Ordinära differentialekvationer 2

1.1 Exponentialfunktionen

Vi erinrar oss att exponentialfunktionen uppfyller

$$D\exp(x) = \exp(x), \quad \exp(0) = 1.$$

Det betyder att $u(x) = \exp(x)$ är lösning till begynnelsevärdesproblemet

$$\begin{aligned} &u'(x) = u(x), \quad x \in [0, b] \quad (b > 0) \\ &u(0) = 1. \end{aligned}$$

Man kan visa att lösningen till detta problem är unik (vi återkommer till detta) så att $u(x) = \exp(x)$ är den enda lösningen. Vi ska använda detta begynnelsevärdesproblem för att konstruera $u(x) = \exp(x)$ för $x \geq 0$. För $x < 0$ kan vi sedan definiera $\exp(x) = 1/\exp(-x)$.

Mer allmänt kan vi betrakta begynnelsvärdesproblemet

$$\begin{aligned} &u'(x) = cu(x), \quad x \in [0, b] \\ &u(0) = u_0. \end{aligned}$$

Det är lätt kolla att lösningen är $u(x) = u_0 \exp(cx)$. Notera att u är växande om $c > 0$, avtagande om $c < 0$ och konstant om $c = 0$.

Ännu mer allmänt kan vi betrakta

$$\begin{aligned} &u'(x) = cu(x), \quad x \in [a, b] \\ &u(a) = u_a. \end{aligned} \tag{1}$$

Det är lätt kolla att lösningen är $u(x) = u_a \exp(c(x - a))$.

Vi konstruerar lösningen till (1) med hjälp av Eulers metod. Vi börjar med att dela in intervallet $[a, b]$ i N stycken delintervall av längden $h = (b - a)/N$:

$$\begin{aligned} &a = x_0 < x_1 < x_2 < \dots x_{i-1} < x_i < \dots < x_{N-1} < x_N = b, \\ &x_i = a + hi, \ h = (b - a)/N = x_i - x_{i-1}. \end{aligned}$$

Sedan utgår vi från approximationen

$$\frac{u(x_i) - u(x_{i-1})}{h} \approx u'(x_{i-1}) = cu(x_{i-1}),$$

vilket leder till

$$u(x_i) \approx u(x_{i-1}) + hcu(x_{i-1}).$$

Vi beräknar nu en approximativ lösning enligt

$$\begin{aligned} &U(x_0) = u_a \\ &U(x_i) = U(x_{i-1}) + hcU(x_{i-1}). \end{aligned}$$

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

Konvergens

Man kan visa att $U(x)$ konvergerar mot en unik lösning $u(x)$ till (1) då antalet delintervall $N \to \infty$. Beviset är upplagt enligt samma princip som för bisektionsalgoritmen och fixpunktsiterationen.

Vi delar intervallet $[a, b]$ med hjälp av halvering. Vi låter n beteckna antalet halveringar, antalet delintervall blir $N = 2^n$, och för varje n får vi en approximativ lösning $U_n(x)$:

$$
\begin{aligned}
&n = 1, \quad N = 2^1 = 2, \quad U_1(x)\\
&n = 2, \quad N = 2^2 = 4, \quad U_2(x)\\
&n = 3, \quad N = 2^3 = 8, \quad U_3(x)\\
&n = 4, \quad N = 2^4 = 16, \quad U_4(x)\\
&\text{och så vidare.}
\end{aligned}
$$

Beviset går sedan enligt följande.

1. Algoritmen Eulers metod ger en följd av approximativa lösningar $\{U_n(x)\}_{n=1}^{\infty}$.
2. Vi visar att $U_n(x)$ är en Cauchy-följd, dvs $U_n(x) - U_m(x) \to 0$ då $m, n \to \infty$. Vi får alltså ett reellt tal (decimalutveckling) $u(x) = \lim_{n\to\infty} U_n(x)$.
3. Vi visar att u löser (1), dvs u är deriverbar med $u' = cu$ och $u(a) = u_a$.
4. Vi visar att lösningen till (1) är unik.

Detta bevis är ganska långt och lite för svårt och vi nöjer oss med att nämna dessa steg. Vi har nu konstruerat en ny funktion $u(x)$ som kallas $\exp(x)$.

Implementering i Matlab

Algoritmen är

$$
\text{initiera:} \begin{cases} x_0 = a \\ U(x_0) = u_a \end{cases}
$$

$$
\text{uppdatera:} \begin{cases} \text{while } x_i < b \\ x_i = x_{i-1} + h \\ U(x_i) = U(x_{i-1}) + hcU(x_{i-1}). \end{cases}
$$

Övning 1. Skriv ett program `myexp.m` med anropet `[x,U]=myexp(c,I,ua,h)` som löser begynnelsevärdesproblemet (1). Du skall använda programskalet `myexp.m`.

Börja med $c = 1$, $I = [0, 1]$, $u_a = 1$. Plotta både den analytiska lösningen u och den approximativa lösningen U i samma figur. Använd stort steg h, till exempel, $h = 10^{-1}$, när du först skriver programmet så blir det lättare att se vad som händer. Tag sedan litet h, till exempel, $h = 10^{-3}$. Obs att Matlabs `exp` är inte en exakt lösning, det är också en approximativ lösning, dock beräknad med hög noggrannhet.

Plotta till sist flera lösningar med olika värden på c, positiva och negativa. $\square$

1.2 Trigonometriska funktioner

Vi vet att de trigonometriska funktionerna uppfyller

$$\begin{aligned}
&D\sin(t) = \cos(t),\ D^2\sin(t) = -\sin(t),\ \sin(0) = 0,\\
&D\cos(t) = -\sin(t),\ D^2\sin(t) = -\cos(t),\ \cos(0) = 1.
\end{aligned}$$

(Notera att vi nu går över till att skriva den oberoende variabeln som t istället för x. Anledningen till detta är att den ofta är tiden i ett begynnelsevärdesproblem.) Det betyder att $u(t) = \sin(t)$ löser begynnelsevärdesproblemet

$$\begin{aligned}
&u''(t) = -u(t), \quad t \in [0, b], \quad (b > 0)\\
&u(0) = 0,\ u'(0) = 1,
\end{aligned}$$

och att $u(t) = \cos(t)$ löser begynnelsevärdesproblemet

$$\begin{aligned}
&u''(t) = -u(t), \quad t \in [0, b], \quad (b > 0)\\
&u(0) = 1,\ u'(0) = 0,
\end{aligned}$$

Mer allmänt har vi

$$\begin{aligned}
&u''(t) = -c^2u(t), \quad t \in [0, b], \quad (b > 0)\\
&u(0) = u_0,\ u'(0) = u_1,
\end{aligned}$$

med lösningen

$$u(t) = u_0\cos(ct) + \frac{u_1}{c}\sin(ct). \tag{2}$$

Verifiera detta genom att derivera och genom att sätta in $t = 0$!

Denna differentialekvation är av andra ordningen. För att kunna använda vår algoritm skriver vi om den som ett system av två differentialkevationer av första ordningen. Vi inför två nya variabler

$$w_1 = u,\ w_2 = u'.$$

Vi deriverar

$$\begin{aligned}
w_1' &= u' = w_2,\\
w_2' &= u'' = -c^2u = -c^2w_1.
\end{aligned}$$

Begynnelsevillkoren blir $w_1(0) = u_0$, $w_2(0) = u_1$. Vi ser att w_1 och w_2 löser begynnelsevärdesproblemet

$$\begin{cases} w_1' = w_2, \\ w_2' = -c^2w_1, \end{cases} \quad \begin{cases} w_1(0) = u_0, \\ w_2(0) = u_1. \end{cases}$$

Vi skriver detta på matrisform:

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}' = \begin{bmatrix} w_1' \\ w_2' \end{bmatrix} = \begin{bmatrix} w_2 \\ -c^2 w_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -c^2 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, \quad \begin{bmatrix} w_1(0) \\ w_2(0) \end{bmatrix} = \begin{bmatrix} u_0 \\ u_1 \end{bmatrix}.$$

Med beteckningarna

$$A = \begin{bmatrix} 0 & 1 \\ -c^2 & 0 \end{bmatrix}, \quad w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, \quad w_0 = \begin{bmatrix} u_0 \\ u_1 \end{bmatrix},$$

får vi slutligen

$$\begin{aligned} &w'(t) = Aw(t), \quad t \in [0, b], \\ &w(0) = w_0. \end{aligned} \tag{3}$$

Vår algoritm (Eulers metod) blir nu

$$\text{initiera: } \begin{cases} t_0 = 0 \\ W(t_0) = w_0 \end{cases}$$

$$\text{uppdatera: } \begin{cases} \text{while } t_i < b \\ t_i = t_{i-1} + h \\ W(t_i) = W(t_{i-1}) + hAW(t_{i-1}). \end{cases}$$

Övning 2. Skriv ett program `mytrig.m` med anropet `[t,W]=mytrig(c,I,w0,h)` som löser begynnelsevärdesproblemet (3). Du skall använda programskalet `mytrig.m`.

I Matlab sparas $W(t_i)$ som i:te kolonnen `W(:,i)` i matrisen `W`, den blir av typ $2 \times (N + 1)$. Matrisen skapas kolonnvis. Till exempel blir initieringen `W(:,1)=w0`. När beräkningen är klar bör du transponera matriserna `t` och `W` bland annat för att plottning ska fungera smidigt.

Tag först $c = 1$ och $w_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

Det finns nu två sätt att plotta lösningen.

1. Vi kan plotta de två *lösningskurvorna* $y = w_1(t)$, $y = w_2(t)$ med kommandot `plot(t,W)`.

2. Vi kan plotta w_2 mot w_1 med kommandot `plot(W(:,1),W(:,2))`. Sådana kurvor med olika begynnelsepunkter bildar ett *fasporträtt* för differentialekvationssystemet.

□

1.3 Hyperboliska funktioner

Vi vet att de hyperboliska funktionerna uppfyller

$$\begin{aligned} &D \sinh(t) = \cosh(t), \; D^2 \sin(t) = \sinh(t), \; \sinh(0) = 0, \\ &D \cosh(t) = \sinh(t), \; D^2 \sinh(t) = \cosh(t), \; \cosh(0) = 1. \end{aligned}$$

Det betyder att både $u(t) = \sinh(t)$ och $u(t) = \cosh(t)$ uppfyller differentialekvationen

$$u''(t) = u(t).$$

Övning 3. Visa hur denna ekvation skrivs om till ett system av två differentialekvationer av första ordningen och hur vi får ett begynnelsevärdesproblem av typen (3) fast med matrisen

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Gör motsvarande ändring i programmet `mytrig.m` och använd detta till att plotta de två kurvorna $y = \sinh(t)$, $y = \cosh(t)$. $\square$

`2006-11-16 /stig`