Laborationstillfälle 4 – Numerisk lösning av ODE

Målsättning vid labtillfälle 4: Klara av laborationsuppgift 3. Läs först texten om differensmetoder och gör övningarna. Läs avsnittet Högre ordningens differentialekvationer om du har tid, annars spara det till ett senare tillfälle. De frivilliga uppgifterna på slutet bygger på detta avsnitt.

Vi skall se på begynnelsevärdesproblem för första ordningens ordinär differentialekvation

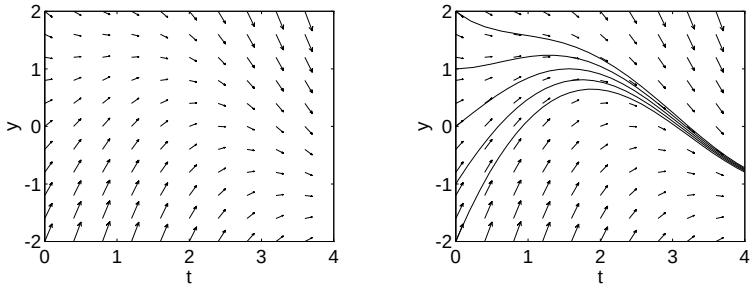
$$\begin{cases} y' = f(t,y),\ a \le t \le b \\ y(a) = c \end{cases}$$

där f en given funktion och c en given konstant. Lösningen till problemet är en funktion $t \mapsto y(t)$.

Exempel 1. I den vänstra figuren nedan har vi ritat riktningsfältet till

$$\begin{cases} y' = -y(t) + \sin(t) + \cos(t),\ 0 \le t \le 4 \\ y(0) = c \end{cases}$$

och i den högra lösningskurvorna för några olika värden på c.



Vi ser hur lösningskurvorna följer riktningsfältet.

Med **ode45** kan vi få fram en numerisk lösning. Vi beskriver högerledet i differentialekvationen och beräknar sedan lösningen för t.ex. $y(0) = 1$.

```
>> myrslok=inline('-y+sin(t)+cos(t)','t','y')
myrslok =
Inline function:
myrslok(t,y) = -y+sin(t)+cos(t)
>> y0=1; [t,y]=ode45(myrslok,0:0.1:4,y0);
>> plot(t,y)
```

I ode45-kommandot är första argumentet funktionen $f(t,y)$, andra argumentet är önskat lösningsintervall [a b] (så kan det också skrivas in, så bestämmer ode45 självt en steglängd) och det tredje är begynnelsevärdet $y(a)$. I stället för en inline-funktion kan man ha en funktionsfil, en s k ode-fil. Den ska ha formen fun(t,y) även om bara den ena variabeln syns i funktionsuttrycket. Detta gäller också inline-funktionen, vilket åstadkoms på det sätt som framgår av ovanstående exempel: inline('...','t','y'). Se även "help ode45"!

Differensmetoder

Vi skall approximera lösningen $y(t)$ till differentialekvationen på ett nät $t_n = a + nh$ för $n = 0, 1, \cdots, N$, med steglängden $h = \frac{b-a}{N}$.

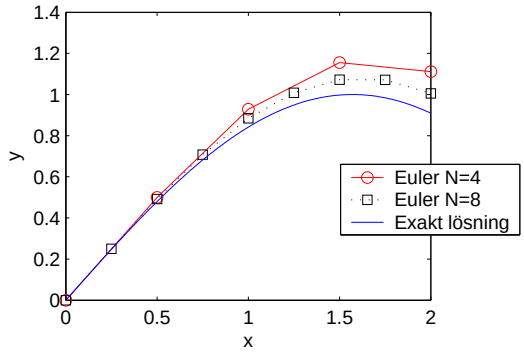
Låt y_n beteckna approximationen av $y(t_n)$ och ersätt $y'(t_n)$ enligt

$$\frac{y(t_{n+1}) - y(t_n)}{h} \approx y'(t_n) = f(t_n, y(t_n)) \Rightarrow$$

$$y(t_{n+1}) \approx y(t_n) + hf(t_n, y(t_n))$$

Detta ger den s.k. Euler (framåt)metoden

$$y_{n+1} = y_n + hf(t_n, y_n)$$



Geometrisk tolkning: Utgående från begynnelsevärdet följer metoden riktningsfältet med korta steg.

Övning 1. Rita riktningsfältet för differentialekvationen i exempel 1. Använd sedan Eulers metod för att beräkna approximationer av lösningar för några olika värden på c. Tag steglängden h så pass liten att ni får bra approximationer.

En förbättring av Eulers metod får man om man efter ett "provsteg" beräknar lutningen i den erhållna punkten, dvs $f(t_n + h, y_n + hf(t_n, y_n))$ och sedan använder medelvärdet av denna och den föregående $f(t_n, y_n)$ istället för den sistnämnda. Denna metod kallas *Heuns metod*:

$$y_{n+1} = y_n + \frac{h}{2}\{k_1 + k_2\}, \quad k_1 = f(t_n, y_n), \; k_2 = f(t_n + h, y_n + hk_1)$$

Metoderna ovan är alla konvergenta, dvs. tar vi tillräckligt liten steglängd h kan vi få godtyckligt bra approximation på ett ändligt intervall.

Övning 2. Använd Heuns metod för att approximera $y(0.8)$, där $y(t)$ är lösningen till $y' = 1 + t\,y^2, \; y(0) = 0$. Tag stegen $h = 0.2$ och $h = 0.1$. Jämför med vad du får med Eulers metod. Kan du se att Heun är mer nogrann än Euler? Noggrannare gäller $y(0.8) = 0.91996546$ (korrekt avrundat).

Vid konstant steglängd gäller att Eulers har ett lokalt fel (på varje steg) som är av storleksordningen $\mathcal{O}(h^2)$ och ett globalt fel (på hela lösningsintervallet) $\mathcal{O}(h)$, medan Heuns metod ger ett lokalt fel $\mathcal{O}(h^3)$ och ett globalt fel som är $\mathcal{O}(h^2)$.

Bra program för att lösa begynnelsevärdesproblem är adaptiva, dvs. de *varierar* steglängden h för att bli *effektiva*. Dessa program ser till att de lokala felen blir små, vilken betydelse detta får på det globala felen beror på differentialekvationen och det får *användaren* av programmet hålla reda på själv! Exempel på en adaptiv lösare är **ode45** i MATLAB.

Högre ordningens differentialekvationer

Högre ordningens differentialekvationer kan skrivas om som system av första ordningen. Dessa system kan sedan lösas numeriskt. Vissa problem ger naturligt ett system av differentialekvationer (se t ex *Volterra-Lotkas* ekvationer sist i denna laborationshandledning).

Exempel 2. Fastspänd bladfjäder som vi knäpper på. Vibrationen beskrivs av

$$\begin{cases} x'' = -x^3 - 0.05x' \\ x(0) = 1, \quad x'(0) = 0 \end{cases}$$

Om vi låter $v = x'$ kan ekvationen skrivas

$$\begin{cases} x' = v \\ v' = -x^3 - 0.05v \\ x(0) = 1 \\ v(0) = 0 \end{cases}$$

För att komma till standardformen låter vi $y_1 = x$ och $y_2 = v$ och får

$$\begin{cases} y_1' = y_2 \\ y_2' = -y_1^3 - 0.05y_2 \\ y_1(0) = 1 \\ y_2(0) = 0 \end{cases}$$

Nu får vi standardformen

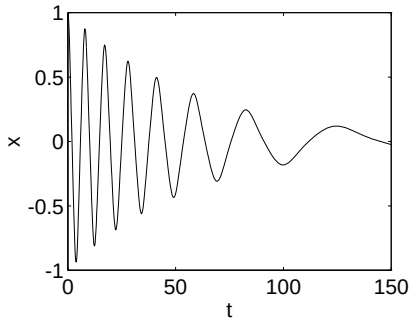
$$\begin{cases} y' = f(t,y) \\ y(0) = c \end{cases}, \quad y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad f(t,y) = \begin{bmatrix} y_2 \\ -y_1^3 - 0.05y_2 \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

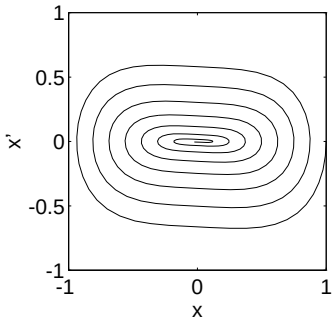
Vi använder **ode45** för att beräkna en approximation av lösningen.

```
>> f=inline('[y(2); -y(1)^3-0.05*y(2)]','t','y');
>> tspan=[0 150];              % Begynnelse- och sluttid
>> c=[1; 0];                   % Begynnelsevärden
>> [t,y]=ode45(f,tspan,c);     % Lösningen
```

Ritar lösning $t \mapsto (t, x(t))$ och fasporträtt $t \mapsto (x(t), x'(t)) = (x(t), v(t))$

```
>> plot(t,y(:,1))
>> plot(y(:,1),y(:,2))
```





Övning 3. Sriv om följande differentialekvationer till system av 1:a ordningen.

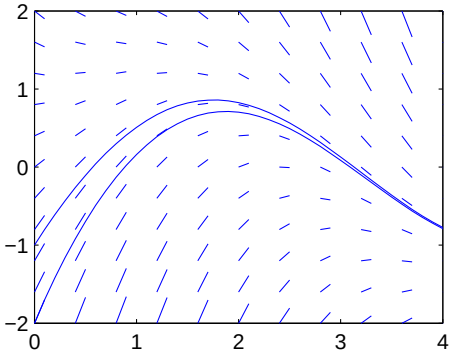
(a). $y'' = t + y + y'$ **(b).** $y''' = y'' + ty$ **(c).** $y''' = -yy''$

Lösningar till övningsuppgifter

1.

```
myrslok=inline('-y+sin(t)+cos(t)','t','y');
s=0.1;
for t=0:0.4:4
   for y=-2:0.4:2
      dt=1; dy=myrslok(t,y);              % I punkten (t,y) ritar vi ett
      plot([t,t+s*dt],[y,y+s*dy])         % litet streck i riktningen (1,y').
      hold on
   end
end

for c=[-1 -2]
   h=0.1;
   t=0:h:4; y=zeros(size(t));
   y(1)=c;
   for k=1:length(t)-1
      y(k+1)=y(k)+h*myrslok(t(k),y(k));
   end
   plot(t,y)
end
```



När vi lärt oss lite mer linjär algebra (läsperiod 3) kan vi använda funktionen **quiver** för att enklare rita riktningsfält.

```
t=linspace(0,4,10); y=linspace(-2,2,10);
[T,Y]=meshgrid(t,y); DT=ones(size(T)); DY=myrslok(T,Y);
quiver(T,Y,DT,DY)
```

Förklaring av funktionerna **meshgrid** och **quiver** väntar vi med.

2.
```
f=inline('1+t*y^2','t','y');
yexakt=0.91996546;
c=0;
for h=[0.2 0.1]
   t=0:h:0.8;  y=zeros(size(t)); z=zeros(size(t));
   y(1)=c; z(1)=c;                                  % y=Euler framåt, z=Heun
   for k=1:length(t)-1
      y(k+1)=y(k)+h*f(t(k),y(k));
      z(k+1)=z(k)+h/2*(f(t(k),z(k))+f(t(k+1),z(k)+h*f(t(k),z(k))));
   end
   disp(['h=',num2str(h),' Euler fel=',num2str(abs(y(end)-yexakt)),
                         ' Heun fel=',num2str(abs(z(end)-yexakt))])
end
```

Ger utskriften

```
h=0.2 Euler fel=0.060149 Heun fel=0.00527
h=0.1 Euler fel=0.034519 Heun fel=0.0014334
```

Vi ser att felet i Euler är $\mathcal{O}(h)$ medan felet i Heun är $\mathcal{O}(h^2)$.

3. **(a).** Låt $y_1 = y, y_2 = y'$. **(b).** Låt $y_1 = y, y_2 = y', y_3 = y''$. **(c).** Låt $y_1 = y, y_2 = y', y_3 = y''$.

$$\begin{cases} y_1' = y_2 \\ y_2' = t + y_1 + y_2 \end{cases} \qquad \begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ y_3' = y_3 + t + y_1 \end{cases} \qquad \begin{cases} y_1' = y_2 \\ y_2' = y_3 \\ y_3' = -y_1 y_3 \end{cases}$$

Laborationsuppgift 3.

1. Skriv en funktionsfil **euler.m** som löser en ode av 1:a ordningen: $y' = f(x, y)$. Indata skall vara f(x,y) (t ex en fördefinerad inline-funktion), lösningsintervallet $[a, b]$, begynnelsevärdet $y(a)$ och steglängden h. Utdata skall vara matrisen $[x\ y]$ som utgör en värdetabell för lösningen. Om man skriver $[x\ y] = euler(...)$ i kommandofönstret, kan man sedan plotta lösningen (vilket förstås även kan begäras i funktionsfilen). Till skillnad från i övning 1 ska du inte rita något riktningsfält.

(a). Lös uppgift 9.16(a) och 9.16(b) i läroboken (*Forsling/Neymark*) med programmet **euler.m** i intervallet $[1, 10]$.
Testa också att lösa problemet med **ode45**.

(b). Lös uppgift 9.17 i ett lämpligt intervall $[0, c]$. Välj själv konstanter, observera att $a \neq b$! Välj intervallet så att kurvan "hinner" visa upp sig väl. Tillverka en serie lösningskurvor i samma figur (jfr sid 388 i läroboken), använd begynnelsevärden mellan a och b.
Om du vill testa med **ode45**, kom ihåg att inlinefunktionen (eller motsvarande funktionsfil om du använder en sådan) till formen måste ha variablerna x och y, även om bara y ingår i funktionsuttrycket.

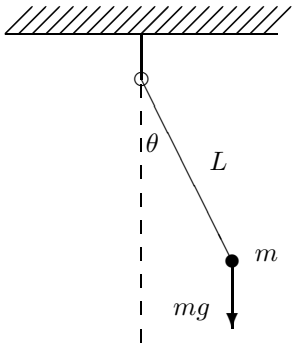
Frivilliga uppgifter (redovisas ej).

2. En *plan pendel*. En masspunkt med massan m hänger i en viktlös smal stav av längden L.

Med beteckningarna i figuren och Newtons andra lag får vi rörelseekvationen

$$\theta'' = -\frac{g}{L}\sin\theta - k\theta'$$

Här är θ pendelutslaget (radianer) och g är tyngdaccelerationen ($\approx 10m/s^2$). Termen $k\theta'$ är en dämpningsterm ($k > 0$). Välj ett värde på k och ett värde på L. Vi vill bestämma lösningen för olika begynnelseutslag θ_0 då vi släpper pendeln från vila.



Följ lösningskurvan med **ode45** för några olika begynnelseutslag, t.ex. $\theta_0 = 30, 50, \cdots, 110°$.

Rita två bilder som visar lösningarna $t \mapsto (t, \theta(t))$ och fasporträtten $t \mapsto (\theta(t), \theta'(t))$ för de olika begynnelseutslagen.

3. *Volterra-Lotkas* ekvationer för samspel mellan rovdjur och bytesdjur:

$$y_1' = (1 - \frac{y_2}{50})y_1, \quad y_2' = (-1 + \frac{y_1}{100})y_2$$

y_1 är bytesdjur, t ex harar, y_2 är rovdjur, t ex rävar. Låt antalet bytesdjur vid tiden noll vara betydligt större än antalet rovdjur. Pröva lite olika begynnelsevillkor, plotta i något fall dels båda lösningskurvorna i samma plot, dels faskurvan i en ny plot.

Nya **kommandon och funktioner** i MATLAB som vi använt vid detta labtillfälle.

ones – ger en matris fylld med ettor.

meshgrid – ger matriser för användning i samband med t.ex **quiver**.
quiver – ritar ett fält
hold – håller kvar grafik.

ode45 – beräknar approximativ lösning till ett system av 1:a ordningens ode.
