



KTH Engineering Sciences

Laboration 2

Interpolation, integration och differentialekvationer

Sista dag för bonuspoäng, se kursplanen. Kom väl förberedd och med välordnade papper till redovisningen. Numeriska resultat ska finnas noterade. Båda i laborationsgruppen ska kunna redogöra för teori och algoritmer!

1. Interpolation och minstakvadratanpassning

I almanackan kan man hitta tider för solens upp- och nedgång för några orter i Sverige. Tabellen nedan är uträknad ur almanackan och anger dagens längd i Stockholm den första dagen i varje månad under sommarhalvåret (tiden är angiven decimalt):

Månad:	1 april	1 maj	1 juni	1 juli	1 aug	1 sep
Dagnr :	91	121	152	182	213	244
Solen uppe:	13.18	15.78	17.97	18.38	15.53	14.07

a) Interpolera punkterna med ett femtegradspolynom. Rita de sex givna punkterna och polynomkurvan med fin diskretisering (dagligen från dag 91 till dag 244). Hur länge är solen uppe på nationaldagen den 6 juni? Hur länge är den uppe 15 augusti?

b) Anpassa ett andragradspolynom i minstakvadratmening till den givna datan. Plotta på samma sätt som ovan och ange soltiden 6 juni och 15 augusti enligt denna modell?

c) Tabellen kompletteras med vinterhalvårets värden:

Månad:	1 jan	1 feb	1 mars	1 okt	1 nov	1 dec
Dagnr :	1	32	60	274	305	335
Solen uppe:	6.13	8.02	10.42	11.43	8.73	6.55

Använd det trigonometriska uttryck $c_1 + c_2 \cos \omega t + c_3 \sin \omega t$ där $\omega = 2\pi/365$, som modellfunktion. Varför bör detta kunna ge god anpassning? Rita resultatet i en tvårutors subplot med de tolv punkterna och kurvresultatet (dagligen från dag 1 till dag 365) i första rutan. I andra rutan ritas residualvektorns tolv komponenter mot de tolv givna dagnumren. Beräkna felkvadratsumman samt nationaldagens soltid enligt denna modell.

2. Numerisk integration

Konturen för en rotationssymmetrisk lur definieras av funktionskurvan

$$y(x) = \frac{1 - \frac{1}{\pi} \arctan[p(x-1)]}{2 - \cos(\pi x)}, \quad 0 \leq x \leq L,$$

där p är en formparameter och L är luren längd. Luren uppstår genom att kurvan roteras kring x -axeln och rotationsvolymen är

$$V = \pi \int_0^L y^2 dx.$$

Vi önskar beräkna volymen för en lur med längden $L = 2.6$ och $p = 1$.

a) Implementera de sammansatta versionerna av trapetsregeln och Simpsons formel. Beräkna integralen med steglängderna $h = 0.1$ och $h = 0.05$. Hur många siffror verkar tillförlitliga i resultaten för de två metoderna?

b) Undersök hur approximationsfelet E_h för metoderna beror på steglängden h . (Notera att antal punkter måste vara udda i Simpson-fallet.) Plotta E_h som funktion av h och uppskatta från dem metodernas noggrannhetsordning. Jämför resultatet med de teoretiska noggrannhetsordningarna. Använd MATLABs kommando `loglog` för plottarna, gärna tillsammans med `grid`-kommandot. (För att beräkna felet kan ni först göra en mycket noggrann referenslösning med Simpsons metod, på liknande sätt som i Lab 1.)

Prova också att uppskatta noggrannhetsordningen genom att för små h studera kvoten

$$\frac{I_h - I_{h/2}}{I_{h/2} - I_{h/4}},$$

där I_h är integralapproximationen med steglängd h . Se föreläsningsanteckningarna om noggrannhetsordning.

c) Lös integralen med MATLAB-kommandot `quad` som använder en adaptiv version av Simpson. (Gör `help quad` för mer info.) Hur liten tolerans behövs för att få samma noggrannhet som din Simpson-implementation med 1000 punkter? Hur många funktionsevalueringar gör `quad` då? (Detta ges av andra returargumentet.) Prova med både små och stora värden på formparametern p . När är `quad` mer effektiv än vanliga Simpson-metoden? Varför?

En fin tredimensionell lur bild gör man så här: Låt $\mathbf{x}$ och $\mathbf{f}$ vara kolumnvektorer för konturkurvan $y(x)$. Skapa en radvektor för rotationsvinkeln $0 \leq \varphi \leq 2\pi$ med lagom steg, tex $2\pi/30$. Bilda matriser $\mathbf{X}$, $\mathbf{Y}$ och $\mathbf{Z}$:

$$\mathbf{X} = \mathbf{x} * \text{ones}(\text{size}(\mathbf{f}_i)); \mathbf{Y} = \mathbf{f} * \cos(\mathbf{f}_i); \mathbf{Z} = \mathbf{f} * \sin(\mathbf{f}_i);$$

Skriv `mesh(X,Y,Z)` som ger en nätfigur eller välj `surf(X,Y,Z)` eller `surfl(X,Y,Z)` som ger en fylld 3D-figur (gör gärna `help surfl`). Testa andra möjligheter: `axis equal`, `axis off`, `shading interp`, etc.

3. Ordinära differentialekvationer

Vi studerar den dämpade linjära oscillatorn

$$mz'' + \alpha z' + kz = 0, \quad z(0) = 1, \quad z'(0) = 0.$$

Ekvationen modellerar till exempel den dämpade svängningen av en massa på en fjäder om fjädern följer Hookes lag och friktionen är proportionell mot hastigheten. Här är z avvikelsen från jämvikt, m massan, α friktionskoefficienten och k fjäderkonstanten. (Detta följer från Newtons andra lag $F = mz''$ och $F = -kz - \alpha z'$.)

a) Skriv om differentialekvationen till ett system av första ordningens ekvationer. Välj $m = k = 1$ och $\alpha = 10$. För detta stora värde på koefficienten α är systemet överdämpat. Lös problemet med Framåt Euler-metoden och tidssteget $h = 0.1$ fram till $t = 20$. Plotta lösningen $z(t)$.

Bestäm teoretisk hur litet tidssteget h måste vara för att Framåt Euler ska vara absolutstabil. (Se föreläsningssanteckningarna.) Verifiera gränsen experimentellt. Plotta ett exempel på en instabil lösning.

b) Välj nu istället det mindre värdet $\alpha = 0.2$, som ger ett underdämpat system. Lös problemet igen med Framåt Euler till $t = 20$. Variera tidssteget och plotta dina numeriska approximationer av $z(t)$ för olika (mindre och mindre) tidssteg i samma figur med hjälpa av `hold on`. Förklara hur man kan se från bilden att Framåt Euler är en första ordnings metod.

Gör nu samma körning med Runge-Kutta 4-metoden. Försök uppskatta hur mycket längre tidssteg du kan använda med denna metod jämfört med Framåt Euler och fortfarande få (ungefär) samma noggrannhet.

Plotta även lösningen i fasplanet, dvs plotta kurvan $(x(t), y(t))$ med $x(t) = z(t)$, $y(t) = z'(t)$. Visualisera rörelsen med kommandot `comet(z,zp)` där vektorn $\mathbf{z}$ innehåller z -värdena och $\mathbf{zp}$ innehåller värdena på z' .

c) Slutligen studerar vi Duffings ekvation som är en lite mer komplicerad version av oscillatorn ovan,

$$mz'' + \alpha z' + kz + z^3 = 6 \cos(t/2), \quad z(0) = 1, \quad z'(0) = 0.$$

Här är kraften $F = -kz - \alpha z' - z^3$ relativt sett större ju mer fjädern töjs ut; problemet är olinjärt. Vi har också lagt till ett högerled $6 \cos(t/2)$ som motsvarar en yttre kraft som forcerar oscillationen.

Lös Duffings ekvation fram till $t = 20$ med MATLABs inbyggda kommando `ode45` som använder en adaptiv metod baserad på inbäddade Runge-Kutta-metoder av hög ordning. Plotta lösningen som funktion av tiden och även lösningen i fasrummet. Visualisera med `comet(z,zp)`.

Efter lång tid konvergerar lösningen mot en periodisk bana. Finn banan genom att lösa differentialekvationen numeriskt och plotta den i fasrummet. (Beräkna tex lösningen över lång tid $t \in [0, 100]$ och plotta bara slutet av banan.)

Experimentera gärna också med mindre värden på α och hitta liknande lösningar. Mer komplicerade periodiska banor och även helt kaotiska banor existerar. Allmänt kan olinjära differentialekvationer ge mycket komplicerade lösningar. Duffings ekvation är ett bra exempel på detta.

4. Randvärdesproblem

Följande differentialekvationsproblem beskriver temperaturfördelningen $T(x)$ i en cylindrisk stav av längden L och med tvärsnittsarean A .

$$-\frac{d}{dx}\left(k\frac{dT}{dx}\right) = Q(x), \quad T(0) = t_0, \quad T(L) = t_1. \quad (1)$$

Vänster och höger ändpunkt på staven har den konstanta temperaturen t_0 resp t_1 . Konstanten k är stavens värmeledningsförmåga och $Q(x)$ är den värmemängd som per tidsenhet och volymenhet genereras i staven, tex genom radioaktivitet. Antag att $L = 2$ [m], $k = 2.5$ [J/(K · m · s)], $t_0 = 300$ [K], $t_1 = 400$ [K] och $Q(x)$ [J/(s · m³)] är funktionen

$$Q(x) = 300e^{-(x-\frac{L}{2})^2} + 3000e^{-1000(x-\frac{L}{4})^2}, \quad 0 \leq x \leq L.$$

Differentialekvation och randvillkor (1) kan lösas numeriskt med hjälp av finita differensmetoden. Om vi diskretiserar intervallet $[0, L]$ enligt $x_i = ih$, $i = 0, 1, 2, \dots, n, n+1$, där $h(n+1) = L$ och approximerar andraderivatan med centraldifferens erhålles

$$\frac{-T_{i-1} + 2T_i - T_{i+1}}{h^2} = \frac{1}{k}Q(x_i), \quad i = 1, 2, \dots, n.$$

Tillsammans med randvillkoren $T_0 = t_0$ och $T_{n+1} = t_1$ leder diskretiseringen till ett linjärt ekvationssystem

$$AT = b,$$

där A är en $n \times n$ -matris, T är en n -vektor med temperaturvärden i det inre av intervallet och b är en n -vektor som beror av bla randvärdena t_0 och t_1 samt $Q(x_i)$ -värdena.

- a) Skriv ner matrisen A för $n = 4$ med papper och penna. Vilken struktur har matrisen A ?
- b) Skriv ett MATLAB-program som löser randvärdesproblemet för en given storlek n . Matrisen A kommer endast att ha ett fåtal nollskilda element. Matrisen skapar du tex med hjälp av MATLABs kommando `diag(v)`, som genererar en diagonalmatris med vektorn v på diagonalen, samt generaliseringen `diag(v,p)` som genererar en matris med vektor v på diagonal nummer p , där p kan vara både positiv och negativ. Verifiera att matrisen stämmer överens med resultatet i uppgift a) för $n = 4$.
- c) Räkna ut temperaturen T genom att lösa det linjära ekvationssystemet $AT = b$ med backslash, `\`. Pröva först med $n = 10$ delintervall. Fördubbla successivt upp till $n = 80$ (eventuellt längre). Plotta de erhållna resultaten i samma figur. Lösningen T kommer att innehålla temperaturen i alla punkter x_i utom i randpunkterna. Du ska plotta temperaturen som funktion av x på *hela* intervallet $0 \leq x \leq L$.

Hur stor är den minimala och maximala temperaturen, samt medeltemperaturen? Uppskatta tillförlitligheten hos värdena.

2D1240, Numeriska metoder gk II

Laboration 2 redovisad och godkänd!

Datum:

Namn:.....

Godkänd av