

# Ordinära differentialekvationer

## 1 Inledning

Vi skall se på *begynnelsevärdesproblem* för första ordningens differentialekvation

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

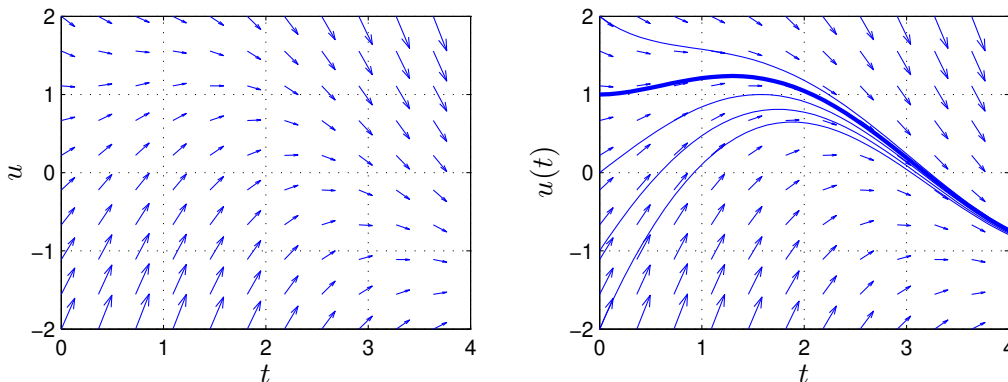
där  $f$  en given funktion och  $u_a$  en given konstant.

Som exempel tar vi problemet

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

med analytisk (exakt) lösning  $u(t) = \sin(t) + u_0 e^{-t}$ .

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 riktningfältet.



Metoder för att beräkna numeriska (approximativa) lösningar till differentialekvationer bygger på idén att försöka följa riktningfältet så *noggrant* och *effektivt* som möjligt.

## 2 Differensmetoder

Vi skall approximera lösningen  $u(t)$  till differentialekvationen på ett nät  $t_n = a + nh$  för  $n = 0, 1, \dots, N$ , 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

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

Detta ger **Eulers framåtmetod**

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

Utgående från begynnelsevärdet försöker metoden följa riktningsfältet med korta steg.

Det finns många olika sätt att approximera derivator och de ger alla upphov till metoder för att lösa differentialekvationer. Mer komplicerade metoder ger större noggrannhet och effektivitet. Vi nöjer oss dock med den enkla Eulers metod.

### 3 Rita riktningsfält

Vi skall se hur man kan rita riktningsfält till differentialekvationen

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

Ett riktningsfält består i en samling punkter  $(t_i, u_j)$  i  $tu$ -planet (ett gitter) där vi i varje punkt ritat 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 ritat vi riktningsfältet till det inledande begynnelsevärdesproblemet

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

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 ritat 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,12); u=linspace(-2,2,12);
>> [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.** Rita riktningsfältet till följande begynnelsevärdesproblem

$$\begin{cases} u' = -u(t) + \sin(5t) + \cos(2t), & 0 \leq t \leq 5 \\ u(0) = u_0 \end{cases}$$

### 4 Eget program i MATLAB

Vi skall nu beskriva hur man kan beräkna en numerisk lösning till begynnelsevärdesproblemet

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

De metoder vi sett på går alla att använda men vi nöjer vi oss med Euler framåtmetoden, som är den enklaste av dessa metoder.

Vi bildar först ett nät med nodpunkterna  $t_n = a + nh$ ,  $n = 0, 1, \dots, N$ , och steglängden  $h = \frac{b-a}{N}$ . Detta ger en uppdelning av intervallet  $a \leq t \leq b$  i  $N$  stycken lika långa delintervall

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

Vi beräknar sedan en approximativ lösning enligt

$$\begin{aligned} U(t_0) &= u_a \\ U(t_{n+1}) &= U(t_n) + hf(t_n, U(t_n)). \end{aligned}$$

Genom att förbinda punkterna  $(t_n, U(t_n))$  med räta linjer får vi en graf och funktionen  $U(t)$  blir definierad också mellan beräkningsnoderna  $t_n$ .

I MATLAB måste  $U(t_n)$  representeras av en vektor  $U$  med  $N$  komponenter. Med andra ord,  $U(n)$  skall innehålla approximationen av  $u(t_n)$  för beräkningsnoden (tidpunkten)  $t_n$ .

Vi ser på vårt inledande exempel och tar  $u(0) = 1$ . Så här enkel blir MATLAB-koden

```
>> f=@(t,u)-u+sin(t)+cos(t);
>> a=0; b=4; ua=1;
>> N=100; h=(b-a)/N;
>> t=a+(0:N)*h; 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)
```

Resultatet ser vi i högra figuren på första sidan, där vi även ritat upp lösningar för några andra begynnelsevärden.

**Uppgift 2.** Vi ser återigen på begynnelsevärdesproblemet

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

Lös problemet med Euler framåtmetoden. Rita en graf av lösningen i en figur som dessutom innehåller riktningsfältet.

**Uppgift 3.** Skriv ett program `min_ode` med anropet `[t,U]=min_ode(f,I,ua,h)` som löser begynnelsevärdesproblemet med Euler framåtmetoden. Använd det programskal du finner på studiohemsidan. In- och ut-variablerna förklaras i programskalet.

**Uppgift 4.** Testa programmet på följande begynnelsevärdesproblem. För varje exempel måste du skriva en funktionsfil av typen `function y=funk(t,u)`. Lös först begynnelsevärdesproblemet analytiskt (dvs. som en formel med penna och papper). Rita både den analytiska lösningen  $u$  och den approximativa lösningen  $U$  i samma figur.

$$(a). \quad \begin{cases} u'(t) = t^2, & 1 \leq t \leq 3 \\ u(1) = 1 \end{cases}$$

$$(b). \quad \begin{cases} u'(t) = u(t), & 0 \leq t \leq 2 \\ u(0) = 1 \end{cases}$$

$$(c). \quad \begin{cases} u'(t) = -tu(t), & 0 \leq t \leq 3 \\ u(0) = 1 \end{cases}$$

$$(d). \quad \begin{cases} u'(t) = -5u(t), & 0 \leq t \leq 1 \\ u(0) = 2 \end{cases}$$

## 5 Färdiga program i MATLAB

Det finns färdiga funktioner i MATLAB för att lösa differentialekvationer, en sådan funktion är `ode45` för "vanliga" begynnelsevärdesproblem, en annan är `ode15s` för s.k. *styva* problem. (Ett styvt begynnelsevärdesproblem beskriver förlopp eller processer vilka utspelas under tidsintervall av mycket olika storleksordning. T.ex. inom kemisk reaktionsteknik är styva problem vanliga.). Med `ode45` kan vi beräkna en lösning till vårt inledande exempel för t.ex.  $u(0) = 1$  enligt

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

Här blir `t` en kolonnvektor, med  $t$ -värden mellan  $a$  och  $b$ , där lösningen är beräknad och `U` är en kolonnvektor med den beräknade lösningen för de olika  $t$ -värdena.

**Uppgift 5.** För en sista gång ser vi på begynnelsevärdesproblemet

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

Lös problemet med `ode45`. Rita en graf av lösningen i en figur som även innehåller riktningsfältet. Beräkna sedan en lösning med `min_ode` och rita upp den i samma figur. Använd olika färg för de olika graferna. Ser du någon skillnad mellan kurvorna, ta då och minska steget  $h$  i `min_ode` och rita ut även den nya lösningen.