Tentamen i grundkurs i numeriska metoder - del 2

Rättas endast om del 1 är godkänd. Betygsgräns (inkl bonuspoäng): 10p D, 20p C, 30p B, 40p A. Maximal poäng 50 + bonuspoäng (max 4p). Miniräknare är ej tillåten på denna tentamen. Svar skall motiveras och uträkningar redovisas. Korrekt svar utan motivering eller med felaktig motivering medför poängavdrag. Då algoritmbeskrivning begärs, avses om inte annat anges beskrivning i Matlab. Eftersom miniräknare ej är tillåten är det tillåtet att lämna enkla beräkningsuttryck oförenklade.

Uppgift 1 Man vill lösa ekvationssystemet

$$\begin{aligned} 10\sin(x+y)+xy &= 5 \\ \cos(xy)+10y &= 0 \end{aligned}$$

med Newtons metod.

(a) **3p** Inför lämpliga beteckningar och formulera Newtons metod för detta ekvationssystem

(b) **4p** Skriv ett program som bestämmer lösningen till den första ekvationen när $y=0$.

(c) **5p** Använd (b) som startgissning (dvs vi antar att en lösning har ett litet y-värde) och skriv ett MATLAB-program som löser det olinjära ekvationssystemet.

Uppgift 2 Följande differentialekvation ska lösas

$$y'''(x)+\alpha y(x)^2=\beta x,\ \ 1\leq x\leq 4.$$

där $\alpha=0.1$. Antag att begynnelsevärdena är givna som $y(1)=1, y'(1)=0.5$, $y''(1)=4$.

(a) **6p** Antag att $\beta=3$. Skriv ett MATLAB-program som löser differentialekvationen på det givna intervallet. Basera programmet på framåt Euler med N steg och steglängd h. Skriv programmet i den här funktionen:

```
function [y4]=solve_ode(y1,yp1,ypp1,beta,N)
# Löser differentialekvationen med givna begynnelsedata
# y1 är begynnelsevillkor y(1)=y1
# yp1 är begynnelsevillkor y'(1)=yp1
# ypp1 är begynnelsevillkor y''(1)=ypp1
# beta är parameter i differentialekvationen
# y4 är approximationen vid y(4)
# N är antalet diskretiseringsintervall
```

(b) **4p** Skriv ett MATLAB-program som med hjälp av `solve_ode` bestämmer β så att $y(4)=0$. Bestäm startgissningar/startgissning genom att lösa problemet exakt för $\alpha=0$.

(c) **5p** Med vilken noggrannhetsordning approximerar metoden i (a) värdet $y(4)$? (Ingen härledning krävs.) Skriv ett program som anropar `solve_ode` och använder Richardsonextrapolation för att öka noggrannheten med $h=0.4$ och $h=0.2$.

Var god vänd!

Uppgift 3 Genom mätningar har vi fått följande datavärden

t	-1	-0.5	-0.25	0	0.25	1
y	9.3	5.3	5.1	4.9	5.4	11.4

(a) **6p** Skriv ett MATLAB-program som bestämmer minstakvadratanpassningen av

$$g(t) = \alpha_1 + \alpha_2 t + e^{\alpha_1 t^2}$$

OBS: Om du använder MATLAB-operatorn \ måste du ange vilket matematiskt problem som MATLAB löser i detta steg.

(b) **5p** Utvidga ditt MATLAB-program i (a) så att det bestämmer minimum av minstrakvadratanpassningen. Ange tydligt hur du kommit fram till startvärden. Denna uppgift kan lösas även om (a) inte lösts.

Uppgift 4 Ingenjören Ingvar har utvecklat en numerisk metod för att lösa linjära ekvationssystem med n obekanta när matrisen är tridiagonal, och har gjort det tillgängligt i MATLAB med programmet:

```
function x=ingvars_metod(T,b,n)
% T är en tridiagonal matris
% b är högerledvektorn i det linjära ekvationssystemet  T x = b
% n är antalet obekanta i stystemet
% x är lösningen till ekvationssystemet
```

På en viss dator kommer Ingvar fram till att beräkningstiden för hans metod kan väl uppskattas till $\alpha n + \beta$ millisekunder där $\alpha = 1$ och $\beta = 10$. Han vill använda det för att lösa differentialekvationen

$$\mathbf{y}'(t) = T\mathbf{y}(t), \quad \mathbf{y}(0) = \mathbf{y}_0,$$

vid tidpunkt $t = 1$ där T är en tridiagonal $n \times n$ matris och $\mathbf{y}_0$ en given vektor. Matrisen T och vektorn $\mathbf{y}_0$ är tillgängliga i MATLAB. Vi antar att beräkningstiden för att multiplicera T med en vektor är $\alpha n/3$ och att addera två vektorer är $\alpha n/10$. Ett steg av bakåt Euler innebär lösningen av ett linjärt ekvationssystem.

(a) **4p** Skriv ett MATLAB program som beräknar lösningen med framåt Euler med $N = 10$ steg och $h = 1/N$.

(b) **4p** Skriv ett MATLAB program som beräknar lösningen med bakåt Euler med hjälp av Ingvars metod med $N = 10$ steg och $h = 1/N$.

(c) **2p** Vad är beräkningstiden för (a) respektive (b)?

(d) **2p** Ingvar upptäcker att problemet är styvt och att man behöver mycket finare diskretisering med framåt Euler för samma noggrannhet med bakåt Euler. Om man med framåt Euler väljer $N = 1000$ behöver man för samma noggrannhet endast $N = 10$ med bakåt Euler. Vilken metod är snabbast för att uppnå den noggrannheten för stora n-värden? Ge ett tydligt resonemang med tydliga antaganden om beräkningstider.