

# Ordinära differentialekvationer – fortsättning

Analys och Linjär Algebra, del B, K1/Kf1/Bt1

## 1 Inledning

Vi skall se lite mer på system av ordinära differentialekvationer av typen

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

Exempelvis skall vi se på s.k. styva problem och metoder som är lämpliga för sådana. Styva problem uppstår ofta i ekvationer som beskriver kemiska reaktioner. Avslutningsvis skall vi se hur högre ordningens differentialekvationer kan skrivas om som system av första ordningens ekvationer.

## 2 Allmänt system av ODE

Vi skall lösa  $m$  differentialekvationer med tillhörande begynnelsevärden

$$\begin{cases} u_1'(t) = f_1(t, u_1(t), \dots, u_m(t)), & t \in [a, b], & u_1(a) = u_{a1}, \\ \vdots & & \vdots \\ u_m'(t) = f_m(t, u_1(t), \dots, u_m(t)), & t \in [a, b], & u_m(a) = u_{am}. \end{cases}$$

Om vi inför vektorerna

$$\mathbf{u}(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ u_m(t) \end{bmatrix}, \quad \mathbf{f}(t, \mathbf{u}) = \begin{bmatrix} f_1(t, u_1(t), \dots, u_m(t)) \\ \vdots \\ f_m(t, u_1(t), \dots, u_m(t)) \end{bmatrix}, \quad \mathbf{u}_a = \begin{bmatrix} u_{a1} \\ \vdots \\ u_{am} \end{bmatrix}$$

så kan begynnelsevärdesproblemet skrivas på standardform

$$\begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)), & t \in [a, b], \\ \mathbf{u}(a) = \mathbf{u}_a. \end{cases} \quad (1)$$

Här är  $\mathbf{f} : \mathbb{R}^{m+1} \rightarrow \mathbb{R}^m$  den oftast icke-linjära funktion som beskriver differentialekvationen och vektorn  $\mathbf{u}_a$  ger begynnelsevillkoret.

## 3 Några lösningsmetoder

Vi ska nu beskriva hur man kan konstruera en lösning  $\mathbf{u}$  för varje kontinuerlig funktion  $\mathbf{f}$ , dvs. konstruera en unik lösning till begynnelsevärdesproblemet (1).

Vi har redan i en tidigare studio-övning tittat på **Eulers framåtmetod**. Vi började med att dela in intervallet  $[a, b]$  i  $N$  stycken delintervall av längden  $h = (b - a)/N$ :

$$a = t_0 < t_1 < t_2 < \dots < t_{i-1} < t_i < \dots < t_{N-1} < t_N = b, \\ t_i = a + hi, \quad h = (b - a)/N = t_i - t_{i-1}.$$

Vi beräknade sedan en approximativ lösning enligt

$$\mathbf{U}(t_0) = \mathbf{u}_a \\ \mathbf{U}(t_i) = \mathbf{U}(t_{i-1}) + h\mathbf{f}(t_{i-1}, \mathbf{U}(t_{i-1})).$$

Metoden kallas *explicit* därför att i varje steg beräknar vi den nya kolonnvektorn  $\mathbf{U}(t_i)$  som en explicit funktion av  $t_{i-1}$  och den förra kolonnvektorn  $\mathbf{U}(t_{i-1})$ .

Vi kan istället använda **Eulers bakåtmetod**

$$\mathbf{U}(t_0) = \mathbf{u}_a \\ \mathbf{U}(t_i) = \mathbf{U}(t_{i-1}) + h\mathbf{f}(t_i, \mathbf{U}(t_i)).$$

Denna metod är *implicit* därför att vi måste lösa ut den nya vektorn  $\mathbf{U}(t_i)$  ur ett icke-linjärt ekvationssystem. Hur man löser sådana ekvationssystem kommer vi lära oss i ALA-C.

Även **trapetsmetoden**

$$\mathbf{U}(t_0) = \mathbf{u}_a \\ \mathbf{U}(t_i) = \mathbf{U}(t_{i-1}) + \frac{h}{2}(\mathbf{f}(t_{i-1}, \mathbf{U}(t_{i-1})) + \mathbf{f}(t_i, \mathbf{U}(t_i))),$$

är en implicit metod.

Implicita metoder är, trots att de är mer komplicerade att skriva program för och varje tidssteg kräver mer beräkningar, mer effektiva på en typ av begynnelsevärdesproblem som kallas styva problem.

Ett begynnelsevärdesproblem kallas *styvt* om det beskriver förlopp eller processer vilka utspelas under tidsintervall av mycket olika storleksordning. T.ex. inom kemisk reaktionsteknik är styva problem vanliga. Det finns färdiga program för sådana problem i MATLAB.

## 4 ODE-lösare i MATLAB

MATLAB har flera program som löser ODE med samma anrop som `min_ode` utom att man inte behöver ange steget; det väljs adaptivt av programmet. Till exempel,

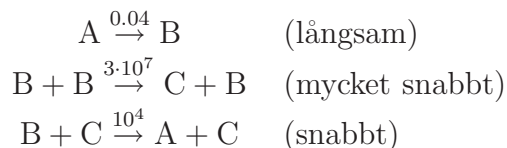
```
>> [t,U]=ode45(f,I,ua)
>> [t,U]=ode15s(f,I,ua)
```

Här är `ode15s` avsedd för styva problem.

Som exempel tar vi: Robertsons ekvation (kemisk kinetik)

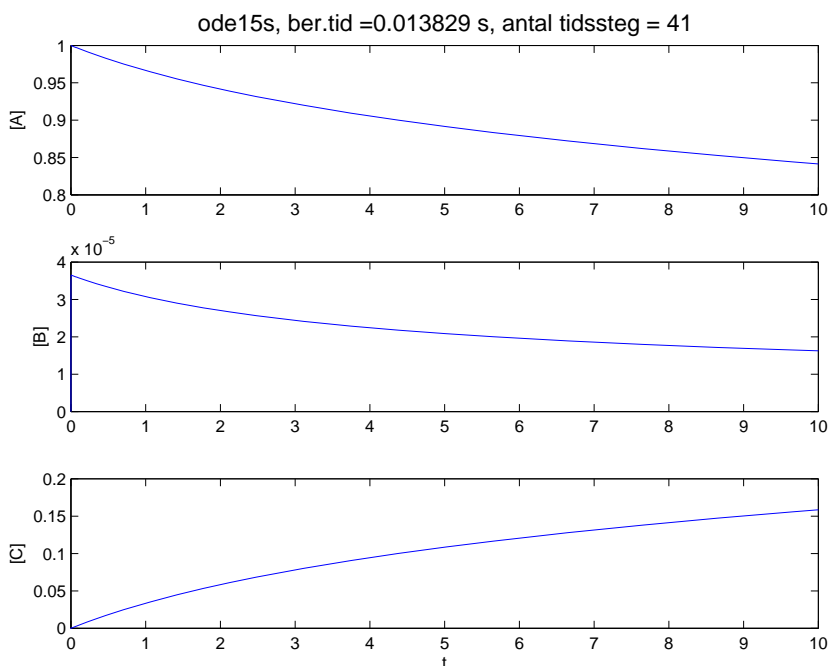
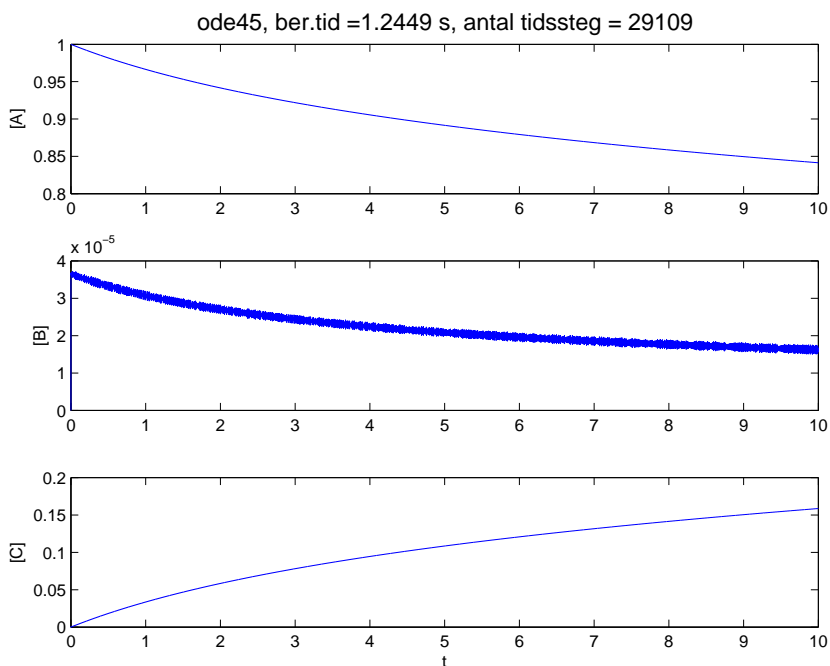
$$\begin{cases} u_1'(t) = -0.04u_1(t) + 10^4u_2(t)u_3(t), & u_1(0) = 1, \\ u_2'(t) = 0.04u_1(t) - 10^4u_2(t)u_3(t) - 3 \cdot 10^7u_2(t)^2, & u_2(0) = 0, \\ u_3'(t) = 3 \cdot 10^7u_2(t)^2, & u_3(0) = 0, \end{cases}$$

där  $u_1 = [A]$ ,  $u_2 = [B]$ ,  $u_3 = [C]$  är koncentrationen av ämnena A, B, C som reagerar enligt



Problemet blir styvt eftersom det har mycket olika tidsskalor.

Vi använder en vanlig lösare `ode45` och en styv lösare `ode15s`. Vi jämför antal tidssteg och beräkningstid, dvs. vilken effektivitet vi får.



Detta var ett renodlat *kvalitativt* exempel för att belysa vikten av att ha olika typer av lösare av differentialekvationer. Lite senare i denna studio-övning finns en lite större uppgift som handlar om en konkret kemisk reaktion, dvs. ett *kvantitativt* exempel.

## 5 Första ordningens system

Vi ser på ett exempel från epidemiologi, nämligen Kermack-McKendricks differentialekvationer

$$\begin{cases} u_1'(t) = -c u_1(t) u_2(t) \\ u_2'(t) = c u_1(t) u_2(t) - d u_2(t) \\ u_3'(t) = d u_2(t) \end{cases}$$

som beskriver utvecklingen av en epidemi.

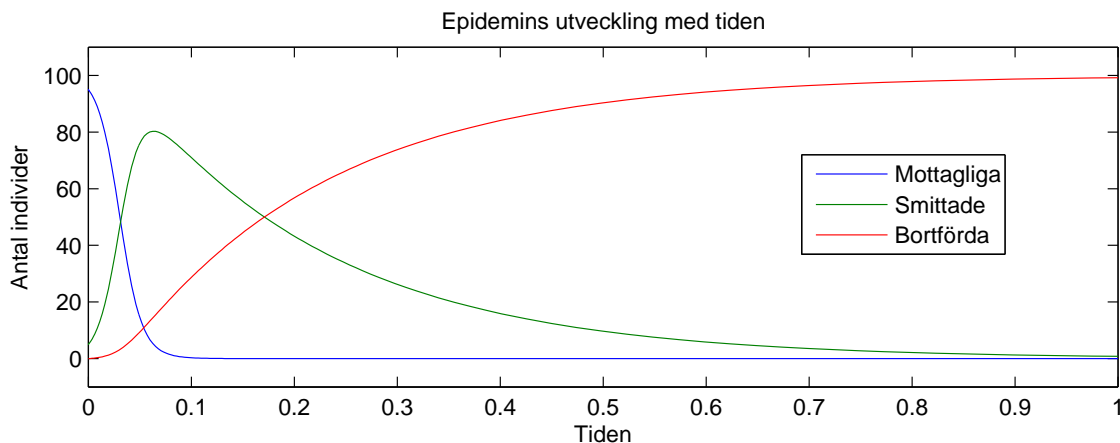
Här står  $u_1(t)$ ,  $u_2(t)$ ,  $u_3(t)$  för antal mottagliga, smittade respektive bortförda. Parametrarna  $c$  och  $d$  beskriver smittspridning- respektive bortförandegraden.

Vi tar  $c = 1$  och  $d = 5$  samt begynnelsestillståndet  $u_1(0) = 95$ ,  $u_2(0) = 5$  och  $u_3(0) = 0$ . Hur utvecklas epidemin med tiden? Vi beskriver differentialekvationen med

```
function uprim=epidemi(t,u)
c=1; d=5;
uprim=[-c*u(1)*u(2); c*u(1)*u(2)-d*u(2); d*u(2)];
```

beräknar och ritar lösningen med

```
>> u0=[95;5;0];
>> [t,u]=ode45(@epidemi,[0,1],u0);
>> plot(t,u)
>> title('Epidemins utveckling med tiden')
>> xlabel('Tiden')
>> ylabel('Antal individer')
>> legend('Mottagliga','Smittade','Bortförda')
```



Vissa likheter med ett visst projekt?

Vid en tidigare studio-övning betraktade vi en population av bytesdjur (kaniner) som lever tillsammans med en population rovdjur (rävar). Vi lät  $u_1(t)$  respektive  $u_2(t)$  beteckna antalet kaniner

respektive rävar vid tiden  $t$ . Som matematisk modell för populationernas utveckling använde vi Volterra-Lotka-ekvationerna:

$$\begin{cases} u_1'(t) = a u_1(t) - b u_1(t) u_2(t) \\ u_2'(t) = -c u_2(t) + d u_1(t) u_2(t) \end{cases} \quad (2)$$

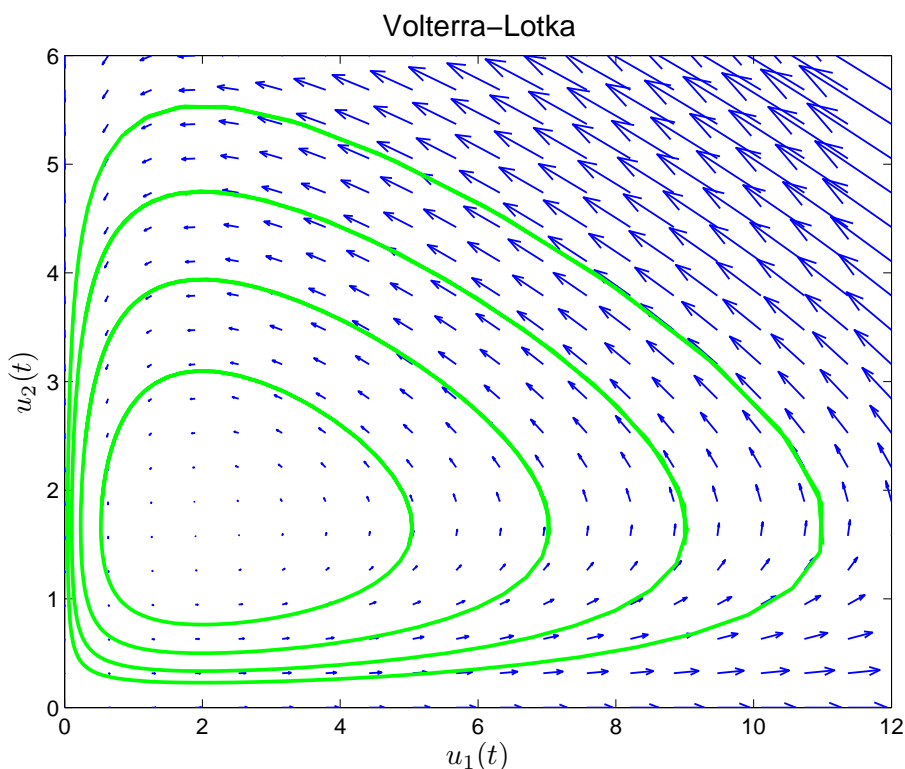
Med vektorbeteckningar får vi

$$\begin{cases} \mathbf{u}' = \mathbf{f}(t, \mathbf{u}) \\ \mathbf{u}(0) = \mathbf{u}_0 \end{cases}, \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad \mathbf{f}(t, \mathbf{u}) = \begin{bmatrix} a u_1 - b u_1 u_2 \\ -c u_2 + d u_1 u_2 \end{bmatrix}, \quad \mathbf{u}_0 = \begin{bmatrix} u_{01} \\ u_{02} \end{bmatrix} \quad (3)$$

För att få en lite bättre uppfattning av vilka egenskaper systemet (3) har så skall vi titta närmare på högerledet. Eftersom funktionen  $\mathbf{f} = \mathbf{f}(\mathbf{u})$  bara beror av  $\mathbf{u}$  och inte av  $t$ , differentialekvationen kallas då *autonom*, så kan vi rita ett s.k. *riktningsfält*.

För varje initialvärde till (3) så har vi en (okänd) lösning  $\mathbf{u}(t)$ . Vi tänker oss ett nät av punkter. I varje punkt  $\mathbf{u}$  ger  $\mathbf{u}'(t)$  den riktning som en lösning genom den punkten går åt. Vi ritar en pil i denna riktning. Då får vi ett riktningfält.

Genom att rita ut detta riktningfält får vi ungefärlig information om hur lösningen  $\mathbf{u}(t)$  till ekvationen beter sig för samtliga möjliga startvärden. Vi kan alltså skapa kurvan  $\mathbf{u}(t)$  genom att sätta pennan på den startpunkt vi önskar och sedan följa pilarna.



Vi ritar några lösningskurvor i riktningfältet med (funktionen `volterra` gjorde vi i en tidigare studio-övning)

```
>> ua=[11;1.5];
>> [t,U]=ode45(@volterra,[0,80],ua);
>> plot(U(:,1),U(:,2),'g','LineWidth',2)
```

```

>> hold on
>> ua=[9;1.5];
>> [t,U]=ode45(@volterra,[0,80],ua);
>> plot(U(:,1),U(:,2),'g','LineWidth',2)
...
>> hold off

```

Hur skall vi modifiera Volterra-Lotka-ekvationerna för att beskriva en överpopulation hos kani-  
nerna. Hur påverkas fasporträttet. I nästa läsperiod skall vi se hur man ritar riktningsfält.

**Uppgift 1.** Vi skall se på en typ av "Belousov-Zhabotinsky reaction" (se Atkins & Jones sid 636).  
Det vi ser på är den kemiska klockan som är ett något exotiska exempel på ett periodiskt förlopp  
i en kemin reaktion. Materialet bygger på Field och Noyes, J. Chem. Phys. 60: 1877-1884 (1974).

För en blandning av

1.25 M	H <sub>2</sub> SO <sub>4</sub>
0.0125 M	KBrO <sub>3</sub>
0.001 M	Ce(NH <sub>4</sub> ) <sub>2</sub> (NO <sub>3</sub> ) <sub>5</sub>
0.025 M	CH <sub>2</sub> (COOH) <sub>2</sub>

härleder Field och Noyes massbalanser för mellanprodukterna HBrO<sub>2</sub>, Br<sup>-</sup> och Ce<sup>4+</sup> med dimen-  
sionslösa koncentrationer  $u_1(t)$ ,  $u_2(t)$  respektive  $u_3(t)$ ;

$$\begin{cases} u_1'(t) = s(u_2(t) - u_1(t)u_2(t) + u_1(t) - qu_1(t)^2), & u_1(0) = 4 \\ u_2'(t) = s^{-1}(u_3(t) - u_2(t) - u_1(t)u_2(t)), & u_2(0) = 1.1 \\ u_3'(t) = w(u_1(t) - u_3(t)), & u_3(0) = 4 \end{cases}$$

där  $s = 77.27$ ,  $q = 8.375 \cdot 10^{-6}$  och  $w = 0.1610$ .

Integrerar man över långa tidsperioder ser man att lösningen är periodisk med en periodlängden  
på ungefär 300 tidsenheter (48.3 sekunder).

Blandningen kommer att växla färg. Först kommer den vara klar, sedan växlar den till gult, blir  
klar igen, blir sedan åter gul, och så vidare. Man kan tillsätta en indikator så att färgväxlingen  
blir mellan blått och rött.

Problemet är styvt så vi använder den styva lösaren `ode15s` i MATLAB. Beräkna och rita upp  
lösningen under en period. Därefter studerar ni det initiala förloppet genom att beräkna och rita  
upp lösningen under de 5 första tidsenheterna. Rita ut logaritmerna av koncentrationerna, så att  
vi ser även små koncentrationer (<sup>10</sup>log ges av `log10` i MATLAB). Använd `xlabel`, `ylabel` och  
`legend` för att göra tydliga grafer.

## 6 Linjära system

En viktig speciell typ av system av differentialekvationer är *linjära system*

$$\begin{cases} \mathbf{u}'(t) = \mathbf{A}\mathbf{u}(t), & t \in [a, b], \\ \mathbf{u}(a) = \mathbf{u}_a. \end{cases} \quad (4)$$

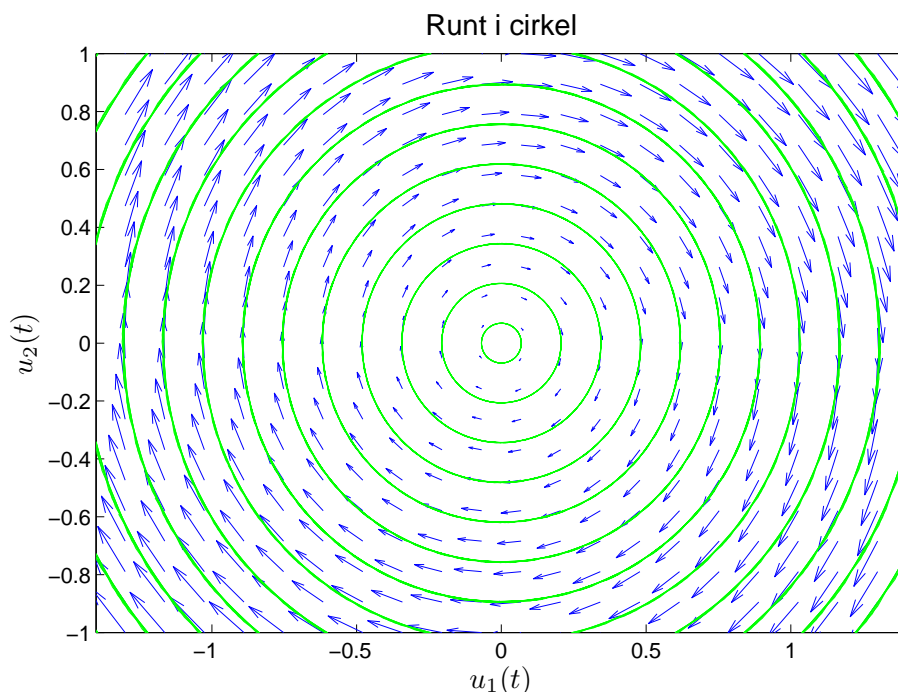
Här är  $\mathbf{A}$  en kvadratisk matris ( $n \times n$ ), dvs.  $\mathbf{f}(t, \mathbf{u}) = \mathbf{A}\mathbf{u}$  är en linjär funktion. Till skillnad  
från de flesta icke-linjära system så kan vi lösa de linjära analytiskt (dvs. exakt) med den s.k.  
egenvärdesmetoden, vi återkommer till detta i ALA-C.

Som exempel på ett linjärt system tar vi

$$\begin{cases} \mathbf{u}' = \mathbf{A}\mathbf{u} \\ \mathbf{u}(0) = \mathbf{u}_0 \end{cases}, \quad t \geq 0, \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad \mathbf{u}_0 = \begin{bmatrix} u_{01} \\ u_{02} \end{bmatrix} \quad (5)$$

Vi ritar några lösningskurvor i riktningsfältet

```
>> A=[0 1;-1 0]; f=@(t,u)A*u;
>> r=1; u0=[0;r];
>> [t,U]=ode45(f,[0,10],u0);
>> plot(U(:,1),U(:,2),'g','LineWidth',2)
>> hold on
>> r=0.8; u0=[0;r];
>> [t,U]=ode45(f,[0,10],u0);
>> plot(U(:,1),U(:,2),'g','LineWidth',2)
...
>> hold off
```



## 7 Högre ordningens differentialekvationer

Högre ordningens differentialekvationer

$$u'' = f(t, u, u'), \quad u''' = f(t, u, u', u''), \quad \dots$$

kan om vi låter

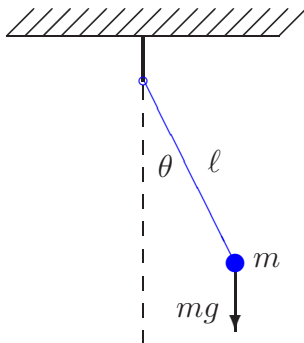
$$u_1 = u, \quad u_2 = u', \quad u_3 = u'', \quad \dots$$

skrivs om som *system* av första ordningens ekvationer

$$\mathbf{u}' = \mathbf{f}(t, \mathbf{u}), \quad \text{där} \quad \mathbf{u}(t) = (u_1(t), u_2(t), \dots)$$

Systemet får lika många ekvationer som ordningen på ursprungliga ekvationen.

Som exempel tar vi en s.k. matematisk pendel. En masspunkt med massan  $m$  hänger i en viktlös smal stav av längden  $\ell$ .



Med beteckningarna i figuren och Newtons andra lag får vi rörelseekvationen

$$m\ell\ddot{\theta}(t) = -mg\sin(\theta(t))$$

Vi vill bestämma lösningen för olika begynnelseutslag  $\theta_0$ , dvs.  $\theta(0) = \theta_0$ , då vi släpper pendeln från vila, dvs.  $\dot{\theta}(0) = 0$ .

Om vi låter  $\varphi = \dot{\theta}$ , dvs. inför vinkelhastigheten, kan ekvationen skrivas

$$\begin{cases} \dot{\theta} = \varphi, & \theta(0) = \theta_0 \\ \dot{\varphi} = -\frac{g}{\ell}\sin(\theta), & \varphi(0) = 0 \end{cases}$$

För att komma till standardform låter vi  $u_1 = \theta$  och  $u_2 = \varphi$  och får

$$\begin{cases} u_1' = u_2, & u_1(0) = \theta_0 \\ u_2' = -\frac{g}{\ell}\sin(u_1), & u_2(0) = 0 \end{cases}$$

Nu har vi standardformen

$$\begin{cases} \mathbf{u}' = \mathbf{f}(t, \mathbf{u}) \\ \mathbf{u}(0) = \mathbf{u}_0 \end{cases}, \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad \mathbf{f}(t, \mathbf{u}) = \begin{bmatrix} u_2 \\ -\frac{g}{\ell}\sin(u_1) \end{bmatrix}, \quad \mathbf{u}_0 = \begin{bmatrix} \theta_0 \\ 0 \end{bmatrix}$$

Vi beskriver differentialekvationen i MATLAB med funktionen

```
function f=pendel(t,u,g,l)
f=[u(2)
   -g/l*sin(u(1))];
```

Följer lösningskurvorna med `ode45` för några olika begynnelseutslag och ritar en bild som visar lösningarna  $t \mapsto (t, \theta(t))$  och det s.k. *fasporträttet*  $t \mapsto (\theta(t), \dot{\theta}(t))$  för de olika begynnelseutslagen.

```
g=9.81; l=0.1; theta0=[30:20:110]*pi/180;
col=['g','b','m','c','r'];
tspan=linspace(0,1,200);
```

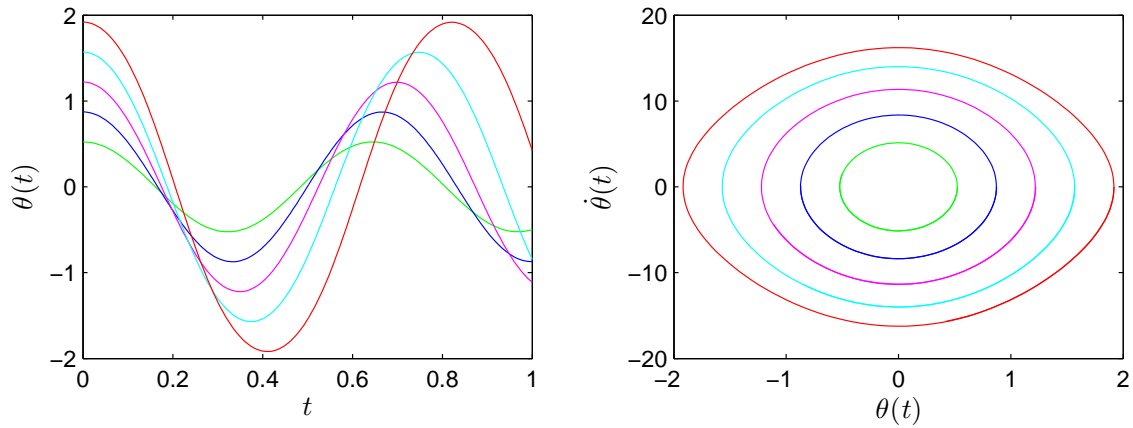
```
for k=1:length(theta0)
    u0=[theta0(k);0];
    [t,U]=ode45(@(t,u)pendel(t,u,g,l),tspan,u0);
    subplot(1,2,1), plot(t,U(:,1),col(k)), hold on
    subplot(1,2,2), plot(U(:,1),U(:,2),col(k)), hold on
end
```



```

subplot(1,2,1), hold off
xlabel('$t$', 'interpreter', 'latex', 'fontsize', 12)
ylabel('$\theta(t)$', 'interpreter', 'latex', 'fontsize', 12),
subplot(1,2,2), hold off
xlabel('$\theta(t)$', 'interpreter', 'latex', 'fontsize', 12)
ylabel('$\dot{\theta}(t)$', 'interpreter', 'latex', 'fontsize', 12)

```



Från figuren ser vi att periodlängden ökar med ökande begynnelseutslag.

**Uppgift 2.** En dämpad matematisk pendel beskrivs av

$$\begin{cases} m\ell \ddot{\theta}(t) = -mg \sin(\theta(t)) - c\ell \dot{\theta}(t), & t \geq 0 \\ \theta(0) = \theta_0, & \dot{\theta}(0) = 0 \end{cases}$$

där  $c$  är dämpningskonstanten. Lös problemet för  $\ell = 0.1$ ,  $m = 0.1$  och  $c = 0.2$  för samma begynnelseutslagsvinklar som i exemplet ovan. Rita upp lösningarna och fasporträtten.

## 8 Redovisning

Denna vecka skall uppgift 1-2 redovisas för handledaren.