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). Betygen för A och B reduceras med wiki-bonuspoängen (max 1p). 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. Stora vektorer och matriser behöver kan beskrivas med punkter ($\cdots$, etc) om strukturen framgår tydligt.

Uppgift 1 (13p) Vi bestämma värden på de grekiska variablerna så att vi anpassar funktionen

$$f(t) = \alpha t^2 + \sin(\beta_1 t + 1)e^{\gamma_1 t} + \sin(\beta_2 t + 2)e^{\gamma_2 t} + \cdots + \sin(\beta_{10} t + 10)e^{\gamma_{10} t}$$

på bästa sätt (i minstakvadratmening) till datapunkter

n	1	2	3	$\cdots$	100
t	0.1	0.2	0.3	$\cdots$	10.0
y	3.1	3.5	2.9	$\cdots$	13.10

Detta genomförs genom formulering som olinjärt överbestämt ekvationssystem.

(a) Formulera problemet som ett överbestämt olinjärt ekvationssystem genom att ange $\mathbf{g}$, $z_1, \ldots, z_N$ och N så att

$$\mathbf{g}(z_1, \ldots, z_N) \approx \mathbf{0}.$$

(b) Skriv ett matlab-program med Gauss-Newtons metod löser detta problem. Använd vektorn som innehåller ettor (`ones(N,1)`) som startvärde. Jacobianen behöver ej programmeras utan skrivs som formler. Antag att t-värden och y-värden finns sparade i vektorerna `t` och `y`.

Lösning: a) Vi ansätter $N = 21$ och

$$\begin{aligned} z_1 &= \alpha \\ z_2 &= \beta_1 \\ &\vdots \\ z_{11} &= \beta_{10} \\ z_{12} &= \gamma_1 \\ &\vdots \\ z_{21} &= \gamma_{10} \end{aligned}$$

Därmed blir

$$\mathbf{g}(z_1,\ldots,z_N)=\begin{bmatrix} z_1t_1^2+\sin(z_2t_1+1)e^{z_{12}t_1}+\sin(z_3t_1+2)e^{z_{13}t_1}+\cdots+\sin(z_{11}t_1+10)e^{z_{21}t_1}-3.1 \\ \vdots \\ z_1t_{100}^2+\sin(z_2t_{100}+1)e^{z_{12}t_{100}}+\sin(z_3t_{100}+2)e^{z_{13}t_{100}}+\cdots+\sin(z_{11}t_{100}+10)e^{z_{21}t_{100}}-10.00 \end{bmatrix}$$

b)

Jakobianen blir

$$J(z)=\begin{bmatrix} t_1^2 & t_1\cos(z_2t_1+1)e^{z_{12}t_1} & \cdots & t_1\sin(z_{11}t_1+10)e^{t_1z_{21}} \\ \vdots & \vdots & & \vdots \\ t_{100}^2 & t_{100}\cos(z_2t_{100}+1)e^{z_{12}t_{100}} & \cdots & t_{100}\sin(z_{11}t_{100}+10)e^{t_{100}z_{21}} \end{bmatrix}$$

som vi säger är sparad i funktionen `J(z)`. Gauss-Newton blir nu

```
N=1;
z=ones(N,1);
TOL=1e-10
while (norm(dz)<TOL)
   dz=-J(z)\g(z);
   z=z+dz;
end
```

Uppgift 2 (18) Följande system av differentialekvationer är givna

$$\mathbf{y}'(t)=\begin{bmatrix}-2 & 1\\ 1 & \alpha\end{bmatrix}\mathbf{y}(t)+\begin{bmatrix}1\\2\end{bmatrix},\quad \mathbf{y}(0)=\begin{bmatrix}1\\1\end{bmatrix} \tag{1}$$

där

$$\mathbf{y}(t)=\begin{bmatrix}y_1(t)\\y_2(t)\end{bmatrix}$$

(a) Skriv ett program som beräknar $\mathbf{y}(h),\mathbf{y}(2h),\ldots,\mathbf{y}(10)$ med Euler framåt för ett givet värde α. Formulera det som implementationen av en funktion

```
function Y=compute_y(alpha,n,t)
% Beräknar en matrix Y av storlek 2 x n vars kolumner motsvarar
% lösningen till differentialekvationen för x-värden h,2h,...,t.
```

(b) Skriv ett program som anropar `compute_y` och beräknar

$$\int_0^{10} y_1(t)\,dt$$

för $h=0.1$ och $\alpha=0$. Använd en numerisk metod som har noggrannhetsordning (minst) två. Du får inte använda `quad` eller `integral`.

(c) Anpassa Newtons metod i flera variabler så att metoden beräknar ett α-värde och t värde så att $y_1(t)=0$ och $y_2(t)=1$. En sådan lösning finns nära $t=10$ och $\alpha=-2$.

Lösning: a)

```
function Y=compute_y(alpha,n,t)
    y0=[1;1];
    h=t/n;
    Y=zeros(2,n);
    Y(:,1)=y0+h*A*y0;
    for i=2:n
       Y(:,i)=Y(:,i-1)+h*A*Y(:,i-1);
    end
```

b) (Huvudidé) Använd trapetsregeln på första raden i `Y`. Kom ihåg att lägga till halva första punkten och dra bort sista punkten om man summerar `Y(1,:)`.

```
Y=compute_y(0,n,10)
yy=Y(1,:)
h=10/n
T=sum(yy)*h+1*h/2-yy(end)*h/2;
```

c) Detta är vår målfunktion som vi vill hitta nollan till

```
f=@(x) compute_y(x(1),n,x(2)-[0;1]
```

Beräkna Jacobianmatrisen numeriskt:

```
d=sqrt(eps); % Något litet
J=@(x) [(compute_y(x(1)+h,n,x(2))-compute_y(x(1)-h,n,x(2))/(2*h),(compute_y(x(1),n,x(2)+h)-
```

Nu kan vi köra den Newton's metod i flera variabler som vanligt

```
x=[10;-2];
dx=inf;
while norm(dx)>TOL
   dx=J(x)\f(x);
   x=x-dx;
end
```

Uppgift 3 (19) Givet är denna finita differensapproximation

$$f'''(t) \approx \frac{-f(t)+3f(t+h)-3f(t+2h)+f(t+3h)}{h^3}$$

(a) Härled approximationens noggrannhetsordning.

(b) Använd denna approximation för att generalisera och anpassa matrismetoden till randvärdesproblemet

Obs: Detta är ett tredje ordningens randvärdesproblem.

$$\begin{aligned} y'''(x) &= \pi + y(x), \quad 0 \le x \le 1 \\ y(0) &= 2, \\ y(1) &= 3, \\ y'(1) &= 0. \end{aligned}$$

För att approximera $y'(1)$ får du använda (valfri) finit differensmetod. Härled ett linjärt ekvationssystem

$$A\mathbf{y} = \mathbf{b}$$

där

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix} \approx \begin{bmatrix} y(h) \\ y(2h) \\ \vdots \\ y(1-h) \end{bmatrix}$$

och ange tydligt vad $A \in \mathbb{R}^{(n-1)\times(n-1)}$ och $\mathbf{b} \in \mathbb{R}^{n-1}$ är. Du behöver inte lösa det linjära ekvationssystemet. Ingen programmering krävs för denna uppgift.

Lösning: När vi använder approximationen för

- $x = 0$

$$-y_0 + 3y_1 - 3y_2 + y_3 = h^3(\pi + y_0)$$

dvs

$$3y_1 - 3y_2 + y_3 = h^3(\pi + 2) + 2$$

- $x = h$

$$-y_1 + 3y_2 - 3y_3 + y_4 = h^3(\pi + y_1)$$

dvs

$$-y_1 - h^3 y_1 + 3y_2 - 3y_3 + y_4 = h^3\pi$$

- $x = 2h$

$$-y_2 - h^3 y_2 + 3y_3 - 3y_4 + y_5 = h^3\pi$$

- $\vdots$

- $x = h(n-3)$

$$-y_{n-3} + 3y_{n-2} - 3y_{n-1} + y_n = h^3(\pi + y_{n-3})$$

dvs

$$-y_{n-3} - h^3 y_{n-3} + 3y_{n-2} - 3y_{n-1} = h^3\pi - 3$$

- $x = h(n-2)$

$$-y_{n-2} + 3y_{n-1} - 3y_n + y_{n+1} = h^3(\pi + y_{n-2})$$

Med hjälp av approximationen $y'(1) = 0$ får vi $(y_{n+1} - y_n)/h = 0$ dvs

$$-y_{n-2} - h^3 y_{n-2} + 3y_{n-1} = h^3\pi - 2y_n = h^3\pi + 6$$

Ekvationerna ovan kan formuleras som

$$\begin{bmatrix} 3 & -3 & 1 & & & & & \\ -1-h^3 & 3 & -3 & 1 & & & & \\ & & -1-h^3 & 3 & -3 & 1 & & \\ & & & \ddots & \ddots & \ddots & \ddots & \\ & & & & -1-h^3 & 3 & -3 & 1 \\ & & & & & -1-h^3 & 3 & -3 \\ & & & & & & -1-h^3 & 3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} h^3(\pi+2)+2 \\ h^3\pi \\ h^3\pi \\ \vdots \\ h^3\pi \\ h^3\pi-3 \\ h^3\pi-6 \end{bmatrix}$$