Project 5 - Computational Physics - FYS3150

Modellering av solsystemet ved bruk av ordinære differensiallikninger

Fysisk institutt, Universitet i Oslo, Norge.
Philip Niane og Rohullah Akbari
https://github.com/philipkarim/Philip-and-Rohullah-ComFys

Abstrakt

I denne artikkelen er de numeriske metodene Forward Euler og Velocity Verlet sammenliknet, hvor det ble vist at Velocity Verlet er den mest presise, med lave relative feil for små tidssteg Δt. Forward Euler viste også små numeriske feil ved små nok Δt, men mistet presisjon ved lengre intervaller. Det ble også vist at ved å simulere et system med Jupiter, Sola og Jorda vil det å gi Jupiter en hypotetisk stor masse, påvirke og slenge Jordkloden ut av banen rundt Sola. Itillegg til dette ble hele solsystemet simulert og vist hvordan den totale energien og angulærmomentet holder seg tilnærmet konservert. Til slutt ble det prøvd å se hvordan Merkurs endring i perihelium kan forklares ved generell relativitetsteori, løsningsmetoden ga en relativ feil på 93% som er en del høyere enn forventet.

Innhold

1 Introduksjon **2**

2 Teori **2**
2.1 Newtons gravitasjonslov 2
2.2 Algoritmer 4
2.3 Litt av fysikken bak solsystemet 6
2.4 Relativitet 8

3 Metode **9**
3.1 Oppbygning av kode 9
3.2 Forward Euler 9

3.3 Velocity Verlet 10

4 Resultater **10**
4.1 System med Sola og Jorda 10
4.2 System med Sola, Jorda, og Jupiter 13
4.3 System med hele solsystemet 15
4.4 System med Sola og Merkur 16

5 Diskusjon **16**
5.1 System med Sola og Jorda 16
5.2 System med Sola, Jorda og Jupiter 17
5.3 System med hele solsystemet 18
5.4 System med Merkur og Sola 18

6 Konklusjon **18**

7 Appendiks **18**
7.1 Utledning av Forward Euler 18
7.2 Utledning av Velocity Verlet 19
7.3 Utledning av betingelsen for en sirkulær bane rundt sola 19
7.4 Utledning av unnslipnignshastighet 20
7.5 Utledning for relasjon for fast massesenter 20

8 Bibliografi **21**

1 Introduksjon

I dette prosjektet er målet å lage en modell for å simulere solsystemet og dets flere planeter. For å simulere dette tas det utgangspunkt i Newtons gravitasjonslov uttrykt som ordinære differensiallikninger. Løsningsmetoden er ved bruk av to svært populære numeriske metoder Velocity Verlet og Forward Euler. Metodene skal brukes og sammenliknes ved å se hvordan små fysiske endringer kan påvirke resten av systemet. Itillegg skal hypotetiske 'Hva hvis...' scenarioer utforskes og lekes med. Itillegg vil også generell relativitetsteori bli brukt til å modellere og forklare Merkurs bane rundt sola, og dens variasjon i periheliumsposisjonen med tiden.

2 Teori

2.1 Newtons gravitasjonslov

Newtons gravitasjonslov er utgangspunktet for tiltrekningskrefter mellom objekter. I solsystemet kan det antas at de eneste kreftene som virker på planetene solsystemet er tiltrekningskreftene mellom legemer. Newtons gravitasjonslov sier at alle legemer med masse tiltrekker hverandre, og kan beregnes ved følgende formel:

$$\vec{F_G} = \frac{Gm_1m_2}{r^2}\frac{\vec{r}}{r} \tag{1}$$

Hvor G er gravitasjonskonstanten, m_1 og m_2 er massene til objektene, og r er avstanden mellom legemene. Ved flere legemer i samme system beregnes kraften ved summen av kreftene som påvirker legemet. I et system kun bestående av Sola og Jorda kan følgende uttrykk for kraften brukes[2]:

$$\vec{F_G} = \frac{GM_\odot M_{Earth}}{r^2} \tag{2}$$

Hvor $M_\odot$ er massen til Sola, M_{Earth} er massen til Jorda og r er avstanden mellom dem.

Newtonsgravitasjonslov kan skrives som parvise ordinære differensiallikninger ved å utvide likningene ved bruk av Newtons andre lov[2]: Kreftene i x og y retningene kan dermed skrives som:

$$\frac{d^2x}{dt^2} = \frac{F_x}{M_E}, \tag{3}$$

og

$$\frac{d^2y}{dt^2} = \frac{F_y}{M_E}, \tag{4}$$

Introduserer polarkoordinater $x = r\cos(\theta)$ og $y = r\sin(\theta)$ som gir r=$\sqrt{x^2+y^2}$ og uttrykker kreftene i de to retningene:

$$F_x = -\frac{GM_\odot M_E}{r^2}\cos(\theta) = -\frac{GM_\odot M_E}{r^3}x \tag{5}$$

$$F_y = -\frac{GM_\odot M_E}{r^2}\sin(\theta) = -\frac{GM_\odot M_E}{r^3}y \tag{6}$$

Som kombinert med (3), (4) og Newtons andre lov gir de følgende ordinære differensiallikninger:
X retningen:

$$\frac{dv_x}{dt} = -\frac{GM_\odot}{r^3}x \tag{7}$$

og

$$\frac{dx}{dt} = v_x \tag{8}$$

Y retning:

$$\frac{dv_y}{dt} = -\frac{GM_\odot}{r^3}y \tag{9}$$

og

$$\frac{dy}{dt} = v_y \tag{10}$$

2.1.1 Større avstand mellom planetene

Ved å uttrykke tiltrekningskraften som en funksjon av eksponenten til avstanden er det mulig å se hvordan planeters bevegelse avhenger av tiltrekningskraften. Følgende uttrykk kan brukes[2]:

$$\vec{F_G} = \frac{GM_{\odot}M_{Earth}}{r^{\beta}} \tag{11}$$

Som viser at ved større verdier for β vil tiltrekningskraften minke.

2.2 Algoritmer

Før algoritmene kan implementeres i programmet må det litt diskretisering til. Ved å se på et gitt tidsintervall $[t_{start}, t_{slutt}]$ kan dette intervallet brukes til å bestemme steglengde h:

$$h = \frac{t_{slutt} - t_{start}}{N} \tag{12}$$

Hvor N er antall steglengder som skal brukes i beregningene. Ved å øke tiden med en steglengde h for hver kalkulasjon kan posisjon, hastighet og akselerasjon beregnes som funksjoner av de bestemte tidsverdiene:
For posisjon:

$$r_i = r(t_i) \tag{13}$$

For hastighet:

$$v_i = v(t_i) \tag{14}$$

og akselerasjon:

$$a_i = a(t_i) \tag{15}$$

Det er også verdt å minne om dotnotasjon, hvor antall 'dotter' bestemmer antall ganger det er derivert med hensyn på tid. $\ddot{\boldsymbol{x}} = \dot{\boldsymbol{v}} = \boldsymbol{a}$

2.2.1 Forward Euler

Forward Euler er den første hovedmetoden som skal brukes til beregninger. Eulers metode går ut på å beregne posisjon eller hastighet en steglengde fremover ved bruk av initialbetingelser før steget tas. Ved bruk av følgende uttrykk kan posisjon og hastighet innenfor et satt tidsintervall beregnes, utledning kan ses i appendiks:

$$x_{i+1} \approx x_i + hv_i \tag{16}$$

og

$$v_{i+1} \approx v_i + ha_i \tag{17}$$

Ved bruk av Forward Euler beregnes først posisjonen i neste steg ved bruk av initialhastigheten og initialposisjonen, videre beregnes hastigheten i neste punkt,

og tiden øker til neste tidssteg for så beregne seg fremover i tid. Akselerasjonen beregnes for hvert tidssteg for å uttrykke hastigheten. Akselerasjonen beregnes ved bruk av Newtons andre lov dersom kraften er kjent.[1]

2.2.2 Verlet metoden

For å løse andreordens differensiallikninger er det mulig å bruke den kjente Verlet algoritmen. Verlet algoritmen er mye brukt i blant annet molekylær dynamikk. Verlet algoritmen tar utgangspunkt i en Taylor ekspansjon av posisjonen som videre utledes til følgende uttrykk[1][5]:

$$x_{i+1} = 2x_i - x_{i-1} + h^2\ddot{x}_i + O(h^4) \tag{18}$$

Der (O^4) er feilleddet. I likning (18) avhenger neste posisjon av å vite to posisjoner fra før av. Ved å introdusere en av hovedmetodene som skal brukes i denne artikkelen blir det mulig å beregne seg fremover i tid uten å måtte vite initialbetingelser for to tidspunkter, nemlig Velocity Verlet metoden. Velocity Verlet metoden bruker følgende formler til å beregne posisjon og hastighet, hvor utledning kan ses i appendiks:

$$x_{i+1} \approx x_i + hv_i + \frac{h^2}{2}\dot{v}_i \tag{19}$$

og

$$v_{i+1} \approx v_i + \frac{h}{2}\left(\dot{v}_{i+1} + \dot{v}_i\right) \tag{20}$$

2.2.3 FLOPs

For å beregne antall floating point operations må matematiske operasjoner telles ved å se hvor mange ganger mulitplikasjon, divisjon, adisjon og subtraksjon blir utført.

For Forward Euler vil det bli tatt utgangspunkt i likning (16) og (17) uten å tenke på økning av tiden. Dermed må antall matematiske operasjoner som skal utføres N antall ganger telles. I likning (16) for posisjonen utføres det 3 operasjoner per gang denne kjøres. Itillegg vil likning (17) for hastigheten inneholde 3 operasjoner hver gang algoritmen kjøres. Totalt 3N+3N=6N FLOPs for Forward Euler algoritmen.

For Velocity Verlet vil det bli tatt utgangspunkt i likning (19) og (20). Antall matematiske operasjoner som skal utføres N antall ganger telles. I likning (19) for posisjonen utføres det 7 operasjoner per gang denne kjøres. Itillegg vil likning (20) for hastigheten inneholde 5 operasjoner hver gang algoritmen kjøres. Totalt 5N+7N=12N FLOPs for Velocity Verlet algoritmen, som er dobbelt så mange FLOPs som for Forward Euler.

2.3 Litt av fysikken bak solsystemet

Planetene vil itillegg til å bli sett på som punktpartikler, bli antatt å ligge i samme plan, dermed vil z koordinaten bli utelatt ved definering av posisjoner og hastigheter.

2.3.1 Enheter og verdier

Når det gjelder enheter i dette prosjektet vil en grei skalering være å bruke lengde enheten AU som er gjennomsnittslengden mellom Sola og Jorda. 1AU=$1.5{\cdot}10^{11}m$. I tillegg vil det være naturlig å skalere massene til planetene (og Pluto) med hensyn på Solas masse. Oversikt over massene hentet fra kan ses i tabell 1.

Himmellegeme	Masse (kg)	Avstand til Sola (AU)
Sola	$2{\cdot}10^{30}$	
Jorda	$6{\cdot}10^{24}$	1
Jupiter	$1.9{\cdot}10^{27}$	5.20
Mars	$6.6{\cdot}10^{23}$	1.52
Venus	$4.9{\cdot}10^{24}$	0.72
Saturn	$5.5{\cdot}10^{26}$	9.54
Merkur	$3.3{\cdot}10^{23}$	0.39
Uranus	$8.8{\cdot}10^{25}$	19.19
Neptun	$1.03{\cdot}10^{26}$	30.06
Pluto	$1.31{\cdot}10^{22}$	39.53

Tabell 1: Planeter, Pluto og Sola: Oversikt over vekt og avstand til Sola [2]

I tillegg vil initialverdier bli hentet fra NASA sine nettsider ved beregninger på hele solsystemet[3].

2.3.2 Sirkulær bane rundt Sola

Det vil bli antatt at Jorda har en sirkulær bane rundt sola, ved sirkulære bevegelser er følgende relasjon tilstede:

$$F_G = \frac{M_{Earth}v^2}{r} = \frac{GM_{\odot}M_{Earth}}{r^2} \tag{21}$$

Hvor v er hastigheten til Jorda og gir følgende, utledning kan ses i appendiks:

$$v = 2\pi\frac{AU}{yr} \tag{22}$$

Vet at gravitasjonskrefter er konservative. Hastigheten vil teoretisk være bevart rundt hele banen, og avstanden vil konstant være 1 AU ved sirkulære baner. Dermed kan initialposisjon settes lik $\vec{r}$=(1,0) som betyr at hastigheten kun vil være rettet i y retning i dette punktet. Altså $\vec{v}$=(0,2π).

2.3.3 Energi og angulærmoment

De eneste kreftene som virker på planetene er konservative tiltrekningskrefter mellom legemene i systemet. Dermed vil det ikke fungere noen ytre krefter på systemet, som betyr at både energien og angulær moment vil være bevart i modellen.[4]

Den totale energien kan dermed uttrykkes som summen av kinetisk energi og potensiell energi:

$$E_{tot} = \sum_i E_K + \sum_i E_P \tag{23}$$

Hvor kinetisk energi E_k er avhengig av masse og hastighet:

$$E_K = \frac{mv^2}{2} \tag{24}$$

Potensiell energi mellom to legemer er avhengig av massene til objektene uttrykt som $masse_1$ og $masse_2$ og avstanden mellom dem uttrykt som r.

$$E_p = -\frac{Gm_1m_2}{r} \tag{25}$$

Ved flere legemer vil den potensielle energien være summen av alle bidragene fra objektene i systemet.

I modellen vil planetene har angulær moment. Ved beregning av angulær moment kan følgende formel brukes:

$$L = \vec{r} \times \vec{p} = \vec{r} \times m\vec{v} \tag{26}$$

Hvor det totale angulærmomentet kan uttrykkes som summen av angulærmomentene til enkeltobjektene:

$$L_{tot} = \sum_i (\vec{r_i} \times m_i\vec{v_i}) \tag{27}$$

2.3.4 Unnslipningshastighet

For at det skal være mulig for jordkloden å unnslippe banen rundt Sola må initialhastigheten økes så den kinetiske energien til jordkloden blir lik den potensielle energien fra gravitasjonskraften. Dermed fås den teoretiske unnslipningshastigheten ved å sette den kinetiske energien lik den potensielle energien. Dette gir en teoretisk unnslipningshastighet på følgende, utledning kan ses i appendiks:

$$V_{Unn} = \sqrt{8}\pi\frac{AU}{yr} \tag{28}$$

2.3.5 Solas hastighet

Ved modellering av Sola og kun én omgitt planet, vil solas masse være såpass stor iforhold til den omgitte planeten at det vil være forsvarlig å anta at sola står stille. Ved modellering av hele solsystemet vil det være litt flere bidrag da det blir mange legemer i samme system. Selv om påvirkningen fra planetene på Sola er liten er kreftene fortsatt til stede, ved modellering av hele solsystemet vil ikke sola lenger bli gitt en hastighet lik 0, men istedet bli gitt en hastighet som lar systemets massesenter stå stille. Dette gjøres ved å sette systemets bevegelsesmengde lik 0:

$$\vec{P_{Tot}} = 0 \tag{29}$$

Som gir følgende relasjon for Solas hastighet for et fast massesenter, utledning kan ses i appendiks:

$$\vec{v}_{\odot} = -\frac{1}{M_{\odot}} \sum_{i=2} m_i v_i \tag{30}$$

Hvor massene og hastighetene bestemmes av de omringende planetene.

2.4 Relativitet

Generell relativitetsteori vil også bli brukt i artikkelen. Dette vil bli brukt til å se hvordan Merkur sin avstand til sola endres med tiden. Den nærmeste avstanden til sola kalles perihelium. Merkur er solas nærmeste planet som betyr at kraften som virker på merkur er veldig stor og gir en stor hastighet. Merkurs bane rundt sola er formet som en ellipse. Med tiden vil ellipsen roteres sakte men sikkert. Teoretisk vil perihelium roteres 43 buesekunder per århundre.[2]

For å beregne gravitasjonskraften mellom Sola og Merkur relativistisk, kan følgende formel brukes:

$$F_G = \frac{GM_{\odot}M_{Merkur}}{r^2}(1 + \frac{3l^2}{r^2c^2}) \tag{31}$$

Hvor r er avstanden, c er lysets hastighet i vakuum og l er angulærmomentet til Merkur per enhetsmasse gitt som $l = |\vec{r} \times \vec{v}|$

Videre kan periheliumvinkelen uttrykkes som funksjon av koordinatene til periheliumet ved følgende formel:

$$tan\theta_p = \frac{y_p}{x_p} \tag{32}$$

3 Metode

I denne seksjonen vil det bli presentert hvordan algoritmene fra teorien er implementert. Koden er inspirert fra eksempelkoden til Morten[6].

3.1 Oppbygning av kode

Koden består hovedsakelig av to klasser. Den ene klassen heter *solsystem*. I denne klassen ligger de fleste hovedfunksjonene. Her blir blant annet nye objekter lagt til. I *solsystem* blir også akselerasjonen kalkulert, algoritmene blir kjørt og beregning av Merkur sitt perihelium. Den andre klassen heter *celestialbody*. Her blir variabler som masse, posisjon, hastighet og akselerasjon lagt til.

I *main.cpp* bestemmer brukeren initial variabler som for eksempel antall steg og tid. Kombinert med bruk av klassen *solsystem* og funksjonen *lag-body*, er det mulig å legge til flere objekter i systemet. Videre kjører programmet algoritmen med funksjonen *solsystem.kjoring-algoritme*.

solsystem.cpp består av følgende funksjoner *lag-body* som lagrer nye objekter, *kalkulering-akselerasjon* som beregner akselerasjonen og energien dersom det finnes flere enn et objektet, *kjoring-algoritme* som kjører algoritmen og *merkur-presesjon* som beregner periheliumet til Merkur.

celestialBody.cpp mottar og legger til initial betingelser for de ulike objektene. Det er denne filen som brukes i *solsystem.cpp* for å lagre objekter.

algoritme.cpp inneholder selve algoritmene som blir brukt, Forward-Euler og Velocity Verlet algoritmene. Disse algoritmene blir kjørt via funksjonen *kjoring-algoritme* i *solsystem.cpp*-filen.

3.2 Forward Euler

Denne algoritmen er blitt implementert med hensyn på seksjonen (2.2.1). I *main.cpp*-filen blir algoritmen kjørt N antall ganger. Likningene (16) og (17) blir direkte implementert her:

$$x_{i+1} \approx x_i + hv_i \longrightarrow body.xpos+ = h \cdot body.xhas$$

$$v_{i+1} \approx v_i + ha_i \longrightarrow body.xhas- = h \cdot 4\pi^2 \cdot body.xpos/(body.r \cdot body.r \cdot body.r)$$

der

$$a = \frac{-4\pi^2}{r^3} \tag{33}$$

Samme metoden gjentas for y-aksen i koden.

3.3 Velocity Verlet

Her er likningene (19) og (20) direkte implementert på følgende måte:

$$x_{i+1} \approx x_i + hv_i + \frac{h^2}{2}\dot{v}_i \longrightarrow body.xpos = body.xpos + h \cdot body.xhas + (h \cdot h/2) \cdot body.xaks$$

$$v_{i+1} \approx v_i + \frac{h}{2}(\dot{v}_{i+1} + \dot{v}_i) \longrightarrow body.xhas = body.xhas + (h/2) \cdot body.xaks + (h/2) \cdot nest - xaks$$

der $body.xaks$ blir beregnet med hensyn på x_i og $nest - xaks$ blir beregnet med hensyn på x_{i+1}. Disse variablene blir beregnet via ligning (33). For studering av periheliumet til Merkur brukes et relativistisk uttrykk for akselerasjonen:

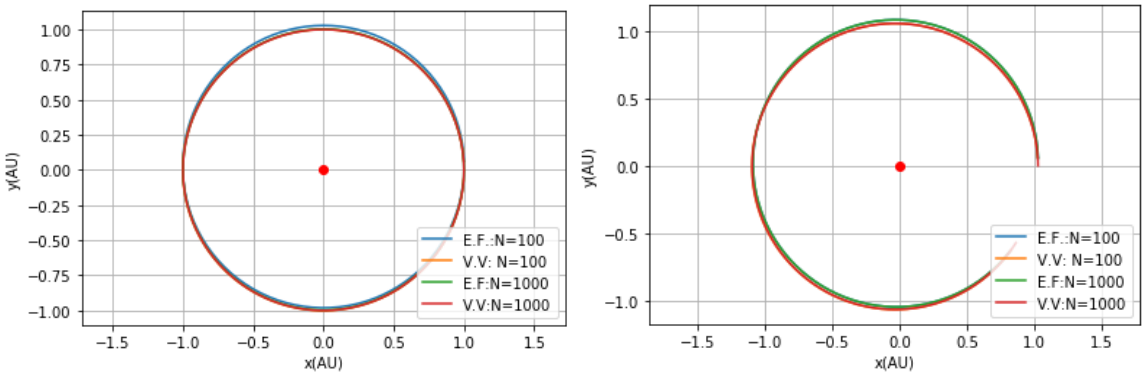
$$a = \frac{-4\pi^2}{r^3} \cdot 1 + 3\frac{(x \cdot v_y)^2 + (y \cdot v_x)^2}{r^2 c^2} \tag{34}$$

der c er lysetshastighet.

4 Resultater

4.1 System med Sola og Jorda

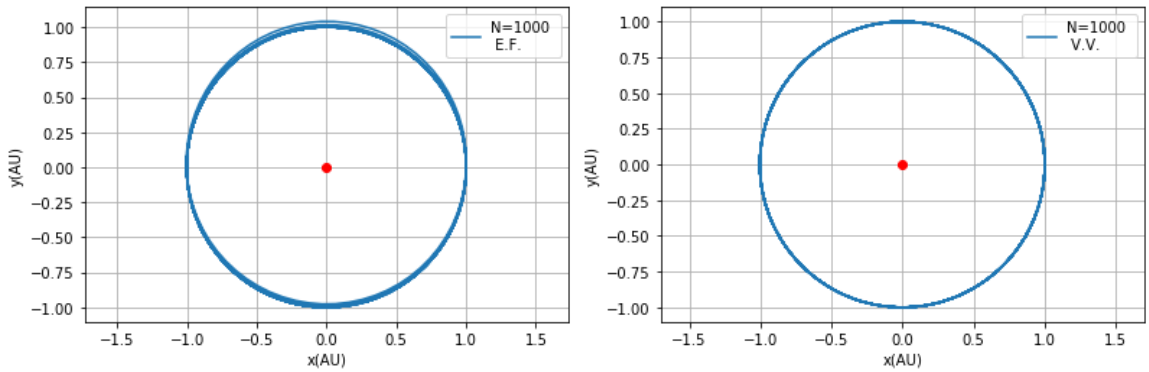
4.1.1 Stabilitet i Forward Euler og Velocity Verlet



(a) (b)

Figur 1: Resultater for Velocity Verlet og Forward Euler plottet for 1 år med N=100 og N=1000. I figur (a) er initialposisjonen (1,0), mens i figur (b) er initialposisjonen (1.03,0). Initialhastighet er $(0,2\pi)$i begge figurene

4.1.2 Modellering for lengre perioder



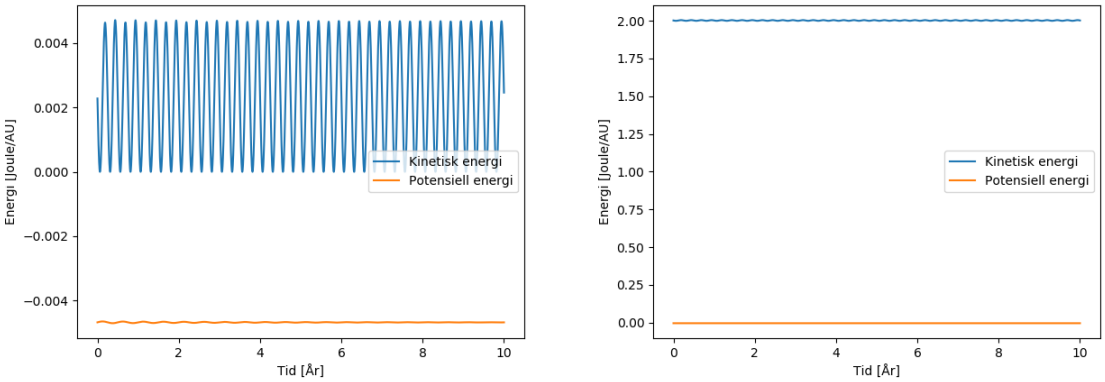
(a) (b)

Figur 2: Figurene viser resultater for simulering for 15 år frem i tid ved N=1000. I figur (a) er Euler Forward brukt, mens i figur (b) er Velocity Verlet brukt.

	Tid (s)		Relativ feil (AU)	
Tidssteg Δt	Forward Euler	Velocity Verlet	Forward Euler	Velocity Verlet
10^{-2}	$6.24 \cdot 10^{-4}$	$6.31 \cdot 10^{-4}$	$1.05 \cdot 10$	$7.11 \cdot 10^{-2}$
10^{-3}	$6.40 \cdot 10^{-5}$	$6.35 \cdot 10^{-5}$	$6.81 \cdot 10^{-3}$	$6.37 \cdot 10^{-3}$
10^{-4}	$6.17 \cdot 10^{-6}$	$6.11 \cdot 10^{-6}$	$6.34 \cdot 10^{-4}$	$6.29 \cdot 10^{-4}$
10^{-5}	$6.15 \cdot 10^{-7}$	$7.23 \cdot 10^{-7}$	$6.29 \cdot 10^{-5}$	$6.28 \cdot 10^{-5}$

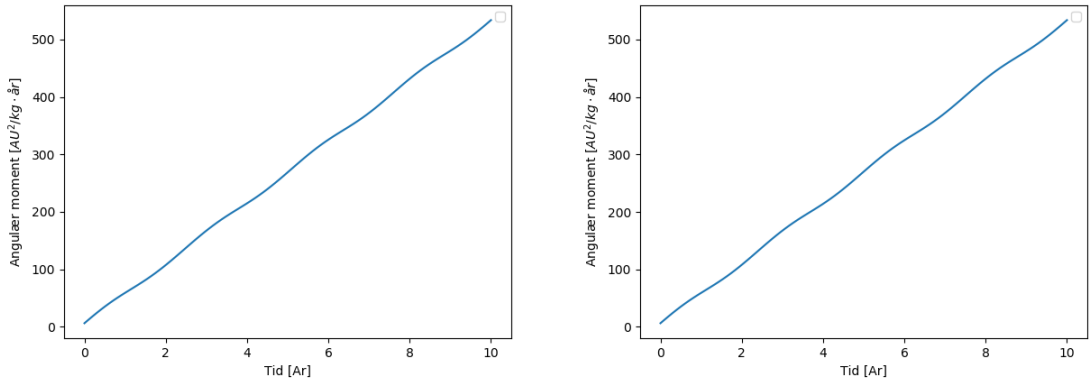
Tabell 2: Tabell som viser tiden det tar å kjøre algoritmene, sammmen med den relative feilen basert på initial- og sluttposisjon etter ett år. Resultatene er beregnet ved forskjellige tidssteg.

4.1.3 Energi og angulærmoment



(a) (b)

Figur 3: Resultater over den kinetiske energien og potensielle energien. Forward Euler er brukt i (a), mens Velocity Verlet er brukt i (b). Δt=1E-3

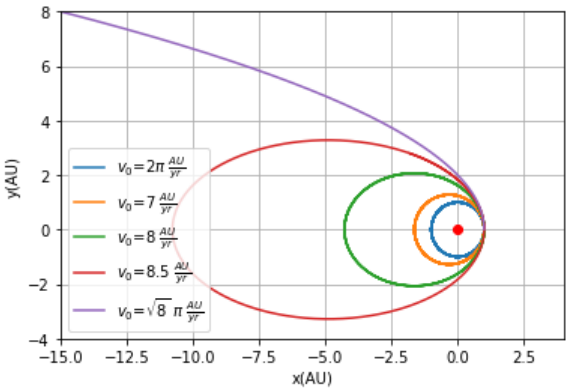


(a) (b)

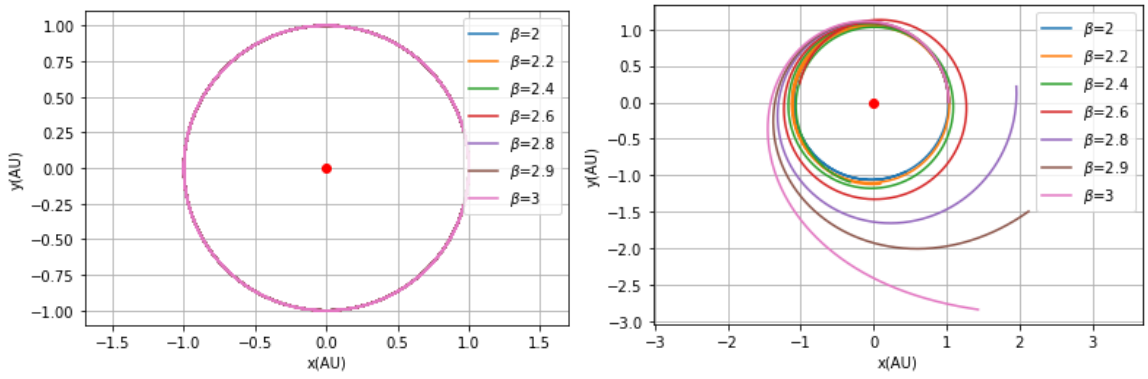
Figur 4: Resultater over det angulære momentet for de forskjellige metodene. Figur (a) viser Forward Euler, mens Velocity Verlet er brukt i (b). Δt=1E-3

4.1.4 Unnslippshastighet og lengere avstand fra sola

Her etterlates Forward Euler bak, hvor resten av resultatene er beregnet ved bruk av Velocity Verlet.



Figur 5: Jordas bane rundt sola, ved forksjellige initialhastigheter $(0,V_0)$. Initialposisjonen er satt til (1,0)

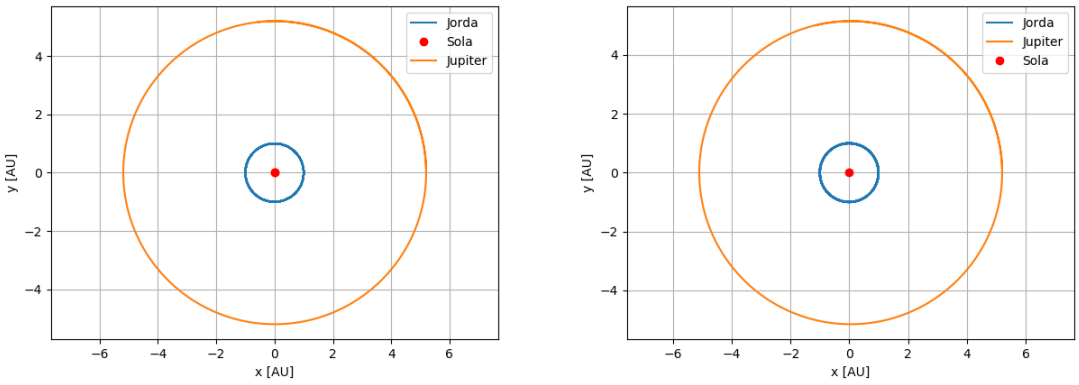


(a) (b)

Figur 6: Resultater over Jordas bane rundt sola med forskjellige β verdier i likning 11. I figur (a) er initialposisjonen (1,0), i figur (b) er initialposisjonen (1.03,0), initialhastigheten er $(0,2\pi)$. Brukt tidssteg Δt=4E-4

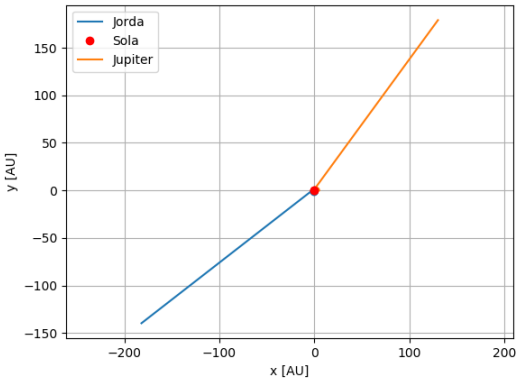
4.2 System med Sola, Jorda, og Jupiter

Ved å fylle på systemet med Jupiter itillegg til Jorda vil systemet forandre seg.



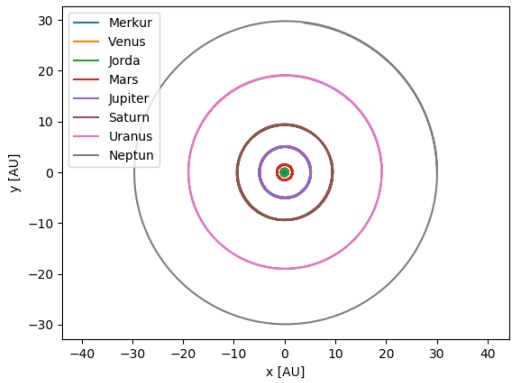
(a) (b)

Figur 7: Resultater fra Jordas og Jupiters bane rundt sola. I (a) er Jupiters egentlige masse brukt, i (b) er Jupiters masse forstørret med en faktor 10. Brukt tidssteg Δt=1.5E-3



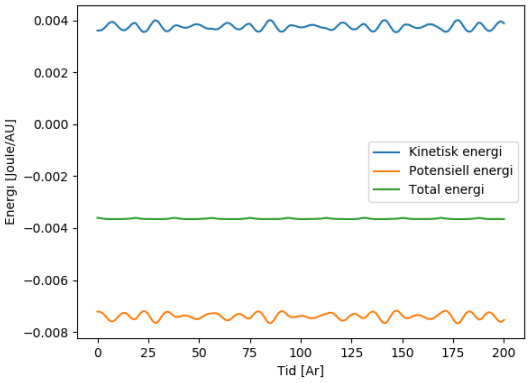
Figur 8: Resultater fra Jordas og Jupiters bane rundt Sola. Jupiters masse er forstørret med en faktor 1000. Brukt tidssteg Δt=1.5E-3

4.3 System med hele solsystemet

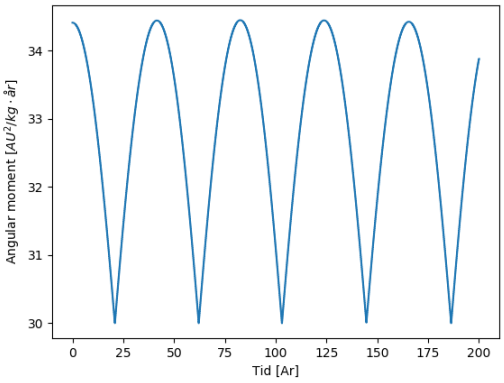


Figur 9: Resultater fra simulering av solsystemet over 200 år. Δt=2E-2

4.3.1 Energi og angulærmoment



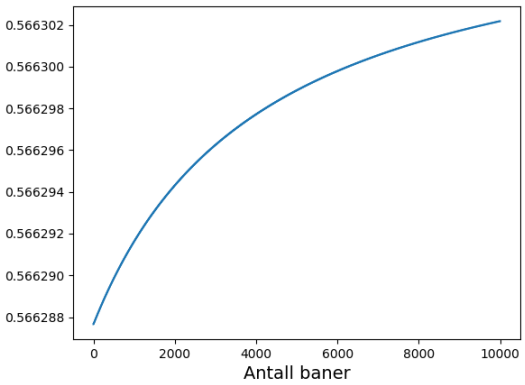
(a)



(b)

Figur 10: Resultater over den kinetiske energien og potensielle energien for hele systemet over 200 år. I (a) er energien plottet, mens i (b) er angulær momentet. Δt=1E-2

4.4 System med Sola og Merkur



Figur 11: Periheliumsvinkelen som funksjon av antall baner rundt sola. Hvor θ_p er langs y aksen

5 Diskusjon

5.1 System med Sola og Jorda

I systemet der kun Sola og Jorda var til stedet ble Forward Euler og Velocity Verlet algoritmene sammenliknet. I figur 1 ble begge metodene plottet for Δt=1E-2 og Δt=-3. Plottene viser relativt liten forskjell, bortsett fra at Forward Euler for Δt=1E-2 ligger bitte litt utenfor banen til resten av kjøringene. Dette viser fortsatt at begge metodene er forsvarlige å bruke så lenge tidsstegene er små nok, men kortere tidssteg fører naturligvis til et større behov for datakraft. Ser også at når Jorda starter lenger ut, følger resten av metodene etter som viser at metodene også fungerer for forskjellige inititialposisjoner.

I figur 2 hvor algoritmene ble kjørt for lengre tidsperioder på 15 år er det lett å se at plottet fra Forward Euler har en 'tykkere' bane som betyr at algoritmen er mer upresis en Velocity Verlet som har en relativt tynn og fin bane.

Da det kan være litt vanskelig å se avvikene i figur 1 der det er mange tidssteg, ble også metodenes presisjon uttrykt ved flere forskjellige størrelser for tidsstegene i tabell 2 ved å se på initial- og sluttposisjon. Etter ett år vil jorda være tilbake der den startet. Altså er det mulig å se på avviket mellom initial- og sluttposisjon iløpet av ett år som den relative feilen. Tiden ble også tatt ved kjøring av de forskjellige metodene. Ifølge tabellen gir Velocity Verlet de resultatene med lavest relativ feil hvor feilen er på omtrent samme størrelse som Δt. Forward Euler har relative feil som ligger såvidt over Velocity Verlet for $\Delta t < 10^{-2}$. Tiden det tar å kjøre algoritmene øker, jo mindre tidsstegene blir. Dette er naturlig da

kortere tidssteg fører til at flere verdier blir beregnet innenfor tidsintervallet. Tiden det tar å utføre Velocity Verlet er omtrent lik som Forward Euler, bortsett fra for $\Delta t = 10^{-5}$ hvor Velocity Verlet bruker lengst tid. Det er forventet at Velocity Verlet burde brukt tydelig lenger tid enn Forward Euler da Forward har færre FLOPs. En årsak til at tidene er såpass like kan være avrundingsfeil i programmet da det opereres med veldig små tider.

I figur 3 kan den kinetiske og den potensielle energien ses for Forward Euler i (a), mens energien kan ses for Velocity Verlet algoritmen i (b). Total energi er summen av kinetisk energi og potensiell energi. Dermed er det tydelig at Forward Euler ikke bevarer den totale energien siden den kinetiske energien varierer relativt mye. For Velocity Verlet er den totale energien tilnærmet konstant da grafene for kinetisk og potensiell energi ser ut til å være ganske konstante. Dermed bevarer Velocity Verlet, energien. Når det gjelder angulær momentet øker angulærmomentet med tiden i figur 4. Det som er litt rart er at (a) ser helt lik ut som (b) noe som kan være grunnet en liten feil i formelen for angulær moment. Det forventes å se en nogenlunde bevaring av angulær moment da Sola går i sirkel rundt sola, dette er hvertfall forventet for Velocity Verlet da energien er tilnærmet konservert.

Ved å variere initialhastighetene er det mulig å prøve seg frem til initialhastigheten som må til for å unnslippe Solas gravitasjonsfelt. I figur 5 er Jordas hastighet testet i intervallet $[2\pi, \sqrt{8}\pi]$, hvor $\sqrt{8}\pi$ er den beregnede unnslipningshastigheten som trengtes for å unnslippe banen. Ut ifra figuren stemmer modellen med beregningene hvor de testede initialhastighetene holder seg i bane rundt Sola selv opp til $8.5\frac{AU}{yr}$. Banene mister sirkelformen og går over til å bli mer og mer elliptiske jo større initialhastigheten blir og til slutt unnslipper banen.

Så ble det sett på hvordan banene endres ved å minke kraften ved økende radius mellom Sola og Jorda ved å endre faktoren avhengig av radius i gravitasjonskraften, $\frac{1}{r^2}$ til $\frac{1}{r^\beta}$. Ved standard initialbetingelser er relasjonen for sirkelbevegelse oppfylt for alle β[2,3] som kan ses i figur 6 a. Dermed ble jorda satt i en initialposisjon litt lenger vekk fra sola. Resultatet kan ses i figur 6 b. Når initialposisjonen er satt til (1.03,0) viser bevegelsene at jo større verdi for β jo mindre tiltrekningskraft påvirker jorda, som videre fører til at jo større β jo større sjanse er det å unnslippe grvitasjonsfeltet til Sola og bryte ut av banen sin.

5.2 System med Sola, Jorda og Jupiter

I figur 7 er Jupiter, Jorda og Sola plottet i samme system. I (a) er massen til Jupiter satt til å være den reelle massen, mens i (b) er massen til Jupiter satt til å være 10 ganger større enn den reelle massen. (a) og (b) viser lite forskjell som kan hinte om en liten feil i programmet, eventuelt upassende verdier for Δt. Teoretisk sett ville den større massen til Jupiter økt tiltrekningskraften mellom Jorda og Jupiter og dermed påvirket Jorda til å endre orbitalbanen sin.

I figur 8 er Jupiters masse forstørret med en faktor 1000. Ut ifra plottet vil både Jorda og Jupiter bli sendt ut av banen. Hvor det at jordkloden blir sendt ut av banen passer med forventningene da tiltrekningskreftene vil øke med massen.

5.3 System med hele solsystemet

I figur 9 ble det antatt at sola er fast, som gir fine og runde baner, dette er fordi programmet ikke klarte å produsere riktige resultater for et fast massesenter. Det som forventes ved et fast massesenter er at sola beveger seg i henhold til resten av systemet for å beholde det fiksede massesenteret, samtidig som det forventes å være lett å se at planetbanene ikke er sirkulære.

Energien i totalsystemet kan ses i figur 10. I (a) vises total energi som ganske konstant som stemmer med teorien. Når det gjelder angulær momentet i (b) varierer angulær momentet stabilt mellom $\approx 35\ \frac{AU^2}{kg\text{år}}$ og $30\frac{AU^2}{kg\text{år}}$. Dette tyder på konservert angulær moment med litt forstyrrelser i intervallet som kan være avrundings- og presisjonsfeil eller påvirkning fra initialverdiene.

5.4 System med Merkur og Sola

Når det gjelder et system der kun Merkur og Sola er tilstede kan periheliumet ses som funksjon av radianer θ_p i figur 11. Siden målet er å finne antall buesekunder per århundre, kan radianer i starten subtraheres fra radianene etter 100 år. Som gir ut ifra figur 11:

$$0.566302rad - 0.566288rad = 0.000014rad = 2.89'' \tag{35}$$

Dette er en del mindre enn den teoretiske verdien på 43" med en relativ feil på 93%. Grunner til dette kan være pga for store tidssteg og numeriske avrundingsfeil.

6 Konklusjon

I denne artikkelen er det vist hvordan Forward Euler kan sammenliknes med Velocity Verlet med tanke på tid, stabilitet og presisjon. Itillegg er det sett på hvordan det er mulig å simulere solsystemet ved å løse ordinære differensiallikninger ved bruk av numeriske metoder. Det er også fokusert på hvordan enkelte planeter oppfører seg i forskjellige systemer. Relativitetsteori er også blitt brukt til å prøve å forklare variasjoner i orbitalbane hos raske planeter. Alt i alt er det sett på hvordan hypoteser og ideer kan simuleres ved bruk av numeriske metoder, der kun kreativiteten setter grenser.

7 Appendiks

7.1 Utledning av Forward Euler

For å bestemme uttrykk for Euler metoden, tas det utgangspunkt i følgende Taylor ekspansjon:

$$x_{i+1} = x(t_i) + h(f(t_i, x_i) + \dots f^{(p-1)}(t_i, x_i)\frac{h^{p-1}}{p!}) + O(h^{p+1}). \tag{36}$$

Hvor x er posisjonen og O(h^{p+1}) er feilleddet.
Siden $\boldsymbol{a}$= $\dot{\boldsymbol{v}}$=$\ddot{\boldsymbol{x}}$ kan likning 36 kan både posisjonen, og hastighet uttrykkes som:

For posisjon:

$$x_{i+1} = x_i + hv_i + O(h^2) \tag{37}$$

For hastighet:

$$v_{i+1} = v_i + ha_i + O(h^2) \tag{38}$$

7.2 Utledning av Velocity Verlet

For å bestemme et uttrykk for hastigheten tas det utgangspunkt i Taylor ekspansjonen til posisjonen:

$$x_{i+1} = x_i + h\dot{x}_i + \frac{h^2}{2}\ddot{x}_i + O(h^3) \tag{39}$$

Som tilsvarer følgende uttrykk for hastigheten:

$$v_{i+1} = v_i + h\dot{v}_i + \frac{h^2}{2}\ddot{v}_i + O(h^3) \tag{40}$$

Vet videre at $\dot{v}$ er kjent gjennom Newtons andre lov, dermed må $\ddot{v}$ uttrykkes via en ekspansjon av $\dot{v}_{i+1}$:

$$\dot{v}_{i+1} = \dot{v}_i + h\ddot{v}_i + O(h^2) \tag{41}$$

Som kan skrives uttykkes uten feilleddet:

$$h\ddot{v}_i \approx \dot{v}_{i+1} - \dot{v}_i \tag{42}$$

Dermed kan dette uttrykket brukes i 40, som gir det endelige uttrykket for hastogheten:

$$v_{i+1} = v_i + \frac{h}{2}\left(\dot{v}_{i+1} + \dot{v}_i\right) + O(h^3) \tag{43}$$

7.3 Utledning av betingelsen for en sirkulær bane rundt sola

Som sagt i teoridelen vil en sirkulær bane ha følgende relasjon:

$$\frac{M_{Earth}v^2}{r} = \frac{GM_{\odot}M_{Earth}}{r^2} \tag{44}$$

Ved å uttrykke Jordas hastighet fås følgende uttrykk:

$$v = \sqrt{\frac{GM_{\odot}}{r}} \tag{45}$$

Videre kan $GM_\odot$ uttrykkes ved følgende relasjon:

$$v^2 r = GM_\odot = 4\pi^2 \frac{AU^3}{yr^2} \tag{46}$$

Ved å sette 46 inn i 45 fås følgende verdi for hastigheten:

$$v = \sqrt{\frac{4\pi^2 AU^3}{AU^2 yr^2}} = 2\pi \frac{AU}{yr} \tag{47}$$

7.4 Utledning av unnslipnignshastighet

For å finne unnslippningshastigheten settes den kinetisk energien lik den potensielle energien.

$$|E_k| = |E_p| \tag{48}$$

Som gir følgende:

$$\frac{M_{Earth} v_{unn}^2}{2} = \frac{GM_\odot M_{Earth}}{r} \tag{49}$$

Som videre uttrykt som unnslipningshastigheten gir:

$$v_{unn} = \sqrt{\frac{2GM_\odot}{r}} \tag{50}$$

Hvor r=1AU og $GM_\odot$=$4\pi^2$, gir unnslipningshastigheten:

$$v_{unn} = \sqrt{\frac{2 \cdot 4\pi^2 \frac{AU^3}{yr^2}}{1AU}} = \sqrt{8}\pi \frac{AU}{yr} \tag{51}$$

7.5 Utledning for relasjon for fast massesenter

For å få et fast massesenter i systemet må bevegelsesmengden i systemet være 0:

$$\vec{P_{Tot}} = 0 \tag{52}$$

Som kan uttrykes som summen av bevegelsesmengder i systemet:

$$\sum_i \vec{P_i} = 0 \tag{53}$$

Som er følgende:

$$\sum_i m_i \vec{v_i} = 0 \tag{54}$$

Videre kan solas bevegelsesmengde trekkes ut fra summen:

$$M_{\odot}\vec{v}_{\odot} + \sum_{i=2} m_i\vec{v_i} = 0 \tag{55}$$

Som gir følgende relasjon for Solas hastighet i et fast massesenter:

$$\vec{v}_{\odot} = -\frac{1}{M_{\odot}}\sum_{i=2} m_i v_i \tag{56}$$

8 Bibliografi

[1] Morten Hjorth-Jensen, august 2015, Computational Physics Lecture Notes,8 Differential equations
(https://github.com/CompPhysics/ComputationalPhysics/ blob/master/doc/Lectures/lectures2015.pdf)

[2] Morten Hjorth-Jensen, November 2019 Project 5,
(https://github.com/CompPhysics/ComputationalPhysics/blob/master/doc/Projects /2019/Project5/pdf/Project5.pdf).

[3] Nasa, Jet Propulsion Labortory, Solar System Dynamics, sist endret 2019,
(https://ssd.jpl.nasa.gov/horizons.cgitop)

[4] Wikipedia, Anguar momentum, sist endret Desember 2019,
(https://en.wikipedia.org/wiki/Angular_momentum)

[5] Hannes Jonsson, 2000, Classical dynamics
(http://physics.drexel.edu/ valliere/PHYS305/Diff_Eq_Integrators/Verlet_Methods/Diffrntleqn3.pdf)

[6] Morten Hjorth-Jensen, 2017,OOexamples
(https://github.com/CompPhysics/ComputationalPhysics/tree/master/doc/Programs/OOExamples/solar-system)

Link til githubmappen: https://github.com/philipkarim/Philip-and-Rohullah-ComFys