



Forelesning VIII

IMAT2150: Høsten 2023

Kent Holing, NTNU (13. september 2023)

LÆRINGSMÅL for dagens forelesning

IVP = Initial Value Problem

- System av første ordens ligninger/Høyere ordens ligninger (forts.)(S: 6.3)
- Runge Kutta metoder/Anvendelser (S: 6.4)

La

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$$

være et såkalt **system** av ordinære differentialligninger (som består av flere skalare ligninger) .

La

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$$

være et såkalt **system** av ordinære differentialligninger (som består av flere skalare ligninger) .

Slike systemer kan alltid omskrives til det **autonome** systemet¹

$$\mathbf{z}' = \mathbf{F}(t, \mathbf{z})$$

$$\mathbf{z} = \begin{bmatrix} t \\ \mathbf{y} \end{bmatrix}, \quad \mathbf{F}(\mathbf{z}) = \begin{bmatrix} 1 \\ \mathbf{f}(t, \mathbf{y}) \end{bmatrix}.$$

¹I et autonomt system er ikke $\mathbf{f}(t, \mathbf{y})$ **eksplisitt** avhengig av t .

Vi kan også alltid skrive om høyere ordens ligninger til første ordens systemer.

Vi kan også alltid skrive om høyere ordens ligninger til første ordens systemer.

Eksempel:

Matematisk pendel¹ (andre ordens differentialligning)

$$m L \theta'' = F = -mg \sin \theta$$

¹Vi definerer de forskjellige størrelsene i pendelligningen presist senere.

Vi kan også alltid skrive om **høyere ordens ligninger** til **første ordens systemer**.

Eksempel:

Matematisk pendel¹ (andre ordens differentialligning)

$$m L \theta'' = F = -mg \sin \theta$$

Sett $\theta_1 = \theta$ og $\theta_2 = \theta'$ og få systemet

$$\theta_1' = \theta_2$$

$$\theta_2' = -g/L \sin \theta_1$$

av første orden med $\theta = [\theta_1, \theta_2]^T \in \mathbb{R}^2$.²

¹Vi definerer de forskjellige størrelsene i pendelligningen presist senere.

²Vi har utelatt initialbetingelsene for $\theta(0)$ og $\theta'(0)$ her.

Fra høyrere ordens ligninger til system av første orden

Generelt tilfelle

A single differential equation of higher order can be converted to a system. Let

$$y^{(n)} = f(t, y, y', y'', \dots, y^{(n-1)})$$

be an n th-order ordinary differential equation. Define new variables

$$y_1 = y$$

$$y_2 = y'$$

$$y_3 = y''$$

$$\vdots$$

$$y_n = y^{(n-1)},$$

and notice that the original differential equation can be written

$$y_n' = f(t, y_1, y_2, \dots, y_n).$$

Fra høyere ordens ligninger til system av første orden

Generelt tilfelle

Taken together, the equations

$$y_1' = y_2$$

$$y_2' = y_3$$

$$y_3' = y_4$$

$$\vdots$$

$$y_{n-1}' = y_n,$$

$$y_n' = f(t, y_1, \dots, y_n)$$

Convert the third-order differential equation

$$y''' = a(y'')^2 - y' + yy'' + \sin t$$

to a system.

Set $y_1 = y$ and define the new variables

$$y_2 = y'$$

$$y_3 = y''.$$

Then, in terms of first derivatives, (6.37) is equivalent to

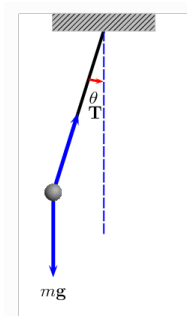
$$y_1' = y_2$$

$$y_2' = y_3$$

$$y_3' = ay_3^2 - y_2 + y_1y_3 + \sin t.$$

$$\ddot{\theta} + \left(\frac{b}{m}\right) \cdot \dot{\theta} + \left(\frac{g}{L}\right) \cdot \sin(\theta) = 0$$

$$\theta(0) = \theta_0 \text{ og } \dot{\theta}(0) = \Omega_0$$



Mathematical model:-

$$\ddot{\theta} + \left(\frac{b}{m}\right) \cdot \dot{\theta} + \left(\frac{g}{L}\right) \cdot \sin(\theta) = 0$$

where,

$\ddot{\theta}$ = Angular Acceleration`

$\dot{\theta}$ = Angular Velocity`

θ =Angular Displacement`

b = Damping Factor`

g = Acceleration due to gravity = $9.81 \text{ (m/s}^2\text{)}$ `

L = Length of pendulum`

m = Mass of bob`

IMAT2150:

Pendel eksempel

Matematisk modell **uten demping** (SAUER, s. 306)

Her er $b = 0$, $\theta(0) = \pi/2$ og $\dot{\theta}(0) = 0$.

Merk her er Eulers metode brukt med tidsstegene $h = 0.01$ og $h = 0.001$.

(Jfr. modelligningen på forrige lysark.)

Den største steglengden gir en unøyaktig løsning som gir økning i energien.

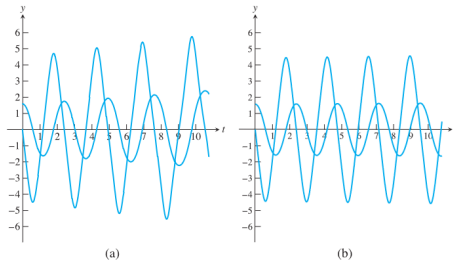


Figure 6.11 Euler Method applied to the pendulum equation (6.40). The curve of smaller amplitude is the angle y_1 in radians; the curve of larger amplitude is the angular velocity y_2 . (a) Step size $h = 0.01$ is too large; energy is growing. (b) Step size $h = 0.001$ shows more accurate trajectories.

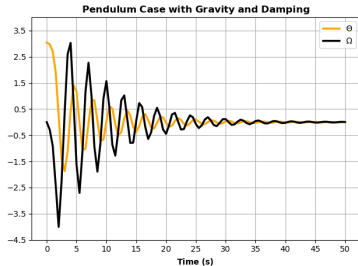
IMAT2150:

Pendel eksempel

Matematisk modell med dempning

Merk `scipy.integrate.odeint` er brukt.

```
1 import numpy as np
2 from scipy.integrate import odeint
3 def pend(y, t, b, c):
4     theta, omega = y
5     dydt = [omega, -b*omega - c*np.sin(theta)]
6     return dydt
7
8 b = 0.25
9 c = 5.0
10 y0 = [np.pi - 0.1, 0.0]
11
12 t = np.linspace(0, 20, 101)
13 t = np.linspace(0, 50, 101)
14 t = np.linspace(0, 10, 101)
15
16 sol = odeint(pend, y0, t, args=(b, c))
17 sol1=sol[:,0]
18 sol2=sol[:,1]
```



Tre himmellegemer med masse m_1 , m_2 og m_3 , og posisjoner $p_1 = (x_1, y_1, z_1)$, $p_2 = (x_2, y_2, z_2)$ og $p_3 = (x_3, y_3, z_3)$ er gitt med innbyrdes avstander

$$\begin{aligned}r_{12} &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \\r_{23} &= \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2} \\r_{13} &= \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2} .\end{aligned}$$

Tyngdekraften mellom legemene er da

$$F_i \text{ p\aa } j = \frac{Gm_i m_j}{r_{ij}^3} (x_i - x_j, y_i - y_j, z_i - z_j) .$$

Akselerasjonen til legemene er

$$\begin{aligned}a_1 &= \frac{F_{2 \rightarrow 1} + F_{3 \rightarrow 1}}{m_1} \\a_2 &= \frac{F_{1 \rightarrow 2} + F_{3 \rightarrow 2}}{m_2} \\a_3 &= \frac{F_{1 \rightarrow 3} + F_{2 \rightarrow 3}}{m_3} .\end{aligned}$$

Første ordens system for 3 himmelleger

$$p'_1 = v_1$$

$$v'_1 = \frac{Gm_2}{r_{12}^3}(p_2 - p_1) + \frac{Gm_3}{r_{13}^3}(p_3 - p_1)$$

$$p'_2 = v_2$$

$$v'_2 = \frac{Gm_1}{r_{12}^3}(p_1 - p_2) + \frac{Gm_3}{r_{23}^3}(p_3 - p_2)$$

$$p'_3 = v_3$$

$$v'_3 = \frac{Gm_1}{r_{12}^3}(p_1 - p_3) + \frac{Gm_2}{r_{23}^3}(p_2 - p_3)$$

"The orbit that put men on the moon" [Arenstorf, NASA (1963)]

Kilde:

<https://www.johndcook.com/blog/2020/02/08/arenstorf-orbit/>

His orbit is a special case of the three body problem where two bodies are orbiting in a plane, i.e. the Earth and the Moon, along with a third body of negligible mass relative to the other bodies, i.e. the satellite

The system of differential equations for the Arenstorf orbit are

$$\begin{aligned}x'' &= x + 2y' - \mu' \frac{x + \mu}{D_1} - \mu \frac{x - \mu'}{D_2} \\y'' &= y - 2x' - \mu' \frac{y}{D_1} - \mu \frac{y}{D_2}\end{aligned}$$

where

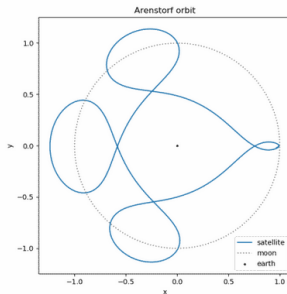
$$\begin{aligned}D_1 &= ((x + \mu)^2 + y^2)^{3/2} \\D_2 &= ((x - \mu')^2 + y^2)^{3/2}\end{aligned}$$

Here the Earth is at the origin and the Moon is initially at (0, 1). The mass of the Moon is $\mu = 0.012277471$ and the mass of the Earth is $\mu' = 1 - \mu$.

The initial conditions are

$$\begin{aligned}x(0) &= 0.994 \\x'(0) &= 0 \\y(0) &= 0 \\y'(0) &= -2.001585106\end{aligned}$$

"The orbit that put men on the moon" [Arenstorf, NASA (1963)]



I found the equations above in [1] which sites the original paper [2]. Note that the paper was written in 1963, seven years before the Apollo missions. Also, before leaving NASA Arenstorf mapped out a rescue orbit. This orbit was later used on Apollo 13.

EKSEMPLER på bruk av himmellegememodeller:

[Simulering av et 2-legemers tilfelle 1](#) (@YouTube)

[Simulering av et 2-legemers tilfelle 2](#) (@YouTube)

[Simulering av et 3-legemers tilfelle](#) (Kilde: Hans Rivertz, NTNU)

At så **intrikate vekselvirkninger** kan oppstå ved bevegelse av system med så få legemer som 2 og 3 er virkelig fascinerende (og kanskje overraskende for mange).

ERRATUM: Øvre grense i Σ i uttrykket for k_r nedenfor skal være $r - 1$ og ikke r .¹⁰

Runge-Kutta-metoder

- Systematisering av steget Euler $\rightarrow$ Heun.
- Vi beregner $k_1, k_2, k_3, \dots, k_s$ verdier inni intervallet. Disse blir beregnet av metoder med lavere orden.
- Til slutt bruker vi en vektet sum av disse k 'ene.
- Genrelt kan metodene skrives som

$$k_r = f \left(t_n + c_r h, y_n + h \sum_{j=1}^r a_{rj} k_j \right), \quad r = 1, 2, 3, \dots, s$$

$$y_{n+1} = y_n + h \sum_{r=1}^s b_r k_r$$

Her kalles s antall *nivåer*.

¹⁰Det ville gitt en implisitt RK metode, som vi ikke studerer her. Feilen er rettet opp nedenfor (lysark 20).

Runge-Kutta-metoder

- Finnes uendelig mange metoder av vilkårlig orden (og finne metoder med høy orden er langt fra trivielt dog) og dette er metoder som OFTE brukes i praksis.
- Disse metodene kan klassifiseres ved
 - matrisen $\underline{A}$ (hvor mye vi skal bruke av hver k -verdi på hvert nivå).
 - vektoren $\underline{c}$ (på hvilket tidspunkt prøver vi å tilnærme tangenten på dette nivået?).
 - vektoren $\underline{b}$ (vektene av hver k -verdi i siste steg).
- Denne informasjonen presenteres ofte i et *RK-tableaux* på formen

$$\begin{array}{c|c} \underline{c} & \underline{A} \\ \hline & \underline{b}^T \end{array}$$

Runge-Kutta-metoder

- Vi skal kun se på eksplisitte metoder. Dette betyr at
 - Det første nivået vil alltid være kun en evaluering i starttidspunktet - det er den eneste informasjonen vi har tilgjengelig.
 - Matrisen $\underline{A}$ må være strengt nedretriangulær (ingen diagonal heller).
- Våre tableaux blir altså på formen

0	0	0	0	...
c_2	a_{21}	0	$\vdots$	$\vdots$
c_3	a_{31}	a_{32}	0	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
	b_1	b_2	b_3	...

Runge-Kutta-metoder

- Altså kan metodene være skrives på formen

$$k_1 = f(t_n, y_n)$$

$$k_r = f\left(t_n + c_r h, y_n + h \sum_{j=1}^{r-1} a_{rj} k_j\right), \quad r = 2, 3, \dots, s$$

$$y_{n+1} = y_n + h \sum_{r=1}^s b_r k_r$$

Runge-Kutta-metoder

- Her skal vi se på en veldig ofte brukt metode, ERK4.
- Som navnet antyder er dette en eksplisitt Runge-Kutta metode av orden 4.
- Tableauxet er gitt ved

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
$\frac{1}{2}$	0	0	1	
1	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

ERRATUM: Ang. kommentarern om Simpsons integrasjonsmetode nedenfor , så skal det være for ikke autonome systemer, dvs. $y'(t) = f(t)$.

Runge-Kutta-metoder

- Hvis vi nå bruker informasjonen kan vi skrive metoden eksplisitt som

$$k_1 = f(t_n, y_n)$$

$$k_2 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right)$$

$$k_4 = f(t_n + h, y_n + hk_3)$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

- Denne metoden sammenfaller med Simpsons metode for autonome systemer.

Oppsummering - eksplisitte metoder

Metode	Funksjonsevalueringer	Globalfeil	Lokalfeil
Euler	1	1	2
Heun	2	2	3
RK4	4	4	5
RKF4	5	4	5
RKF5	6	5	6
RKF	6	4	5

Legg merke til at RKF4 er en dårlig metode for "ren" 4. orden da vi bruker 5 funksjonsevalueringer.

Vi skal bruke RK4 på $y'(t) = y$ med $y(0) = 1$, som har den eksakte løsningen e^t .

¹Litt komplisert å vise, men se lysark 26. (Buthcer sine konvergensbetingelser for (eksplisitte) RK-metoder med orden opp til og med 4.)

IMAT2150:

Bakgrunn for RK4 metoden

Vi skal bruke RK4 på $y'(t) = y$ med $y(0) = 1$, som har den eksakte løsningen e^t .

RK4-tilnærmingen til løsningen med et skritt av lengde h er gitt på første linje nedenfor.

$$1 + h + \frac{h^2}{2} + \frac{h^3}{6} + \frac{h^4}{24}$$
$$1 + h + \frac{h^2}{2} + \frac{h^3}{6} + \frac{h^4}{24} + O[h]^5$$

¹Litt komplisert å vise, men se lysark 26. (Buthcer sine konvergensbetingelser for (eksplisitte) RK-metoder med orden opp til og med 4.)

IMAT2150:

Bakgrunn for RK4 metoden

Vi skal bruke RK4 på $y'(t) = y$ med $y(0) = 1$, som har den eksakte løsningen e^t .

RK4-tilnærmingen til løsningen med et skritt av lengde h er gitt på første linje nedenfor.

$$\begin{array}{l} 1 + h + \frac{h^2}{2} + \frac{h^3}{6} + \frac{h^4}{24} \\ 1 + h + \frac{h^2}{2} + \frac{h^3}{6} + \frac{h^4}{24} + \frac{h^5}{120} + \mathcal{O}(h^5) \end{array}$$

Vi ser at dette sammenfaller med Taylor-rekken for e^t opp til og med $\mathcal{O}(h^4)$ (linje 2).

¹Litt komplisert å vise, men se lysark 26. (Buthcer sine konvergensbetingelser for (eksplisitte) RK-metoder med orden opp til og med 4.)

IMAT2150:

Bakgrunn for RK4 metoden

Vi skal bruke RK4 på $y'(t) = y$ med $y(0) = 1$, som har den eksakte løsningen e^t .

RK4-tilnærmingen til løsningen med et skritt av lengde h er gitt på første linje nedenfor.

$$1 + h + \frac{h^2}{2} + \frac{h^3}{6} + \frac{h^4}{24} + \frac{h^5}{120} + \mathcal{O}(h^6)$$

Vi ser at dette sammenfaller med Taylor-rekken for e^t opp til og med $\mathcal{O}(h^4)$ (linje 2).

Dette er ingen tilfeldighet, men ligger i selve konstruksjonen av RK-metoder:

RK-metoder av orden k er slik at tilnærmingen til løsningen av $y' = f(t, y)$ sammenfaller med Taylor-rekken av den eksakte løsningen opp til og med $\mathcal{O}(h^k)$.

Dette oppnås med bruk av lineære kombinasjoner av fire $f(.,.)$ verdier.¹

¹Litt komplisert å vise, men se lysark 26. (Buthcer sine konvergensbetingelser for (eksplisitte) RK-metoder med orden opp til og med 4.)

Skjemaene nedenfor kalles også **Butcher-tabeller**.¹

Skjema for Eulers metode

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

Skjema for Trapesmetoden

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array}$$

Skjema for Midtpunktsmetoden

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ \hline & 0 & 1 \end{array}$$

Skjema for RK4

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

¹"Butcher tableaux".

Convergence Order for Runge-Kutta

For Runge-Kutta methods, one can prove:

The method is of order p with $p \leq 4$ if all of the conditions in the table on the left up to and including p are fulfilled. (all sums run from 1 to s .)

Remark.

Similar conditions for higher order exist, but get a bit more complicated.

Example. Let's check Heun's method.

p	conditions
1	$\sum_i b_i = 1$
2	$\sum_i b_i c_i = 1/2$
3	$\sum_i b_i c_i^2 = 1/3$ $\sum_{i,j} b_i a_{ij} c_j = 1/6$
4	$\sum_i b_i c_i^3 = 1/4$ $\sum_{i,j} b_i c_i a_{ij} c_j = 1/8$ $\sum_{i,j} b_i a_{ij} c_j^2 = 1/12$ $\sum_{i,j,k} b_i a_{ij} a_{jk} c_k = 1/24$

For Heuns metode ("improved Euler"/eksplisitt trapesmetode) har vi $b_1 = b_2 = \frac{1}{2}$ og $c_1 = 0$ og $c_2 = 1$, som stemmer med at metoden har orden 2. (Betingelsene 1 og 2 er oppfylt, men det er ikke betingelse 3.) **Sjekk selv for RK4!**

```
1 # Python program to implement the RK4 method
2 # Example "dy / dx = (x - y)/2"
3 def dydx(x, y):
4     return ((x - y)/2)
5
6 def RK4(x0, y0, x, h):
7     n = (int)((x - x0)/h)
8     y = y0
9     # Iterate for number of iterations
10    for i in range(1, n + 1):
11        k1 = h * dydx(x0, y)
12        k2 = h * dydx(x0 + 0.5 * h, y + 0.5 * k1)
13        k3 = h * dydx(x0 + 0.5 * h, y + 0.5 * k2)
14        k4 = h * dydx(x0 + h, y + k3)
15        y = y + (1.0 / 6.0)*(k1 + 2 * k2 + 2 * k3 + k4)
16        x0 = x0 + h
17    return y
```

```
1 import numpy as np
2 def RK4OnAutSyst( F , Yinit , h , n ):
3     m = Yinit.size;
4     W = np.zeros([m,n+1])
5     W[:,0] = Yinit
6     for i in range(0,n):
7         s1=F(W[:,i]);
8         s2=F(W[:,i]+(h/2)*s1);
9         s3=F(W[:,i]+(h/2)*s2);
10        s4=F(W[:,i]+h*s3);
11        W[:,i+1]=W[:,i]+h*(s1+2*s2+2*s3+s4)/6
12    return W
```

```
1 import numpy as np
2 from numpy import *
3 def runge_kutta_system(f,y0,t0,t_end,h):
4     num_steps=int((t_end-t0)/h)
5     t=np.linspace(t0,t_end,num_steps+1)
6     y=np.zeros((len(t),len(y0)))
7     y[0]=y0
8     for i in range(num_steps):
9         k1=h*np.array(f(t[i],y[i]))
10        k2=h*np.array(f(t[i]+0.5*h,y[i]+0.5*k1))
11        k3=h*np.array(f(t[i]+0.5*h,y[i]+0.5*k2))
12        k4=h*np.array(f(t[i]+h,y[i]+k3))
13        y[i+1]=y[i]+(k1+2*k2+2*k3+k4)/6.0
14    return t.tolist(),y.tolist()
15
16 def system_of_odes(t, y):
17     dy1dt,dy2dt=y[1],-9.82*np.sin(y[0])
18     return [dy1dt,dy2dt]
19
20 t0,t_end,h,y0= 0.0,0.5,0.5,[1.0, 0.0]
21 t, y=runge_kutta_system(system_of_odes,y0,t0,t_end,h)
22 print(t)
23 print(y)
```

Runge-Kutta-metoder

- Finnes uendelig mange metoder av vilkårlig orden (og finne metoder med høy orden er langt fra trivielt dog) og dette er metoder som OFTE brukes i praksis.
- Disse metodene kan klassifiseres ved
 - matrisen $\underline{A}$ (hvor mye vi skal bruke av hver k -verdi på hvert nivå).
 - vektoren $\underline{c}$ (på hvilket tidspunkt prøver vi å tilnærme tangenten på dette nivået?).
 - vektoren $\underline{b}$ (vektene av hver k -verdi i siste steg).
- Denne informasjonen presenteres ofte i et *RK-tableaux* på formen

$$\begin{array}{c|c} \underline{c} & \underline{A} \\ \hline & \underline{b}^T \end{array}$$

Runge-Kutta-metoder

- Vi skal kun se på eksplisitte metoder. Dette betyr at
 - Det første nivået vil alltid være kun en evaluering i starttidspunktet - det er den eneste informasjonen vi har tilgjengelig.
 - Matrisen $\underline{A}$ må være strengt nedretriangulær (ingen diagonal heller).
- Våre tableaux blir altså på formen

0	0	0	0	...
c_2	a_{21}	0	$\vdots$	$\vdots$
c_3	a_{31}	a_{32}	0	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
	b_1	b_2	b_3	...

Runge-Kutta-metoder

- Altså kan metodene være skrives på formen

$$k_1 = f(t_n, y_n)$$

$$k_r = f\left(t_n + c_r h, y_n + h \sum_{j=1}^{r-1} a_{rj} k_j\right), \quad r = 2, 3, \dots, s$$

$$y_{n+1} = y_n + h \sum_{r=1}^s b_r k_r$$

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
$\frac{1}{2}$	0	0	1	
1	0	0	1	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

$$k_1 = f(t_n, y_n)$$

$$k_2 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right)$$

$$k_4 = f(t_n + h, y_n + hk_3)$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

ERRATUM: Sløyf siste uttrykk på neste siste linje!

Eksempel

$1/2$

$$\theta'' + g \sin \theta = 0$$

$$\theta(0) = \theta_0 = 1 \text{ (radianer)}$$

$$\theta'(0) = 0$$

$$Y = \begin{pmatrix} u \\ v \end{pmatrix}$$

$$g = 9.82$$

$$Y' = f(Y) = \begin{pmatrix} v \\ -g \sin u \end{pmatrix}$$

Bruk RK4 til å beregne en
tiln. til $\theta(1/2)$ med et
skritt. ~~og~~ ($h = 1/2$)

$$k_1 = f(Y_0) = f_1\left(\begin{pmatrix} u_0 \\ v_0 \end{pmatrix}\right) = f_1\left(\begin{pmatrix} v_0 \\ u_0 \end{pmatrix}\right)$$

$$k_1 = \begin{pmatrix} 0 \\ -g \sin 1 \end{pmatrix}$$

Husk at $f(Y) = f((u, v)^T) = (v, -g \sin u)^T$.

$$k_2 = f\left(Y_0 + \frac{h}{2} k_1\right) = f\left(\begin{pmatrix} u_0 + \frac{h}{2} \cdot 0 \\ v_0 - \frac{h}{2} g \sin u \end{pmatrix}\right)$$

$$k_2 = f\left(\begin{pmatrix} u_0 \\ -\frac{h}{2} g \sin u \end{pmatrix}\right)$$

$$k_2 = \begin{pmatrix} -\frac{h}{2} g \sin u \\ -g \sin u \end{pmatrix}$$

$$k_1 = (0, -8.263)^T$$

$$k_2 = (-2.066, -8.263)^T$$

Vi får på samme vis $2(2)$

$$k_3 = (-2.066, -4.659)^T$$

$$k_4 = (-2.285, 0.324)^T$$

(Vis det selv!)

Da gir RK4 med skritt lengde $h = 1/2$

$$Y_1 = \begin{pmatrix} u_1 \\ v_1 \end{pmatrix} = \begin{pmatrix} 0.121 \\ -2.80 \end{pmatrix}$$

$$\underline{\theta(1/2)} \doteq u_1 = \underline{0.121}$$

$$Y_1 = \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4)$$

Bestem hvor lang tid det tar for pendelen å komme fra utgangspunktet ($\theta = \theta_0$) til $\theta = 0$ for første gang.

15	0.468750000	0.19620558488032824	-2.9413664191990190
16	0.500000000	0.10349855836259253	-2.9872001598291722
17	0.531250000	9.8022760316733709E-003	-3.0045829551580843
18	0.562500000	-8.3987861770073807E-002	-2.9931977675898267
18	0.562500000	-8.3987861770073807E-002	-2.9931977675898267
0.53444042348972409			
19	0.593750000	-0.17697474173958039	-2.9532525433433605
20	0.625000000	-0.26827583237754354	-2.8854680078409118
21	0.656250000	-0.35703862987856144	-2.7910360509347347
22	0.687500000	-0.44245440003379544	-2.6715532782964093

Ovenfor er det utdrag av resultat på bruk av RK4 på pendelproblemet med skrittlengde $h = 1/32$.

La θ skifte fortegn fra positiv til negativ for første gang i tabellen ved $t = \tilde{t} = 0.5625$.

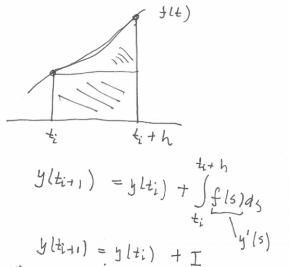
Da er $\tilde{t} - u(\tilde{t})/\nu(\tilde{t})$ en mye bedre tilnærming til det det spørres om. (Hvorfor?)

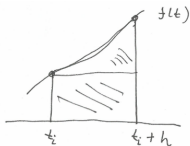
Svar: 0.5344 sekunder.

IMAT2150:

Spesialtilfelle av en-skrittsmetoder

$f(t, y) = f(t)$ er kun avhengig av t





$$y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_i+h} \underbrace{f(s)}_{y'(s)} ds$$

$$y(t_{i+1}) = y(t_i) + I$$

- 1) tilnærmer I med rektangelmetoden (areal $\Rightarrow$)
gir Eulers metode
- 2) tilnærmer I med trapesmetoden får vi forbedret Euler (eksplicit trapes metode) (areal $\Rightarrow$ + //)

IMAT2150:

Spesialtilfelle av en-skrittsmetoder

$f(t, y) = f(t)$ er kun avhengig av t

Spesialtilfelle av

RK4

Anta at $f(t, y)$ ikke
avhenger av y .

$$): \underline{y'(t) = f(t)}$$

Se oppgave 6, 4, 7 i SAUER.

Da vil $k_2 = k_3$ i RK4 og

RK4 er ekvivalent med

Simpsons metode for å tilnærme

$$\int_{t_i}^{t_i+h} f(t) dt = \frac{h}{6} \left(f(t_i) + 4f\left(t_i + \frac{h}{2}\right) + f(t_i+h) \right) \\ = \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4)$$

IMAT2150:

Spesialtilfelle av en-skrittsmetoder

$f(t, y) = f(t)$ er kun avhengig av t

Spesialtilfelle av

RK4

Anta at $f(t, y)$ ikke
avhenger av y .

1: $y'(t) = f(t)$

Se oppgave 6, 4, 7 i SAUER.

Da vil $k_2 = k_3$ i RK4 og

RK4 er ekvivalent med

Simpsons metode for 2 tilnær

$$\begin{aligned} \int_{t_i}^{t_i+h} f(t) dt &= \frac{h}{6} \left(f(t_i) + 4f\left(t_i + \frac{h}{2}\right) + f(t_i+h) \right) \\ &= \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4) \end{aligned}$$

$$k_1 = f(t_i)$$

$$k_2 = f\left(t_i + \frac{h}{2}\right)$$

$$k_3 = f\left(t_i + \frac{h}{2}\right)$$

$$k_4 = f(t_i+h)$$

$$(k_3 = k_2)$$

IVP er løst eksakt på side 291 i Sauer i eksempel 6.6.

Apply Runge–Kutta of order four to the initial value problem

$$\begin{cases} y' = ty + t^3 \\ y(0) = 1 \end{cases} \quad (6.51)$$

Computing the global truncation error at $t = 1$ for a variety of step sizes gives the following table:

steps n	step size h	error at $t = 1$
5	0.20000	2.3788×10^{-5}
10	0.10000	1.4655×10^{-6}
20	0.05000	9.0354×10^{-8}
40	0.02500	5.5983×10^{-9}
80	0.01250	3.4820×10^{-10}
160	0.00625	2.1710×10^{-11}
320	0.00312	1.3491×10^{-12}
640	0.00156	7.2609×10^{-14}

Illustrasjon av konvergens av RK4 (SAUER 6.51)

Figuren viser tydelig at RK4 er en metode av orden 4.

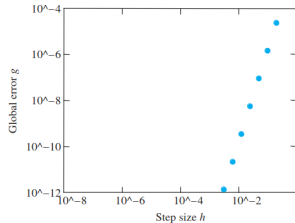
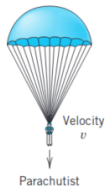


Figure 6.15 Error as a function of step size for Runge-Kutta of order 4. The difference between the approximate solution of (6.5) and the correct solution at $t = 1$ has slope 4 on a log-log plot, so is proportional to h^4 , for small h .

Kent Holing: Modellering av fallskjermhopping

Modellering av fallskjermhopping

Kent Holing (2023)



Vi skal diskutere modellering av fallskjermhopping ved å løse problemet nedenfor I

IMAT2150:

Modellering av fallskjermhopping ved hjelp av en RK4 type metode

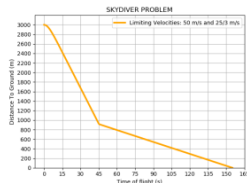
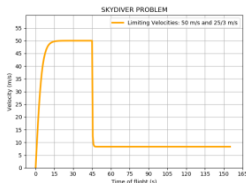
Svar på spørsmålene nedenfor: a) ca. 45s, b) ca. 925m og c) ca. 155s.

Problem

En fallskjerm er designet slik at den åpner seg så snart fallhastigheten ikke endrer seg mer, og er tilpasset hopperen slik at landingshastigheten innstiller seg på 30 km/t.

En mann hopper fra 3000 meters høyde og skjermen utløses etter en tid i fritt fall når hastigheten hans har stabilisert seg på 180 km/t.

- a) Hvor lang tid tar det før skjermen løser seg ut?
- b) Hvor langt over bakken er mannen når landingshastigheten slår inn?
- c) Hvor lang tid tar det (totalt) før mannen er på bakken?

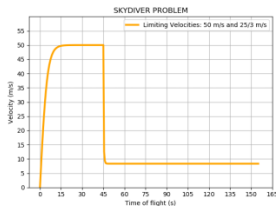


Modellering av fallskjermhopping ved hjelp av en RK4 type metode

Numerisk løsningsmetode

Vi har her brukt startverdi problemet (1) $v' = g - cv^2$ og $v(0) = 0$ for $g = 9,81 \text{ m/s}^2$ med $z(0)$ og $v(0)$ gitt². Ligningen skrives om på standard måte som et system av to koblete ordinære differentialligninger, som vi løser med en adaptiv RK metode³.

Vi ser av figuren til høyre at problemet gir opphav til en veldig rask endring i hopperens hastighet rett etter at fallskjermen åpner seg før den flater ut før landing⁴. Å fange effektivt opp store og raske endringer (som i andre tilfeller også kan være ekte diskontinuiteter) i løsninger er ekstra krevende for numeriske løsningsmetoder. Vi har valgt den såkalte Cash-Karp RK metoden som er spesielt implementert for å løse slike problemer for ordinære differentialligninger⁵.



Modellering av luftmotstanden skjer ved å velge passende konstanter $c = g/v^2$ i (1) som oppnår null akselerasjon i grensetilfellene 1) og 2):

- 1) Før skjermen slår seg ut bruker vi $c = c_1 = g/50^2 \text{ m}^{-1}$.
- 2) Etter at skjermen har slått seg ut bruker vi $c = c_2 = g/(25/3)^2 \text{ m}^{-1}$.

IMAT2150:

Cash-Karp RK metode

(Brukt i modelleringen av fallskjermhopping ovenfor)

IKKE PENSUM!

In [numerical analysis](#), the **Cash-Karp method** is a method for solving [ordinary differential equations](#) (ODEs). It was proposed by Professor Jeff R. Cash ^[1] from [Imperial College London](#) and Alan H. Karp from [IBM Scientific Center](#). The method is a member of the [Runge-Kutta](#) family of ODE solvers. More specifically, it uses six function evaluations to calculate fourth- and fifth-order accurate solutions. The difference between these solutions is then taken to be the error of the (fourth order) solution. This error estimate is very convenient for [adaptive stepsize](#) integration algorithms. Other similar integration methods are [Fehlberg](#) (RKF) and [Dormand-Prince](#) (RKDP).

The [Butcher tableau](#) is:

0						
1/5	1/5					
3/10	3/40	9/40				
3/5	3/10	-9/10	6/5			
1	-11/54	5/2	-70/27	35/27		
7/8	1631/55296	175/512	575/13824	44275/110592	253/4096	
	37/378	0	250/621	125/594	0	512/1771
	2825/27648	0	18575/48384	13525/55296	277/14336	1/4

The first row of b coefficients gives the fifth-order accurate solution, and the second row gives the fourth-order solution.

Dette gir en enkel og effektiv måte å **estimere den lokale feilen** på underveis i beregningen.

$$w_{i+1} = w_i + h \left(1 - \frac{1}{2\alpha} \right) f(t_i, w_i) + \frac{h}{2\alpha} f(t_i + \alpha h, w_i + \alpha h f(t_i, w_i))$$

$$w_{i+1} = w_i + h \left(1 - \frac{1}{2\alpha} \right) f(t_i, w_i) + \frac{h}{2\alpha} f(t_i + \alpha h, w_i + \alpha h f(t_i, w_i))$$

RK metodeklassen ovenfor (for $\alpha \neq 0$) er (eksplisitt) trapesmetode for $\alpha = 1$ og midtpunktmetoden for $\alpha = 1/2$.

$$w_{i+1} = w_i + h \left(1 - \frac{1}{2\alpha} \right) f(t_i, w_i) + \frac{h}{2\alpha} f(t_i + \alpha h, w_i + \alpha h f(t_i, w_i))$$

RK metodeklassen ovenfor (for $\alpha \neq 0$) er (eksplisitt) trapesmetode for $\alpha = 1$ og midtpunktmetoden for $\alpha = 1/2$.

Skjema for Trapesmetoden

0	0	0
1	1	0
		1/2
		1/2

Skjema for Midtpunktsmetoden

0	0	0
1/2	1/2	0
		0
		1

ØLS.

$$\text{Gitt } y'' = \pm \sin y$$

$$y(0) = 0, \quad y(1) = 1$$

For å bruke RK4 må vi jobbe med et IVP.

Løsningsstrategi:

Løs IVP med 1) $y'(0) = 0,8$
og 2) $y'(0) = 1,0$

Vi gjør det med RK4 med $h=1$.

ERRATUM: Vi skal ha $u = t$, ikke $u = 1$.

$$Y = (u, v, w)^T$$

$$u = 1, \quad v = y, \quad w = y'$$

$$Y' = f(Y) = (1, w, u \sin v)^T$$

$$Y_0 = (u_0, v_0, w_0)^T = (0, 0, z)^T$$

$$z = 0.8 \text{ og } z = 1.0$$

Husk at $f((u, v, w)^T) = (1, w, u \sin v)^T$.

$$\underline{h} z = 0.8:$$

$$\vec{K}_1 = f(y_0) = (1, 0.8, 0)^T$$

$$\vec{K}_2 = f(y_0 + \frac{h}{2} \vec{K}_1) = (1, 0.8, 0.194709)^T$$

$$\vec{K}_3 = f(y_0 + \frac{h}{2} \vec{K}_2) = (1, 0.897355, 0.194709)^T$$

$$\vec{K}_4 = f(y_0 + \frac{h}{2} \vec{K}_3) = (1, 0.994709, 0.78168)^T$$

$$y(1) \approx X_1 = y_0 + \frac{h}{6} (\vec{K}_1 + 2\vec{K}_2 + 2\vec{K}_3 + \vec{K}_4)$$

$$\underline{(2)} = 0.865$$

For $z = 1.0$:

RK4 gir da $y(1) = 1.08$

Hvordan velge en ny z
slik at vi forbedrer $y(1)$ tiln.?

La ny $z = z^*$

$$\frac{z^* - 1.0}{1 - 1.08} = \frac{0.8 - 1.0}{0.865 - 1.08}$$

$$z^* = 0.924$$

Kjør så RK4 med $y'(0) = z^*$

Det gir $y(1) = \underline{\underline{1.00044}}$