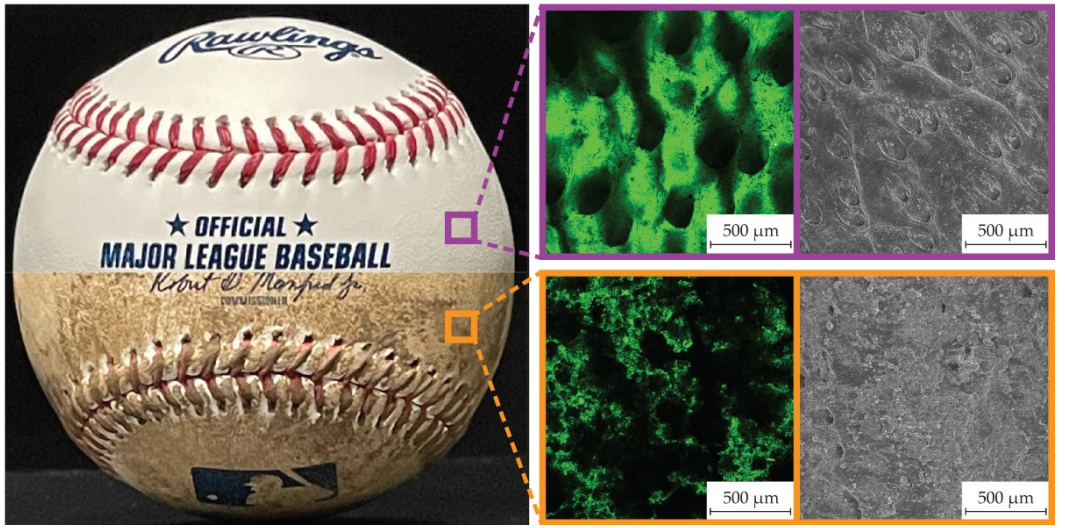
FYS1100 Mekanikk og modellering, vår 2025

Forelesning uke 8: Energiprinsippet. Numerisk derivasjon og integrasjon (AMS 11.4, 6.2, 7.5)

Læringsmål: Vite hva **numerisk derivasjon** og **integrasjon** er og hvordan det brukes i praksis. Se på betydningen av størrelsen på ***tidsintervallet***.





Beskjeder:

- ✓ Vi skal programmere i seminartimen, ta med laptop
- ✓ Neste uke: ***ingen undervisning***, midtveis-eksamensuke 😊 (Info om midtveis-eksamen straks!)
- ✓ ***Eksamenstrening*** i stedet for diskusjonsoppgaver de første 45 minuttene i gruppetimene denne uka! 😊
- ✓ Oblig3: legges ut denne uka (frist uka etter midtveiseksamen). *Husk å gi beskjed om dere trenger utsettelse på oblig*!

Mandag 10.03.2025

Midtveiseksamen ☺️

Skriftlig, digital eksamen med Inspera:
https://www.uio.no/studier/eksamen/inspera/index.html

1. Tid og sted: tirsdag 18. mars kl. 15-18, Silurveien. OBS: sjekk hvor du skal møte i Studentweb! ***Møt i god tid!***
2. Digital eksamen med 20 flervalgsoppgaver, PC-er er på plassene på Silurveien (NB: ikke lov å ha med egen PC til eksamen!)
3. Pensum: alt vi har gjennomgått så langt + denne ukas temaer: grovt sett fra og med kapittel 3 til og med kapittel 11 i læreboka, og notatet om diff-ligninger, og noen delkapitler fra Kalkulus-boka (se liste som legges ut i Canvas)
4. Tillatte hjelpemidler: godkjent kalkulator og Rottmann
5. Formelark er tilgjengelig i Inspera

«Trøsterunde» 🤗:
Om dere lurer på noe i selve oppgaveteksten, kommer jeg innom og går en runde ca. kl. 16 og ca kl. 17. Jeg kan *ikke* svare på om dere har gjort riktig! Kun avklare selve oppgaveteksten.

Formelark FYS1100

$\sum \vec{F} = m\vec{a} = \frac{d\vec{p}}{dt}$, hvor $\vec{p} = m\vec{v} = m\frac{d\vec{r}}{dt}$ og $\vec{a} = \frac{d\vec{v}}{dt} = \frac{d^2\vec{r}}{dt^2}$

Konstant $\vec{a}$: $\vec{v} = \vec{v}_0 + \vec{a}t$, $\vec{r} = \vec{r}_0 + \vec{v}_0 t + \frac{1}{2}\vec{a}t^2$, $v^2 - v_0^2 = 2\vec{a}\cdot(\vec{r} - \vec{r}_0)$

Konstant α: $\omega = \omega_0 + \alpha t$, $\theta = \theta_0 + \omega_0 t + \frac{1}{2}\alpha t^2$, $\omega^2 - \omega_0^2 = 2\alpha(\theta - \theta_0)$

Baneakselerasjon: $\vec{a} = \frac{d\vec{v}}{dt}\hat{u}_T + \frac{v^2}{r}\hat{u}_N$

Rotasjon: $\vec{v} = \vec{\omega} \times \vec{r}$, $\vec{a} = \vec{\alpha} \times \vec{r} + \vec{\omega} \times (\vec{\omega} \times \vec{r})$

Galilei-transformasjon, referansesystemer: $\vec{r} = \vec{R} + \vec{r}'$, $\vec{v} = \vec{V} + \vec{v}'$, $\vec{a} = \vec{a}'$

Fjærkraft: $F(x) = -k(x - x_0)$, luftmotstand: $\vec{F}_v = -k\vec{v}$ eller $\vec{F}_v = -D|\vec{v}|\vec{v}$

Statisk friksjon: $|\vec{F}_s| \leq \mu_s N$, dynamisk friksjon: $|\vec{F}_d| = \mu_d N$

Arbeid: $W_{AB} = \int_{t_A}^{t_B} \vec{F}\cdot\vec{v}\,dt = \int_A^B \vec{F}\cdot d\vec{r} = K_B - K_A$, kinetisk energi: $K = \frac{1}{2}mv^2$

Potensiell energi for gravitasjon: $U = mgy$, for fjærkraft: $U = \frac{1}{2}k(x - x_0)^2$

Effekt: $P = \lim_{\Delta t \to 0}\frac{\Delta W}{\Delta t} = \vec{F}\cdot\vec{v}$

Konservativ kraft: 1D: $F(x) = -\frac{dU(x)}{dx}$, 3D: $\vec{F} = -\vec{\nabla}U(\vec{r})$

Impuls: $\vec{J} = \int_{t_0}^{t_1}\vec{F}dt = \Delta\vec{p} = \vec{p}(t_1) - \vec{p}(t_0)$ Rakettligningen: $\vec{F}^{\text{ext}} + \vec{v}_{\text{rel}}\frac{dm}{dt} = m\vec{a}$

Massesenter: $\vec{R} = \frac{1}{M}\sum_i m_i\vec{r}_i = \frac{1}{M}\int_M \vec{r}dm$, $M = \sum_i m_i = \int_M dm$

Kraftmoment: $\vec{\tau} = \vec{r} \times \vec{F}$, spinn: $\vec{L} = \vec{r} \times \vec{p}$

Spinnsats: $\vec{\tau} = \frac{d\vec{L}}{dt}$ Stive legemer: $L_z = I_z\omega_z$, $\tau_z = I_z\alpha_z$

Kinetisk rotasjonsenergi: $K = \frac{1}{2}I\omega^2$ Treghetsmoment: $I = \sum_i m_i\rho_i^2 = \int_M \rho^2 dm$

Parallellakseteoremet: $I = I_{cm} + Md^2$ Rullebetingelse: $V = \omega R$

Gravitasjon: $\vec{F}(\vec{r}) = -G\frac{m_1 m_2}{r^2}\hat{u}_r$, $U(r) = -G\frac{m_1 m_2}{r}$; G er gravitasjonskonstanten

Lorentz-transformasjon: $x' = \gamma(x - ut)$, $y' = y$, $z' = z$, $t' = \gamma\left(t - \frac{u}{c^2}x\right)$, $\gamma = \frac{1}{\sqrt{1-\frac{u^2}{c^2}}}$

Lorentz-transformasjon for hastighet: $v' = \frac{v-u}{1-\frac{u}{c^2}v}$

Tidsdilatasjon: $\Delta t = \gamma\Delta t_0$; lengdekontraksjon: $l = \frac{l_0}{\gamma}$; relativistisk kinetisk energi: $K = m_0c^2(\gamma - 1)$, hvor m_0 er hvilemassen.

Taylor-rekke til en funksjon $f(x)$ om funksjonsargumentet $x = a$:

$T_n(f;a)(x) = f(a) + \frac{1}{1!}(x-a)\frac{df(x=a)}{dx} + \frac{1}{2!}(x-a)^2\frac{d^2f(x=a)}{dx^2} + \cdots + \frac{1}{n!}(x-a)^n\frac{d^nf(x=a)}{dx^n}$

Restledd: $R_n(f;a)(x) = \frac{(x-a)^{n+1}}{(n+1)!}\cdot\frac{d^{n+1}f(x=\zeta)}{dx^{n+1}}$ hvor $\zeta \in [x,a]$ er ukjent.

Difflikninger: Separabel 1.ordens lineær diff-likn.:
$q(x)\cdot\frac{dx}{dt} = p(t) \Rightarrow \int q(x)\,dx = \int p(t)\,dt \Rightarrow Q(x) = P(t) + C$, med $\frac{dQ}{dx} = q(x)$ og $\frac{dP}{dt} = p(t)$
2.ordens homogene lineære diff-likn. med konstante koeffisienter:
$\frac{d^2x(t)}{dt^2} + p\frac{dx(t)}{dt} + qx(t) = 0 \Rightarrow$ Karakteristisk likn. $\lambda^2 + p\lambda + q = 0$ med generelt to røtter $\lambda_1$og λ_2.
Generelle løsninger: To reelle røtter: $x(t) = C_1e^{\lambda_1 t} + C_2e^{\lambda_2 t}$ En reell rot: $x(t) = C_1e^{\lambda t} + C_2te^{\lambda t}$
To komplekse røtter, $\lambda_1 = a + ib, \lambda_2 = a - ib : x(t) = e^{at}(C_1\cos(bt) + C_2\sin(bt))$

Ikke-konservative krefter og energiprinsippet

[✏️ lysbilde med skriblerier på iPad ✏️]

Vi kan dele opp nettokraften vår i

- Den konservative kraften (eller summen av dem): $\vec{F}$
- Den ikke-konservative kraften (eller summen av dem): $\vec{f}$

$\vec{F}_{net} = \vec{F} + \vec{f}$

Arbeidet: $W_{0,1} = \int_{t_0}^{t_1} (\vec{F}+\vec{f})\cdot\vec{v}\,dt$

$= W_F + W_f$

Vi vet at for en konservativ kraft, kan vi finne et potensial slik at:

$W_F = U_0 - U_1$

$\Rightarrow W_{0,1} = W_F + W_f = U_0 - U_1 + W_f$

Arbeid-energi-teoremet: $W_{0,1} = K_1 - K_0 = U_0 - U_1 + W_f$

$K_1 + U_1 = K_0 + U_0 + W_f$

$E_1 \qquad E_0$

Forskjell i mek.energi:

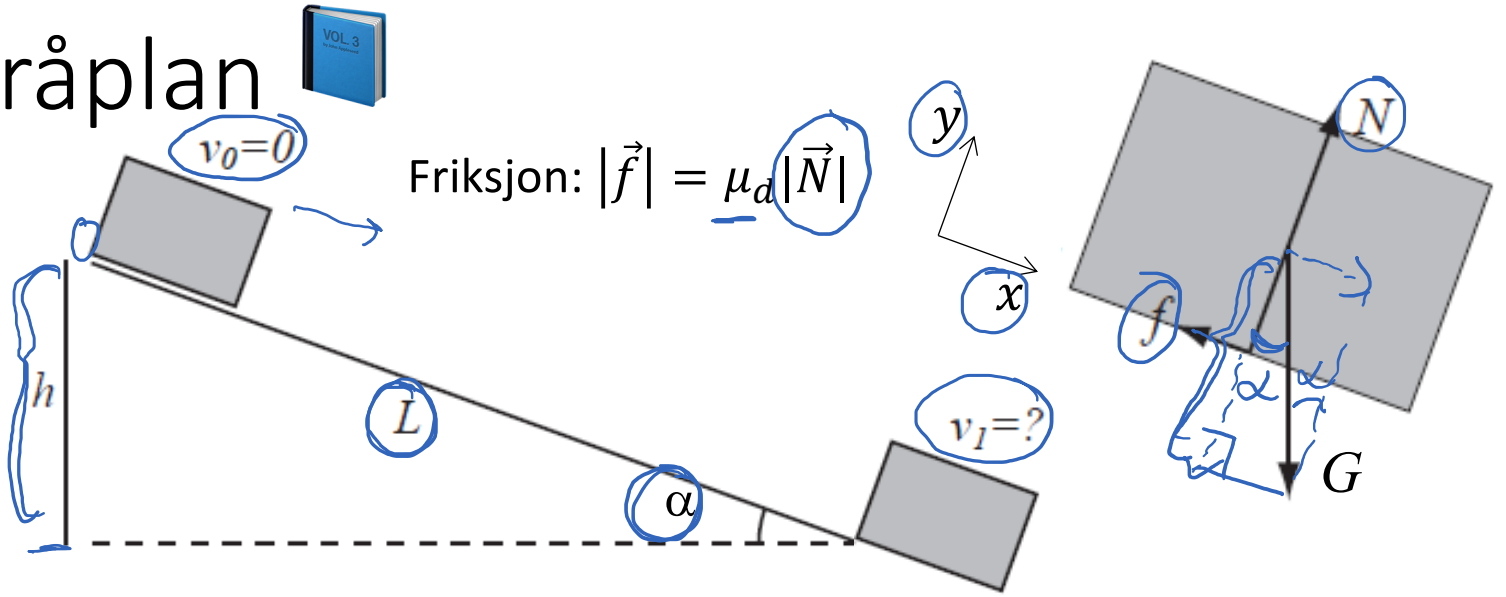
$\Rightarrow \Delta E = E_1 - E_0 = W_f = \int_{t_0}^{t_1} \vec{f}\cdot\vec{v}\,dt$

Forandring i den mekaniske energien = arbeid gjort av *ikke*-konservative krefter 😀

Eksempel: bok på et skråplan

[✏️ lysbilde med skriblerier på iPad ✏️]

Ei bok sklir på et skråplan med helnings-vinkel α og med friksjons-koeffisient μ_d. Vi ser bort fra luftmotstand. Hva er farten v_1 til boka når den har sklidd ned skråplanet?



Lengde h

N2L i y-retning:

$$\sum F_y = N - mg\cos\alpha = 0$$

$$\Rightarrow N = mg\cos\alpha \quad \Rightarrow \text{Friksjonskraft: } f = -\mu_d mg\cos\alpha$$

Potensiell energi: $U_0 = mgh = mgL\sin\alpha$, $U_1 = 0$ (valgt selv!)

Forandring i mekanisk energi: $\Delta E = (K_1 + U_1) - (K_0 + U_0) = W_f = \int f\,dx$

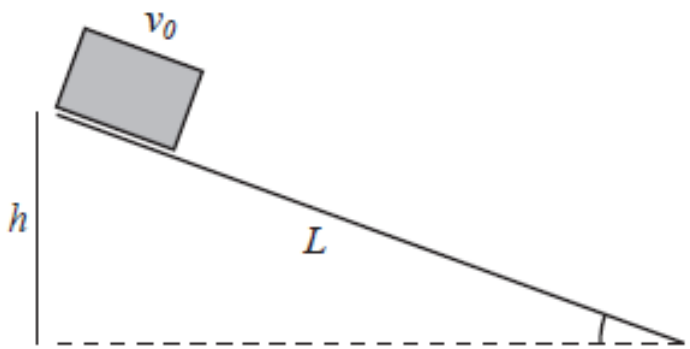
$$\Rightarrow \Delta E = \frac{1}{2}mv_1^2 + 0 - (0 + mgL\sin\alpha) = -\mu_d mg\cos\alpha \cdot L$$

$$\frac{1}{2}mv_1^2 - mgL\sin\alpha = -\mu_d mgL\cos\alpha \Rightarrow v_1 = \sqrt{2gL(\sin\alpha - \mu_d\cos\alpha)}$$

Hvor ble det av energien ΔE? 🤔

Termisk energi ☕

[✏️ lysbilde med skriblerier på iPad ✏️]



Friksjon ⇒ *atomære* vibrasjoner
kinetisk og potensiell energi på *mikroskopisk nivå*

Mikroskopiske bevegelser ⇒ varme

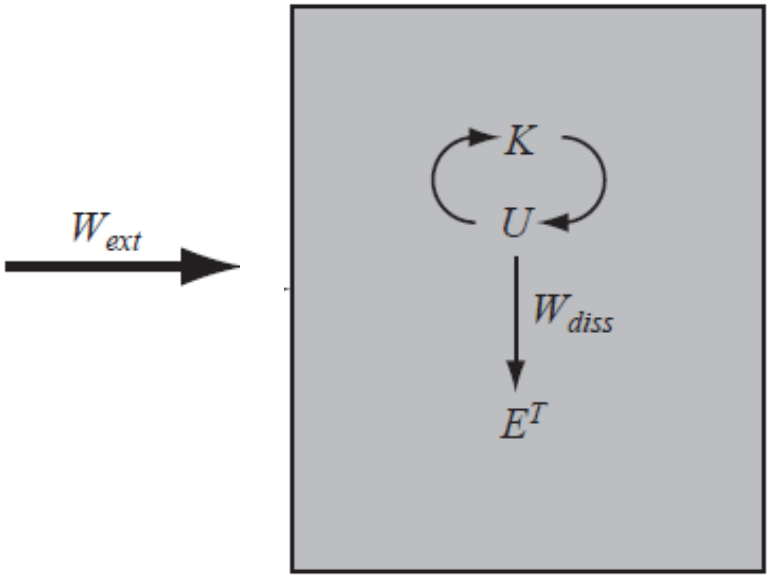
Friksjon *varmer* boka og planet
⇒ temperaturen i systemet «bok + skråplan» øker

termiske energi

Energien i hele systemet er bevart: $E_{tot} = K_0 + U_0 + E_0^T = K_1 + U_1 + E_1^T$

Arbeid fra en ytre kraft:
- jeg skyver boka opp
- jeg løfter boka opp

$\Delta E_{ext} = W_{ext}$



Lukket system:

- konservative krefter
  kinetisk ⇔ potensiell energi
- ikke–konservative krefter
  (dissipative krefter)
  mekanisk ⇒ termisk energi

Praktisk tips til ukas hovedtema ☺️

Hans Petter Langtangen: «A Primer on Scientific Programming with Python»

https://link.springer.com/book/10.1007/978-3-642-30293-0%20

3.1 Functions 99

3.1.7 Multiple Return Values

Python functions may return more than one value. Suppose we are interested in evaluating both $y(t)$ and $y'(t)$:

$$y(t) = v_0 t - \frac{1}{2} g t^2,$$
$$y'(t) = v_0 - g t\,.$$

To return both y and y' we simply separate their
a comma in the `return` statement:

```
def yfunc(t, v0):
    g = 9.81
    y = v0*t - 0.5*g*t**2
    dydt = v0 - g*t
    return y, dydt
```

Calling this latter `yfunc` function makes a need for two
of the assignment operator because the function returns

```
position, velocity = yfunc(0.6, 3)
```

il å oversette)

As an example, consider a function for computing the second-order derivative of a function $f(x)$ numerically:

$$f''(x) \approx \frac{f(x-h) - 2f(x) + f(x+h)}{h^2}, \qquad (3.3)$$

where h is a small number. The approximation (3.3) becomes exact in the limit $h \to 0$. A Python function for computing (3.3) can be implemented as follows:

```
def diff2nd(f, x, h=1E-6):
    r = (f(x-h) - 2*f(x) + f(x+h))/float(h*h)
    return r
```

Kap. 3.1.12

Praktisk tips til ukas hovedtema ☺️

Joakim Sundnes: «Introduction to Scientific Programming with Python»

Gratis å laste ned fra UiO-nettet! 👍 NB! Python3 🥰

https://link.springer.com/book/10.1007/978-3-030-50356-7

Kapittel 4: Functions and Branchings



Numerisk derivasjon

[✏️ lysbilde med skriblerier på iPad ✏️]

Definisjonen av den deriverte:

$$\frac{df(x)}{dx} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

Numerisk derivasjon: diskretiserer f(x) og x

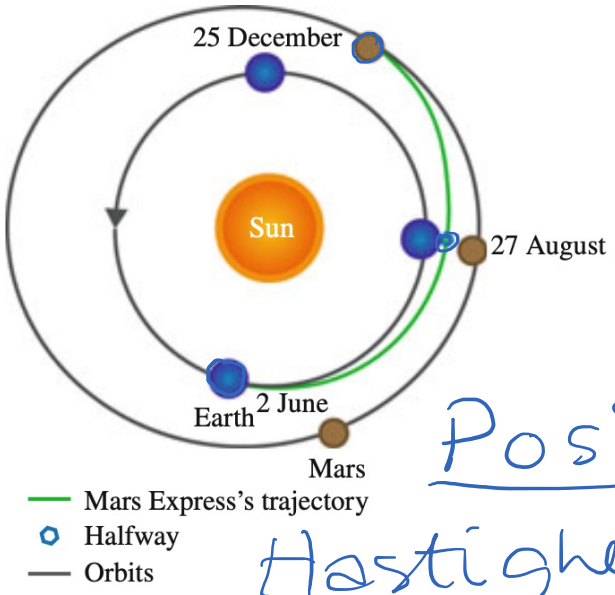
Newtons kvotient: $\frac{df}{dx} \approx \frac{f(x+\Delta x) - f(x)}{\Delta x}$, lite Δx

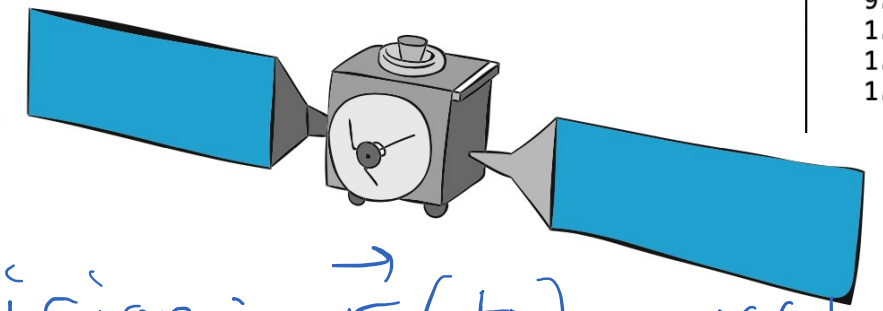
Numerisk implementering: $\frac{\Delta f}{\Delta x} = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$

I praksis kan vi ikke regne ut for alle mulige verdier, vi må gjøre et (klokt) utvalg 🤓

Eksempel: Mars Express

[✏️ lysbilde med skriblerier på iPad ✏️]





Datafil ("low resolution", marsexpresslr.txt):

marsexpresslr.txt

Tid (dager)	x (km)	y (km)	z (km)
4.9532291e-02	-2.8034186e+07	-1.4922374e+08	-3.5527680e+04
3.0066101e+01	5.4163451e+07	-1.4711959e+08	-2.6206946e+05
6.0082670e+01	1.2452776e+08	-1.1167004e+08	-4.2939533e+05
9.0099239e+01	1.7188367e+08	-5.4904690e+07	-5.1677779e+05
1.2011581e+02	1.9351086e+08	1.0496006e+07	-5.2653357e+05
1.5013238e+02	1.9154329e+08	7.4880556e+07	-4.7213981e+05
1.8014894e+02	1.6995059e+08	1.3176488e+08	-3.7005167e+05

30-dagers intervaller

Posisjon: $\vec{r}(t_i) = x(t_i)\hat{i} + y(t_i)\hat{j} + z(t_i)\hat{k}$

Hastighet: $\vec{v}(t_i) \approx \frac{\vec{r}(t_i+\Delta t) - \vec{r}(t_i)}{\Delta t} = \frac{\vec{r}(t_{i+1}) - \vec{r}(t_i)}{\Delta t}$

Akselerasjon: $\vec{a}(t_i) \approx \vec{v}(t_i) - \vec{v}(t_{i-1})$

$= \frac{\vec{r}(t_{i+1}) - \vec{r}(t_i) - [\vec{r}(t_i) - \vec{r}(t_{i-1})]}{(\Delta t)^2} = \frac{\vec{r}(t_{i+1}) - 2\vec{r}(t_i) + \vec{r}(t_{i-1})}{(\Delta t)^2}$

Eksempel: Mars Express: forflytning og endring i $\vec{v}$

```python
import numpy as np
import matplotlib.pyplot as plt

# Lese datafil
t, x, y, z = np.loadtxt('marsexpresslr.txt', usecols = [0, 1, 2, 3], unpack = True)
n = len(t)

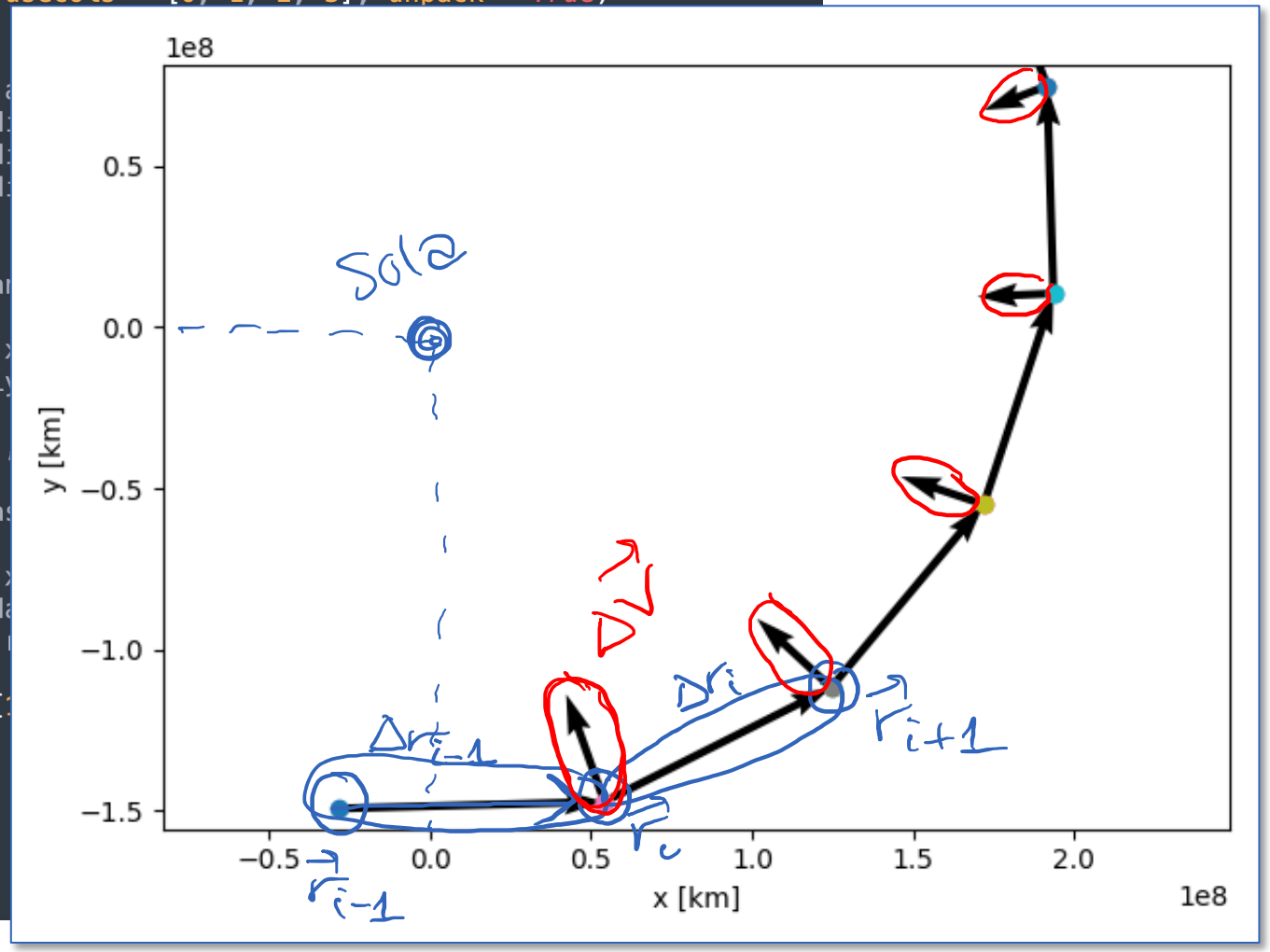
r = np.zeros((n,3),float) # Posisjonsvektor, a
r[:, 0] = x # For alle rader fyller vi x-verdi
r[:, 1] = y # For alle rader fyller vi y-verdi
r[:, 2] = z # For alle rader fyller vi z-verdi

# Vi vil gjerne plotte forflytningen i xy-plan
for i in range(n-1):
    plt.plot(r[i, 0], r[i, 1], 'o') # Plotte x
    dr = r[i+1, :] - r[i, :] # Regne ut forfly
    # Vi plotter kun i xy-planet
    plt.quiver(r[i, 0], r[i, 1], dr[0], dr[1],

# Så vil vi også gjerne regne ut endring i has
for i in range(1,n-1):
    plt.plot(r[i, 0], r[i, 1], 'o') # Plotte x
    # Regne ut endring i hastighet for alle da
    dr2 = (r[i+1, :] - r[i, :]) - (r[i, :] - r
    # Igjen plotter vi kun i xy-planet
    plt.quiver(r[i, 0], r[i, 1], dr2[0], dr2[1

plt.xlabel('x [km]')
plt.ylabel('y [km]')
plt.axis('equal')
plt.tight_layout()
plt.show()
```

Regner ut $\Delta\vec{r}$



Python-script: marsexpress.py

Fellesoppgave: Mars Express: hastighet og akselerasjon

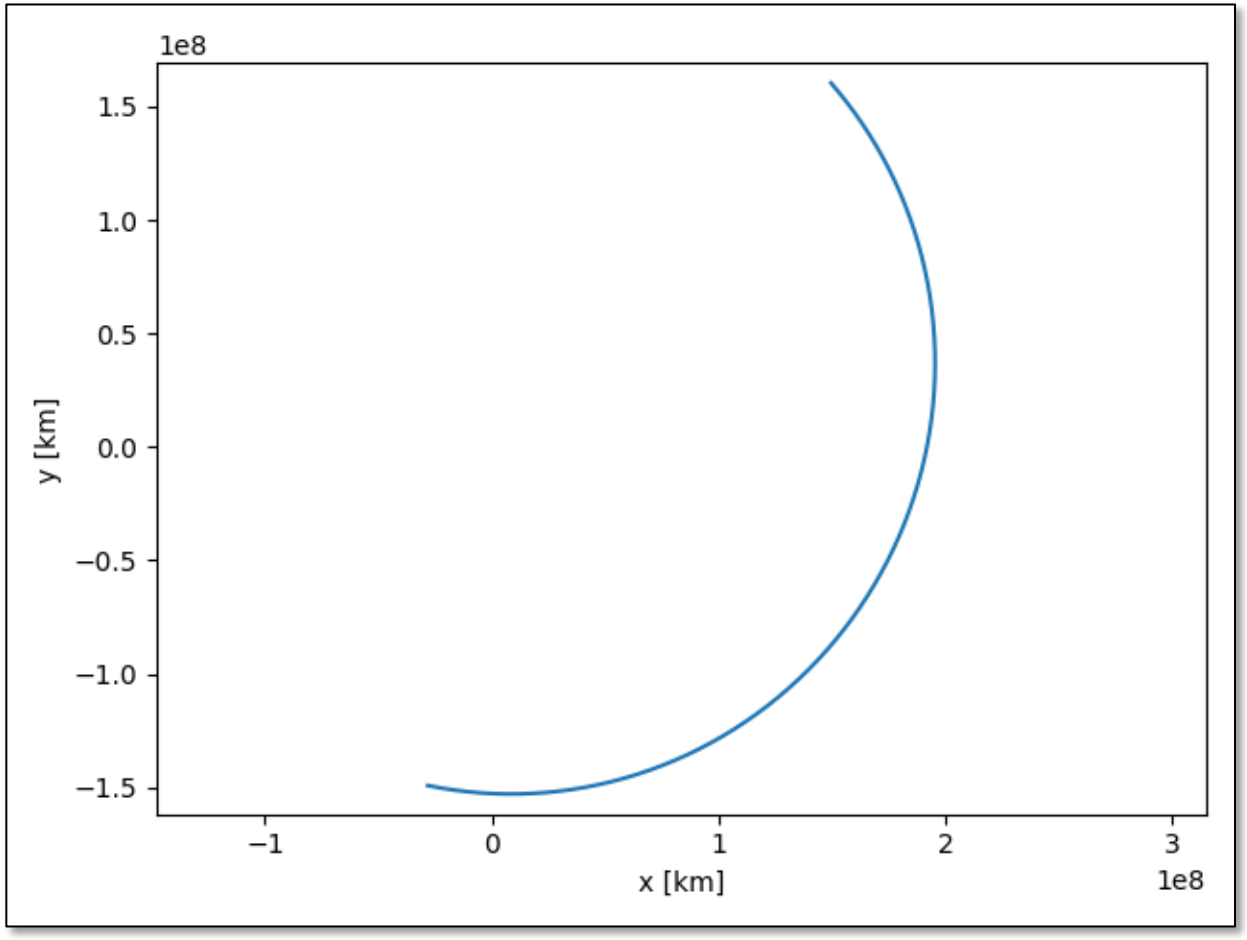
Datafil ("high resolution", marsexpresshr.txt) på semestersiden:

Tid (dager) x (km) y (km) z (km)

marsexpresshr.txt

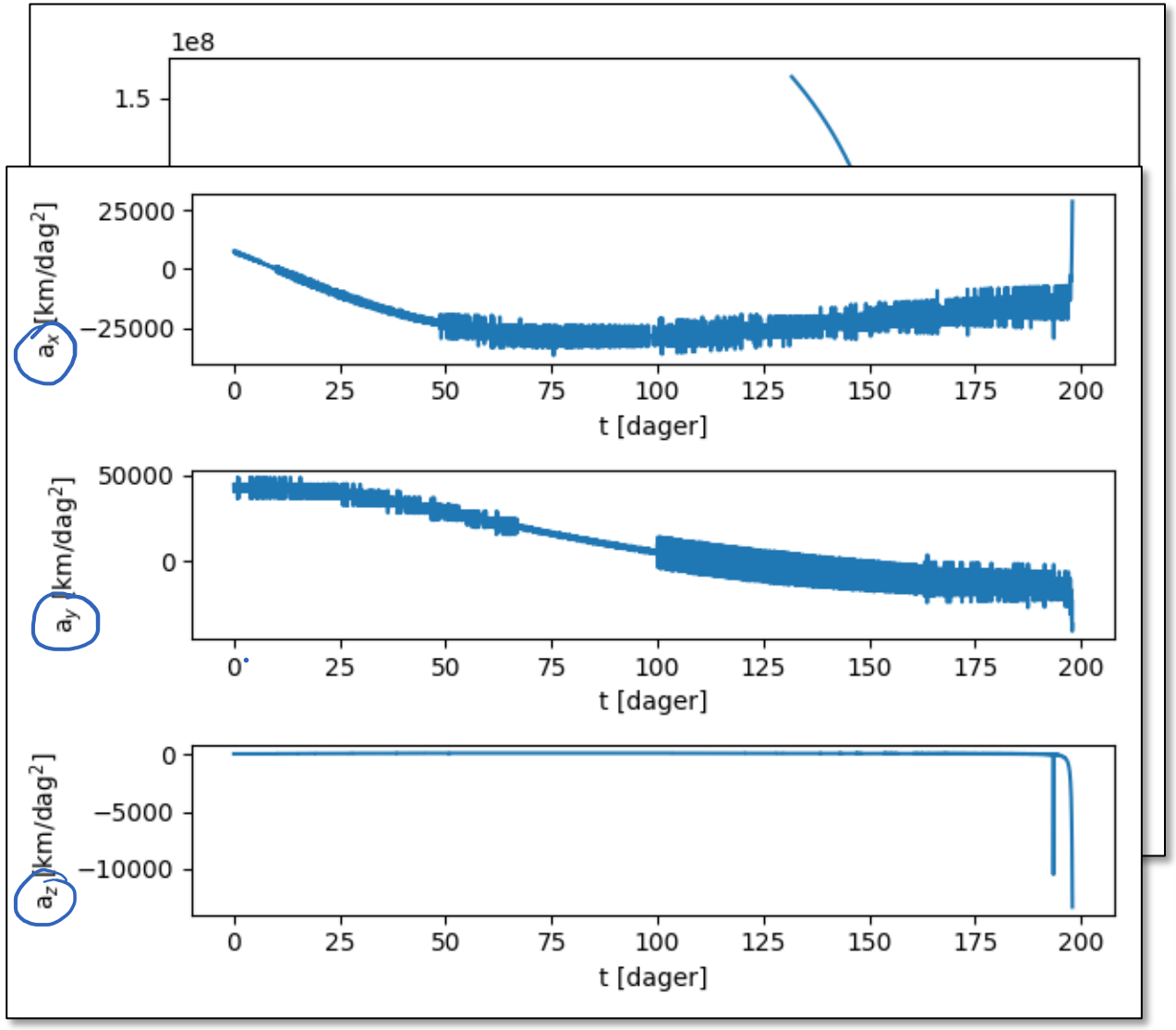
```
  0.0000000e+00  -2.8169824e+07  -1.4919588e+08  -3.5126537e+04
  4.9532291e-02  -2.8034186e+07  -1.4922374e+08  -3.5527680e+04
  9.9064583e-02  -2.7898530e+07  -1.4925149e+08  -3.5928763e+04
  1.4859687e-01  -2.7762857e+07  -1.4927914e+08  -3.6329785e+04
  1.9812917e-01  -2.7627166e+07  -1.4930669e+08  -3.6730749e+04
  2.4766146e-01  -2.7491458e+07  -1.4933413e+08  -3.7131653e+04
  2.9719375e-01  -2.7355732e+07  -1.4936147e+08  -3.7532497e+04
  3.4672604e-01  -2.7219989e+07  -1.4938870e+08  -3.7933282e+04
  3.9625833e-01  -2.7084229e+07  -1.4941583e+08  -3.8334007e+04
  4.4579062e-01  -2.6948452e+07  -1.4944286e+08  -3.8734673e+04
  4.9532292e-01  -2.6812658e+07  -1.4946978e+08  -3.9135279e+04
  5.4485521e-01  -2.6676847e+07  -1.4949660e+08  -3.9535825e+04
  5.9438750e-01  -2.6541019e+07  -1.4952331e+08  -3.9936312e+04
  6.4391979e-01  -2.6405175e+07  -1.4954992e+08  -4.0336740e+04
  6.9345208e-01  -2.6269313e+07  -1.4957643e+08  -4.0737107e+04
```

Fellesoppgave: Mars Express: hastighet og akselerasjon

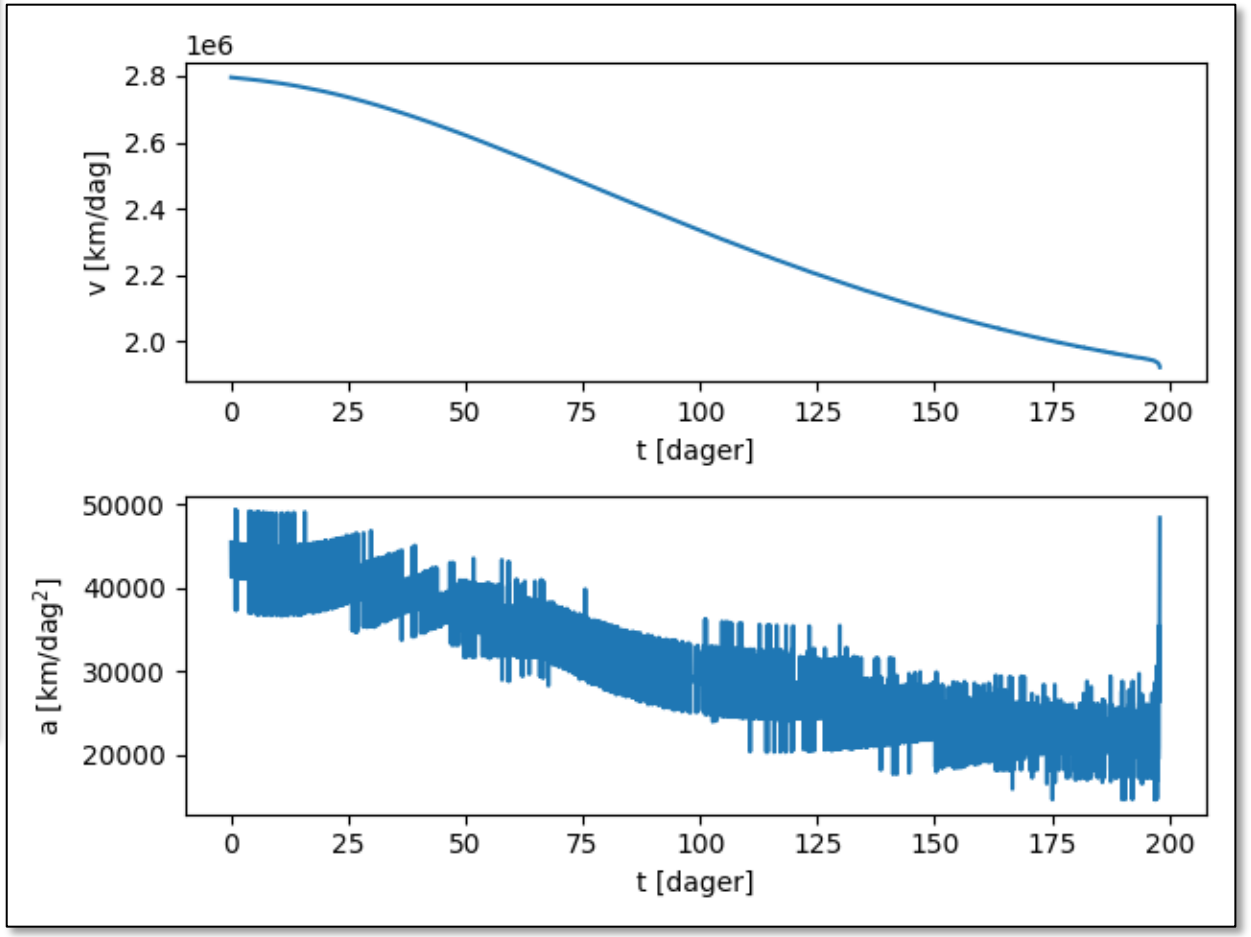


Python-script: marsexpress_v_and_a.py

Fellesoppgave: Mars Express: hastighet og akselerasjon



Hva skjer med akselerasjonen?? 🤯🤔



Python-script: marsexpress_v_and_a.py

Numerisk integrasjon

[✏️ lysbilde med skriblerier på iPad ✏️]

En funksjon $F(x)$ som er den integrerte av en annen funksjon $f(x)$: $f(x) = \frac{dF(x)}{dx}$

$\Rightarrow f(x)dx = dF$ ($dF \to \Delta F$)

For en endelig endring i F, $dF \to \Delta F$:

(i) $\Delta F \approx F(x+\Delta x) - F(x)$

(ii) $\Delta F \approx f(x) \cdot \Delta x$

} Δx liten!

Setter (i)=(ii): $F(x+\Delta x) - F(x) = f(x)\cdot\Delta x$

$\Rightarrow F(x+\Delta x) \approx F(x) + f(x)\Delta x$

Numerisk: $F(x_{i+1}) \doteq F(x_i) + f(x_i)\cdot\Delta x$

“Forward Euler”

Bedre metode: **Euler-Cromer** 👍

Takk for i dag! 😀