

## System av differentialekvationer

### 1 Inledning

Vi skall se på system av differentialekvationer av typen

$$\begin{cases} \mathbf{u}'(t) = \mathbf{f}(t, \mathbf{u}(t)), & a \leq t \leq b \\ \mathbf{u}(a) = \mathbf{u}_a \end{cases}$$

där

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

Detta är precis samma typ vi började med i förra laborationen, fast nu har vi vektorer. De metoder vi såg på för en enda differentialekvation fungerar även för system av ekvationer.

Vi skall också se på högre ordningens differentialekvationer och hur de kan skrivas om som system av första ordningens ekvationer. Dessa system kan vi sedan lösa.

### 2 System av differentialekvationer

Som exempel på ett system av ekvationer tar vi: *Populationsdynamik* – Vi betraktar population av bytesdjur (kaniner) som lever tillsammans med en population rovdjur (rävar). Låt  $u_1(t)$  respektive  $u_2(t)$  beteckna antalet kaniner respektive rävar vid tiden  $t$ . En enkel matematisk modell för populationernas utveckling ges av Volterra-Lotka-ekvationerna:

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

Koefficienterna  $a, b, c, d$  är positiva. Termen  $a u_1(t)$  representerar netto-födelse-dödstalet i en ensam kaninpopulation. Termen  $-c u_2(t)$  är motsvarande för rävarna. Termen  $-b u_1(t) u_2(t)$  är antalet kaniner som blir uppätta per tidsenhet. Termen  $d u_1(t) u_2(t)$  är antalet rävar per tidsenhet som överlever på grund av tillgång på föda. Observera teckenkombinationen i ekvationerna. Vad blir lösningen om populationerna är ensamma ( $b = d = 0$ )?

Vår differentialekvation har formen

$$\begin{cases} \mathbf{u}' = \mathbf{f}(t, \mathbf{u}) \\ \mathbf{u}(0) = \mathbf{u}_0 \end{cases}$$

där

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad \mathbf{f}(t, \mathbf{u}) = \begin{bmatrix} a u_1 - b u_1 u_2 \\ -c u_2 + d u_1 u_2 \end{bmatrix}, \quad \mathbf{u}_0 = \begin{bmatrix} u_1(0) \\ u_2(0) \end{bmatrix}$$

Vi skall använda `ode45` för att beräkna en numerisk lösning.

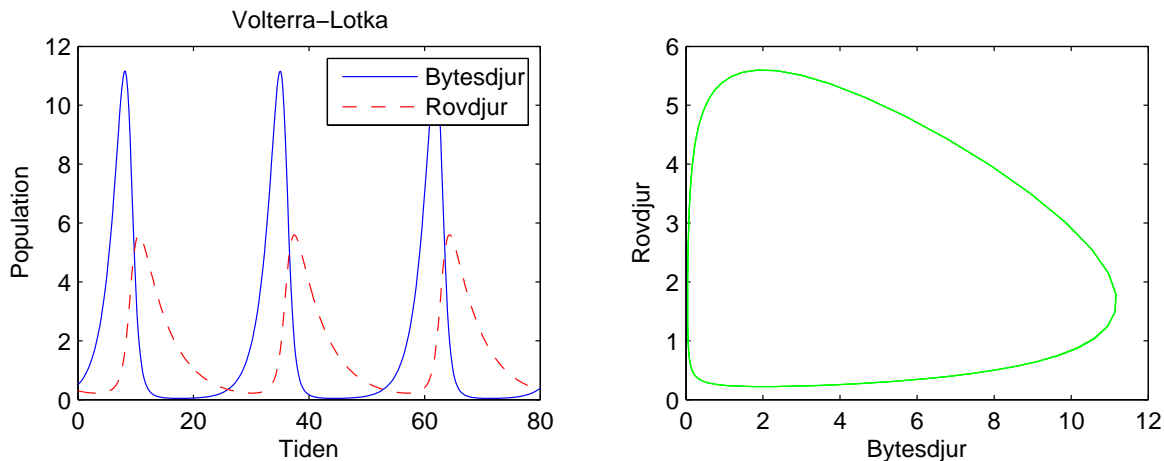
Först beskriver vi högerledet i differentialekvationen med en funktion

```
function f=volterra(t,u)
    a=0.5; b=0.3; c=0.2; d=0.1;
    f=[ a*u(1)-b*u(1)*u(2)
        -c*u(2)+d*u(1)*u(2)];
```

och sedan löser vi med `ode45` och ritar upp enligt

```
>> u0=[0.5;0.3];
>> tspan=linspace(0,80,400);
>> [t,U]=ode45(@volterra,tspan,u0);
>> plot(t,U(:,1),t,U(:,2),'r--')
>> legend('Bytesdjur','Rovdjur')
>> xlabel('Tiden'), ylabel('Population')
>> title('Volterra-Lotka')
```

Vi får bilden nedan till vänster.



Skillnaden mot i förra laborationen är att när vi använder `ode45` blir `U` inte en vektor utan en matris. Vi finner  $u_1(t)$  och  $u_2(t)$  (för olika tidpunkter  $t_n$ ) som `U(:,1)` respektive `U(:,2)`, dvs. första och andra kolonnen i `U`.

I bilden ovan till höger har vi ritat även upp ett s.k. *fasporträtt*, dvs. man ritar antal rovdjur mot antalet bytesdjur.

```
>> plot(U(:,1),U(:,2),'g')
>> xlabel('Bytesdjur'), ylabel('Rovdjur')
```

**Uppgift 1.** Lös Volterra-Lotka-ekvationerna med `ode45`. Ändra koefficienterna till  $a = 0.5$ ,  $b = 0.3$ ,  $c = 0.2$ ,  $d = 0.05$ .

Läs gärna lite mer om Volterra-Lotka-ekvationerna i Stewart avsnitt 9.6. Där ser vi hur man kan modifiera ekvationerna om man vill beskriva hur bytesdjuren kan drabbas vid ett överutnyttjande av resurser (kaninerna får ont om föda)?

Om  $b = 0$ , dvs. bytesdjuren slipper träffa på rovdjuren, så beskriver ekvationen för bytesdjuren en exponentiell tillväxt  $u_1'(t) = au_1(t)$ . Det är rimligt att bl.a. mängden föda och levnadsutrymme

kommer begränsa populationens tillväxt. Därför skulle man kunna modifiera ekvationen med en begränsningsterm, dvs. samma modifiering som ledde till den logistiska ekvationen (Stewart avsnitt 9.4). Vi får ekvationen

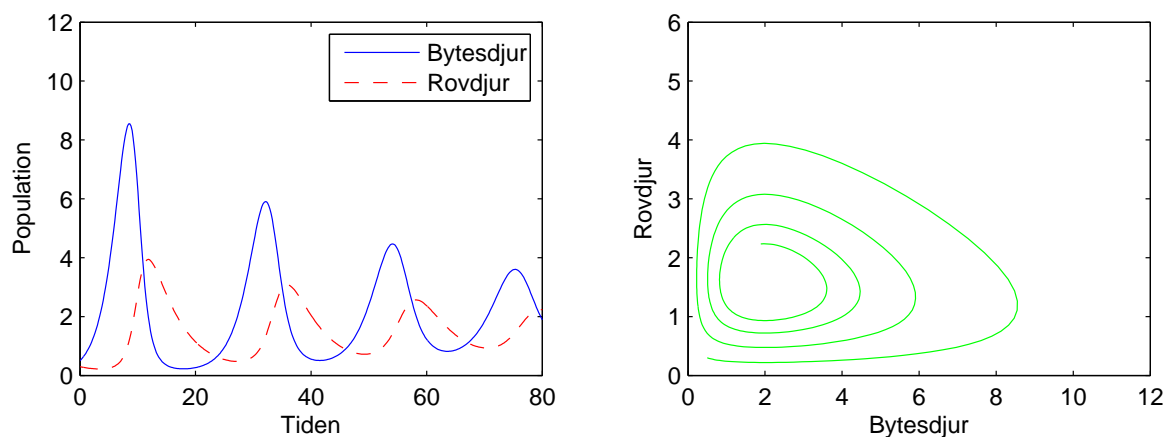
$$u_1'(t) = au_1(t) \left( 1 - \frac{u_1(t)}{M} \right)$$

där  $M$  är en konstant som ger den största mängden individer ekosystemet kan livnära.

Vi har kommit fram till följande modifierade Volterra-Lotka-ekvationer

$$\begin{cases} u_1'(t) = a u_1(t) \left( 1 - \frac{u_1(t)}{M} \right) - b u_1(t) u_2(t) \\ u_2'(t) = -c u_2(t) + d u_1(t) u_2(t) \end{cases}$$

Dessa löser vi för  $M = 30$  och ritar upp resultatet.



Vi ser att vi får en utdämning med tiden. Lösningens banan går i spiral in mot ett jämviktsläge.

### 3 Högre ordningens differentialekvationer

Högre ordningens differentialekvationer kan skrivas om som system av första ordningens ekvationer. Vill vi skriva om t.ex. en andra ordningens differentialekvation

$$u'' = f(t, u, u')$$

så låter vi  $u_1 = u$ ,  $u_2 = u'$  och får systemet

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

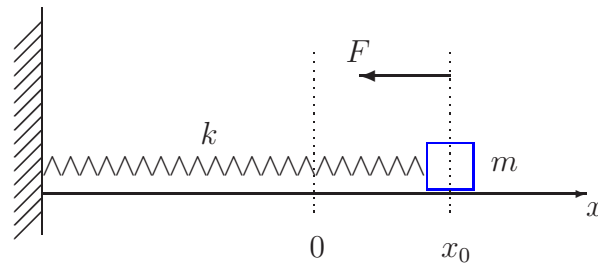
där

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad \mathbf{f}(t, \mathbf{u}) = \begin{bmatrix} u_2 \\ f(t, u_1, u_2) \end{bmatrix}$$

Systemet får lika många ekvationer som ordningen på ursprungliga ekvationen.

Som exempel ser vi på en *svängande fjäder*. Antag att vi har en kloss med massan  $m$  som är fastsatt i en linjär fjäder med fjäderkonstanten  $k$ . Detta innebär att kraften då fjädern sträcks en sträcka  $x$  blir  $-kx$  enligt Hookes lag. Fjäders andra ända är fastsatt i en fix punkt och klossen kan röra sig utan friktion längs en horisontell linje.

Vid tiden  $t = 0$  släpps klossen från vila, på avståndet  $x_0$  från jämviktspunkten, och man vill beräkna den fortsatta rörelsen.



Vi inför ett koordinatsystem enligt figuren ovan med origo där klossen har sitt jämviktsläge. Då  $x$  betecknar klossens läge får vi rörelseekvationen  $m x'' = -kx$  från Newtons andra lag ( $F = ma$ ). Detta är en differentialekvation av 2:a ordningen.

Vidare har vi begynnelsevärdena  $x(0) = x_0$ , läget vid tiden  $t = 0$ , och  $x'(0) = 0$ , klossen i vila vid tiden  $t = 0$ .

Vi har begynnelsevärdesproblemet

$$\begin{cases} x'' = -\frac{k}{m}x \\ x(0) = x_0, \quad x'(0) = 0 \end{cases}$$

Om vi låter  $v = x'$ , dvs. inför hastigheten, kan differentialekvationen med begynnelsevärden skrivas

$$\begin{cases} x' = v, & x(0) = x_0 \\ v' = -\frac{k}{m}x, & v(0) = 0 \end{cases}$$

dvs. som ett system av ekvationer av 1:a ordningen.

För att komma till standardform låter vi  $u_1 = x$  och  $u_2 = v$  och får

$$\begin{cases} u_1' = u_2, & u_1(0) = x_0 \\ u_2' = -\frac{k}{m}u_1, & u_2(0) = 0 \end{cases}$$

Med vektorbeteckningar kan detta skrivas

$$\begin{cases} \mathbf{u}' = \mathbf{f}(t, \mathbf{u}) \\ \mathbf{u}(0) = \mathbf{u}_0 \end{cases}, \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad \mathbf{f}(t, \mathbf{u}) = \begin{bmatrix} u_2 \\ -\frac{k}{m}u_1 \end{bmatrix}, \quad \mathbf{u}_0 = \begin{bmatrix} x_0 \\ 0 \end{bmatrix}$$

Vi tar t.ex.  $m = 0.1$  kg,  $k = 0.12$  N/m och  $x_0 = 0.1$  m.

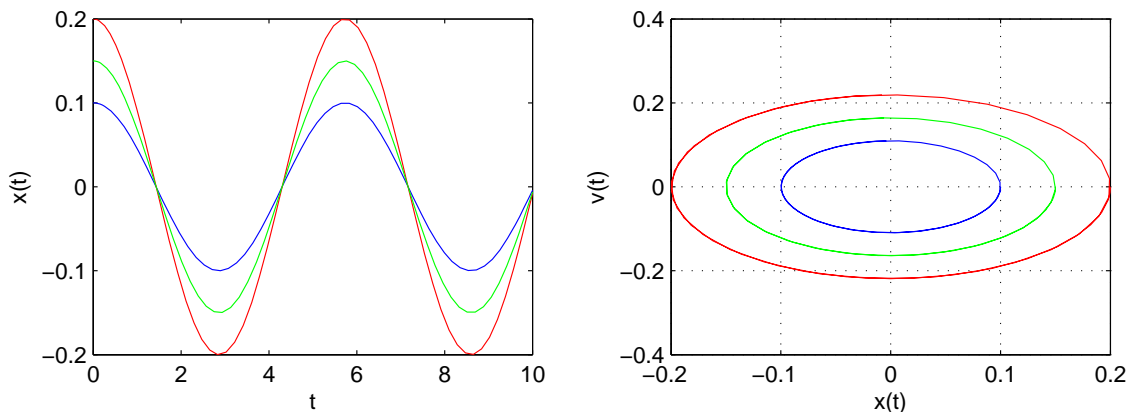
Vi beräknar lösning med `ode45` och ritar en bild som visar läget  $x(t)$  med

```
>> m=0.1; k=0.12; x0=0.1; v0=0; tspan=[0 10]; u0=[x0;v0];
>> f=@(t,u) [u(2);-k*u(1)/m];
>> [t,U]=ode45(f,tspan,u0);
>> plot(t,U(:,1),'b')
>> xlabel('t'), ylabel('x(t)')
```

och fasporträttet  $(x(t), v(t))$  med

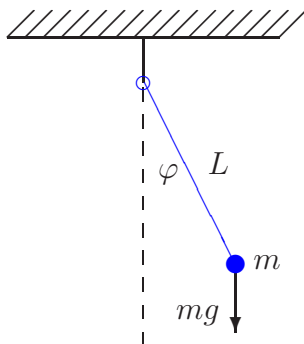
```
>> plot(U(:,1),U(:,2),'b')
>> xlabel('x(t)'), xlabel('v(t)')
```

I bilden har vi även ritat ut lösningar för några andra begynnelsevärden.



Ekvationen för den svängande fjädern kan vi lösa analytiskt (räkna ut en formel för lösning). Vi avslutar med att se på en uppgift med en differentialekvation som inte har någon analytisk lösning.

**Uppgift 2.** Den matematiska pendeln. 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

$$mL \ddot{\varphi}(t) = -mg \sin(\varphi(t))$$

Vi vill bestämma lösningen för olika begynnelseutslag  $\varphi_0$ , dvs.  $\varphi(0) = \varphi_0$ , då vi släpper pendeln från vila, dvs.  $\dot{\varphi}(0) = 0$ . Tag  $L = 0.1$  m och begynnelseutslagen  $\varphi_0 = 30^\circ, 45^\circ, 60^\circ$ . Använd ode45. Tänk på att du måste omvandla grader till radianer.