

Korreksjoner ved prosjektoppgaven mandag kl 1240

Feil 1: a(n), (pluss en parentes for lite)

Det snek seg inn en feil i uttrykket for akselerasjon i hjelpeskrevet nesten øverst s 3. Opprinnelig tekst:

```
a(n) = - ( G*M / (norm(r(n,:))^3 ) * r(n,:); % Beregning av akselerasjon ut fra de kreftene
% som finnes. Egentlig ikke nødvendig å lagre alle a-ene i en array, så en enkel variabel
% for a er å foretrekke.
v(n+1,:) = v(n,:) + a(n)*delta_t;
r(n+1,:) = r(n,:) + v(n+1,)*delta_t;
```

Korrigert tekst:

```
a = - ( G*M / (norm(r(n,:)))^3 ) * r(n,:); NB Her var det en parentes for lite opprinnelig!
v(n+1,:) = v(n,:) + a*delta_t;
```

Da bør a heller **ikke** være deklart på forhånd gjennom noe `a=zeros(N,1)` eller `a=zeros(N,3)`; for a blir med den korrigerte koden bare en enkel array med tre elementer, svarende til `ax`, `ay`, `az`.

Oppklaring 1: a i AU/jorår-enheter

Koden ovenfor:

```
a = - ( G*M / (norm(r(n,:))^3 ) * r(n,:);
```

er korrekt så lenge man er innenfor SI-enheter, men når vi går over til astronomiske enheter og jordår, kan vi ikke bruke G og M i SI-enheter uten videre. For å gjøre en lang historie kort kan vi si at vi da må erstatte $G*M$ med en "konvertert $G*M$ " slik at uttrykket inne i løkka blir:

```
a = - ( GM_konvertert / (norm(r(n,:))^3 ) * r(n,:);
```

mens `GM_konvertert` kan finnes ved å bruke tredje ligning side 5 i hjelpeskrevet, det vil si:

```
GM_konvertert = G*M*(jordaar^2) / (AU)^3;
```

Her har jeg skrevet "jordaar" i stedet for `k_tid` og `AU` i stedet for `k_lengde`, i et forsøk på å få enda bedre fram hvilke størrelser som inngår (jordaar er antall sekund i et jordår, og `AU` er antall meter i en astronomisk enhet).

Merk at man bør legge beregningen av `GM_konvertert` FØR løkken (ikke inne i løkken) for å spare regnetid. Det er fullt mulig siden ingen av størrelsene ved beregning av `GM_konvertert` endrer seg gjennom beregningene. Uttrykket for `a` derimot må selvfølgelig ligge inne i løkka.

Legg forøvrig merke til at vi i første del av prosjektoppgaven bare har bruk for $G*M_{sol}$, og at det er oppgitt `GMsol` med stor presisjon (langt bedre presisjon enn `G` er kjent med). Det er derfor unødvendig å regne seg tilbake til `Msol` ved å dividere `GMsol` med `G`, for så å multiplisere med `G` igjen.

I siste del av oppgaven (tre-legemer) kan man bruke masseforholdet sol/jupiter for å lage `GMjupiter`, igjen uten at vi behøver å gå veien om `G`. Men dette er en smaksak.

Feil 2: Et komma for mye

I interpolasjonsuttrykket er det sneket seg inn et komma for mye (to steder). Korrekte uttrykk skal være:

$$\frac{\delta}{\Delta t} = \frac{-y_n}{y_{n+1} - y_n}$$

Da er den nokså nøyaktige tiden for passering av xz-planet gitt ved:

$$t_{kryssing} = t_n - \frac{y_n}{y_{n+1} - y_n} \cdot \Delta t$$

Oppklaring 2: Delta_t

Mange lurer på hvor lang delta_t bør være og hvor mange arrayelementer N man bør ha. Man må da tenke over det fysiske systemet vi skal gjøre beregninger på. Det vil antakelig være tilstrekkelig å beskrive en omdreining ved hjelp av 1000 punkter. Men siden Jupiter beveger seg betydelig langsommere enn jorda, kan det være ok å starte med å velge delta_t lik 1/100 jordår og se om dette siden svarer til omtrent/minst 1000 punkter for Jupiter (og det vil det).

For å finne antall punkter i arrayene vil det da være lurt med tanke på senere del av oppgaven å definere total_tid som vi ønsker å gjøre beregninger for, og dernest beregne antall punkter. I punkt b ser vi at vi skal gjøre beregninger for 50 år, og da vil koden kunne se slik ut:

```
total_tid = 50;
delta_t = 1/100;
N = total_tid/delta_t;
```

Tips om test-utskrifter

Når vi har lagt inn initialbetingelsene, f.eks.slik:

```
r(1,:) = [-5.7582951908421E+11 -5.7045119979515E+11 1.5253585405754E+10];
```

og så har gjort konverteringen til AU slik:

```
r(1,:) = r(1,)/AU;
```

kunne det vært ok å ta en testutskrift for å se at alt gikk bra. Men dersom vi forsøker å gjøre det ved å sløyfe semikolonet her, dvs slik:

```
r(1,:) = r(1,)/AU
```

så får vi ikke bare ut de tre tallene vi ønsker, men HELE den todimensjonale r-arrayen! For å få en brukbar utskrift bør vi heller lage en ekstra linje med en enkel variabel (som da blir omgjort til en endimensjonal array med tre elementer), slik:

```
test_r = r1(1,)/AU
```

Med denne linjen får vi ut på skjermen bare de tre elementene vi ønsker.

