KTH Matematik

Tentamen del 2
SF1511, 2017-03-16, kl 8.00-11.00,
Numeriska metoder och grundläggande programmering

Del 2, Max 50p + bonuspoäng (max 4p).
Inga hjälpmedel. Rättas endast om del 1 är godkänd.
Betygsgränser inkl bonuspoäng: 10p D, 20p C, 30p B, 40p A.

P1. Givet differentialekvationen

$$\frac{d^3y}{dx^3} - \cos(x)\frac{dy}{dx} - \alpha\frac{d^2y}{dx^2} = 0, \qquad y(0) = 10, \quad \frac{dy}{dx}(0) = 0, \quad \frac{d^2y}{dx^2}(0) = 1.$$

(4p) **a.** Skriv om problemet till ett system av differentialekvationer av första ordningen.

(6p) **b.** Skriv ett MATLAB-program som löser det omskrivna problemet för $\alpha = 0.1$ för x på intervallet 0 till 5 och ritar upp funktionen $w(x) = y(x) + 5y'(x)$.

(7p) **c.** När ett program som löser differentialekvationen kördes för $\alpha = 0.05$ fick man $y(5)$ till 17.46407. När α ändrades till 0.5 blev $y(5) = 24.77647$. Du vill hitta ett α som ger $y(5) = 20$. Man kan göra detta på flera olika sätt. Beskriv en algoritm, tex i form av ett MATLAB-program, som med fem decimalers noggrannhet hittar α. Diskutera hur effektiv din algoritm är. Alltför ineffektiva metoder ger inte full poäng.

P2. Ändarna av en två meter lång stav hålls vid de konstanta temperaturerna 20°C respektive 100°C. Den stationära temperaturfördelningen $u(x)$ i staven bestäms då av differentialekvationen

$$-k\frac{d^2u}{dx^2} = e^x, \qquad u(0) = 20, \quad u(2) = 100,$$

där k är stavens värmeledningsförmåga.

(10p) **a.** Formulera en andra ordningens finit differensmetod för detta randvärdesproblem. Låt $\mathbf{u} = (u_0, \ldots, u_{n+1})^T$ eller $\mathbf{u} = (u_1, \ldots, u_n)^T$ (välj det du föredrar) innehålla approximationerna $u_j \approx u(x_j)$, där $x_j = hj$ och $h = 2/(n+1)$. Metoden leder till ett linjärt ekvationssystem $A\mathbf{u} = \mathbf{b}$ av storlek $n \times n$. Härled uttryck för elementen i A-matrisen och i högerledet $\mathbf{b}$ när $n = 3$. (Ekvationssystemet behöver inte lösas.)

(10p) **b.** Implementera metoden i MATLAB för fallet $k = 5$. Programmet ska räkna ut vektorn $\mathbf{u}$. Det ska vara generellt och kunna anropas med olika värden på n.

(4p) **c.** Utöka ditt MATLAB-program så att det också beräknar en andra ordningens approximation av $u'(1)$.

Var god vänd

P3. Givet $I = \int_0^4 f(x)dx$ med värden enligt tabellen.

x	0	1	2	3	4
f(x)	200	250	281,5	299,5	300

I_h betecknar approximationen av I, då I beräknas med trapetsregeln med steglängden h.
Om någon deluppgift behöver resultatet från en föregående deluppgift och du inte har egna resultat, får du använda en rimlig uppskattning.

(1p) **a.** Beräkna I_4 för hand.

(1p) **b.** Beräkna I_2 för hand.

(1p) **c.** R_2 betecknar en Richardsonextrapolation utifrån I_4 och I_2. Beräkna R_2 för hand.

(1p) **d.** Beräkna I_1 för hand.

(1p) **e.** R_1 betecknar en Richarsonextrapolation utifrån I_2 och I_1. Beräkna R_1 för hand.

(2p) **f.** Gör en Richardsonextrapolation för hand utifrån R_2 och R_1.

(2p) **g.** Skriv ett matlabprogram som beräknar I_h, där steglängden h är en variabel som tilldelas ett värde i början av programmet. Antag att det i samma mapp finns en färdigskriven fil funk.m med indata x och utdata f(x) enligt ovan, d.v.s fAvx = funk(x).
Ovanstående tabell ska alltså inte användas.