Projekt 2 Del A: Finita differensmetoden

Projektet görs i grupper om två och examineras genom ett individuellt Canvasquiz på övningen 23/11 kl 15-17. Skrivtid är 60 minuter (90 minuter om man har förlängd skrivtid genom FUNKA). Innan quizet ska obligatoriska individuella uppgifter utföras i Matlab Grader och Matlabkoden för projektet ska lämnas in gruppvis i Canvas (grupper "Projekt 2A"). I slutet av detta dokument finns mer information om förberedelse för quizet.

Diskretisering med finita differensmetoder

Antag att intervallet $a \leq x \leq b$ är diskretiserat med N lika stora intervall. Avståndet mellan varje punkt ges av *steglängden* $h = (b-a)/N$ och

$$x_j = a + jh, \;\; j = 0, 1, \ldots, N. \tag{1}$$

Notera att $x_0 = a$ och $x_N = b$ ger intervallets ändpunkter.
En funktion $u(x)$ som är definierad på intervallet kan representeras i de diskreta punkterna (1), vi benämner dessa värden

$$u_j = u(x_j), \;\; j = 0, 1, \ldots, N.$$

För att numeriskt approximera derivator av $u(x)$ använder vi finita differenser. Vi kan använda så kallade centrerade differensformler för att approximera båda första och andra derivatan till andra ordningen:

$$u'(x_j) = \frac{u_{j+1} - u_{j-1}}{2h} + \mathcal{O}(h^2), \tag{2}$$

$$u''(x_j) = \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} + \mathcal{O}(h^2). \tag{3}$$

Förstaderivatan av u i punkten x_j, $u'(x_j)$, kan också approximeras genom en framåtdifferens eller bakåtdifferens,

$$u'(x_j) = \frac{u_{j+1} - u_j}{h} + \mathcal{O}(h), \tag{4}$$

$$u'(x_j) = \frac{u_j - u_{j-1}}{h} + \mathcal{O}(h). \tag{5}$$

Om vi vill approximera förstaderivatan i x_0 kan vi bara använda differensformel (4), eftersom de andra uttrycken kräver värden utanför intervallet. Motsvarande i högra änden av intervallet kan vi bara använda (5). Dessa differensformler har dock bara noggrannhetsordning 1. Man kan förstås härleda även högre ordningens skeva (ej centrerade) differensformler, tex:

$$u'(x_j) = \frac{-u_{j+2} + 4u_{j+1} - 3u_j}{2h} + \mathcal{O}(h^2), \tag{6}$$

$$u'(x_j) = \frac{3u_j - 4u_{j-1} + u_{j-2}}{2h} + \mathcal{O}(h^2), \tag{7}$$

U1: Gör uppgift U1 i Matlab Grader (OBLIGATORISK): För funktionen $u(x) = \cos(e^x)$, beräkna approximationen av funktionens förstaderivata vid $x = 0.6$ med differensformlerna (4), (2), och (6) för $h = 2^{-k}, k = 1, 2, \ldots, 8$. Plotta absolutbeloppet av felen som funktion av $1/h$ med Matlab-funktionen `loglog`.

Randvärdesproblem

För ett initialvärdesproblem av högre ordning har ni sett i projekt 1B att det går att skriva om som ett system av första ordningens differentialekvationer för att sedan kunna applicera standard ODE-lösare på problemet. För ett andra ordningens randvärdesproblem kan vi istället använda en finit differensmetod.
Betrakta differentialekvationen

$$u''(x) - 4u(x) = f(x), \quad 0 \leq x \leq 1, \tag{8}$$

med Dirichletrandvillkoren

$$u(0) = \alpha, \quad u(1) = \beta. \tag{9}$$

alternativt med de blandade randvillkoren (Dirichletvillkor i $x = 0$ och Neumannvillkor i $x = 1$),

$$u(0) = \gamma, \quad u'(1) = \delta. \tag{10}$$

Låt oss diskretisera intervallet [0 1] med N delintervall och ersätta $u''(x)$ i (8) med den centrala differensapproximationen (3). Vi låter vektorn $\mathbf{U}$ innehålla den numeriska lösningen. Med $N = 5$ får vi $N + 1 = 6$ punkter och $\mathbf{U} = [U_0, U_1, U_2, U_3, U_4, U_5]^T$. I fallet med Dirichletrandvillkor så har vi U_0 och U_5 givna av randvillkoren. Det är alltså endast fyra okända variabler ($\bar{\mathbf{U}} = [U_1, U_2, U_3, U_4]^T$) som vi ska lösa för och systemmatrisen går att skriva som en $(N-1) \times (N-1)$=4 × 4-matris. Randvillkoren kommer in i den första och den sista ekvationen. Värdena i högerledet betecknas $f_i = f(x_i)$.

För de blandade randvillkoren (10) så diskretiserar vi det högra randvillkoret med den skeva differensformeln (7). I formuleringen av det diskreta problemet väljer vi nu att även lösa för värdet U_N, och att utöka systemet med ekvationen för det högra randvillkoret. Nu blir $\bar{\mathbf{U}} = [U_1, U_2 \ldots, U_{N-1}, U_N]^T$ och systemmatrisen en $N \times N$ matris. Vi får ett ekvationssystem på formen

$$A\bar{\mathbf{U}} = \mathbf{F} \tag{11}$$

där högerledet $\mathbf{F}$ beror både på randvillkoren, punktvisa värden av högerledet $f(x)$, och steglängden h. Vi löser för $\bar{\mathbf{U}}$ och sätter sedan $\mathbf{U} = [U_0\, \bar{\mathbf{U}}^T]^T$.

Förberedelseuppgift (behöver inte redovisas): Med blandade randvillkor enligt (10)
a) Lös differentialekvationen (8) analytiskt då $f(x) = 0$ och $\gamma = 1, \delta = e^2 - e^{-2}$.
b) Gör den förberedande uppgiften för U2 i Matlab Grader (FRIVILLIG).
c) Diskretisera intervallet [0 1] med $N = 5$ delintervall och ersätt $u''(x)$ i (8) med differensapproximationen (3). Diskretisera det högra randvillkoret med approximationen (7). Ställ upp det resulterande linjära ekvationssystemet som ska lösas.

Felet i den numeriska lösningen kan beräknas med diskret 2-norm. För en vektor $v = [v_0, v_1, \ldots, v_N]$ med $N + 1$ komponenter är den diskreta 2-normen definierad som

$$\|v\|_2 = \left(\frac{1}{N+1} \sum_{k=0}^{N} |v_k|^2 \right)^{1/2}. \tag{12}$$

U2: Med blandade randvillkor enligt (10), $f(x) = 0$, $\gamma = 1, \delta = e^2 - e^{-2}$:
a) Diskretisera randvärdesproblemet enligt ovan och formulera det linjära ekvationssystemet som ska lösas på formen (11) för godtyckligt N.
b) Skriv en funktion `diskretisering.m` som givet antal diskretiseringsintervall N returnerar systemmatrisen A och högerledet $\mathbf{F}$ för det diskretiserade problemet. Funktionen testas som en obligatorisk uppgift i Matlab Grader.
c) Skriv ett Matlab-program som beräknar numeriska approximationer för $N = 50, 100, 200, 400, 800, 1600$. Beräkna det diskreta 2-felet i de numeriska approximationerna och beräkna noggrannhetsordning för metoden baserat på dessa resultat. Gör en tabell som innehåller uppmätta fel och beräknade värden på noggrannhetsordningen. För att verifiera att den numeriska lösningen är korrekt, plotta felet över rumsintervallet. Felet bör vara i storleksordningen 10^{-6} om du fått det rätt.

Värmeledningsekvationen i en dimension

Antag att vi har en metallstav på intervallet $x \in [0\ 1]$. Värmeledningsekvationen beskriver hur temperaturen i staven förändras över tid när den genomströmmas av ett värmeöverflöde. Värmeledningsekvationen är en partiell differentialekvation som i en dimension ges av

$$\frac{\partial u}{\partial t} = d\frac{\partial^2 u}{\partial x^2} + f(x,t), \tag{13}$$

för $t \geq 0$ och här för $x \in [0, L]$. $u(x,t)$ är temperaturen i staven vid position x och vid tiden t, d är värmeföringskoefficienten och $f(x,t)$ är en källterm som kan bero på både rumsvariabeln x och på tiden t.
För att kunna lösa värmeledningsekvationen numeriskt behöver vi ett begynnelsevillkor,

$$u(x,0) = u_0(x).$$

Vi behöver också randvillkor för temperaturen vid ändpunkterna $x = 0, L$. Ett Dirichlet villkor betyder fysiskt att stavens ände hålls vid en fix temperatur. Ett homogent Neumann villkor såsom $\frac{\partial u(0,t)}{\partial x} = 0$, har den fysikaliska tolkningen att stången är isolerad vid denna ände, vilket innebär att ingen värme läcker ut.
För en partiell differentialekvation som värmeledningsekvationen vill vi diskretisera både i *rummet* (med avseende på x) och i *tiden* (med avseende på t). På samma sätt som vi diskretserade ranvärdesproblemet kan vi kan först diskretisera andraderivatan i differential ekvationen med finita differensapproximationer. Vi får de semi-diskreta ekvationerna

$$\frac{dU_j(t)}{dt} = d\frac{U_{j+1}(t) - 2U_j(t) + U_{j-1}(t)}{h^2} + f(x_j, t), \quad j = 1, \ldots, N-1, \tag{14}$$

där varje $U_j(t)$ är en kontinuerlig funktion av t. Detta kompletteras med aktuella randvillkor för att bestämma alla $U_j(t)$, $j = 0, \ldots, N$.
Vi kan nu använda metoder som vi tidigare lärt oss för att diskretisera detta system av ODE:er. Tolkningen är något annorlunda jämfört med systemen av ODE:er tex i projekt 1B där de olika komponenterna beskrev två positioner och två hastigheter. Här beskriver alla komponenterna i den numeriska lösningen temperaturen i staven, men vid olika positioner i rummet.

När vi diskretiserar i tiden så inför vi ett tidssteg Δt och diskreta tidsnivåer $t_n = n\Delta t$, $n = 0, 1, 2, \ldots$. Vi betecknar nu vektorn med den numeriska approximationen med

$$\mathbf{U}^n = [U_0(t_n)\, U_1(t_n) \ldots U_N(t_n)]^T.$$

Vi börjar med att betrakta Dirichlet randvillkor

$$u(0,t) = T_L(t), \quad u(L,t) = T_R(t). \tag{15}$$

U3: a) Diskretisera värmeledningsekvationen (13) på $x \in [0, L]$ med randvillkor (15). Använd central differens (3) för diskretisering i rummet och Euler framåt för diskretisering i tiden. Formulera det diskreta problemet för hur du bestämmer hela vektorn $\mathbf{U}^n$ för godtyckligt N och tidsstegsstorlek Δt.
b) Låt $f(x,t) = 0$, $d = 1$, $T_L = 0$, och $T_R = \sin(4\pi L)TT(t)$, där $u(x,t) = XX(x)TT(t)$ är den analytiska lösningen. Med begynnelsevillkoret $u_0(x) = \sin(4\pi x)$ blir den analytiska lösningen $u(x,t) = \sin(4\pi x)e^{-16\pi^2 t}$.
Skriv ett Matlabprogram som löser värmeledningsekvationen enligt ovan för ett tidsintervall $t \in [0\ T]$, givet ett antal tidssteg M ($\Delta t = T/M$), och N. Kör koden med $L = 1.2$ och sluttid $T = 0.03$ för $N = 100$ och $\Delta t = 10^{-5}$. För att verifiera din numeriska lösning: plotta den numeriska lösningen och skillanden mellan din numeriska approximation och den analytiska lösningen som funktion av både x och t (se t ex `surf` i Matlab och extrauppgift i Matlab Grader). Det maximala felet i maxnorm bör vara ungefär $4 \cdot 10^{-4}$ om du fått det rätt.
c) Vad är stabilitetsvillkoret för tidssteget, dvs Δt_{max} i $\Delta t \leq \Delta t_{max}$? Vad händer om du väljer ett något större tidsteg än Δt_{max}? Testa för några olika värden på N och verifiera att det teoretiska villkoret överensstämmer med beteendet du ser i praktiken.

Implicit metod

Eftersom stabilitetsvillkoret för tidssteget blir mycket strikt när man löser värmeledningsekvationen med en explicit metod vill man gärna använda en implicit metod för numerisk lösning. Crank-Nicolson är en andra ordningens noggrann metod i både rummet och tiden. Diskretisering av (13) med Crank-Nicolson ges för rumspunkt x_j av

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{d}{2}\left(\frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{h^2} + \frac{U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}}{h^2}\right) + \frac{1}{2}\left(F_j^n + F_j^{n+1}\right), \tag{16}$$

där $U_j^n = U_j(t_n)$, $F_j^n = f(x_j, t_n)$, $t_n = n\Delta t$ och $j = 1, 2, \ldots, N-1$.

U4: a) Diskretisera värmeledningsekvationen (13) med randvillkor (15) med Crank-Nicolsons metod. Formulera det diskreta problemet för hur du bestämmer hela vektorn $\mathbf{U}^n$ för godtyckligt N och tidsstegsstorlek Δt.
b) Skriv ett Matlabprogram som löser värmeledningsekvationen enligt ovan för ett tidsintervall $t \in [0\ T]$, givet ett antal tidssteg M ($\Delta t = T/M$), och N.
c) Sätt data som givet i **U3b)**. Vad förväntar du dig vad gäller stabilitet för Crank-Nicolsons metod? Gör några experiment som visar att metoden beter sig som du förväntar dig enligt teorin.

d) Gör en konvergensstudie där du låter Δt vara proportionell mot h. Använd diskret 2-norm och redovisa felen och uppmätt noggrannhetsordning vid tiden $T = 0.03$. Vad förväntar ni er för noggrannhetsordning i tid och rum för Crank-Nicolsons metod? Verifiera att resultaten stämmer med teorin.

Inför Canvasquiz

- Canvasquizet kommer bestå av uppgifter baserat på projektuppgifterna. Ni kommer behöva modifiera er kod, exekvera den och svara på frågor om resultaten. Frågorna kan också vara relaterade till teorin i projektet och till uppställningen av systemen i uppgifterna U2 a), U3 a) och U4 a). Ni bör därför formulera systemen så att de enkelt går att modifiera om t ex randvillkor och initialvillkor förändras. Samma sak gäller för koden, försök att skriva en kod som är så enkel som möjligt att modifiera. Detta gör ni t ex genom att definiera variabler (bara en gång!) och anonyma funktioner istället för att hårdkoda in värden. (Exempel: använd variabler `right` och `left` för att definiera höger och vänster randvillkor och låt sedan högerledet i den linjära ekvationssystemet bli ett uttryck baserat på dessa variabler.) Försök även att undvika koduppprepning i den mån det är möjligt (t ex använd for-slinga vid konvergensstudier). På så sätt slipper ni göra ändringar i koden på fler ställen än nödvändigt.
- Individuella obligatoriska uppgifter ska göras i Matlab Grader innan quizet. Dessa kan med fördel göras tidigt, som en förberedelse för projektarbetet.
- Matlabkoden ska lämnas in gruppvis (grupper "Projekt 2A") i Canvas innan quizet. Vi kommer provköra koden för att se att den fungerar. Koden kommer även genomgå plagiatkontroll.