PROSJEKTOPPGAVE I FYS-MEK1110

VÅR 2005

BØRGE SÆTER
(borgesa@student.matnat.uio.no)

Gruppe 23

2 vedlegg: hovedprogram, hjelpeprogram

Rapport:

Oppgave a

I denne oppgaven skal vi plotte jordas bane rundt sola i et diagram med sola i origo. Modelleringstiden er 365.25 døgn, som tilsvarer 15778800 sekunder. Initialbetingelsene for posisjon og hastighet til jorda er gitt, og akselerasjonen er grei å finne når vi vet solas masse: avec = -G*Msol*rvec/|r^3|.

Runge-kuttas metode gir en pen kurve ved tusen steg, og vi ser at vi får en nesten sirkulær bane som starter og ender i samme punkt. Det virker altså som denne metoden modellerer systemet bra.

Oppgave b

Nå skal vi kjøre det samme opplegget som over, men variere antall steg fra 500 til 4000. Man skulle kanskje tro at 4000 steg vil gi en mer eksakt beregning av banen fordi vi da får kortere tidsintervaller mellom hver nye beregning av posisjon, hastighet og akselerasjon. Dette viser seg derimot ikke å være tilfelle. Banen blir rund og hel uansett om vi benytter 500 eller 4000 steg. Resultatet er med andre ord ikke kritisk avhengig av antall steg. Sammenliknet med oblig1, hvor vi benyttet Eulers-metode, viser 4.ordens Runge-Kutta seg å være en relativt robust metode.

Oppgave c

Her skal vi finne tidspunktene hvor banen krysser x- og y-aksen. Dette gjør vi ved å, under beregningene, legge inn en test på om produktet av forrige x-posisjon og denne x-posisjonen (n-1 og n) er negativt. Er dette negativt, har disse posisjonene forskjellig fortegn, og vi har krysset y-aksen. Den samme sjekken gjøres for kryssing av x-aksen. Hvis disse testene slår til, skrives tiden ut på skjermen som antall dager etter beregningene startet (1.desember -04).

De resultatene vi får er:

Krysser y-akse etter:

20.1843 døgn

Som tilsvarer 20. desember 2004

Krysser x-akse etter:

109.1415 døgn

Som tilsvarer 20. mars 2005

Krysser y-akse etter:

201.7519 døgn

Som tilsvarer 20. juni 2005

Krysser x-akse etter:

295.2757 døgn

Som tilsvarer 22. september 2005

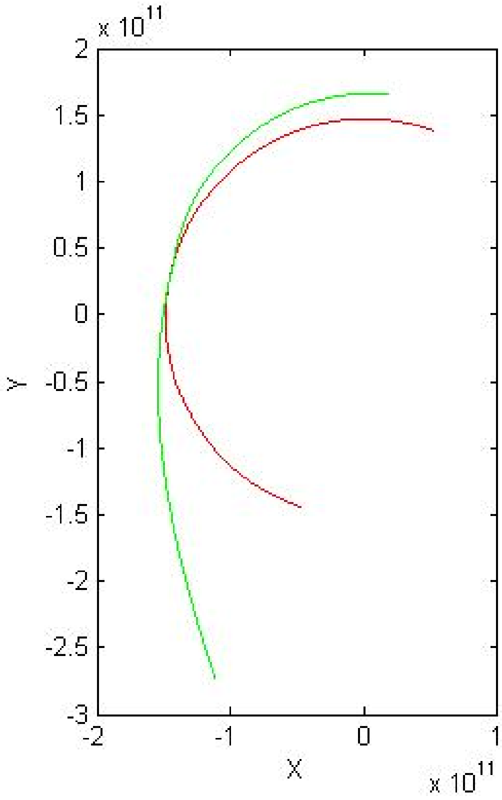
Altså tilsvarer disse datoene (kanskje ikke så uventet) henholdsvis vintersolverv, vårjevndøgn, sommersolverv og høstjevndøgn. Vintersolverv inntreffer egentlig mellom 21. og 22. desember, men ellers stemmer det svært bra.

Dette sier noe om solas posisjon på himmelen. Ved sommer- og vintersolverv står sola henholdsvis på sitt høyeste og laveste på himmelen (motsatt på den sydlige halvkule), vi sier at da ”snur” sola. Ved vår- og høstjevndøgn står sola normalt på ekvator. På disse døgnene er dagen og natta like lange over alt på jorda - altså 12 timer.

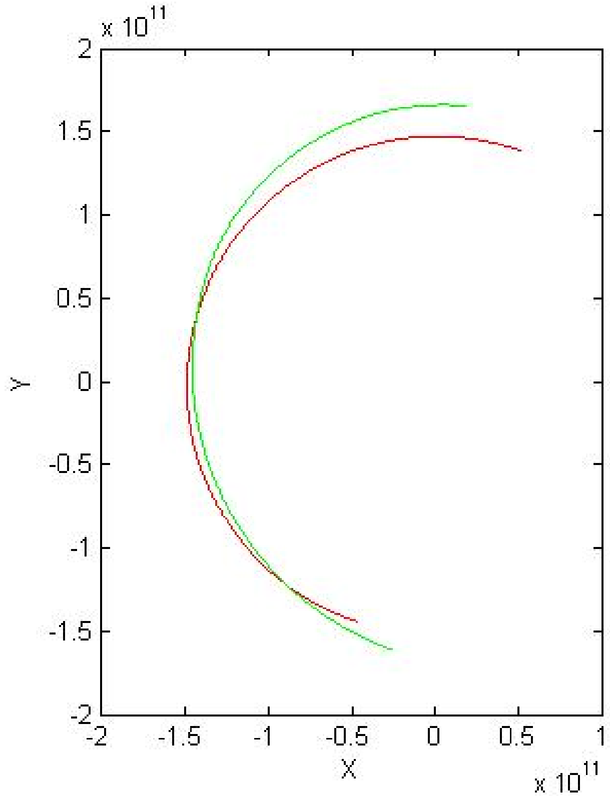
Disse fenomenene skyldes at banen til jorda rundt sola og ekvatorplanet til jorda ikke er parallelle. Dette stemmer jo dermed godt overendes med hvordan vi har definert x- og y-aksene i systemet.

d)
Nå har vi lagt inn startbetingelser og beregninger på banen til Rosetta i tillegg. Dette ble gjort ganske enkelt ved å legge inn seks nye startbetingelser og ellers doble antallet arrayer som blir beregnet. For å finne Rosettas akselerasjon, finner vi først akselerasjonen som virker på Rosetta fra sola på samme måte som vi gjorde for jorda. For å finne avstanden fra Rosetta til jorda tok vi posisjonen til Rosetta i forhold til sola subtrahert med posisjonen til jorda i forhold til sola. Dette brukte vi igjen for å finne akselerasjonen fra jorda på Rosetta, på samme måte som for akselerasjonen fra sola på Rosetta. Til slutt legger vi sammen akselerasjonsvektorene fra sola og jorda for å finne den totale akselerasjonen på Rosetta.

Av resultatet ser vi at Rosettas bane sneier jordas bane før den skytes av gårde igjen bort fra jordbanen i negativ y-retning. Dette er plottet med tusen steg. Figuren ser slik ut:



For å illustrere et interessant poeng viser vi et nytt plott under, hvor massen til sola er satt til å være 1.99E30 i stedet for 1.989E30. Som vi ser, er beregningene ekstremt følsomme for endringer i dataene og man kan således stille seg tvilende til hvor god denne metoden egentlig er til å beregne Rosettas bane. Jordbanen virker å være ganske uforstyrret av denne økningen i solas masse, noe som ikke er så rart når man tenker på jordas store masse i forhold til Rosetta. Da blir denne endringen mer neglisjerbar.

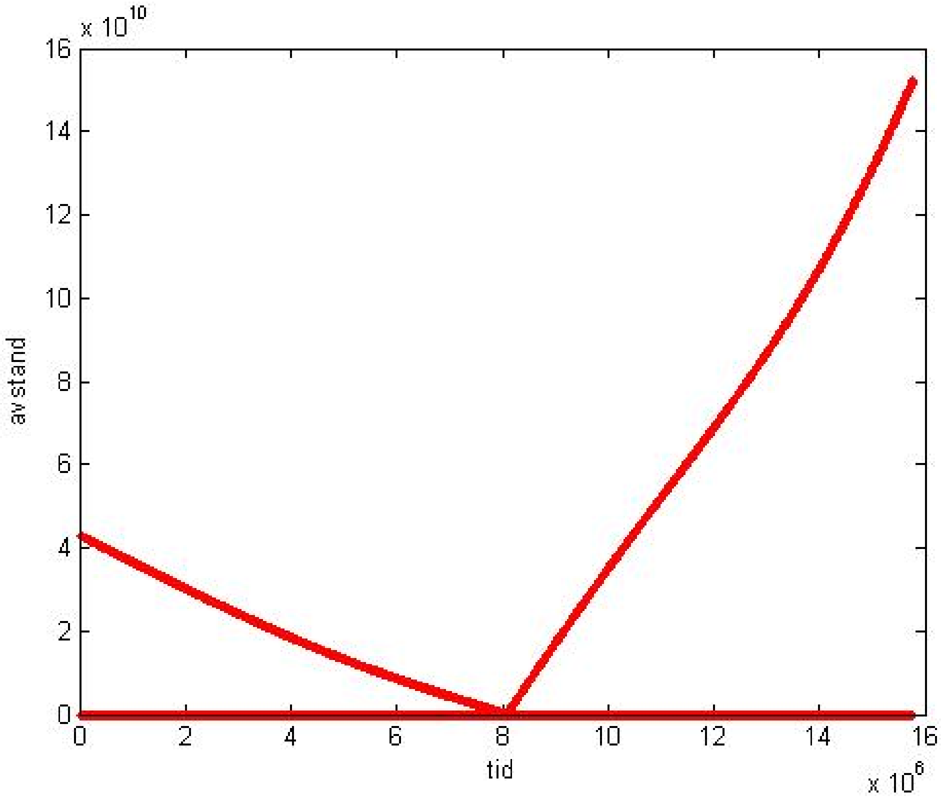


Oppgave e

Nå skal vi, tilsvarende som for oppgave b, kjøre programmet med ulikt antall steg og se hvor mye dette har å si for resultatet. Forskjellen er at nå er også Rosetta med i bergningene, og det viser seg at banen dens er mye mer følsom for endringer i steglengden enn banen til jorda. Utviklingen når man går fra å bruke få steg til å bruke flere er at banen til Rosetta synes å følge jordbanen mer og mer. Hvis man kjører med fire tusen steg går faktisk banen til Rosetta innenfor jordbanen. Hva som er korrekt er ikke helt greit å si i og med at det bygger seg opp større og større feil som følge av unøyaktigheter når man kjører med flere steg. Det som er helt sikkert er i hvert fall at denne metoden ikke er helt ideell.

Oppgave f

I denne oppgaven skal vi plotte avstanden mellom jorda og Rosetta over dette halve året. Dette gjør vi enkelt og greit ved å plotte lengden på differansen mellom avstandene Rosetta-sol og jord-sol. Avstanden legges i en en-dimensjonal array som ”fylles opp” inne i løkka. Plottet viser avstand som funksjon av tid, og er som følger:



Vi ser at avstanden avtar og avtar før den når sin minste verdi på

5.0337e+007 meter, eller 50337 km.

Til sammenlikning er jordradien pluss atomsfæren 6480 km, så den bommer med god margin (i følge disse beregningene).

Når dette skjer har det gått 93.41 dager. Det vil si at den kom nærmest jorda den 4.mars 2005.

Tidspunktet stemmer helt overendes med det som faktisk hendte, men i virkeligheten passerte den bare 1954 km over jorda, eller 8334 meter fra jordas sentrum.

Oppgave g

Nå skal vi forandre programmet slik at det benytter en adaptiv metode for å beregne banen til Rosetta. En slik metode går ut på at man har et eget delprogram som hele tiden analyserer beregningene og finner ut hvor stor steglengde man bør ha. Er endringene i posisjon og hastighet langsomme, kan man bruke lengre tidssteg uten at dette blir noe problem. Skjer derimot endringene fryktelig raskt, må man ha mye kortere tidssteg i denne perioden. Dette gjør at det ikke er helt godt å si hvor mange steg man må ha for å beregne et halvt års omløp rundt sola, men her kan man prøve og feile helt til man får noe man er fornøyd med. Eller man kan la løkka gå helt til man har nådd riktig tid, bare man sørger for å sette av tilstrekkelig med plass i arrayene på forhånd.

Forskjellen i programmet blir at man må innføre en feilmargin som sier noe om hvor stor feil man tolererer. I tillegg går man fra å kjøre rk4 til å kjøre rk4a som er adaptiv. Denne returnerer ikke bare den nye statusen, men også den nye tiden og det nye tidssteget (delta-t endrer seg som sagt i en adaptiv metode). Ellers blir så å si alt som før.

Vi skal nå prøve å gjøre beregningene som vi gjorde i punkt f, og se om vi får noen forskjell. Først legger vi merke til at Rosetta passerer jorda omtrent på samme tid som ved forrige beregning, men nå passerer den kun:

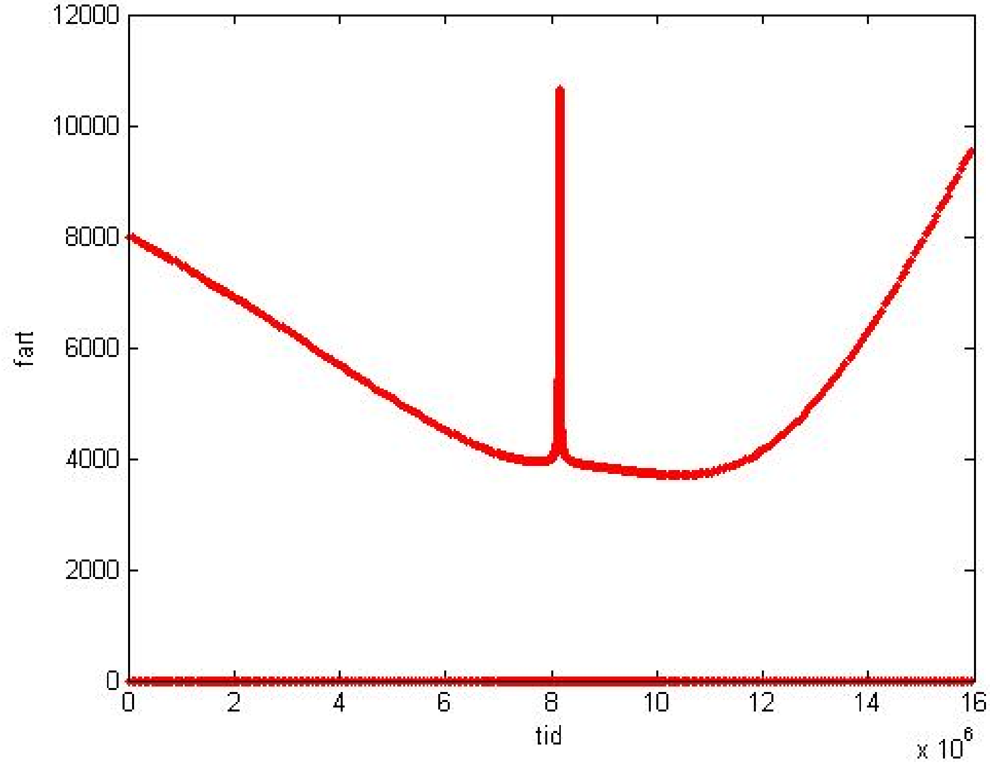
8.1208e+006 m = 8120.8 km fra jordas sentrum, eller 1740.8 km over jordoverflata.

Dette er mye nærmere det som faktisk hendte (1954 km), og det viser seg at beregningene blir forbedret svært mye ved å bruke den adaptive metoden. Rosetta passerer i følge disse beregningene *nærmere* jorda enn den virkelig gjorde!

Når det gjelder spørsmålet om resultatet er kritisk avhengig av angitt feilmargin, synes svaret å være nei. Øker man feilmarginen hundre ganger, forandres den minste avstanden fra jorda med 0.8 kilometer fra 8120.8 til 8121.6 kilometer, noe som ikke kan sies å være en dramatisk endring.

Oppgave h

Nå skal vi plotte den relative farten mellom Rosetta og jorda i den gitte perioden. Dette gjøres på akkurat samme måte som for avstanden mellom jorda og Rosetta, bare at vi nå bruker hastigheten i stedet for posisjonen i utregningene. Resultatet vi får vises under:



Dette er jo litt interessant. Etter hvert som den nærmer seg jorda får den mindre og mindre fart i forhold til jorda. Men på det tidspunktet hvor den er nærmest jorda, får den plutselig en svært stor fart i forhold til jorda. Dette er egentlig ikke så rart når man tenker på hva som faktisk skjer. Det er jo ved dette tidspunktet hvor den får energi fra jorda, den passerer altså svært nærme og svinger litt rundt jorda slik at den får en svært stor hastighet i forhold til jorda akkurat i det den passerer.

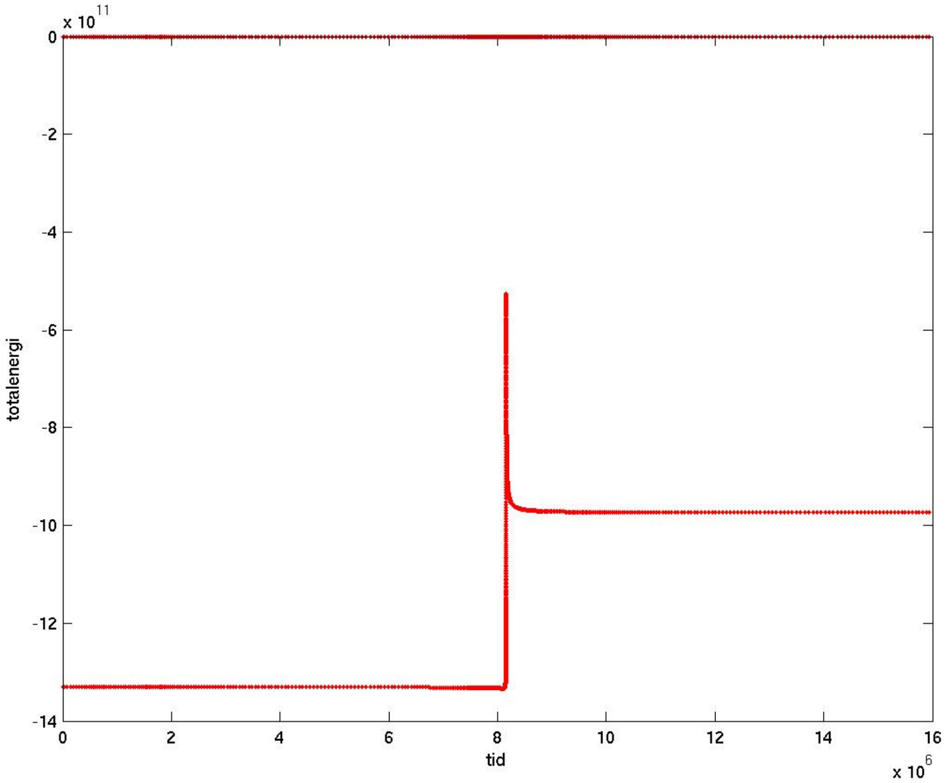
Den maksimale hastigheten den får i forhold til jorda er 1.04*10^4 m/s, eller 38520 km/h. Dette er litt interessant fordi den farta den virkelig hadde i forhold til jorda den 4.mars var 38000 km/h. Beregningene er altså særdeles nøyaktige, spesielt med tanke på de faktorene vi har sett bort fra (gravitasjon fra månen/andre planeter, solvind etc.).

Oppgave i

Nå skal vi beregne den totale mekaniske energien til Rosetta i denne perioden. Dette gjør vi ved å legge sammen Rosettas potensielle energi i forhold til jorda, Rosettas potensielle energi i forhold til sola og den kinetiske energien til Rosetta i forhold til sola.

Den potensielle energien finner vi ved å ta massen til Rosetta ganger massen til sola ganger gravitasjonskonstanten, delt på avstanden mellom de. Vi gjør det samme for den potensielle energien i forhold til jorda, men nå bruker vi jordmassen og avstanden fra Rosetta til jorda. Begge disse energiene er negative. Til slutt finner vi den kinetiske energien ved å finne banefarten, opphøye denne i annen og gange med halve massen til Rosetta.

Plottet til den totale energien til Rosetta ser slik ut:



Energien får altså, ikke uventet, et «hopp» akkurat idet Rosetta passerer jorda. Etter den har passert, har Rosetta større totalenergi enn før den passerte jorda.

Differansen i totalenergi for Rosetta før og etter den hadde passert Jorda ble beregnet til å være

3.5703E11 j

Rosetta har altså «stjålet» 3.5730E11 j mekanisk energi fra jorda.

Oppgave j

Vi skal nå forsøke å forklare hva som menes med «gravity assist». Som vi har sett av beregningene ovenfor går dette ut på å sende et romfartøy i bane svært nær en annen større gjenstand i rommet. Denne gjenstanden har da en kinetsk energi som fartøyet kan «stjele» litt av, ved hjelp av et gravitasjonsstøt. Dette gjør igjen at fartøyet får økt totalenergi i forhold til gjenstanden og sola, slik at den kan vandre litt lengre eller i en litt annen retning. Rosetta bruker altså gravitasjonskraften fra jorda for å bli kasta med en større fart utover i rommet.

Man kan jo også tenke seg at dette fenomenet kan brukes til flere forskjellige formål. Man kan bruke det til å redusere totalenergien til fartøyet (som vi skal se senere), og man kan bruke det til å endre banen til fartøyet når dette måtte ønskes.

Det viktige ved ”gravity-assist” er at man går rundt en gjenstand som har en hastighet i forhold til en annen gjenstand, og får økt den totale energien i forhold til disse to gjenstandene. Det hadde *ikke* virket om vi hadde hatt bare jorda eller bare sola som Rosetta gikk rundt. Men så lenge vi har tre legemer vil dette kunne virke.

Oppgave k

Her skal vi utforske hva som skjer når vi gir Rosetta litt andre initialbetingelser enn de som er gitt originalt. For å gjøre dette, krysser vi en hastighet som er parallell med z-planet med banehastigheten til Rosetta for å få en ny hastighet som er retta rett inn mot eller rett bort fra sola. Resultatet av dette kryssproduktet er at vi legger til en liten hastighet til x- og y-komponeneten av hastigheten til Rosetta. For x-komponenten er dette lille tillegget lik tilleggshastigheten ganger den opprinnelige y-komponeneten. For y-komponenten er dette tillegget lik minus tilleggshastigheten ganger den opprinnelige x-komponeneten av hastigheten.

Vi prøver da med ti forskjellige tilleggshastigheter i intervallet +5E-4 til –5E-4. Det vi legger merke til er at Rosettas minste avstand til jorda avtar når tilleggshastigheten nærmer seg null ovenfra. Når vi legger til negative hastigheter øker først denne minste avstanden for så å avta litt igjen.

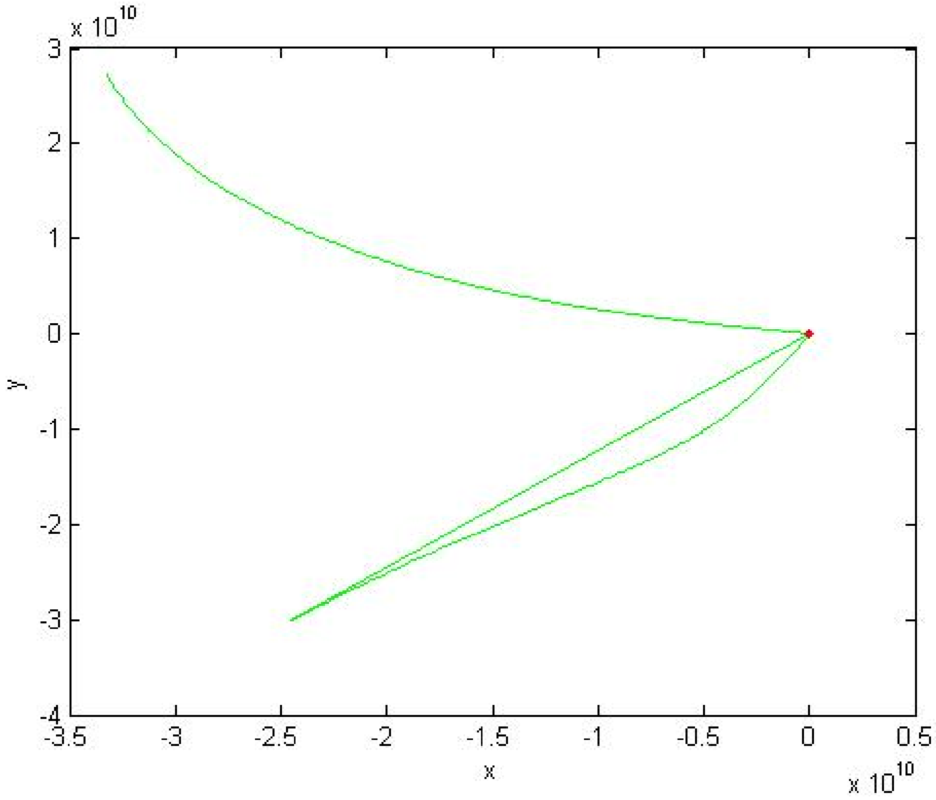
Når det gjelder endringa i total-mekanisk energi, øker den når man legger til positive tilleggshastigheter som blir mindre og mindre i størrelse. Av de initialhastighetene som ble testet var det den opprinnelige som ga størst økning i total-energi, noe som kanskje ikke er så rart i og med at ESA helt sikkert var interessert i å øke energien mest mulig. Man kan se en viss lovmessighet i at passeringer som har liten minsteavstand til jorda gir større økning i totalenergi enn passeringer som har større minimumsavstand til jorda. Rosetta er faktisk den ESA-sonden som har passert nærmest jorda gjennom tidene, så det er nok helt klart noe å vinne på å passere nære jorda.

Hvis vi legger til negative tilleggshastigheter, skjer det noe litt spesielt med endringa i totalenergien – den er negativ. Da har altså Rosetta tapt total-energi etter passeringa av jorda. Den taper mindre og mindre energi jo større negativ verdi tilleggshastigheten har.

Når det gjelder banen til Rosetta legger vi merke til at den kommer innenfor jordbanen ved alle de forskjellige initialbetingelsene, unntatt for den opprinnelige. Tilfeldig?

Oppgave l

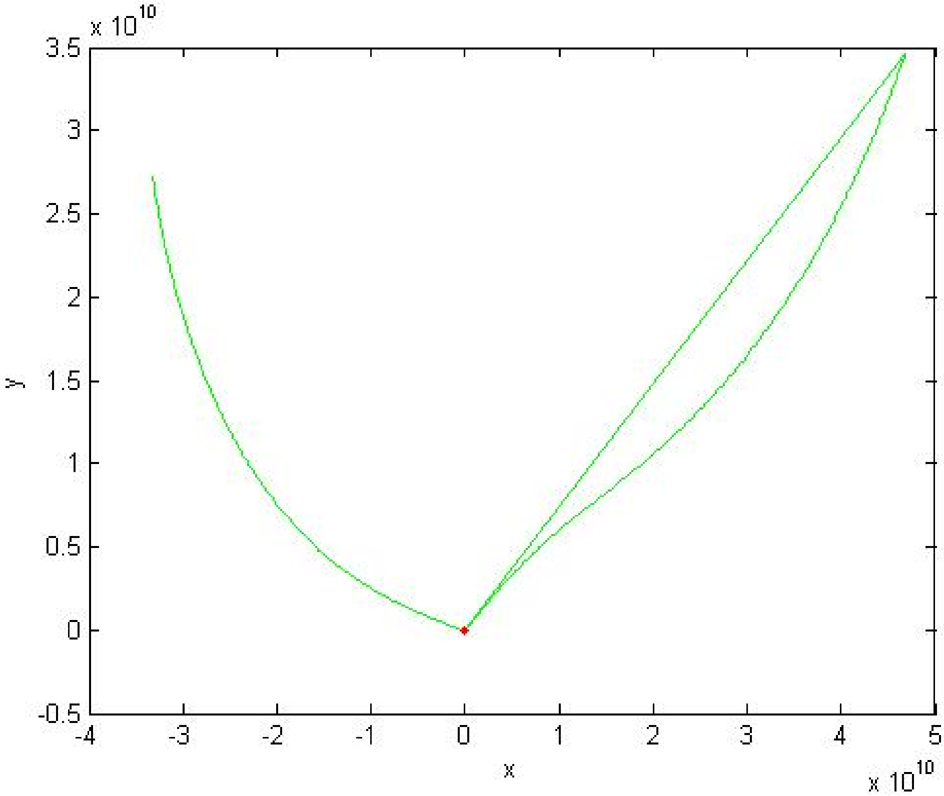
Her skal vi diskutere hva som er det essensielle punktet for om Rosetta skal få økt eller redusert sin totalenergi. For å finne ut av dette har vi plottet posisjonen til Rosetta i forhold til jorden ved å ta avstanden fra Rosetta til sola minus avstanden fra joden til sola komponentvis inne i løkka, og plotta dette i et x-y diagram. For de opprinnelige initialhastighetene til Rosetta ser dette slik ut:



Her er jorda den lille røde prikken og den grønne kurven er banen til Rosetta.

Som vi ser av denne figuren, kommer Rosetta inn nedover fra venstre og gjør en sving rundt jorda. Det man legger merke til er at hvis den skal ha positiv endring i total-energi, må den skytes ut igjen med hastighet i negativ y-retning i forhold til jorda. Hvis den skytes ut med hastighet i positiv y-retning i forhold til jorda, så taper den energi. Dette er egentlig ikke så rart når vi tenker på at jorda på dette tidspunktet også har en hastighet i negativ y-retning i forhold til sola. Hvis da Rosetta kommer ut av støtet med hastighet i negativ y-retning i forhold til sola betyr jo det at den har større fart enn jorda i banen rundt sola. Den har på en måte blitt dytta videre. Men hvis den kommer ut av jordpasseringen med fart i positiv y-retning i forhold til jorda,. har den jo lavere hastighet enn jorda rundt sola. Den har altså på en måte blitt slynget tilbake. Jorda har da vunnet litt energi og Rosetta har tapt litt.

Det viktige er altså at Rosetta blir slynget ut igjen i samme retning som jorda går rundt sola, i forhold til jorda. Det kommer også ganske klart fram av tallene fra oppgave k at avstanden den passerer jorda med har veldig mye å si for hvor stor økningen i energi blir. Til slutt viser vi et eksempel på en passering hvor Rosetta taper totalenergi:



Vi ser her at Rosetta blir slynget ut med en hastighet mot hastigheten til jorda rundt sola, i forhold til jorda.

Oppgave m

Her skal vi forsøke å estimere hvor mye årets lengde for jorda økes fordi Rosetta stjeler litt av energien til jorda under passeringen.

Jorda har i utgangspunktet en kinetisk energi i sin bane rundt sola som er lik:

2.6448e+033 j

Rosetta ”stjeler” en energi lik

3.5703e+011 j

For å finne den nye omløpshastigheten til jorden, må vi trekke den ”stjålne” energien fra den kinetiske energien og finne den nye banefarten ved hjelp av formelen for kinetisk energi . Når vi har den, kan vi bruke uttrykket T = 2*pi*r_jord_sol/v for å finne den nye omløpstiden til jorda omkring sola. Men fordi den energien Rosetta ”stjeler” er totalt neglisjerbar i forhold i til jordas kinetiske energi, er det vanskelig å finne et nytt tall og dermed finne endringen. Men en ting er sikkert, denne endringen kommer ikke til å forandre hverdagen vår!

Helt til slutt bare en liten kommentar utover oppgaven. Hvis man velger å kjøre denne prosessen over mye lengre tid enn et halvt år, ser man, med de opprinnelige initialbetingelsene, at Rosetta begynner å gå i en elliptisk bane rundt sola, og man kan undres på om dette er riktig. Det gir høres jo litt meningsløst ut når den egentlig skal ut og utforske asteroider. Men hvis man ser litt nøyere etter, ser man at banen til Rosetta kommer helt ut til 2.5*10^11 meter fra sola. Det er i dette området banen til Mars befinner seg, og det høres jo ikke så ufornuftig ut i og med at den skal gjøre en liknende manøver rundt Mars som den gjorde rundt jorda, den 26. Februar 2007. Deretter skal den et par turer innom jorda igjen, før den legger ut på reisen mot det egentlige målet.

Det er jo litt morsomt at disse beregningene våre klarer å få fram såpass mange artige detaljer!

```
% Børge Sæter 3/5-05

clear all;
N = 3000;                % Ca antall punkter vi antar å bruke i
beregningene
                         % Brukes her bare for å dimensjonere allokering
                         % av hukommelse til arrayer for plotting o.l.
err = 1.0e-12;           % Feilmargin brukt ved adaptiv Runge-Kutta.
ts = 0.0;                % Velger å starte klokka (t=0) ved forsøkets
start
tn = 31556376/2;         % Omtrentlig tidsrom vi ønsker å studere
bevegelsen i.

rj = zeros(N,3);         % Allokerer plass i hukommelsen for
vj = zeros(N,3);         % posisjon og hastighet,
rsj = zeros(N,3);        % Plass til Rosettas posisjon i forhold til
jorda.
rs = zeros(N,3);         % Plass til Rosettas posisjon i forhold til
sola.
vs = zeros(N,3);         % Plass til Rosettas fart i forhold til sola.
hast = zeros(N);         % Plass til Rosettas fart i forhold til jorda.
avst = zeros(N);         % Plass til Rosettas avstand fra jorda.
totEn = zeros(N);        % Plass til Rosettas totalenergi.
t = zeros(N,1);          % Allokerer plass for faktisk brukt tid i bane
energiFor =0;            % Variabel som skal inneholde totalenergi før
grav. støt.
energiEtter =0;          % Variabel som skal inneholde totalenergi etter
grav. støt.

% Initialbetingelser (posisjon og hastighet i SI-enheter)
r1(1) =  5.166734180331959E+10;
r1(2) = 1.381539034726806E+11;
r1(3) =  -1.604723030678928E+06;
v1(1) =  -2.838631758600434E+04;
v1(2) =  1.032416805998588E+04;
v1(3) =  -1.027156225190673E-01;

% Initialbetingelser for Rosetta (posisjon og hastighet i SI-enheter).
r2(1) =   1.842076556232294E+10;
r2(2) =     1.654310247473529E+11;
r2(3) =    -1.196698444494754E+09;
v2(1) =  -2.647257096491207E+04;
v2(2) =  2.567196158373114E+03;
v2(3) =  3.359804111597520E+01;

% Bestemmer nye initialbetingelser for hastigheten til Rosetta ut i fra
en
% gitt tilleggshastighet.

 tilHast = 0;
 omega = tilHast;
 v2(1) = v2(1) + (v2(2)*omega);
 v2(2) = v2(2) - (v2(1)*omega);

% Hele initialbetingelsene samles i én vektor og paramtre i en annen
status = [r1 v1 r2 v2];
```

```
% Lagrer sentrale data om systemet og sender dette som parameter til
% tilleggsprogrammet.
mSol = 1.989E30;
mJord = 5.979E24;
mSat = 3000;
G = 6.672E-11;
param = [mSol*G, mJord*G];
tid = zeros(N,1);
delta_t=(tn-ts)/(N-1);    % Lengde på tidsstegene. Bare startverdi.

n=0;

minavst = norm(r1(1, :) - r2(1, :)); % Setter første avstanden mellom
Rosetta og jorda som minsteavstanden.
mintid = 0;
while (ts<tn)       % Løkke for å beregne de resterende punktene
    n = n+1;
    rj(n,1) = status(1); rj(n,2) = status(2); rj(n,3) = status(3);
    vj(n,1) = status(4); vj(n,2) = status(5); vj(n,3) = status(6);
    rs(n,1) = status(7); rs(n,2) = status(8); rs(n,3) = status(9);
    vs(n,1) = status(10); vs(n,2) = status(11); vs(n,3) = status(12);
    avst(n) = norm(rs(n,:) - rj(n,:));% Finner avstanden mellom jorda
og Rosetta.
    hast(n) = norm(vs(n,:) - vj(n,:));% Finner hastighten til Rosetta i
forhold til jorden.
    totEn(n) = (-(G*mJord*mSat)/(norm(rs(n, :) - rj(n, :))))...
        + (-(G*mSol*mSat)/(norm(rs(n, :)))) + (0.5 *mSat*norm(vs(n,
:))^2); % Finner Rosettas totalenergi.

if n == 10
    energiFor = totEn(n); % Bestemmer totalenergien før grav. støt.
end;

        % Finner posisjonen til Rosetta i forhold til jorda.
    rsj(n,1) = rs(n,1) - rj(n,1);
    rsj(n,2) = rs(n,2) - rj(n,2);
    rsj(n,3) = rs(n,3) - rj(n,3);


    if n > 1
        % Fyller opp tids-arrayen.
    tid(n) = (tid(n-1) + delta_t);

    %Bestemmer den minste avstanden mellom Rosetta og jorda og tiden
hvor dette inntreffer.
    if avst(n) < minavst
    minavst = avst(n);
    mintid = tid(n);
    end;
    end;

    %Skriver ut når jorda krysser x- og y-aksene.
    %Dette er egentlig IKKE med i koden i den siste delen av oppgaven,
men
    %er lagt til her for at leseren skal kunne se hvordan det er gjort.
    if (n > 1)
    if ((rj(n, 1)*rj((n-1), 1)) < 0)
```

```
        ykryss = t/(60*60*24)

    end;

    if ((rj(n, 2)*rj((n-1), 2)) < 0)

        xkryss = t/(60*60*24)

    end;
end;

    % Finner ny status, tid og tidssteg vha. 4.ordens Runge-Kutas
metode.
    [status ts delta_t] = rk4a(status, ts, delta_t,
err,'tilleggsprog3', param);
end;

% Finner differansen mellom totalenergien før og etter støtet.
energiDiff = totEn(n) - energiFor

% Skriver ut minsteavstanden og tiden hvor den inntreffer
minavst
mintid
% Diverse plot. Valgt 2D
plot(rj(:,1),rj(:,2),'-r');
hold
plot(rs(:,1),rs(:,2),'-b');
daspect([1 1 1]);       % Ønsker å ha lik skalering på alle aksene

 Plotter de forskjellige grafene.
figure;
plot(tid, avst, '.r');
xlabel('tid');
ylabel('posisjon');
figure;
plot(tid,hast, '.r');
xlabel('tid');
ylabel('fart');
figure;
plot(tid,totEn, '.r');
xlabel('tid');
ylabel('totalenergi');
figure;
plot(rsj(:,1), rsj(:,2), '-g');
hold;
plot(0, 0, '.r');
xlabel('x');
ylabel('y');
```

```
% Børge Sæter 3/5-05.

function deriv = corioliskraft3d(status, t, aks)

% Beregner akselerasjonen:
rj = [status(1) status(2) status(3)];  % Henter ut posisjonen
vj = [status(4) status(5) status(6)];  % og hastighet for jord-bane

rs = [status(7) status(8) status(9)];    % Henter ut posisjon
vs = [status(10) status(11) status(12)]; % og hastighet for satellitt-
bane

aj = -aks(1)*rj/(norm(rj)^3); % Finner akselerasjonen på jorda fra
sola.

ass = -aks(1)*rs/(norm(rs)^3); % Finner akselerasjonen fra sola på
Rosetta.
asj = -aks(2)*(rs - rj)/(norm(rs-rj)^3); % Finner akselerasjonen fra
jorda på Rosetta.
as = ass + asj; % Summerer vektorene for akselerasjonen fra jorda og
sola på Rosetta.

% Returnerer de deriverte av statusvektor (dvs hast. og aksel.) for
jorda og Rosetta.
deriv = [vj(1) vj(2) vj(3) aj(1) aj(2) aj(3) vs(1) vs(2) vs(3)...
        as(1) as(2) as(3)];
return;
```