

TENTAMEN, DEL 2
SF1544
NUMERISKA METODER, GRUNDKURS IV
Onsdag 7 juni 2023 kl 14.00–17.00

Rättas endast om Del 1 är godkänd. Maximalt antal poäng på denna del är 50.
Betygsgränser: 10 p D, 20 p C, 30 p B, 40 p A.

Du får arbeta med del 1 och del 2 parallellt under hela skrivtiden.

Svar skall motiveras och uträkningar redovisas detaljerat. Korrekt svar utan motivering eller med felaktig motivering medför poängavdrag.

Inga hjälpmedel är tillåtna (ej heller miniräknare).

P1. Den ickeinjära modellen

$$f(t) = e^{at} (bt + ct^2), \quad (\heartsuit)$$

där a , b och c är parametrar, ska anpassas till datapunkterna i tabellen nedan.

t	0.0	0.1	0.2	0.5	1.0	2.0	4.0
f	0.1	0.2	0.7	1.2	3.0	8.5	21.4

- (7 p) **a)** Antag just nu att $a = 0$ och använd de tre mätvärdena för $t = 0$, $t = 1$ och $t = 2$ för att för hand beräkna parametrarna b och c med minstakvadratmetoden och denna linjära modell.
- (7 p) **b)** Använd värdena från uppgift **a)** som startvärden och formulera Gauss–Newtons metod för att bestämma värden på alla tre parametrarna a , b och c . Definiera och redovisa tydligt de storheter som ingår i metoden.
- (9 p) **c)** Skriv ett program som beräknar parametrarna a , b och c med Gauss–Newtons metod och alla mätpunkterna i tabellen ovan. Programmet ska även beräkna minstakvadratsumman och plotta residual-kurvan för mätpunkterna. Programmet ska beräkna parametrarna med minst 4 säkra decimaler.

Lösningsförslag:

- a)** Med $a = 0$ blir modellen $f(t) = bt + ct^2$. Insättning av mätvärdena i modellen ger det linjära ekvationssystemet

$$\underbrace{\begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 2 & 4 \end{bmatrix}}_A \underbrace{\begin{bmatrix} b \\ c \end{bmatrix}}_{\vec{x}} = \underbrace{\begin{bmatrix} 0.1 \\ 3.0 \\ 8.5 \end{bmatrix}}_{\vec{y}}.$$

(Första kolumnen i A är t och andra kolumnen är t^2 .) Systemet är överbestämt och löses med hjälp av normalekvationerna $A^T A \vec{x} = A^T \vec{y}$. Normalekvationerna blir

$$\begin{bmatrix} 5 & 9 \\ 9 & 17 \end{bmatrix} \begin{bmatrix} b \\ c \end{bmatrix} = \begin{bmatrix} 20 \\ 37 \end{bmatrix},$$

med lösning

$$\begin{cases} b = 1.75, \\ c = 1.25. \end{cases}$$

- b) Gauss–Newtons metod är en metod för att bestämma ett ungefärligt nollställe $\vec{F}(\vec{x}) \approx \vec{0}$ då ekvationen är överbestämd, dvs funktionen $\vec{F}$ har fler komponenter än vektorn $\vec{x}$. Den funktion vi vill hitta ett ungefärligt nollställe till här är residualen, $\vec{F}(\vec{x}) = (f(t_i) - y_i)$, där indexet i går från 1 till 7 och $f(t)$ beror på $\vec{x} = (a, b, c)$. Explicit uppskriven blir funktionen

$$\vec{F}(\vec{x}) = \vec{F}(a, b, c) = \begin{bmatrix} e^{at_1}(bt_1 + ct_1^2) - y_1 \\ e^{at_2}(bt_2 + ct_2^2) - y_2 \\ \vdots \\ e^{at_7}(bt_7 + ct_7^2) - y_7 \end{bmatrix},$$

där $\vec{t} = (t_1, \dots, t_7) = (0, 0.1, 0.2, 0.5, 1, 2, 4)$ och $\vec{y} = (y_1, \dots, y_7) = (0.1, 0.2, 0.7, 1.2, 3, 8.5, 21.4)$.

Metoden kräver även Jacobimatrisen

$$J(\vec{x}) = \begin{bmatrix} \frac{\partial \vec{F}}{\partial a} & \frac{\partial \vec{F}}{\partial b} & \frac{\partial \vec{F}}{\partial c} \end{bmatrix} = \begin{bmatrix} t_1 e^{at_1}(bt_1 + ct_1^2) & t_1 e^{at_1} & t_1^2 e^{at_1} \\ t_2 e^{at_2}(bt_2 + ct_2^2) & t_2 e^{at_2} & t_2^2 e^{at_2} \\ \vdots & \vdots & \vdots \\ t_7 e^{at_7}(bt_7 + ct_7^2) & t_7 e^{at_7} & t_7^2 e^{at_7} \end{bmatrix}.$$

Låt startvärdet vara $\vec{x}_0 = (0, 1.75, 1.25)$. Iterationsformeln för Gauss–Newton lyder

$$\vec{x}_{n+1} = \vec{x}_n + \vec{s}_n, \quad n = 0, 1, 2, \dots$$

där uppdateringen $\vec{s}_n$ är minstakvadratlösningen till det överbestämda linjära systemet $J(\vec{x}_n)\vec{s}_n = -\vec{F}(\vec{x}_n)$.

- c) Förslag på program (Matlab):

```
t = [0, 0.1, 0.2, 0.5, 1, 2, 4]';
```

```
y = [0.1, 0.2, 0.7, 1.2, 3, 8.5, 21.4]';
```

```
% Gauss-Newton enligt ovan, välj tolerans för fyra säkra decimaler
tol = 1e-5; s = 1; x = [0, 1.75, 1.25]';
```

```

while norm(s) > tol
    a = x(1); b = x(2); c = x(3);
    F = [exp(a*t).*(b*t+c*t.^2) - y]; % blir vektor
    J = [t.*exp(a*t).*(b*t+c*t.^2), t.*exp(a*t), t.^2.*exp(a*t)]; % matris
    s = -J\F;
    x = x + s
end

% Plotta residualkurvan och beräkna minstakvadratsumman
a = x(1); b = x(2); c = x(3);
r = [exp(a*t).*(b*t+c*t.^2) - y];
plot(t, r, '.-')
mkvsum = sum(r.^2)

```

P2. Vi vill lösa begynnelsevärdesproblemet

$$y'' - 2y' - y = \sin(2\pi t), \quad y(0.0) = 1.0, \quad y'(0.0) = 0.1 \quad (\diamond)$$

fram till $t = 1.0$.

- (3 p) a) Inför lämpliga beteckningar och skriv om ekvationen ($\diamond$) som ett system av första ordningens differentialekvationer på standardform.
- (2 p) b) Formulera *framåt* Euler för systemet i a).
- (2 p) c) Formulera *bakåt* Euler för systemet i a).
- (8 p) d) Använd *bakåt* Euler och steglängden $h = 0.5$ för att beräkna en approximation av $y(1)$ och $y'(1)$, för hand.

Lösningsförslag:

- a) Låt $u = y$ och $v = y'$ så får vi

$$\begin{aligned} u' &= v, & u(0) &= 1, \\ v' &= \sin(2\pi t) + 2v + u, & v(0) &= 0.1. \end{aligned}$$

Vi kan införa

$$\mathbf{U} = \begin{bmatrix} u \\ v \end{bmatrix}, \quad \mathbf{F}(t, \mathbf{U}) = \begin{bmatrix} v \\ \sin(2\pi t) + 2v + u \end{bmatrix}, \quad \mathbf{U}_0 = \begin{bmatrix} 1 \\ 0.1 \end{bmatrix}$$

och skriva systemet som $\mathbf{U}' = \mathbf{F}(t, \mathbf{U})$, $\mathbf{U}(0) = \mathbf{U}_0$.

- b) Framåt Euler för systemet ges av $\mathbf{U}_{i+1} = \mathbf{U}_i + h\mathbf{F}(t_i, \mathbf{U}_i)$ där $\mathbf{U}$ och $\mathbf{F}$ är definierade i lösningen till a). Utskrivet får vi

$$\begin{bmatrix} u_{i+1} \\ v_{i+1} \end{bmatrix} = \begin{bmatrix} u_i \\ v_i \end{bmatrix} + h \begin{bmatrix} v_i \\ \sin(2\pi t_i) + 2v_i + u_i \end{bmatrix}.$$

c) Bakåt Euler för systemet ges av $\mathbf{U}_{i+1} = \mathbf{U}_i + h\mathbf{F}(t_{i+1}, \mathbf{U}_{i+1})$. Utskrivet får vi

$$\begin{bmatrix} u_{i+1} \\ v_{i+1} \end{bmatrix} = \begin{bmatrix} u_i \\ v_i \end{bmatrix} + h \begin{bmatrix} v_{i+1} \\ \sin(2\pi t_{i+1}) + 2v_{i+1} + u_{i+1} \end{bmatrix}.$$

d) För att beräkna $y(1)$ och $y'(1)$ när steglängden $h = 0.5$ behöver vi ta två steg med bakåt Euler. Om vi börjar med att skriva om bakåt Euler på formen

$$\begin{bmatrix} 1 & -h \\ -h & 1 - 2h \end{bmatrix} \begin{bmatrix} u_{i+1} \\ v_{i+1} \end{bmatrix} = \begin{bmatrix} u_i \\ v_i + h \sin(2\pi t_{i+1}) \end{bmatrix}$$

så får vi

$$\begin{bmatrix} u_{i+1} \\ v_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & -h \\ -h & 1 - 2h \end{bmatrix}^{-1} \begin{bmatrix} u_i \\ v_i + h \sin(2\pi t_{i+1}) \end{bmatrix}.$$

Med $h = 0.5$ får vi

$$\begin{bmatrix} u_{i+1} \\ v_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 0 \end{bmatrix}^{-1} \begin{bmatrix} u_i \\ v_i + 0.5 \sin(2\pi t_{i+1}) \end{bmatrix} = \begin{bmatrix} 0 & -2 \\ -2 & -4 \end{bmatrix} \begin{bmatrix} u_i \\ v_i + 0.5 \sin(2\pi t_{i+1}) \end{bmatrix}.$$

Vi har att $u_0 = 1$, $v_0 = 0.1$ vilket ger oss

$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} 0 & -2 \\ -2 & -4 \end{bmatrix} \begin{bmatrix} u_0 \\ v_0 + 0.5 \sin(2\pi t_1) \end{bmatrix} = \begin{bmatrix} 0 & -2 \\ -2 & -4 \end{bmatrix} \begin{bmatrix} 1 \\ 0.1 + 0.5 \sin(\pi) \end{bmatrix} = \begin{bmatrix} -0.2 \\ -2.4 \end{bmatrix}$$

$$\begin{bmatrix} u_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 & -2 \\ -2 & -4 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 + 0.5 \sin(2\pi t_2) \end{bmatrix} = \begin{bmatrix} 0 & -2 \\ -2 & -4 \end{bmatrix} \begin{bmatrix} -0.2 \\ -2.4 + 0.5 \sin(2\pi) \end{bmatrix} = \begin{bmatrix} 4.8 \\ 10 \end{bmatrix}$$

Så $y(1) \approx u_2 = 4.8$ och $y'(1) \approx v_2 = 10$.

(15 p)

Alternativ uppgift P2. Formulera och bevisa en sats om konvergens av fixpunkt-iterationer.

Lösningförslag: Se Sauer Theorem 1.6 (sida 35) eller anteckningarna från Föreläsning 3, 2022-11-24 (Ekvationslösning, fixpunktsmetoden).

Lös antingen P2 eller alternativa uppgift P2, inte båda. Om lösningar till den alternativa uppgiften lämnas in är det de som kommer bedömas.

P3. Badhuset Glädjen tänker bygga en rutsch-kana till sin pool. Vid designen tänker de använda interpolation och anpassa banan till ett par givna punkter i omgivningen. Du som sommarjobbade teknolog får i uppdrag att titta på designen:

Givna punkter är

x	0.00	0.30	2.10	2.40
f	2.00	1.99	0.52	0.50

- (9 p) a) Man vill att första delen av banan, den mellan $x = 0.00$ och $x = 0.30$, ska vara helt rak så att det blir enkelt att kliva på. Därefter ska den mjukt böja ned för att mjukt sluta med en rak del igen mellan $x = 2.10$ och $x = 2.40$, innan de åkande hamnar i vattnet.

Föreslå lämpliga interpolationspolynom för att uppfylla önskemålen.

Ange detaljerat hur banan ska räknas ut, ställ upp alla de ekvationer som måste lösas, men du behöver inte lösa dem om du inte vill.

(Obs: Om de ekvationer du ställer upp är icke-triviala behöver du ange vilken metod du i så fall tänker använda för att lösa dem.)

- (3 p) b) En person på badhus-firman kan lite Matlab och föreslår en snabbare lösning. I stället för att sommarjobbaren ska spendera tid på att jobba med problemet så kan man anropa en spline-funktion i Matlab med enbart de fyra givna punkterna och få ut koefficienterna för en bana direkt. Denna Matlab-funktion anpassar kubiska splines till samtliga datapunkter.

Kommer den så erhållna rutschkane-banan att uppfylla alla önskemålen? Om inte, vilka villkor uppfylls och vilka uppfylls inte? Kan det bli något problem med spline-rutschbanan?

Lösningsförslag:

- a) Mellan $x = 0$ och $x = 0.3$ vill vi använda linjär interpolation, och detsamma gäller mellan $x = 2.1$ och $x = 2.4$. Dessa polynom kan vi bestämma direkt, till exempel med hjälp av Lagrangeinterpolation, som

$$p_1(x) = 2 \cdot \frac{(x - 0.3)}{(0 - 0.3)} + 1.99 \cdot \frac{(x - 0)}{(0.3 - 0)} = \frac{2(0.3 - x) + 1.99x}{0.3} = \frac{60 - x}{30},$$

för $0 \leq x \leq 0.3$, och

$$\begin{aligned} p_3(x) &= 0.52 \cdot \frac{(x - 2.4)}{(2.1 - 2.4)} + 0.5 \cdot \frac{x - 2.1}{(2.4 - 2.1)} = \frac{0.52(2.4 - x) + 0.5(x - 2.1)}{0.3} \\ &= \frac{19.8 - 2x}{30}, \end{aligned}$$

för $2.1 \leq x \leq 2.4$.

På intervallet $0.3 \leq x \leq 2.1$ måste polynomet $p_2(x)$ mjukt ansluta till de övriga polynomen vid ändpunkterna. Det betyder att polynomet måste uppfylla villkoren

$$p_2(0.3) = p_1(0.3) = 1.99, \quad p_2(2.1) = p_3(2.1) = 0.52$$

och dessutom

$$p_2'(0.3) = p_1'(0.3), \quad p_2'(2.1) = p_3'(2.1).$$

Eftersom det är fyra villkor måste p_2 vara minst av grad 3. Vi ansätter därmed $p_2(x) = ax^3 + bx^2 + cx + d$. Derivatans blir $p_2'(x) = 3ax^2 + 2bx + c$, medan de linjära polynomens derivator blir $p_1' = -1/30$ och $p_3' = -2/30$.

De fyra ekvationer som p_2 ska uppfylla blir:

$$\begin{cases} a \cdot 0.3^3 + b \cdot 0.3^2 + c \cdot 0.3 + d = 1.99, \\ a \cdot 2.1^3 + b \cdot 2.1^2 + c \cdot 2.1 + d = 0.52, \\ 3a \cdot 0.3^2 + 2b \cdot 0.3 + c = -1/30, \\ 3a \cdot 2.1^2 + 2b \cdot 2.1 + c = -2/30. \end{cases}$$

Systemet kan lösas med backslash i Matlab.

Rutchkanans bana ges då av

$$p(x) = \begin{cases} p_1(x), & 0 \leq x \leq 0.3, \\ p_2(x), & 0.3 \leq x \leq 2.1, \\ p_3(x), & 2.1 \leq x \leq 2.4. \end{cases}$$

- b) Om kubiska splines anpassas till samtliga datapunkter kommer kurvan att gå genom de givna datapunkterna, men inget av delsegmenten kommer (i allmänhet) bli helt rakt. Banan uppfyller alltså inte önskemålen om att banan ska vara rak mellan $x = 0$ och $x = 0.3$ samt mellan $x = 2.1$ och $x = 2.4$.

Om banan bildar en grop vid slutet kan det till exempel samlas vatten där, vilket man kanske inte vill. Det kan också vara svårt att kliva på banan om den bildar en kulle i början.

(Som en bonus visas båda banorna tillsammans med datapunkterna i figuren på nästa sida.)

