

Tentamen del 2

Numeriska metoder SF1544 och BE3003

8.00-11.00 13/3 2020

Skriv din anonymiseringskod på varje inlämnat blad.

Inga hjälpmedel är tillåtna (ej heller miniräknare).

Det finns två varianter av uppgift 2. Endast en av dem får lösas.

Råd för att undvika poängavdrag: Skriv lösningar med fullständiga meningar och utförliga motiveringar.

Del 2 rättas endast om del 1 är godkänd.

Betygsgränser: D 10, C 20, B 30 och A 40 poäng, av maximalt 50 poäng på del 2.

1. Vi vill beräkna en lösning till

$$f(x) = \sin(x) - \cos(x) = 0.$$

För att lösa problemet har vi följande två fixpunktsiterationer givna

$$\text{i)} \quad x_{n+1} = x_n - \sin(-x_n) - \cos(x_n)$$

$$\text{ii)} \quad x_{n+1} = x_n - e^{-x_n}(\sin(x_n) - \cos(x_n)).$$

Tips: Det gäller att $\sin(\frac{\pi}{4}) = \cos(\frac{\pi}{4})$.

1a. (3p) Visa att varje fixpunkt till i) och ii) också är en rot till $f(x) = 0$.

För det första fallet använder vi att $\sin(x) = -\sin(-x)$ vilket ger oss att

$$x = x - \sin(-x) - \cos(x) \Leftrightarrow x = x + \sin(x) - \cos(x) \Leftrightarrow \sin(x) - \cos(x) = 0.$$

I det andra fallet har vi att

$$x = x - e^{-x}(\sin(x) - \cos(x)) \Leftrightarrow \sin(x) - \cos(x) = 0,$$

eftersom $e^{-x} \neq 0$ för alla värden på x .

1b. (7p) Givet en startgissning $0 < x_0 < \pi/2$, avgör om iterationerna ovan konvergerar mot $\pi/4$ för alla värden på x_0 , endast för vissa värden på x_0 (ange i sådant fall vilka) eller inte för några värden på x_0 .

Om vi skriver en fixpunktsiteration på formen $x_{n+1} = g(x_n)$ och kallar fixpunkten α (dvs. här är $\alpha = \pi/4$) så gäller enligt medelvärdessatsen att

$$x_{n+1} - \alpha = g'(c)(x_n - \alpha), \quad (\text{A})$$

där c är någon punkt mellan x_n och α .

För fixpunktsiteration i) har vi att

$$g(x) = x + \sin(x) - \cos(x) \quad \Rightarrow \quad g'(x) = 1 + \cos(x) + \sin(x).$$

Eftersom $g'(\pi/4) > 1$ konvergerar inte denna fixpunktsmetod för något $x_0 \neq \pi/4$. Om $x_0 = \pi/4$ konvergerar metoden trivialt eftersom $x_1 = g(x_0) = x_0$ (förutsatt att inga avrundningsfel är med i bilden).

För fixpunktsiteration ii) har vi att

$$g(x) = x - e^{-x}(\sin(x) - \cos(x))$$

och

$$g'(x) = 1 + e^{-x}(\sin(x) - \cos(x)) - e^{-x}(\cos(x) + \sin(x)) = 1 - 2e^{-x}\cos(x).$$

Eftersom $0 < \cos(x) < 1$ och $0 < e^{-x} < 1$ för alla $0 < x < \pi/2$ får vi att $|g'(c)| < K < 1$ för alla c mellan x_0 och α . Ekvation (A) ger därmed att

$$\|x_1 - \alpha\| \leq K\|x_0 - \alpha\|.$$

Därmed kommer x_1 att ligga närmare α än x_0 , vilket ger att

$$\|x_2 - \alpha\| \leq K\|x_1 - \alpha\|.$$

Det följer att $\|x_n - \alpha\| \leq K^n\|x_0 - \alpha\|$, och fixpunktsiterationen ii) konvergerar därför för alla startvärden i det givna intervallet.

2. (15p) Matrisen

$$A = \begin{pmatrix} 3 & 2 \\ 2 & 3 \end{pmatrix}$$

har egenvärdena $\lambda_1 = 1$ och $\lambda_2 = 5$, med tillhörande egenvektorer

$$\mathbf{v}_1 = \begin{pmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \text{ och } \mathbf{v}_2 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}.$$

Kalla resultatet av n steg med potensmetoden med startvektor

$$\mathbf{x}_0 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

för $\mathbf{x}_n$. Bestäm ett n sådant att $\|\mathbf{x}_n - \mathbf{v}_2\| \leq 10^{-7}$, där $\|\cdot\|$ betecknar den Euklidiska normen.

Genom att skriva startvektorn $\mathbf{x}_0$ i egenvektorsbasen har vi att

$$\mathbf{x}_0 = \frac{1}{\sqrt{2}}\mathbf{v}_1 + \frac{1}{\sqrt{2}}\mathbf{v}_2.$$

Liksom i beviset av konvergens av potensmetoden har vi att

$$\mathbf{x}_n = \frac{\lambda_1^n \mathbf{v}_1 + \lambda_2^n \mathbf{v}_2}{\|\lambda_1^n \mathbf{v}_1 + \lambda_2^n \mathbf{v}_2\|} = \frac{\frac{\lambda_1^n}{\lambda_2^n} \mathbf{v}_1 + \mathbf{v}_2}{\|\frac{\lambda_1^n}{\lambda_2^n} \mathbf{v}_1 + \mathbf{v}_2\|}.$$

(Faktorn $1/\sqrt{2}$ kan divideras bort eftersom vektorn ändå normaliseras.)

Eftersom egenvektorerna $\mathbf{v}_1$ och $\mathbf{v}_2$ är ortogonala gäller

$$\|\mathbf{x}_n - \mathbf{v}_2\| \leq \left\| \frac{\lambda_1^n}{\lambda_2^n} \mathbf{v}_1 + \mathbf{v}_2 - \mathbf{v}_2 \right\| = \frac{\lambda_1^n}{\lambda_2^n} = \frac{1}{5^n}.$$

Vi har alltså tillräckligt bra approximation om

$$\frac{1}{5^n} \leq 10^{-7},$$

dvs. om

$$n \geq \frac{7}{\lg 5},$$

där $\lg$ är 10-logaritmen.

Alternativ uppgift 2. (15p) Formulera och bevisa en sats om kvadratisk konvergens för Newtons metod.

Se Sats 1.11 i Sauer.

3. Vi önskar lösa begynnelsevärdesproblemet

$$u'(t) = -\sin(\alpha u(t) + \beta), \quad u(0) = 2, \tag{B}$$

där $\alpha = 10$ och $\beta = 1/8$.

3a. (3p) Formulera metoden Bakåt Euler (Implicit Euler) för ekvation (B).

Bakåt Euler för ekvationen $u' = f(t, u)$ lyder

$$u_{n+1} = u_n + hf(t_{n+1}, u_{n+1}) \quad n = 0, 1, 2, \dots$$

Om vi skriver (B) som $u'(t) = f(t, u)$ får vi att $f(t, u) = -\sin(\alpha u(t) + \beta)$ och Bakåt Euler blir då för den aktuella ekvationen

$$\begin{aligned} u_0 &= 2 \\ u_{n+1} &= u_n - h(\sin(\alpha u_{n+1} + \beta)), \quad n = 0, 1, 2 \dots \end{aligned} \tag{B}$$

- 3b.** (7p) I varje tidssteg med Bakåt Euler behöver en icke-linjär ekvation lösas. Formulera Newtons metod för denna ekvation. Specificera en lämplig startpunkt för Newtoniterationerna i varje tidssteg.

I varje tidssteg behöver vi lösa en olinjär ekvation för u_{n+1} . Från (B) får vi att den olinjära ekvationen ges av

$$f(u_{n+1}) = u_{n+1} + h(\sin(\alpha u_{n+1} + \beta)) - u_n = 0$$

Newtons metod för $f(u_{n+1}) = 0$ ges av

$$u_{n+1}^{k+1} = u_{n+1}^k - \frac{f(u_{n+1}^k)}{f'(u_{n+1}^k)} = u_{n+1}^k - \frac{u_{n+1}^k + h(\sin(\alpha u_{n+1}^k + \beta)) - u_n}{1 + h\alpha \cos(\alpha u_{n+1}^k + \beta)}, k = 0, 1, 2 \dots$$

En lämplig startgissning till Newtons metod i varje tidssteg är $u_{n+1}^0 = u_n$.

- 3c.** (8p) Skriv ett Matlab-program som beräknar en approximation av $u(3)$, dvs. lösningen till ekvation (B) vid tiden $t = 3$. Matlabprogrammet ska använda Bakåt Euler för tidsdiskretiseringen och Newton i varje tidssteg, och ska skriva ut approximationen av $u(3)$ på skärmen. För full poäng krävs att programmet gör ett test för att se till att absolutbeloppet av felet i approximationen av $u(3)$ är maximalt 10^{-3} . Det får antas känt att Newtoniterationerna i varje tidssteg konvergerar.

Felet i approximationen av $u(3)$ härrör från både Newtons metod och Bakåt Euler. Eftersom Newtons metod konvergerar kvadratisk kan dess fel göras totalt obetydligt med litet beräkningsarbete. Genom att sätta toleransen för Newton till 10^{-10} i Matlabkoden nedan kan vi bortse från Newtonfelet.

```
alpha = 10; beta = 1/8;
T = 3;
h = 0.1;
erru = 1; tol = 1e-10;
uold = 100;
```

```

while erru > 1e-3                                % Kontrollerar felet i u(3)
    n = T/h;                                     % mha steglangdshalvering
    u = 2;
    t = 0;
    tvec = 0;
    uvec = u;
    % Eulers metod
    for ii = 1:n
        u0=u; korr = 1;
        % Newtons metod
        while abs(korr) > tol
            korr = (u+h*sin(alpha*u+beta)-u0)/(1+h*alpha*cos(alpha*u + beta));
            u = u-korr;
        end
        t = t+h;
    end
    erru=abs(u-uold);
    uold = u;
    h = h/2;
end

```

-
- 3d.** (7p) Beskriv en algoritm som med användning av Newtons metod bestämmer parametrarna α och β så att lösningen u till (B) uppfyller $u(1/2) = 1.37$ och $u(1) = 1.54$. Tidsdiskretiseringen till differentialekvationen (B) får göras med valfri metod, och valet av tidssteg kan t.ex. väljas som $h = 1/100$. Startvärden till Newtons metod kan tas som $\alpha = 2$ och $\beta = 3$. Förutom en beskrivning av algoritmen i ord och formler ska du också skriva ett Matlabprogram som implementerar algoritmen.
-

Om vi gör N steg med Framåt Euler på intervallet $0 \leq t \leq 1$ så vill vi lösa ekvationssystemet $\mathbf{F}(\alpha, \beta) = \mathbf{0}$, där

$$\mathbf{F} = \begin{pmatrix} u_N - 1.54 \\ u_{N/2} - 1.37 \end{pmatrix}.$$

De numeriska lösningarna u_N och $u_{N/2}$ approximerar alltså $u(1)$ respektive $u(1/2)$. För att använda Newtons metod behövs Jacobimatrisen, dvs.

$$J = \begin{pmatrix} \frac{\partial u_N}{\partial \alpha} & \frac{\partial u_N}{\partial \beta} \\ \frac{\partial u_{N/2}}{\partial \alpha} & \frac{\partial u_{N/2}}{\partial \beta} \end{pmatrix}.$$

Metoden Framåt Euler för differentialekvationen (B) ges av

$$u_{n+1} = u_n + hf(u_n; \alpha, \beta), \quad (C)$$

där $f(u; \alpha, \beta) = -\sin(\alpha u + \beta)$. Derivering av relationen (C) med avseende på α ger

$$\frac{\partial u_{n+1}}{\partial \alpha} = \frac{\partial u_n}{\partial \alpha} + h \left(\frac{\partial f}{\partial \alpha} + \frac{\partial f}{\partial u_n} \frac{\partial u_n}{\partial \alpha} \right). \quad (D)$$

För att beräkna derivatan $\partial u_{n+1}/\partial \alpha$ behövs alltså $\partial u_n/\partial \alpha$. Denna kan beräknas om $\partial u_{n-1}/\partial \alpha$ är känd osv. Eftersom $\partial u_0/\partial \alpha = 0$ kan vi erhålla $\partial u_N/\partial \alpha$ och $\partial u_{N/2}/\partial \alpha$ genom rekursionen i (D). På samma sätt beräknas derivatorna med avseende på β .

Ett alternativt sätt att beräkna jacobimatrisen J är genom att approximera den med finita differensapproximationer. Nedan följer en Matlab-kod som löser uppgiften med hjälp av rekursionen (D).

```
a=2; b=3;
TOL=1e-7;
N=1e3; %Antal tidssteg
h=1/N;

par=[a; b];
parold=par+1;
while norm(par-parold)>TOL
    parold=par;

    %Best av F
    uvec=zeros(1,N+1);
    uvec(1)=1;
    for n=1:N
        uvec(n+1)=uvec(n)-h*sin(a*uvec(n)+b);
    end
    F=[uvec(N/2+1)-1.37; uvec(N+1)-1.54];

    %Best av Jacobian
    pvec=zeros(1,N+1);
    qvec=pvec;
    for n=1:N
        pvec(n+1)=(1-h*a*cos(a*uvec(n)+b))*pvec(n)-h*uvec(n)*cos(a*uvec(n)+b);
        qvec(n+1)=(1-h*a*cos(a*uvec(n)+b))*qvec(n)-h*cos(a*uvec(n)+b);
    end
    J=[pvec(N/2+1) qvec(N/2+1); pvec(N+1) qvec(N+1)];

    %Newtonsteget
    par=par-J\F;
```

```
    a=par(1); b=par(2);  
end  
  
disp(['Parametrarna ges av a: ', num2str(a),', b:', num2str(b)])
```