1 Inledning

Vi skall se på numerisk lösning av ordinära differentialekvationer (ODE), först begynnelsevärdesproblem och sedan randvärdesproblem. Problemen i sig är viktiga i tekniska tillämpningar, men även som grund för att utveckla metoder för partiella differentialekvationer (PDE).

2 Begynnelsevärdesproblem

Vi skall se på metoder för att lösa *begynnelsevärdesproblem* för första ordningens differentialekvation

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

där f en given funktion och u_a en given konstant.

Dessa metoder fungerar lika bra på *system* av första ordningens differentialekvationer

$$\begin{cases} \boldsymbol{u}' = \boldsymbol{f}(t,\boldsymbol{u}),\ a \le t \le b \\ \boldsymbol{u}(a) = \boldsymbol{u}_a \end{cases}$$

där $\boldsymbol{f}$, $\boldsymbol{u}$ och $\boldsymbol{u}_a$ är vektorer.

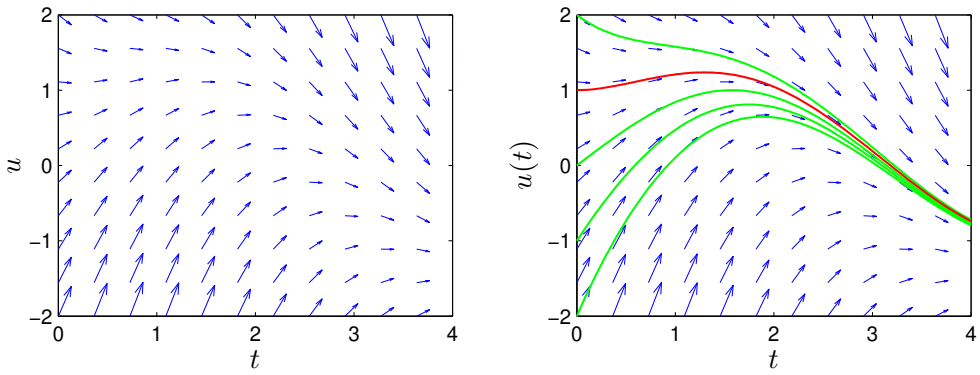
Högre ordningens differentialekvationer skrivas om som system av första ordningens ekvationer.

Som exempel på en första ordningens ekvation tar vi problemet

$$\begin{cases} u'(t) = -u(t) + \sin(t) + \cos(t),\ 0 \le t \le 4 \\ u(0) = u_0 \end{cases}$$

med analytisk (exakt) lösning $u(t) = \sin(t) + u_0 \exp(-t)$.

I den vänstra figuren nedan har vi ritat *riktningsfältet* och i den högra lösningskurvorna för några olika värden på u_0.



Vi ser hur lösningskurvorna följer riktningsfältet. Beräkningsmetoderna som vi skall se på bygger på idén att försöka följa riktningsfältet.

2.1 Rita riktningsfält

Vi skall se hur man kan rita riktningsfält till differentialekvationen

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

Ett riktningsfält består i en samling punkter (t_i, u_j) i tu-planet (ett gitter) där vi i varje punkt ritar en liten pil i den riktning som en lösningskurva $t \mapsto (t, u(t))$ genom punkten har precis i punkten, dvs. en pil i riktningen $(1, u'(t_i)) = (1, f(t_i, u_j))$. Vi skalar pilarna så att vi får en tydlig bild. (För korta pilar och inget syns, för långa pilar och bilden blir grötig.)

Som exempel ritar vi riktningsfältet till det inledande begynnelsevärdesproblemet

$$\begin{cases} u'(t) = -u(t) + \sin(t) + \cos(t),\ 0 \le t \le 4 \\ u(0) = u_0 \end{cases}$$

Först bildar vi ett gitter (grid) med `meshgrid`. I matriserna `T` och `U` kommer vi ha gittrets koordinater. Sedan bildar vi en matris `DT` med ettor, som är första koordinaterna i pilarna, och en matris `DU`, som är andra koordinaterna i pilarna. Slutligen ritar vi ut pilarna med `quiver`, där talet 0.9 är en skalfaktor (lagom långa pilar).

```
>> f=@(t,u)-u+sin(t)+cos(t);
>> a=0; b=4;
>> t=linspace(a,b,20); u=linspace(-2,2,20);
>> [T,U]=meshgrid(t,u);
>> DT=ones(size(T)); DU=f(T,U);
>> quiver(T,U,DT,DU,0.9)
```

Som resultat får vi vänstra figuren på första sidan.

Uppgift 1. Vi skall se på följande begynnelsevärdesproblem i några uppgifter

$$\begin{cases} u' = \cos(3t) - \sin(5t)u,\ 0 \le t \le 5 \\ u(0) = 2 \end{cases}$$

Rita riktningsfältet över området $0 \le t \le 5,\ 0 \le u \le 3$. Välj lagom långa pilar.

2.2 Differensmetoder

Vi skall approximera lösningen $u(t)$ till differentialekvationen på ett nät

$$a = t_0 < t_1 < t_2 < \cdots < t_n < t_{n+1} < \cdots < t_{N-1} < t_N = b$$

där $t_n = a + nh$ för $n = 0, 1, \cdots, N$, med steglängden $h = \frac{b-a}{N}$.

Låter vi u_n beteckna approximationen av $u(t_n)$ och ersätter $u'(t_n)$ med en *framåt differenskvot* $D_+u(t_n)$ så gäller

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

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

Detta ger **Eulers framåtmetod**

$$u_{n+1} = u_n + hf(t_n, u_n)$$

Metoden är *explicit* eftersom alla värden i högerledet är kända. För att ta ett steg till nästa tidsnivå räcker det att sätta in kända värden i högerledet. Med MATLAB kan vi göra det så här enkelt

```
>> f=@(t,u)-u+sin(t)+cos(t);
>> a=0; b=4; ua=1;
>> N=10; h=(b-a)/N;
>> t=linspace(a,b,N+1); u=zeros(size(t));
>> u(1)=ua;
>> for n=1:N
       u(n+1)=u(n)+h*f(t(n),u(n));
   end
>> plot(t,u)
```

Ersätter vi $u(t_{n+1})$ med en *bakåt differenskvot* $D_-u(t_{n+1})$ får vi

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

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

Detta ger **Eulers bakåtmetod**

$$u_{n+1} = u_n + hf(t_{n+1}, u_{n+1})$$

Metoden är *implicit* eftersom u_{n+1}, som är obekant, finns med även i f. För att ta ett steg måste man normalt lösa en icke-linjär ekvation

$$g(w) = w - u_n - hf(t_{n+1}, w) = 0$$

med t.ex. Newtons metod.

Vi kan också *integrera* differentialekvationen från t_n till $t_{n+1} = t_n + h$ och får

$$u(t_{n+1}) = u(t_n) + \int_{t_n}^{t_{n+1}} f(t, u(t))\, dt$$

och kan sedan approximera integralen på olika sätt.

Med *vänster* och *höger rektangelregel* får vi Eulers framåt- respektive bakåtmetod och med *trapetsregeln* får vi den implicita **trapetsmetoden**

$$u_{n+1} = u_n + \frac{h}{2}(f(t_n, u_n) + f(t_{n+1}, u_{n+1}))$$

Om vi i denna metod ersätter u_{n+1} i högerledet med en Euler framåt approximation får vi **Heuns metod**

$$u_{n+1} = u_n + \frac{h}{2}(f(t_n, u_n) + f(t_n + h, u_n + hf(t_n, u_n)))$$

som är en explicit metod.

Vi kan enkelt använda Heuns metod enligt

```
>> f=@(t,u)-u+sin(t)+cos(t);
>> a=0; b=4; ua=1;
>> N=10; h=(b-a)/N;
>> t=linspace(a,b,N+1); u=zeros(size(t));
>> u(1)=ua;
>> for n=1:N
       k1=f(t(n),u(n)); k2=f(t(n+1),u(n)+h*k1);
       u(n+1)=u(n)+h/2*(k1+k2);
   end
>> plot(t,u)
```

Ett begynnelsevärdesproblem som beskriver förlopp eller processer vilka utspelas under tidsintervall av mycket olika storleksordning kallas för *styvt*. T.ex. inom kemisk reaktionsteknik är styva problem vanliga. För styva problem måste implicita metoder används, dvs. av typen Euler bakåtmetoden eller trapetsmetoden. Explicita metoder, som Euler framåtmetoden eller Heuns metod, blir mycket ineffektiva.

Uppgift 2. Vi ser återigen på begynnelsevärdesproblemet

$$\begin{cases} u' = \cos(3t) - \sin(5t)u, \ 0 \le t \le 5 \\ u(0) = 2 \end{cases}$$

Lös problemet med Euler framåtmetoden. Rita en graf av lösningen i en figur som dessutom innehåller riktningsfältet. Tag tillräckligt många tidssteg så att vi får en noggrann lösning.

2.3 Konvergens

Utgående från begynnelsevärdet försöker vi följa riktningsfältet med korta steg för att få en noggrann approximation.

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

För Euler metoderna gäller att om vi halverar steglängden så halveras felet i approximationen. För trapetsmetoden och Heuns metod gäller att om vi halverar steglängden så delas felet i approximationen med fyra.

Vi jämför Euler framåtmetoden med Heuns metod på exemplet från inledningen. Eftersom vi har en formel för den exakta lösningen kan vi ta differensen mellan den och approximationerna för att se hur pass bra de är. Vi ser vid tiden $t = 4$ först för steglängden $h = 0.4$ och sedan successivt halverade steglängder.

	Euler framåt metod		Heuns metod	
h	u_N	$u_N - u(4)$	u_N	$u_N - u(4)$
0.4	$-0.7674846249\ldots$	$0.0289977684\ldots$	$-0.7253391181\ldots$	$-0.0131477382\ldots$
0.2	$-0.7514168883\ldots$	$0.0129300319\ldots$	$-0.7351256618\ldots$	$-0.0033611945\ldots$
0.1	$-0.7446007637\ldots$	$0.0061139073\ldots$	$-0.7376516492\ldots$	$-0.0008352072\ldots$
0.05	$-0.7414601596\ldots$	$0.0029733032\ldots$	$-0.7382793420\ldots$	$-0.0002075143\ldots$
0.025	$-0.7399530614\ldots$	$0.0014662050\ldots$	$-0.7384351731\ldots$	$-0.0000516832\ldots$

Heuns metod är betydligt mer noggrann och mer effektiv.

2.4 Färdiga program i MATLAB

I MATLAB finns flera olika funktioner för lösning av begynnelsevärdesproblem av olika typ, t.ex. finner vi `ode45` som används enligt

`[t,U]=ode45(fun,tspan,u0)` `[t,U]=ode45(fun,tspan,u0,opts)`

där `fun` är en funktion som beskriver differentialekvationens högerled, `tspan` är en vektor som anger tidsintervallet och `u0` ger begynnelsevärdet för lösningen vid tiden `tspan(1)`. Funktionen `ode45` producerar som utdata dels `t`, en vektor som innehåller tidpunkter och dels `U`, en vektor eller matris som innehåller lösningen för de olika tidpunkterna.

Alternativet med `opts` använder vi då vi t.ex. vill ange hur noggrant lösningen skall beräknas. Det finns också möjlighet till avbrottshantering. Man skapar vektorn `opts` med funktionen `odeset`, se hjälptexten.

En annan funktion i MATLAB är `ode15s` avsedd för styva begynnelsevärdesproblem.

Med funktionen `ode45` i MATLAB kan vi beräkna en numerisk lösning till vårt inledande exempel för t.ex. $u(0) = 1$ enligt

```
>> f=@(t,u)-u+sin(t)+cos(t);
>> a=0; b=4; ua=1;
>> [t,u]=ode45(f,[a b],ua);
>> plot(t,u)
```

Uppgift 3. För en sista gång ser vi på begynnelsevärdesproblemet

$$\begin{cases} u' = \cos(3t) - \sin(5t)u, \ 0 \le t \le 5 \\ u(0) = 2 \end{cases}$$

Lös problemet med `ode45`. Rita en graf av lösningen i en figur som även innehåller riktningsfältet och en approximation beräknad med Euler framåtmetoden. Använd olika färger på graferna.

2.5 System av första ordningens differentialekvationer

Med MATLAB kan vi lika lätt lösa system av differentialekvationer

$$\begin{cases} \boldsymbol{u}'(t) = \boldsymbol{f}(t, \boldsymbol{u}(t)), \quad a \le t \le b \\ \boldsymbol{u}(a) = \boldsymbol{u}_a \end{cases}$$

där

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

Skillnaden är att nu blir `U` en matris och vi finner $u_k(t)$ (för olika tidpunkter t_i) som kolonn `U(:,k)`, dvs. kolonn nr k i `U`.

Som exempel tar vi följande system med två ekvationer

$$\begin{cases} u_1'(t) = 2u_1(t) - u_1(t)u_2(t), & u_1(0) = 0.1 \\ u_2'(t) = u_2(t) + 0.4u_1(t)u_2(t) - u_2(t)^2, & u_2(0) = 0.1 \end{cases}$$

Båda ekvationerna beror av både $u_1(t)$ och $u_2(t)$ så vi måste lösa dem samtidigt.

Vi har

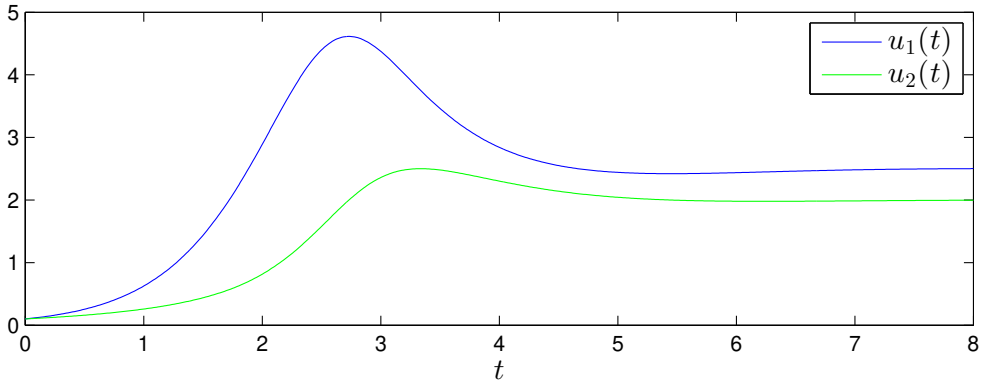
$$\begin{cases} \boldsymbol{u}'(t) = \boldsymbol{f}(t, \boldsymbol{u}(t)), \quad 0 \le t \le T \\ \boldsymbol{u}(0) = \boldsymbol{u}_0 \end{cases}$$

med

$$\boldsymbol{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}, \quad \boldsymbol{f}(t, \boldsymbol{u}) = \begin{bmatrix} 2u_1 - u_1 u_2 \\ u_2 + 0.4 u_1 u_2 - u_2^2 \end{bmatrix}, \quad \boldsymbol{u}_0 = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$$

Vi beskriver högerledet i differentialekvationen, löser med `ode45` och ritar upp lösningen enligt

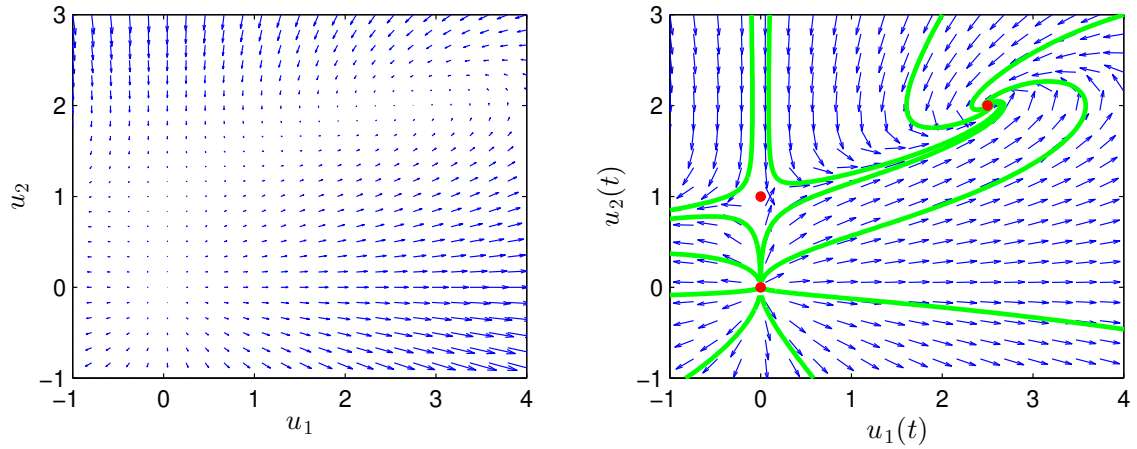
```
>> f1=@(u1,u2)2*u1-u1.*u2;
>> f2=@(u1,u2)u2+0.4*u1.*u2-u2.^2;
>> f=@(t,u)[f1(u(1),u(2))
            f2(u(1),u(2))];
>> T=8; tspan=linspace(0,T,200);
>> u0=[0.1;0.1];
>> [t,U]=ode45(f,tspan,u0);
>> plot(t,U(:,1),'b',t,U(:,2),'g')
```



Vi ritade graferna av $u_1(t)$ och $u_2(t)$ mot t. Nu skall vi rita riktningsfält och s.k. *fasporträtt*, där vi ritar $u_1(t)$ mot $u_2(t)$. Riktningsfältet ritar vi enligt

```
>> u1=linspace(-1,4,25); u2=linspace(-1,3,25);
>> [U1,U2]=meshgrid(u1,u2);
>> s=1.5;          % s - skalfaktor som förlänger pilarna.
>> quiver(U1,U2,f1(U1,U2),f2(U1,U2),s)
```

och får bilden nedan till vänster.



Vi lägger till en lösningsbana med `ode45` enligt

```
>> u0=[0.1;0.1];
>> [t,U]=ode45(f,tspan,u0);
>> plot(U(:,1),U(:,2),'g','LineWidth',2)
```

I figuren ovan till höger har vi löst för några ytterligare startvärden. Vi ser hur lösningarna följer fältet. (För tydlighets skull har vi även sett till att pilarna blev lika långa. När vi sitter vid en dator och kan se på en stor figur behövs inte det.)

Eftersom detta system är autonomt, dvs. har formen $\boldsymbol{u}'(t) = \boldsymbol{f}(\boldsymbol{u}(t))$, så kan vi söka efter jämviktslösningar. (Ett autonomt systems högerled beror inte direkt av t, bara indirekt via $\boldsymbol{u}(t)$.) För jämviktslösningar gäller att $u_1'(t) = 0$ och $u_2'(t) = 0$, dvs. de förändras inte med tiden. I riktningsfältet syns dessa som punkter där riktningsvektorn $\boldsymbol{u}' = (u_1', u_2') = (0, 0)$. Vi ser om det finns några sådana punkter genom att lösa $\boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{0}$, som för vårt problem kan skrivas

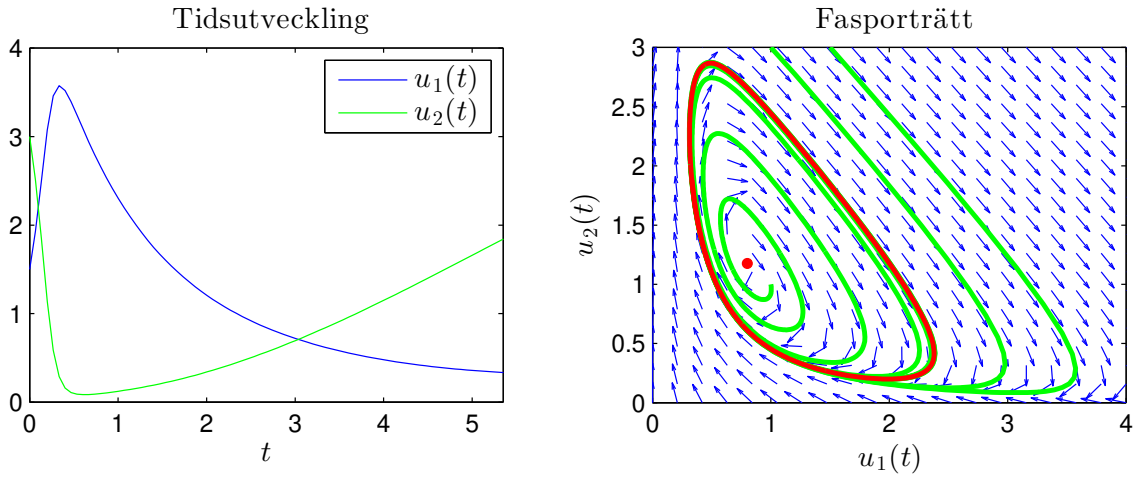
$$\begin{cases} 2u_1 - u_1u_2 = 0 \\ u_2 + 0.4u_1u_2 - u_2^2 = 0 \end{cases}$$

Vi skall lite senare se på numeriska metoder för att lösa ekvationer av den här typen, men denna enkla ekvation kan vi lösa för hand. Den första ekvationen är uppfylld om $u_1 = 0$. Insättning av $u_1 = 0$ i den andra ger $u_2 - u_2^2 = 0$, med lösningarna $u_2 = 0$ och $u_2 = 1$. Därmed är båda ekvationerna uppfyllda samtidigt om $u_1 = u_2 = 0$ och om $u_1 = 0, u_2 = 1$. Alltså är $(u_1, u_2) = (0, 0)$ och $(u_1, u_2) = (0, 1)$ jämviktspunkter. Om å andra sidan $u_1 \neq 0$ ser vi från första ekvationen att $u_2 = 2$. Insättning av $u_2 = 2$ i andra ekvationen ger att $u_1 = 2.5$. Alltså är även $(u_1, u_2) = (2.5, 2)$ en jämviktspunkt. Några fler lösningar finns inte. De tre jämviktspunkterna ser vi i figuren ovan. Uppenbarligen är $(0, 0)$ och $(0, 1)$ instabila jämviktspunkter, medan $(2.5, 2)$ är stabil. Vi har markerat jämviktspunkter med röda prickar.

Låt oss ta ytterligare ett exempel, även detta ett autonomt system.

$$\begin{cases} u_1'(t) = -u_1(t) + 0.04u_2(t) + u_1(t)^2u_2(t) \\ u_2'(t) = 0.8 - 0.04u_2(t) - u_1(t)^2u_2(t) \end{cases}$$

Vi löser på motsvarande sätt. Nedan till vänster $u_1(t)$ och $u_2(t)$ som funktioner av t och till höger fasporträtt med riktningsfält.



Vi har en instabil jämviktspunkt i $(0.8, 1.1765)$, markerad som röd prick, och en stabil periodisk lösning (en gränscykel) som vi ritat ut med rött (den slutna röda öglan).

Uppgift 4. Betrakta följande system

$$\begin{cases} u_1'(t) = \cos(1 + u_1(t) - u_2(t)^2) + 0.3 \\ u_2'(t) = \sin(0.3 + u_1(t) + u_2(t)) - 0.4u_1(t) \end{cases}$$

Lös problemet med `ode45` för begynnelsevärdena $u_1(0) = 1.5$ och $u_2(0) = 3$. Rita upp lösningen som funktion av tiden. Rita ett riktningsfält för ekvationen över området $0 \leq u_1 \leq 3, 0 \leq u_2 \leq 3$. Hur många jämviktspunkter ser du i området? Rita in i riktningsfältet banan för lösningen du just beräknade. Lös också för några andra begynnelsevärden och rita upp även de banorna.

2.6 Högre ordningens differentialekvationer

Högre ordningens differentialekvationer

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

kan skrivas om som system av första ordningen som sedan lösas med de metoder vi sett på.

Vi låter $u_1 = u,\ u_2 = u',\ u_3 = u'', \cdots$, och kan då skriva ekvationen som ett system av första ordningens ekvationer

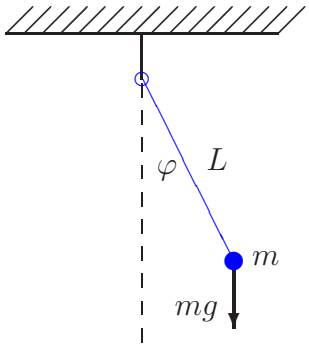
$$\boldsymbol{u}' = \boldsymbol{f}(t, \boldsymbol{u})$$

där

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}, \quad \boldsymbol{f}(t, \boldsymbol{u}) = \begin{bmatrix} u_2 \\ \vdots \\ f(t, u_1, \cdots, u_n) \end{bmatrix}$$

Systemet får lika många ekvationer som ordningen på ursprungliga ekvationen.

Som exempel tar vi 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 φ_0 då vi släpper pendeln från vila, vilket ger begynnelsevillkoren

$$\varphi(0) = \varphi_0, \quad \dot{\varphi}(0) = 0$$

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

$$\begin{cases} \dot{\varphi} = \omega, & \varphi(0) = \varphi_0 \\ \dot{\omega} = -\frac{g}{L}\sin(\varphi), & \omega(0) = 0 \end{cases}$$

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

$$\begin{cases} u_1' = u_2, & u_1(0) = \varphi_0 \\ u_2' = -\frac{g}{L}\sin(u_1), & u_2(0) = 0 \end{cases}$$

Nu har vi standardformen

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

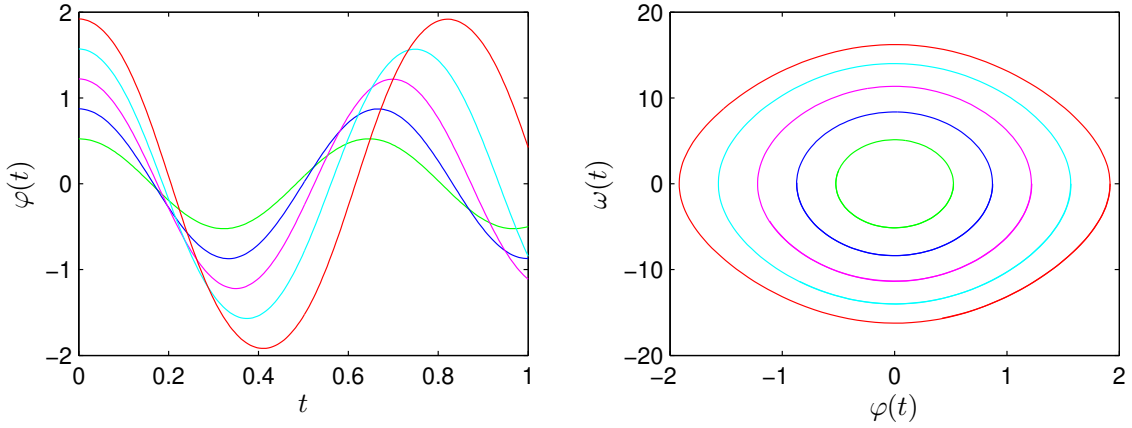
med

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad \boldsymbol{f}(t, \boldsymbol{u}) = \begin{bmatrix} u_2 \\ -\frac{g}{L}\sin(u_1) \end{bmatrix}, \quad \boldsymbol{u}_0 = \begin{bmatrix} \varphi_0 \\ 0 \end{bmatrix}$$

Vi beskriver differentialekvationens högerled i MATLAB och följer lösningskurvorna med `ode45` för några olika begynnelseutslag och ritar en bild som visar lösningarna $(t, \varphi(t))$ och fasporträtten $(\varphi(t), \omega(t))$ för de olika begynnelseutslagen.

```
>> f=@(t,u,g,L)[u(2); -g/L*sin(u(1))];
>> g=9.81; L=0.1;
>> tspan=linspace(0,1,200); phi0=30*pi/180;
>> u0=[phi0;0];
>> [t,U]=ode45(@(t,u)f(t,u,g,L),tspan,u0);
>> subplot(1,2,1), plot(t,U(:,1),'g')
>> subplot(1,2,2), plot(U(:,1),U(:,2),'g')
```

Vi ritar med olika färger så att vi kan se vilka lösningskurvor som hör ihop med vilka fasporträtt.



Från figuren ser vi att periodlängden ökar med ökande begynnelseutslag, något vi såg redan i första laborationen.

Uppgift 5. Låt $L = 0.1$ m och tag begynnelseutslagen $\varphi_0 = 10°, 30°, \cdots, 170°$. Beräkna en approximation av periodlängden T för de olika begynnelseutslagen med hjälp av `ode45`. Använd händelsehantering, se nedan eller läs i hjälptexterna (sök på `odeset`) hur man gör. Rita en graf av T som funktion av φ_0 och jämför med resultatet från uppgift 5, laboration 1.

Men kan låta `ode45` hantera händelser som kan beskrivas som nollställen till funktioner som beror av lösningen till differentialekvationen. Med `odeset` anger vi vilken funktion som beskriver händelsen och detta förmedlas till `ode45` enligt

```
opts=odeset('Events',@funevent)
[t,U]=ode45(fun,tspan,u0,opts)
```

Strukturen på händelsefunktionen är

```
function [value,isterminal,direction]=funevent(t,U)
value=
isterminal=
direction=
```

där `value` skall ange funktionsvärdet som skall övervakas.

Variabeln `isterminal` skall ges värdet 1 om `ode45` skall stoppa då händelsen inträffar, annars skall den ges värdet `0`, och variabeln `direction` skall ges värdet `+1 (-1)` om bara stigande (avtagande) mot noll skall beaktas, annars skall den ges värdet `0`.

T.ex. med `value=U(1), isterminal=1, direction=0` stoppar `ode45` då första komponenten i `U` blir noll för första gången.

Uppgift 6. En dämpad matematisk pendel beskrivs av

$$\begin{cases} mL\,\ddot{\varphi}(t) = -mg\sin(\varphi(t)) - cL\,\dot{\varphi}(t), \quad t \geq 0 \\ \varphi(0) = \varphi_0, \quad \dot{\varphi}(0) = 0 \end{cases}$$

där c är dämpningskonstanten.

Lös problemet för $L = 0.1$, $m = 0.1$ och $c = 0.2$ och några olika begynnelseutslagsvinklar.

3 Randvärdesproblem

Vi skall se på metoder för att lösa *randvärdesproblem* för ordinär differentialekvation av typen

$$\begin{cases} u''(x) = f(x, u, u'),\ a \leq x \leq b \\ u(a) = g_a,\ u(b) = g_b \end{cases}$$

där g_a och g_b är givna värden.

3.1 Differensmetoder

För att lösa problemet gör vi en likformig indelning av intervallet $a \leq x \leq b$ i ett nät

$$a = x_0 < x_1 < \cdots < x_{i-1} < x_i < x_{i+1} < \cdots < x_n < x_{n+1} = b$$

där $x_i = a + ih, i = 0, 1, \cdots, n+1$ med $h = \frac{b-a}{n+1}$.

Vi ersätter u'' med en centraldifferens D_+D_-u och i högerledet ersätter vi u' med differensapproximationen D_0u. Då får vi andra ordningens approximation av alla derivator.

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = f(x_i, u_i, \frac{u_{i+1} - u_{i-1}}{2h}),\ i = 1, 2, \cdots, n$$

Detta resulterar i antingen ett linjärt ekvationssystem $\boldsymbol{Au} = \boldsymbol{b}$ eller ett icke-linjärt ekvationssystem $\boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{0}$. Vi skall i senare laborationer se på hur man löser sådana system.

Som exempel tar vi: *Stationär värmeledning* i en isolerad stav med en värmekälla

Temperaturen $u(x)$ beskrivs av randvärdesproblemet

$$\begin{cases} -\kappa u''(x) = f(x),\ 0 \le x \le L \\ u(0) = g_0,\ u(L) = g_L \end{cases}$$

där g_0 och g_L är temperaturen i stavens ändpunkter, κ är stavens termiska diffusivitet och $f(x)$ är värmekällan.

Inför nätet $x_i = ih$, $i = 0, 1, \cdots, n+1$ med $h = \frac{L}{n+1}$ på $0 \le x \le L$ och låt u_i beteckna approximationer av $u(x_i)$, $i = 0, 1, \cdots, n+1$.

Ersätt $u''(x_i)$ med centraldifferenskvoten

$$D_+D_-u(x_i) = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$$

i de inre nätpunkterna $x_i, i = 1, \cdots, n$, och vi får

$$\kappa(-u_{i+1} + 2u_i - u_{i-1}) = h^2 f(x_i) \quad i = 1, 2, \cdots, n$$

Med matriser kan detta skrivas $\boldsymbol{Au} = \boldsymbol{b}$ med

$$\boldsymbol{A} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} \frac{h^2}{\kappa} f(x_1) + g_0 \\ \frac{h^2}{\kappa} f(x_2) \\ \vdots \\ \frac{h^2}{\kappa} f(x_{n-1}) \\ \frac{h^2}{\kappa} f(x_n) + g_L \end{bmatrix}$$

Matrisen $\boldsymbol{A}$ är exempel på en *gles matris*, det är stora matriser (många obekanta) med få nollskilda element (få kopplingar). Vår matris kallas även för en *bandmatris* eller tridiagonal matris. Glesa matriser bildar vi med `sparse` (allmän gleshetsstuktur) och `spdiags` (bandstruktur).

För vårt exempel bildar vi en vektorer fylld med 1:or, därefter placerar vi in denna vektor (multiplicerad med 2 respektive -1) som diagonaler med `spdiags` enligt

```
>> n=30; ett=ones(n,1);
>> A=spdiags([-ett 2*ett -ett],[-1 0 1],n,n);
```

Diagonalerna sätts ihop som kolonner i en matris med `[-ett 2*ett -ett]`, de måste därför vara lika långa. Kolonnerna placeras som diagonaler i ordningen som ges av vektorn `[-1 0 1]` och det hela skall bli en $n \times n$-matris, därav `n,n` allra sist.

När vi har bildat högerledsvektorn $\boldsymbol{b}$ löser vi med backslash (\). När matrisen är lagrad som gles matris känner MATLAB av att det och lösningen av ekvationssystemet görs effektivt med metoder som tar vara på gleshetsstrukturen.

Uppgift 7. Låt $\kappa = 2$, $L = 1$, $g_0 = 40$, $g_L = 60$ samt $f(x) = 200\exp(-(x - \frac{L}{2})^2)$. Lös värmeledningsproblemet och rita upp lösningen. Tag $n = 30$.

3.2 Inskjutningsmetoden

Ett annat sätt att lösa randvärdesproblem är att skriva om dem som system av första ordningen

$$\begin{cases} \boldsymbol{u}' = \boldsymbol{f}(x, \boldsymbol{u}), \ a \le x \le b \\ \boldsymbol{g}(\boldsymbol{u}(a), \boldsymbol{u}(b)) = \boldsymbol{0} \end{cases} \tag{1}$$

Betrakta motsvarande begynnelsevärdesproblem

$$\begin{cases} \boldsymbol{u}' = \boldsymbol{f}(x, \boldsymbol{u}), \ a \le x \le b \\ \boldsymbol{u}(a) = \boldsymbol{s} \end{cases} \tag{2}$$

Om vi betecknar lösningen till (2) med $\boldsymbol{u}(x, \boldsymbol{s})$ så är denna en lösning till randvärdesproblemet (1) om

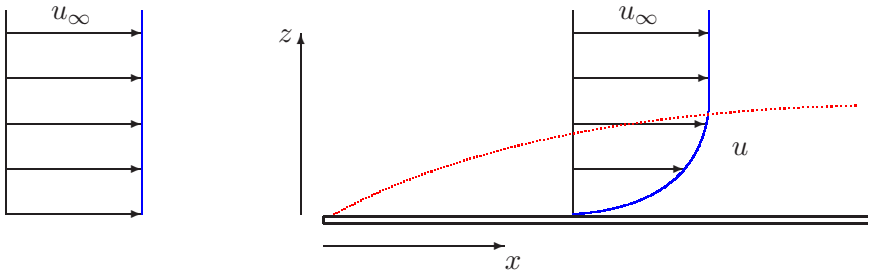
$$\boldsymbol{q}(\boldsymbol{s}) = \boldsymbol{g}(\boldsymbol{s}, \boldsymbol{u}(b, \boldsymbol{s})) = \boldsymbol{0}$$

Detta är ett ekvationssystem i $\boldsymbol{s}$ (linjärt eller icke-linjärt beroende på $\boldsymbol{f}$ och $\boldsymbol{g}$), med lika många lösningar som randvärdesproblemet (1).

En metod för lösning av (1), den s.k. **inskjutningsmetoden**, består helt enkelt i att man löser ekvationen $\boldsymbol{q}(\boldsymbol{s}) = \boldsymbol{0}$. (Varje beräkning av $\boldsymbol{q}$ kräver att man löser (2).)

Uppgift 8. Lös värmeledningsproblemet från uppgift 7 med inskjutningsmetoden.

Som exempel ser vi på: *Gränsskikt* – En tunn platta är placerad i, och parallell med, en strömmande vätska. Vätskan strömmar med hastigheten u_∞ och har viskositet ν. Vi skall bestämma hur u och w, hastighetskomponenterna i x- respektive z-riktningen, varierar i området nära plattan.



Rörelse- och kontinuitetsekvationerna är (som vi hämtar från strömningsläran)

$$\begin{cases} u\frac{\partial u}{\partial x} + w\frac{\partial u}{\partial z} = \nu\frac{\partial^2 u}{\partial z^2} \\ \frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0 \\ u(x,0) = w(x,0) = 0, \ u(x,\infty) = u_\infty \end{cases}$$

Genom att införa en dimensionslös koordinat $\eta = z\sqrt{\frac{u_\infty}{\nu x}}$ och strömfunktion $\psi = \sqrt{\nu x u_\infty} f(\eta)$ sådan att

$$u = \frac{\partial \psi}{\partial z} \ \text{ och } \ w = -\frac{\partial \psi}{\partial x}$$

reduceras rörelse- och kontinuitetsekvationerna till ett randvärdesproblem för en ordinär differentialekvation för strömfunktionen $f(\eta)$

$$\begin{cases} f''' + \frac{1}{2} f f'' = 0 \\ f(0) = f'(0) = 0, \ f'(\infty) = 1 \end{cases} \tag{3}$$

Från ekvationerna för strömfunktionen kan man härleda följande formler för de dimensionslösa hastigheterna U och W:

$$U = \frac{u}{u_\infty} = f', W = \frac{w}{\sqrt{\frac{\nu u_\infty}{x}}} = \frac{1}{2}(\eta f' - f)$$

Vi kan alltså lösa randvärdesproblemet och förutsäga hastigheterna U och W.

Om vi inför variablerna $u_1 = f, u_2 = f'$ och $u_3 = f''$ så kan vi skriva om vår ekvation (3) som ett första ordningens system

$$\begin{cases} u_1' = u_2 \\ u_2' = u_3 \\ u_3' = -\frac{1}{2}u_1 u_3 \\ u_1(0) = u_2(0) = 0,\ u_2(\infty) = 1 \end{cases} \tag{4}$$

Vi ersätter villkoret $u_2(\infty) = 1$ med $u_2(\eta_{\max}) = 1$, där vi tagit $\eta_{\max}$ så stort att lösningen inte förändras mycket om vi tar $\eta_{\max}$ större. ($\eta_{max} = 10$ visar sig duga.)

Detta rättfärdigas fysikaliskt av att hastigheten u är i stort sett u_∞ långt ifrån plattan.

För det till (4) motsvarande begynnelsevärdesproblemet vet vi att $u_1(0) = u_2(0) = 0$. Men vi vet inte vad $u_3(0)$ är, detta värde måsta vi ansätta.

Låt $u_i(\eta, s), i = 1, 2, 3$, beteckna lösningen till begynnelsevärdesproblemet

$$\begin{cases} u_1' = u_2 \\ u_2' = u_3 \\ u_3' = -\frac{1}{2}u_1 u_3 \\ u_1(0) = u_2(0) = 0,\ u_3(0) = s \end{cases} \tag{5}$$

Vi vill välja s så att

$$q(s) = u_2(\eta_{max}, s) - 1 = 0$$

dvs. så att alla randvillkoren blir uppfyllda.

Vi skall alltså lösa ekvationen $q(s) = 0$. Varje gång funktionen q skall beräknas måste man lösa begynnelsevärdesproblemet (5).

Eftersom q är en funktion i en enda variabel kan vi lösa begynnelsevärdesproblemet (5) för en följd av s-värden och rita upp grafen av q.

Vi beskriver differentialekvationen och funktionen $q(s)$ med respektive

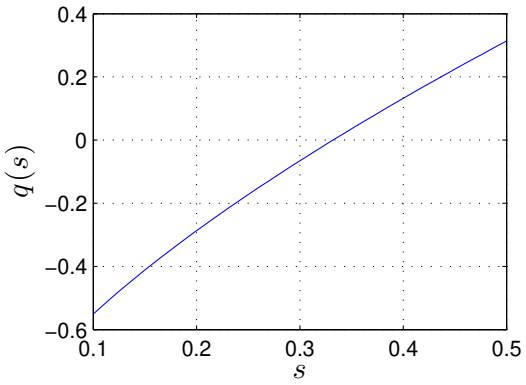
```
function uprim=skikt(eta,u)
uprim=[ u(2)
        u(3)
       -0.5*u(1)*u(3)];
```

```
function q=qskikt(s,etamax)
[eta,u]=ode45(@skikt,[0 etamax],[0;0;s]);
q=u(end,2)-1;
```

Vi ritar grafen till $q(s)$ med

```
>> etamax=10;
>> a=0.1; b=0.5; s=linspace(a,b,50); q=zeros(size(s));
>> for k=1:length(s)
       q(k)=qskikt(s(k),etamax);
   end
>> plot(s,q)
```

Här ser vi grafen av $q(s)$

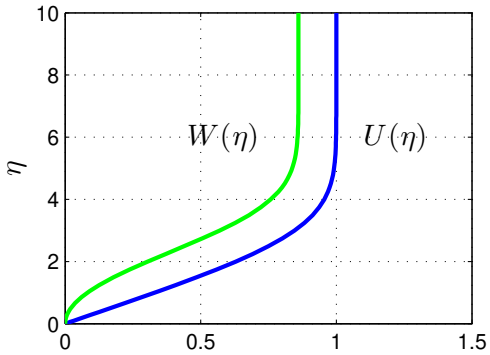


Vi ser att lösningen till $q(s) = 0$ ligger mellan 0.3 och 0.35. Noggrann lösning får vi med

```
>> s=fzero(@(s)qskikt(s,etamax),[0.3 0.35])
s =
    0.3320
```

Vi löser slutligen differentialekvationen för detta s-värde och ritar upp lösningen enligt

```
>> [eta,u]=ode45(@skikt,[0 etamax],[0;0;s]);
>> f=u(1:length(eta),1); fprim=u(1:length(eta),2);
>> U=fprim; W=0.5*(eta.*fprim-f);
>> plot(U,eta,W,eta,'g')
```



Uppgift 9. Lös randvärdesproblemet

$$\begin{cases} u'' + u^2 = 1 \\ u(0) = u(1) = 0 \end{cases}$$

med inskjutningsmetoden. Rita lämpliga grafer.