



Ordinära differentialekvationer och glesa system

Efter den här laborationen skall du känna igen problemtyperna randvärdes- och begynnelsevärdesproblem för ordinära differentialekvationer och kunna lösa dessa med differensmetoder. Du skall kunna analysera noggrannhetsordning och bestämma stabilitetsegenskaper både teoretiskt och experimentellt. Du skall också lära dig att lösa stora linjära ekvationssystem.

1. Begynnelsevärdesproblem

Läsanvisning: GNM Kap 6.1-6.2

a) Givet är följande differentialekvation

$$\frac{dy}{dt} = \cos(t) - 4y, \quad y(0) = 0.6, \quad t \in [0, 30].$$

I denna deluppgift ska ovanstående differentialekvation lösas numeriskt med Eulers framåtmetod, Eulers bakåtmetod och trapetsmetoden. Uppgiften går ut på att undersöka

- hur trunkeringsfelet avtar med steglängden h (för små värden på h), dvs noggrannheten
 - hur den numeriska lösningen uppför sig för 'stora' h -värden (stabilitet)
- (1) Lös först differentialekvationen exakt (analytiskt), dvs med metoder som du lärt dig i matematik. Denna lösning betecknas $y(t)$.
 - (2) Dela in tidsintervallet $[0, 30]$ i n ekvidistanta steg h . Eulerlösningen i en punkt t betecknas $y(t, h)$. Skriv ett Matlabprogram som beräknar Eulerlösningarna för $n = 150$, $n = 300$, $n = 600$ och $n = 1200$. Plotta i samma graf den exakta lösningen $y(t)$ samt de tre Eulerlösningarna. Plotta även i ett loglog-diagram felet i slutpunkten, dvs $|y(30) - y(30, h)|$ som funktion av h . Vilken ordning hos Eulers framåtmetod kan utläsas ur diagrammet?
 - (3) Gör samma beräkningar, grafer, diagram och dra slutsatser för Eulers bakåt-metod.
 - (4) Gör samma beräkningar, grafer, diagram och dra slutsatser för trapetsmetoden.
 - (5) Dela in tidsintervallet $[0, 30]$ i $n = 120, 60, 30$ och 15 ekvidistanta steg. Beräkna de fyra Eulerlösningarna och plotta dem tillsammans med den exakta lösningen i varsin graf. Gör samma sak med Eulers bakåt-metod. Gör samma sak med trapetsmetoden. Vilken slutsats kan du dra beträffande den numeriska stabiliteten för de tre metoderna genom att titta på graferna? För den eller de icke-stabila, vid vilken steglängd h går gränsen?

b) Följande andra ordningens differentialekvation beskriver en pendels rörelse.

$$\frac{d^2\phi}{dt^2} + \frac{g}{L} \sin(\phi) = 0, \quad \phi(0) = \frac{\pi}{4}, \quad \frac{d\phi}{dt}(0) = 0, \quad t = [0, T]$$

Här är ϕ vinkeln och ϕ' vinkelhastigheten. Längden på snöret är $L = 3 \text{ m}$ och tyngdkraften $g = 9.81 \text{ m/s}^2$.

- (1) Skriv om differentialekvationen som ett system av första ordningens differentialekvationer. Systemet skall redovisas på papper.
- (2) Lös systemet med MATLABs inbyggda ode-lösare **ode45**. Välj ett tidsintervall, $[0, T]$, som gör att pendeln hinner svänga ungefär två hela perioder. Plotta vinkel och vinkelhastighet som funktion av tiden.
- (3) Animera pendelns gång. Animeringen kan göras med tex ett anrop av följande MATLAB-kod:

```
function anim(tut,fiut,L);
    for i=1:length(tut)-1
        x0=L*sin(fiut(i));y0=-L*cos(fiut(i));
        plot([0,x0],[0,y0],'-o')
        axis('equal')
        axis([-1 1 -1 0]*1.2*L)
        drawnow
        pause(tut(i+1)-tut(i))
    end;
end
```

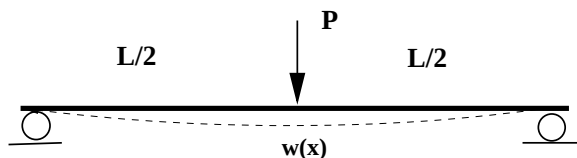
där **tut** är tidpunkterna vid vilka **ode45** har räknat ut lösningen, **fiut** är den uträknade vinkeln vid motsvarande tidpunkt och **L** är pendelns längd.

- (4) Bestäm pendelns svängningstid (dvs period) med minst 1 decimal genom att göra interpolation i den beräknade pendelrörelsen och bestämma interpolationspolynomets nollställe.
- (5) Beror svängningstiden av L ?

2. Randvärdesproblem och stora linjära ekvationssystem

Läsanvisning: GNM Kap 6.3 och 4.1 (glesa matriser).

En homogen balk av höghållfast stål är fritt upplagd horisontellt på två rullstöd med avståndet $L = 2.50 \text{ m}$. Balken har ett cirkulärt tvärsnitt med en radie $r = 4.00 \cdot 10^{-2} \text{ m}$ och en elasticitetsmodul $E = 2.10 \cdot 10^{11} \text{ N/m}^2$. Mitt på balken verkar en nedåtriktad kraft $P = 240 \text{ N}$. Man vill beräkna balkens utböjning, $w(x)$.



För ett tvärsnitt vid läget x finns ett samband mellan momentet $M(x)$ och balkens utböjning $w(x)$. Om kraften P är tillräckligt liten ges detta samband av följande linjära modell

$$\frac{M(x)}{EI} = -w''(x). \quad (1)$$

I är yttröghetsmomentet för balken och ges utav formeln

$$I = \frac{\pi r^4}{4}$$

Momentet beror av kraften P enligt

$$M(x) = \begin{cases} -\frac{Px}{2} & x \leq \frac{L}{2} \\ -\frac{PL}{2}(1 - \frac{x}{L}) & x \geq \frac{L}{2} \end{cases}$$

Randvillkoren, att balken är fritt upplagd, kan skrivas som

$$w(0) = 0, \quad \text{och} \quad w(L) = 0. \quad (2)$$

Ekvation (1) tillsammans med randvillkoren (2) kan lösas numeriskt med hjälp av finita differensmetoden. Om vi diskretiserar balken i $N + 1$ punkter (dvs N delar), $x_j = jh$, $j = 0, 1, \dots, N$, $h = L/N$, och använder en andra ordningens noggrann differens-approximation får vi följande:

$$\frac{w_{j+1} - 2w_j + w_{j-1}}{h^2} = -\frac{M(x_j)}{EI}, \quad j = 1, 2, \dots, N-1 \quad (3)$$
$$w_0 = 0 \quad w_N = 0 \quad (\text{ges av randvillkoren})$$

där w_j är utböjningen i punkten $x = x_j$. Diskretiseringen leder till ett linjärt ekvationssystem

$$A\mathbf{w} = \mathbf{f} \quad (4)$$

där A är av storlek $(N-1) \times (N-1)$ och $\mathbf{w} = (w_1, w_2, \dots, w_{N-1})$ är en vektor med de obekanta utböjningarna. Vektorn $\mathbf{f}$ ges av högerledet i ekvation (3).

a) Skriv ner matrisen A med papper och penna. Vilken struktur har matrisen A ?

b) Skriv ett Matlab-program som löser randvärdesproblemet. Matrisen A kommer endast att ha ett fåtal nollskilda element. (Tips: Skapa matrisen i Matlab med Matlabs `diag` och `eye`).

Räkna ut utböjningen w för fallet $N = 200$. Lösningen får du genom att lösa det linjära ekvationssystemet $A\mathbf{w} = \mathbf{f}$. Du behöver också räkna ut $\mathbf{f}$ som ges av högerledet i ekvation (3) och sambanden för momentet, $M(x)$, och yttröghetsmomentet, I . Använd värdena på L , E , P och r enligt ovan. Lösningen, $\mathbf{w}$, kommer att innehålla utböjningen i alla punkter, x_j utom i randpunkterna där $w = 0$.

Plotta utböjningen som funktion av x . Hur stor är den maximala utböjningen?

c) Backslash-kommandot i MATLAB använder vanlig gausseliminering för att lösa ekvationssystemet. Undersök hur tidsåtgången för gausseliminering beror av systemmatrisens storlek genom att lösa $A\mathbf{w} = \mathbf{f}$ för olika N , tex $N = 200, 400, 800, 1600$. Observera att högerledet måste räknas om för de olika storlekarna. För att mäta tiden kan MATLAB-funktionen `cputime` eller `tic` och `toc` användas (`help` ger mer information). För att få en bra noggrannhet av mätningen av cputiden (eftersom den är kort) bör man kanske upprepa beräkningarna (dvs lösandet av det linjära ekvationssystemet) i en for-slinga (tex 5 eller 20 gånger, beroende på dator) och sedan ta medelvärdet. Plotta tidsåtgången mot antal intervall N i en loglog-plot. Hur beror tidsåtgången av N ? Stämmer detta med teorin för glesa matriser?

d) När en matris är gles (fåtal nollskilda element) kan betydligt effektivare metoder än vanlig gausseliminering användas för att lösa ekvationssystemet. Genom att tala om för Matlab att matrisen är gles kommer bättre metoder automatiskt användas när backslash-operatören anropas. Detta kan ni enkelt göra här genom att skriva `A=sparse(A)`. Gå igenom beräkningarna i deluppgifterna b och c igen. (Tiden är nu ännu kortare, kanske behövs fler varv i for-slingan?) Hur stor tidsvinst gör man i detta fall genom att låta Matlab använda metoder för glesa matriser? Plotta tidsåtgången mot antal delintervall N i en loglog-plot. Hur beror tidsåtgången av N ? Stämmer detta med teorin för glesa matriser?

Tre små kommentarer:

- Kan det vara så att ens resultat inte tycks stämma med teorin i boken? Hur motiverar man sitt (korrekta) resultat då?
- Om man redan från början vet att matrisen skall vara gles kan man istället skapa den med kommandona

```
e = ones(N-1,1)/h^2;
A = spdiags([e -2*e e], -1:1, N-1, N-1);
```

- Faktum är att det egentligen är rätt dumt att dela upp balken i så pyttesmå bitar. Uppdelning i $N = 10, 20$ och 40 följt av ett stegs Richardson-extrapolation ger ett svar med högre noggrannhet än körningarna med $N = 1000$ och 2000 .
— Men med så små N får vi så usla tids-studier ;)

Hur många timmar ungefär har den här laborationen tagit?

En fråga på kursutvärderingen i slutet av kursen kommer att gälla tidsåtgång och laborationsomfång. Tänk redan nu igenom vad som är bra och vad som kan förbättras!

/--- $\mathcal{NC}$ ---/