Differentialekvationer

1 Inledning

Vi skall se på *begynnelsevärdesproblem* för första ordningens differentialekvation

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

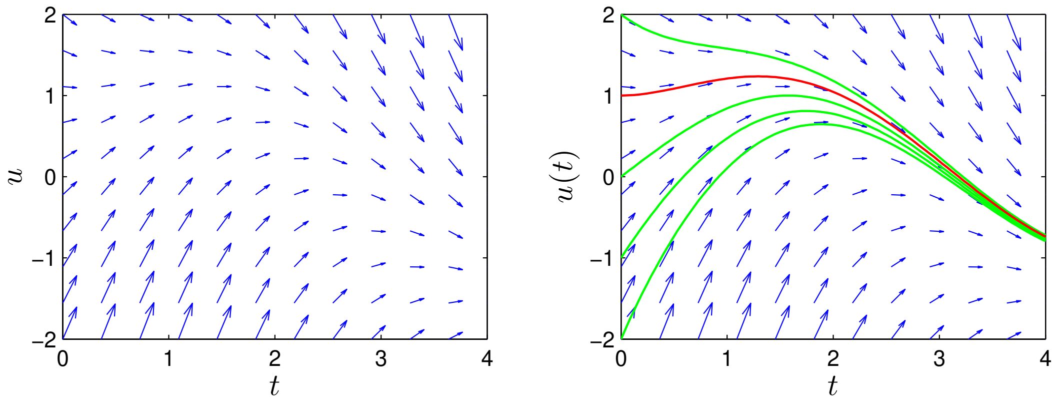
där f en given funktion och u_a en given konstant.

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 $u(t) = \sin(t) + u_0 \exp(-t)$.

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.



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

2 Riktningsfält

Ett riktningsfält består av en samling punkter (t_i, u_j) i tu-planet (ett gitter). I varje punkt ritar vi en liten pil (eller linje) i den riktning som en lösningskurva $(t, u(t))$ genom punkten har precis i punkten, dvs. en pil i riktningen $(1, u'(t_i)) = (1, f(t_i, u_j))$.

Uppgift 1. Rita riktningsfält till följande begynnelsevärdesproblem

$$u' = -u(t) + \sin(5t) + \cos(2t), \ 0 \leq t \leq 5$$

Använd funktionen `riktningsfaelt` på kurshemsidan för att rita riktningfält. Titta gärna på programkoden, den är ganska lätt att förstå.

3 Eulers metod

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

Låter vi u_n beteckna approximationen av $u(t_n)$ och ersätter $u'(t_n)$ med en differenskvot får vi

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

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

Detta ger **Eulers metod**

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

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.

Vi skall nu i detalj beskriva hur man med Eulers metod kan beräkna en numerisk lösning till begynnelsevärdesproblemet

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

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

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

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 $u(0) = 1$. Med Matlab beräknar vi lätt lösningen 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
       U(n+1)=U(n)+h*f(t(n),U(n));
   end
>> plot(t,U)
```

Uppgift 2. Lös följande differentialekvation med begynnelsevillkor

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

med Eulers metod. Rita en graf av lösningen i en figur som dessutom innehåller riktningsfältet.

Uppgift 3. Skriv en funktion med namnet `min_ode` och anropet `[t,U]=min_ode(f,I,ua,h)` som löser begynnelsevärdesproblem med Eulers metod. Du skall använda programskalet `min_ode.m` på kurshemsidan.

Uppgift 4. Testa programmet på följande begynnelsevärdesproblem. Lös först begynnelsevärdesproblemet analytiskt (dvs. som en formel med penna och papper). Rita upp både den analytiska lösningen u och den approximativa lösningen U i samma figur. Tag steglängderna $h = 0.1$ och $h = 0.001$.

(a). $$\begin{cases} u'(t) = t^2, \ 0 \le t \le 3 \\ u(0) = 1 \end{cases}$$

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

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

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

4 Färdiga program i Matlab

Det finns färdiga funktioner i Matlab för att lösa differentialekvationer. En sådan funktion är `ode45` och vi använder den för att beräkna en 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 5. (a). Lös följande differentialekvation med begynnelsevillkor

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

med `ode45`. Rita upp riktningsfält och lösningskurvan med `ode45` i samma bild. Rita sedan även lösningskurvan beräknad med `min_ode` där du tar steglängden $h = 0.001$.

(b). Använd `ode45` enligt exemplet ovan och lös differentialekvationen i (a)-uppgiften. Titta på lösningen som `ode45` ger.

· Kommandot väljer själv steglängd. Hur många steg har valts?

· Används samma steglängd på hela intervallet?

Beräkna en approximation till $u(3)$ med hjälp av `ode45`.