
Matematikk 1000

Øvingsoppgaver i numerikk – leksjon 11

Eulers metode

Løsningsforslag

Oppgave 1 – Samanlikning med analytisk løsning

$$y' = \frac{3}{2}\sqrt{x}y, \quad y(0) = 1 \quad .$$

- a) Kandidat til løsning: $y = e^{x^{3/2}}$.
Vi deriverer denne:

$$y' = e^{x^{3/2}} \cdot (x^{3/2})' = \frac{3}{2}x^{1/2}e^{x^{3/2}} = \frac{3}{2}x^{1/2}y \quad .$$

Av første og siste likhet ser vi at differensiallikninga er oppfylt. Så sjekker vi initialkravet:

$$y(0) = e^{0^{3/2}} = 1 \quad .$$

Dette stemmer også.

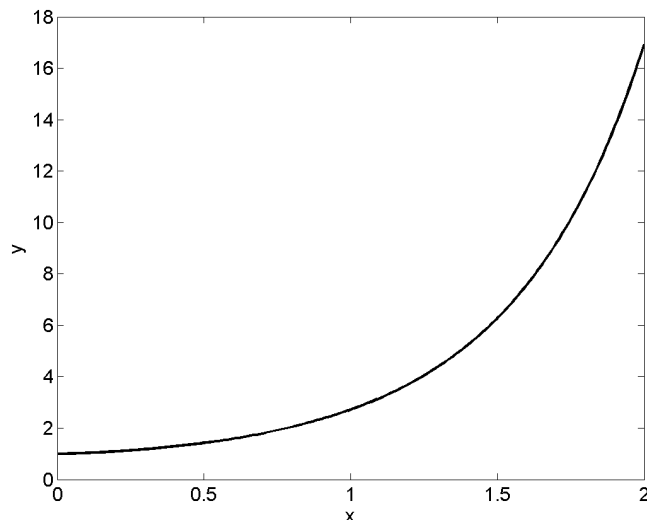
Plottet kan lages slik i MATALAB:

```
>>x=0:1e-2:2;  
>>y=exp(x.^(3/2));  
>>plot(x,y,'linewidth',2)  
>>xlabel('x'); ylabel('y')
```

Resultatet er vist i figur 1.

- b) Vi veit at $P_1(x) = y(a) + y'(a)(x - a)$. Differensiallikninga forteller oss at

$$y'(a) = \frac{3}{2}\sqrt{a}y(a) \quad ,$$



Figur 1: Svaret i deloppgave 1a). Figuren viser et plott av funksjonen $y(x) = \exp(x^{3/2})$.

slik at

$$P_1(x) = y(a) + \frac{3}{2}\sqrt{a}y(a)(x-a) \quad .$$

Om vi set inn $a = x_n$ og $x = x_n + h$, får vi

$$P_1(x_n + h) = y(x_n) + \frac{3}{2}\sqrt{x_n}y(x_n)(x_n + h - x_n) = y(x_n) + \frac{3}{2}\sqrt{x_n}y(x_n)h \quad .$$

c)

$$\begin{aligned} y_0 &= 1 \\ y_{n+1} &= y_n + \frac{3}{2}\sqrt{x_n}y_n h, \quad x_n = nh \quad . \end{aligned} \quad (1)$$

Dersom $y_n = y(x_n)$, vil høyre side i uttrykket over være ei lineær tilnærming – et første ordens Taylor-polynom – til $y(x_n + h)$. Vi har tidligere sett at dette er ei god tilnærming så lenge h er *liten*. Dermed har vi, siden $y_0 = y(x_0)$, at $y_1 \approx y(x_1)$, som igjen gir at $y_2 \approx y(x_2)$ og så vidare. Merk at vi for hver slik iterasjon introduserer ein feil. Denne blir større og større for hvert ledd.

d) Skriptet kan for eksempel se slik ut:

```

1  % Gir start og slutt for x
2  x0=0;
3  xSlutt=2;
4  % Initialverdi for y:
5  y0=1;
6
```

```

7  % Gir antall steg
8  N=input('Gi antall steg: ');
9  % Steglengda
10 h=(xSlutt-x0)/N;
11
12 % Initierer vektor med x-verdier
13 xVektor=x0:h:xSlutt;
14
15 yVektor(1)=y0;
16 for n=1:N
17 % Tilornder x- og y-verdiar:
18   xn=xVektor(n);
19   yn=yVektor(n);
20 % Finner neste y-verdi ved Eulers metode:
21   yVektor(n+1)=yn+3/2*sqrt(xn)*yn*h;
22 end
23 % Skriver x- og y-verktorane til skjerm
24 xVektor
25 yVektor

```

Kommentarene i skriptet burde vere ganske forklarande. Vi har gitt skriptet namnet `EulersMetode.m`. Når vi køyrer det i MATLAB og gir N -verdien 10, får vi:

```

>> EulersMetode
Gi antall steg: 10

```

```

xVektor =

```

```

Columns 1 through 8

```

```

      0      0.2000      0.4000      0.6000      0.8000      1.0000      1.2000      1.4000

```

```

Columns 9 through 11

```

```

      1.6000      1.8000      2.0000

```

```

yVektor =

```

```

Columns 1 through 8

```

```

      1.0000      1.0000      1.1342      1.3494      1.6629      2.1091      2.7419      3.6429

```

```

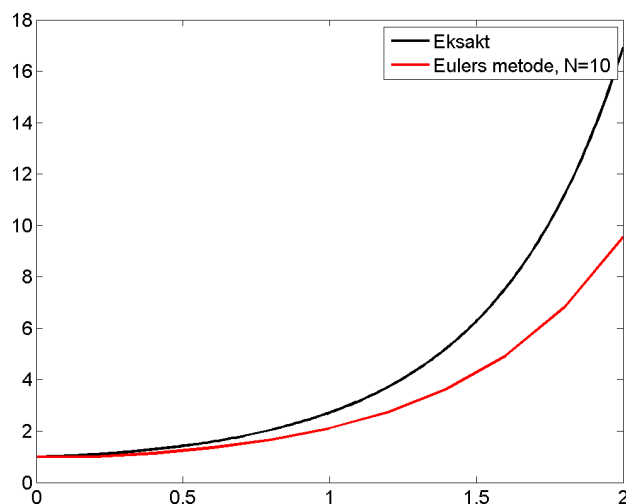
Columns 9 through 11

```

```

      4.9360      6.8091      9.5498

```



Figur 2: Svaret i deloppgave 1e). Figuren viser den analytiske løsningen av initialverdiproblemet sammen med ei numerisk løsning basert på $N = 10$ steg.

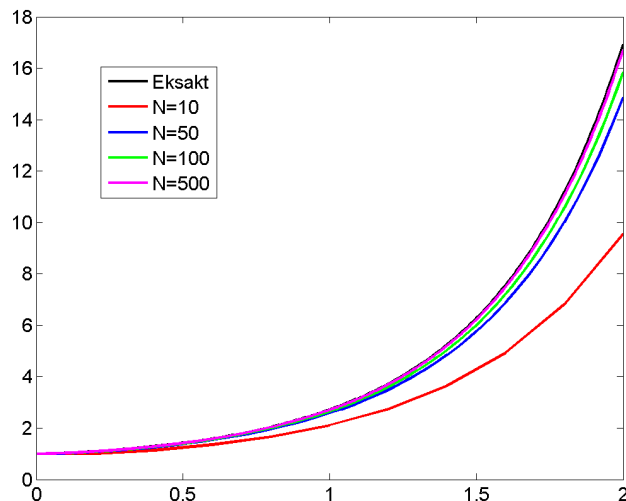
Denne måten å presentere svaret på er jo ikke så informativ. Det er nok bedre å gjøre det grafisk.

- e) Om vi legger følgende til i skriptet over, får vi de plottene vi ønsker:

```
% Vi plottar resultatet sammen med den analytiske løsningen
% Eksakt, analytisk løsning:
xEks=x0:1e-2:xSlutt;
yEks=exp(xEks.^(3/2));
% Plotter eksakt løsning
plot(xEks,yEks,'k-', 'linewidth',2)
hold on
% Plotter numerisk løsning
plot(xVektor,yVektor,'r-', 'linewidth',2)
hold off
```

Vi kjører skriptet igjen og setter $N = 10$. Figuren vi da får, er vist i figur 2. Som vi ser, er ikke de to grafene særlig like. Da er vel h for stor, da – eller, sagt på en annen måte: N er for liten.

- f) Når vi gjentar det vi gjorde i forrige del-oppgave med stadig høyere N -verdier, får vi grafene vist i figur 3. Den numeriske løsningen vår ser ut til å begynne å nærme seg den eksakte. Vi ser også at den numeriske løsningen konsekvent ligger *under* den eksakte, og avviket blir større og større når x øker. Først når $N = 500$ begynner det å bli vanskeleg å sjå skilnad på den numeriske og den eksakte løsningen.



Figur 3: Svaret i deloppgave 1f). Vi ser numeriske løysingar for ulike steglengder.

- g) Vi skal vise at den generelle formelen $y_{n+1} = y_n + F(x_{n+1}, y_{n+1})h$ i vårt tilfelle kan skrivest slik:

$$y_{n+1} = \frac{y_n}{1 - \frac{3}{2}\sqrt{n+1}h^{3/2}} \quad .$$

Vi setter inn at $F(x, y) = \frac{3}{2}\sqrt{x}y$ og at $x_n = nh$:

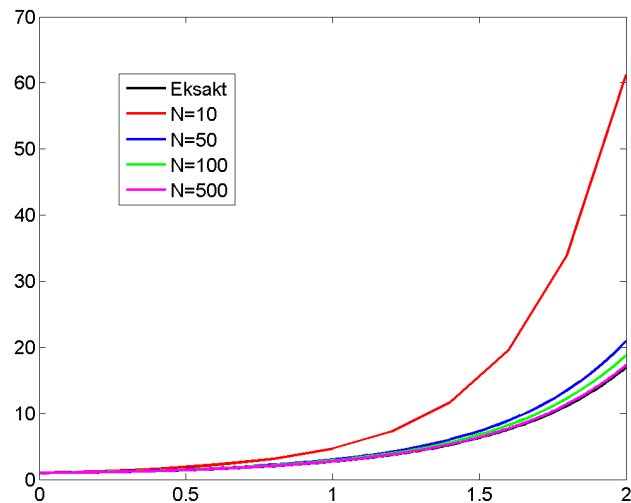
$$\begin{aligned} y_{n+1} &= y_n + \frac{3}{2}\sqrt{x_{n+1}}y_{n+1}h = y_n + \frac{3}{2}\sqrt{n+1}h y_{n+1}h = \\ &= y_n + \frac{3}{2}\sqrt{n+1}y_{n+1}h^{3/2} \\ y_{n+1} \left(1 - \frac{3}{2}\sqrt{n+1}h^{3/2}\right) &= y_n \\ y_{n+1} &= \frac{y_n}{1 - \frac{3}{2}\sqrt{n+1}h^{3/2}} \quad . \end{aligned}$$

- h) Vi endrer linje 21 i skriptet frå deloppgåve d) til:

$$\text{yVektor}(n+1) = y_n / (1 - 3/2 * \text{sqrt}(n+1) * h^{(3/2)});$$

Med denne endringa gjentar vi det vi gjorde i deloppgave f). Resultatet er vist i figur 4. Som vi ser, gir slett ikke denne metoden noko bedre resultat enn framover-metoden. Til forskjell frå framover-metoden, gir bakover-metoden her ein graf som konsekvent ligger *over* den eksakte løysinga. Som med framover-metoden, blir feilen konsekvent større med økende x .

- i) De to eksemplene oppgava siktar til, er midpunktsformelen for numerisk derivasjon og at midpunktssleksjoner gir langt bedre integral-estimer



Figur 4: Svaret på deloppgåve 1h). Vi ser her numeriske løsninger basert på Eulers bakover-metode for ulike steglengder sammen med den eksakte løsninga.

enn høyre- og venstre-seleksjonar for Riemann-summar basert på regulære partisjoner.

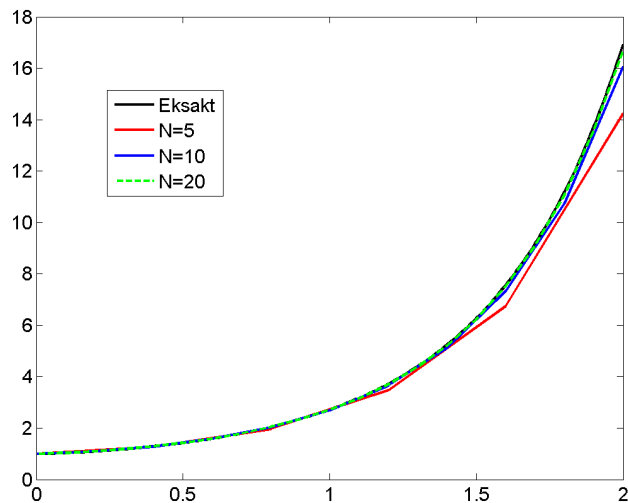
Når vi skal implementere denne metoden, er det bare det som står i `for`-løkka som trenger å endras i forhold til “framover-skriptet”. Denne kan implementeres slik:

```
for n=1:N
% Tilordner x- og y-verdier
    xn=xVektor(n);
    yn=yVektor(n);
    xnHatt=xn+h/2;
    ynHatt=yn+3/2*sqrt(xn)*yn*h/2;
% Finner neste y-verdi ved Eulers midtpunk-metode
    yVektor(n+1)=yn+3/2*sqrt(xnHatt)*ynHatt*h;
end
```

Når vi lager tilsvarende plott som i figur 3 og -4, får vi figur 5 som resultat. Vi ser her at estimatet nærmer seg den eksakte løsninga langt raskere enn for framover- og bakovermetodene; vi får ei numerisk løsning som nesten ligger oppå den eksakte med bare $N = 20$!

Oppgave 2 – Utan “fasit”

I disse oppgavene har vi valgt å gjøre skriptet som implementerer Eulers metode litt mer generiske i den forstand at vi lager ei egen funksjonsfil for $F(x, y)$. På



Figur 5: Svaret på deloppgave 1j). Figuren viser tydeleg at Eulers midtpunktsmetode har helt andre konvergensegenskaper enn framover- og bakover-metodene. (Legg merke til at fargene tilsvarer andre N -verdier enn i figur 3 og -4.)

den måten er det ganske smidig å bruke det same skriptet for å løse forskjellige differensiallikningar. I tillegg tar vi med ei linje som fikserer størrelsen på `yVektor` (linje 17). Skriptet vi bruker for framover-metoden ser slik ut (det er svært likt det vi brukte i oppgave 1):

```

1  % Tar bort gamle variable
2  clear all
3
4  % Gir start og slutt for x
5  x0=0;
6  xSlutt=3;
7  % Initialverdi for y:
8  y0=1;
9
10 % Gir antall steg
11 N=input('Gi antall steg: ');
12 % Steglengda
13 h=(xSlutt-x0)/N;
14
15 % Initierer vektor med x-verdier og y-verdier
16 xVektor=x0:h:xSlutt;
17 y=zeros(1,(N+1))
18 yVektor(1)=y0;      % Initialkravet
19
20 for n=1:N

```

```

21 % Tilornder x- og y-verdiar
22   xn=xVektor(n);
23   yn=yVektor(n);
24   yDeriv=DerivFunk(xn,yn);
25 % Finner neste y-verdi ved Eulers metode
26   yVektor(n+1)=yn+yDeriv*h;
27 end
28
29 % Plotter løsninga
30 plot(xVektor,yVektor,'b-', 'linewidth',2)

```

Skriptet heter `EulerFramoverFunk.m`. Merk at initialkravet og maksimal x -verdi fremdeles er hard-koda; dette kravet må endres hver gang vi skal løse et nytt initialverdiprobem. Funksjonsfila som gir $F(x, y)$, heter `DerivFunk.m`. For deloppgave 2a) ser den slik ut:

```

function F=DerivFunk(x,y)

% Funksjon som blir bruka av skripta EulersMetode.m og
% og EulerMidpunkt.m. Funksjonen er høgresida i diff.-
% likninga  $y'=F(x,y)$ 

F=sin(x+y^2);

```

Midpunkts-metoden er implementert akkurat likt; det er bare `for`-løkka som er ulik. Den ser slik ut:

```

for n=1:N
% Tilornder x- og y-verdiar
   xn=xVektor(n);
   yn=yVektor(n);
% Tilordner x-hatt og y-hatt
   xnHatt=xn+h/2;
   ynHatt=yn+DerivFunk(xn,yn)*h/2;
% Finner neste y-verdi ved Eulers midtpunkt-metode
   yVektor(n+1)=yn+DerivFunk(xnHatt,ynHatt)*h;
end

```

a)

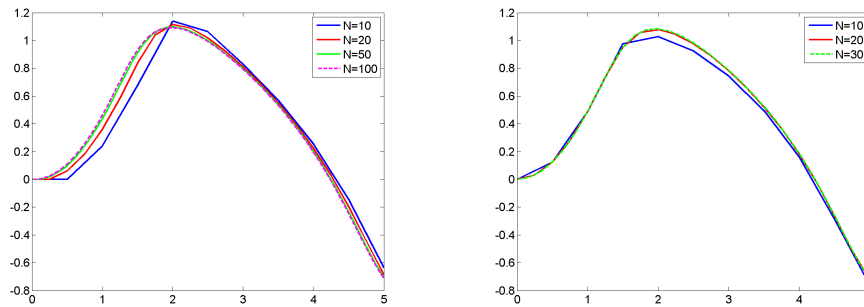
$$y' = \sin(x + y^2), \quad y(0) = 0, \quad x \in [0, 5] \quad .$$

Vi kjører skriptet fleire ganger med stadig høyere N (lavere h):

```

>> EulerFramoverFunk
Gi antall steg: 10
h =

```



Figur 6: Svaret på deloppgåve 2a). Figuren viser hvordan svaret konvergerar med Eulers framover-metode (venstre) og med Eulers midtpunktsmetode (høgre)

```
0.5000
```

```
>> EulerFramoverFunk
```

```
Gi antall steg: 20
```

```
h =
```

```
0.2500
```

```
>> EulerFramoverFunk
```

```
Gi antall steg: 50
```

```
h =
```

```
0.1000
```

```
>> EulerFramoverFunk
```

```
Gi antall steg: 100
```

```
h =
```

```
0.0500
```

Etterpå gjør vi tilsvarende med midtpunktsforlemlen. Resultatene er vist i figur 6. For framover-metoden ser vi ut til å trenge minst $N = 100$, mens $N = 30$ ser ut til å vere nok for midtpunktsmetoden.

b)

$$y' + x^2 \cos(x/2) = \sqrt{y}, \quad y(0) = 1, \quad x \in [0, 4] \quad .$$

Likninga er ekvivalent med

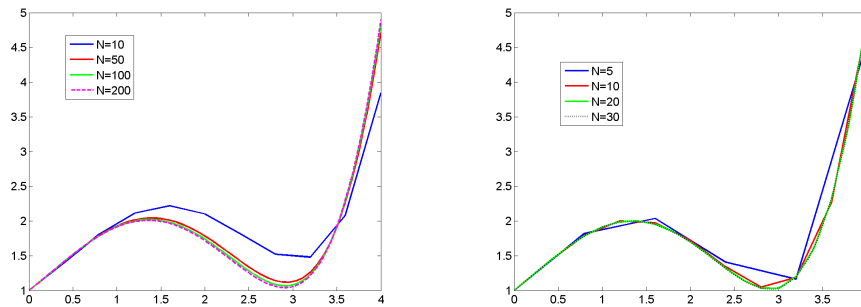
$$y' = \sqrt{y} - x^2 \cos(x/2) \quad .$$

Vi endrer de første radene i skripta til følgende:

```
% Tar bort gamle variable
```

```
clear all
```

```
% Gir start og slutt for x
```



Figur 7: Svaret på deloppgave 2b). Som i figur 6 viser figuren hvordan svaret konvergerer med Eulers framover-metode (venstre) og med midtpunktmetoden (høgre).

```
x0=0;
xSlutt=4;
% Initialverdi for y:
y0=1;
```

Videre endrer vi siste linje i funksjonsfila til

```
F=sqrt(y)-x^2*cos(x/2);
```

Konvergesplottene er vist i figur 7. Framover-resultatet begynner å se konvergent ut med $N = 200$ ($h = 0.02$), mens $N = 30$ ($h = 0.13$) ser ut til å vere rikeleg med midpunktsmetoden.

Ekstra: Oppgave 3 – konvergensbevis

a) Vi deriverer:

$$y' = y_0 \cdot (e^x)' = y_0 e^x = y(x) \quad .$$

Vi ser at differensiallikninga er oppfylt. Initialkravet er også oppfylt:

$$y(0) = y_0 e^0 = y_0.$$

Vi regner ut $y(1)$:

$$y(1) = y_0 e^1 = y_0 e \quad .$$

b) I følge Eulers metode får vi at

$$y_1 = y_0 + y_0 h = y_0(1 + h)$$

$$y_2 = y_1 + y_1 h = y_1(1 + h) = y_0(1 + h) \cdot (1 + h) = y_0(1 + h)^2$$

$$y_3 = y_2 + y_2 h = y_2(1 + h) = y_0(1 + h)^2 \cdot (1 + h) = y_0(1 + h)^3$$

$\vdots$

...og så videre. Etterhvert blir det ganske tydelig at vi for hver iterasjon multipliserer forrige ledd med $(1+h)$ slik at

$$y_n = y_0(1+h)^n \quad .$$

c)

$$\lim_{N \rightarrow \infty} y_N = \lim_{N \rightarrow \infty} y_0(1+h)^N = y_0 \lim_{N \rightarrow \infty} \left(1 + \frac{1}{N}\right)^N = y_0 e \quad .$$

d) Eulers bakover-metode gir her følgende formel:

$y_{n+1} = y_n + y_{n+1} h \Leftrightarrow y_{n+1}(1-h) = y_n \Leftrightarrow y_{n+1} = y_n/(1-h)$. Det gir at

$$\begin{aligned} y_1 &= \frac{y_0}{1-h} \\ y_2 &= \frac{y_1}{1-h} = \frac{\frac{y_0}{1-h}}{1-h} = \frac{y_0}{(1-h)^2} \\ &\vdots \\ y_n &= \frac{y_0}{(1-h)^n} \end{aligned}$$

Grenseverdien blir:

$$\lim_{N \rightarrow \infty} y_N = \lim_{N \rightarrow \infty} \frac{y_0}{(1-h)^N} = y_0 \lim_{N \rightarrow \infty} \left(1 - \frac{1}{N}\right)^{-N} = y_0 \lim_{u \rightarrow 0} (1+u)^{1/u} = y_0 e \quad .$$

Vi har her innført variabelbyttet $u = -1/N$.