quad i = 1, \dots, N.$$

$$\mathbf{P}_0 = \sum_{i=0}^N \mathbf{p}_i = \sum_{i=0}^N m_i \mathbf{v}_i = 0 \Rightarrow m_0 \mathbf{v}_0 + \sum_{i=1}^N m_i \mathbf{v}_i = 0 \Rightarrow \mathbf{v}_0 = -\frac{1}{m_0} \sum_{i=1}^N m_i \mathbf{v}_i.$$

# 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] Dirk Brouwer. *On the accumulation of errors in numerical integration*. *The Astronomical Journal*, 46:149–153, 1937.
- [5] John Charles Butcher. *Numerical Methods for Ordinary Differential Equations*. Second edition, Wiley, 2008.
- [6] 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.
- [7] 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.
- [8] Robert M Corless and Nicolas Fillion. *A graduate introduction to numerical methods*. *AMC*, 10:12, 2013.
- [9] 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.
- [10] Ariadna Farrés, Jacques Laskar, Sergio Blanes, Fernando Casas, Joseba Makazaga, and Ander Murua. *High precision symplectic integrators for*

- the solar system*. Celestial Mechanics and Dynamical Astronomy, 116(2): 141–174, 2013.
- [11] 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.
- [12] Toshio Fukushima. *Reduction of round-off error in symplectic integrators*. The Astronomical Journal, 121(3):1768, 2001.
- [13] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006.
- [14] 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.
- [15] 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.
- [16] Nicholas J Higham. *Accuracy and stability of numerical algorithms*. Siam, 2002.
- [17] Nicholas J Higham. *Programming languages: An applied mathematics view*. 2015.
- [18] MP Calvo JM Sanz-Serna. Numerical Hamiltonian problems. Chapman and Hall, 1994.
- [19] 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.
- [20] Jacques Laskar and Philippe Robutel. *High order symplectic integrators for perturbed hamiltonian systems*. Celestial Mechanics and Dynamical Astronomy, 80(1):39–62, 2001.
- [21] Harold F Levison and Martin J Duncan. *The long-term dynamical behavior of short-period comets*. Icarus, 108(1):18–36, 1994.
- [22] Robert I McLachlan. *Composition methods in the presence of small parameters*. BIT Numerical Mathematics, 35(2):258–268, 1995.



- [23] A Morbidelli. *Modern integrations of solar system dynamics*. Annual Review of Earth and Planetary Sciences, 30(1):89–112, 2002.
- [24] 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.
- [25] Nobelprize.org. *The nobel prize in chemistry 2013*, May 2014.
- [26] Michael L Overton. *Numerical computing with IEEE floating point arithmetic*. Siam, 2001.
- [27] 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*.
- [28] Mark Sofroniou and Giulia Spaletta. *Derivation of symmetric composition constants for symmetric integrators*. Optimization Methods and Software, 20(4-5):597–613, 2005.
- [29] Pat H Sterbenz. *Floating-point computation*. Prentice Hall, 1973.
- [30] Gerald J Sussman and Jack Wisdom. *Chaotic evolution of the solar system*. Technical report, DTIC Document, 1992.
- [31] Jack Wisdom and Matthew Holman. *Symplectic maps for the n-body problem*. The Astronomical Journal, 102:1528–1538, 1991.