



KTH Engineering Sciences

Tentamen 2020-08-14 Del 2

SF1547 – Numeriska metoder

Max antal poäng på denna del är 50. Rättas endast om del 1 är godkänd. Betygsgränser: 10p D, 20p C, 30p B, 40p A. Svar skall motiveras och uträkningar redovisas. Korrekt svar utan motivering eller med felaktig motivering medför poängavdrag.

Inga hjälpmedel är tillåtna (ej heller miniräknare).

Varje uppgift på Del 2 ska skrivas på separata papper. (Deluppgifter kan skrivas på samma papper.)

Skriv uppgiftsnummer, sidnummer, namn och personnummer på varje papper.

P1. Heuns metod är en andra ordningens numerisk metod för att lösa differentialekvationer på formen $y' = f(y)$. Den lyder

$$y_{n+1} = y_n + h \frac{f(y_n) + f(y_n + hf(y_n))}{2}.$$

(a) Skriv ett MATLAB-program som löser följande ordinära differentialekvation (10 p)

$$\frac{d^2y}{dt^2} - \left(\frac{dy}{dt}\right)^2 + \alpha e^y = 0, \quad y(0) = 0, \quad \frac{dy}{dt}(0) = 0,$$

med Heuns metod från $t = 0$ till $t = 10$. Använd $\alpha = 0.75$ och valfri steglängd mindre än $1/2$. Programmet ska skriva ut $y(10)$.

Lösning: Vi skriver först om ODEn som ett första ordningens system. Med $u = y$, $v = y'$ och $\mathbf{u} = (u, v)$ får vi

$$\mathbf{u}' = \begin{pmatrix} u \\ v \end{pmatrix}' = \begin{pmatrix} v \\ v^2 - \alpha e^u \end{pmatrix} =: \mathbf{f}(\mathbf{u}), \quad \mathbf{u}(0) = \begin{pmatrix} u(0) \\ v(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Vi kan nu tidsstega lösningen med Heuns metod given i uppgiften. En Matlab-lösning kan se ut som:

```
n = 100; T = 10; h = T/n; t=0:h:T; al=0.75;
f = @(u) [u(2); u(2)^2-al*exp(u(1))];
u = [0;0];
U = u;
for tt = t(1:end-1)
    u = u + h*(f(u)+f(u+h*f(u)))/2;
    U = [U u]; % Sparar lösningen i U-vektorn för senare uppgifter
end
u(1)
```

Kommentar: När f är dyr att evaluera är det bättre att spara det uträknade värdet $f(\mathbf{u}_n)$ för att minska beräkningskostnaden. Raden $\mathbf{u} = \mathbf{u} + \mathbf{h}*(f(\mathbf{u})+f(\mathbf{u}+\mathbf{h}*f(\mathbf{u}))) / 2$; byts då mot

```

    us = f(u);
    u = u + h*(us+f(u+h*us))/2;
end

```

- (b) Skriv ett MATLAB-program som bestämmer parametern α i differentialekvationen så att (10 p)

$$\int_0^{10} y(t)^2 + y'(t)^2 dt = 50.$$

Värdet på detta α ligger nära 0.6. För full poäng ska programmet vara effektivt och integralen ska approximeras med noggrannhetsordning två. Du får återanvända delar av ditt program i deluppgift (a), men inte använda inbyggda funktioner för integration. Man ska kunna välja parametrar i programmet så att svaret får godtyckligt litet fel.

Lösning: Vi formulerar först om programmet i deluppgift (a) som en funktion av α som ska vara noll vid det korrekta värdet. Vi lägger också till rader som beräknar integralen med trapetsregeln:

```

function r = G(a1)
n = 100; T = 10; h = T/n; t=0:h:T;
f = @(u) [u(2); u(2)^2-a1*exp(u(1))];
u = [0;0];
U = u;
for tt = t(1:end-1)
    u = u + h*(f(u)+f(u+h*f(u)))/2;
    U = [U u];
end
y = U(1,:); yp = U(2,:);
I = h*(sum(y.^2+yp.^2)-(y(1)^2+yp(1)^2+y(end)^2+yp(end)^2)/2); % Integralen
r = I-50; % Ska vara noll när vi hittat rätt alfa
end

```

Vi använder sedan sekantmetoden för att hitta rätt α , med startgissningar nära 0.6:

```

tol = 1e-10;
a1 = 0.5; a2 = 0.75;
g1 = G(a1); g2 = G(a2);
while abs(a1-a2)>tol
    an = a2 - g2*(a2-a1)/(g2-g1);
    a1 = a2; g1 = g2;
    a2 = an; g2 = G(a2);
end
an

```

Genom att välja `tol` mindre och `n` större i koden kan vi få felet så litet vi vill.

- (c) Heuns metod har noggrannhetsordningen två. Genom Richardson-extrapolation kan man få fram en högre ordnings approximation av $y(10)$. Förklara i viss detalj hur man kan göra det med hjälp av MATLAB-programmet i deluppgift (a). (5 p)

Lösning: Vi skriver först programmet i deluppgift (a) som en funktion av antal tidssteg n :

```

function y = y10(n)
T = 10; h = T/n; t=0:h:T; al=0.75;
f = @(u) [u(2); u(2)^2-al*exp(u(1))];
u = [0;0];
for tt = t(1:end-1)
    u = u + h*(f(u)+f(u+h*f(u)))/2;
end
y = u(1);
end

```

I Richardsonextrapolation använder man lösningen beräknad med h och $h/2$, kallade u_h och $u_{h/2}$, för att bestämma ett värde med högre noggrannhetsordning enligt:

$$u_{\text{Rich}} = u_{h/2} - \frac{u_h - u_{h/2}}{2^p - 1},$$

där p är metodens noggrannhetsordning. I vårt fall är $p = 2$ eftersom Heun är av ordning två. Genom att anropa programmet två gånger med n och $2n$ får vi därför fram den bättre approximationen med:

```

n=100;
y1 = y10(n);
y2 = y10(2*n);
yrich = y2 - (y1-y2)/3

```

Detta blir en tredje ordningens approximation.

P2. Vi vill lösa följande differentialekvation

$$u''(x) + (1 - x^2)u(x) = f(x), \quad 0 < x < 2,$$

och

$$u(0) = 1, \quad u(2) = 2.$$

- (a) Formulera en andra ordningens finit differensmetod för detta randvärdesproblem. (10 p)
Låt $\mathbf{u} = (u_1, \dots, u_n)^T$ 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} . (Ekvationssystemet behöver inte lösas.)

Lösning: I varje inre punkt, $j = 1, \dots, n$, approximerar vi andraderivatan i ekvationen med andra ordningens centraldifferenskvoter,

$$u_{xx}(x_j) \approx \frac{u_{j-1} - 2u_j + u_{j+1}}{h^2}.$$

Insatt i ekvationen ger detta

$$\frac{u_{j-1} - 2u_j + u_{j+1}}{h^2} + (1 - x_j^2)u_j = f(x_j), \quad j = 1, \dots, n.$$

Multiplitera med h^2 och samla ihop termerna

$$u_{j-1} + (h^2(1 - x_j^2) - 2)u_j - u_{j+1} = h^2 f(x_j), \quad j = 1, \dots, n. \quad (1)$$

Utnyttja randvillkoren för att eliminera u_0 och u_{n+1} :

$$u_0 = 1, \quad u_{n+1} = 2.$$

Detta ger för $j = 1$,

$$(h^2(1 - x_1^2) - 2)u_1 - u_2 = h^2 f(x_1) - 1 \quad (2)$$

och för $j = n$,

$$u_{n-1} + (h^2(1 - x_n^2) - 2)u_n = h^2 f(x_n) - 2. \quad (3)$$

Tillsammans ger (1,2,3) det linjära ekvationssystemet $A\mathbf{u} = \mathbf{b}$ med

$$A = \begin{pmatrix} h^2(1 - x_1^2) - 2 & 1 & & & & & \\ 1 & h^2(1 - x_2^2) - 2 & 1 & & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & h^2(1 - x_{n-1}^2) - 2 & & \\ & & & & 1 & & h^2(1 - x_n^2) - 2 \end{pmatrix},$$

och högerledet

$$\mathbf{b} = \begin{pmatrix} h^2 f(x_1) - 1 \\ h^2 f(x_2) \\ \vdots \\ h^2 f(x_{n-1}) \\ h^2 f(x_n) - 2 \end{pmatrix}.$$

- (b) Metoden i deluppgift (a) implementeras i MATLAB. (Du behöver inte göra det själv!) Till ditt förfogande får du MATLAB-funktionen $\mathbf{u} = \text{RVP}(\mathbf{n}, \mathbf{f})$ som returnerar Lösningsvektorn \mathbf{u} givet \mathbf{n} (antal obekanta) och vektorn $\mathbf{f} = (f(x_1), \dots, f(x_n))^T$ (högerledsfunktionen evaluerad i gridpunkterna). (10 p)

Antag att f är det femtegradspolynom som interpolerar de uppmätta värdena

x	0	0.4	0.8	1.2	1.6	2
$f(x)$	0.1	1.3	1.4	1.0	0.3	-0.4

Varje mätvärde har en absolut felgräns på 0.1. Detta ger en osäkerhet i lösningen u . Skriv ett MATLAB-program som uppskattar felgränsen för värdet $u(1)$. Programmet får använda funktionen $\mathbf{u} = \text{RVP}(\mathbf{n}, \mathbf{f})$ men inte inbyggda funktioner för interpolation. Antalet obekanta n är valfritt, men måste vara betydligt större än 4.

För full poäng krävs ett effektivt program som anropar $\mathbf{u} = \text{RVP}(\mathbf{n}, \mathbf{f})$ så få gånger som möjligt.

Lösning: Programmet måste först kunna interpolera sex punkter med ett femtegradspolynom. Vi kallar interpolationspunkterna i tabellen (X_j, Y_j) . Interpolationen gör man tex genom att ansätta

$$f(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4 + c_5 x^5,$$

och lösa Vandermondesystemet

$$\begin{bmatrix} 1 & X_0 & \cdots & X_0^5 \\ 1 & X_1 & \cdots & X_1^5 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & X_5 & \cdots & X_5^5 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_5 \end{bmatrix} = \begin{bmatrix} Y_0 \\ Y_1 \\ \vdots \\ Y_5 \end{bmatrix}.$$

Från detta kan vi beräkna f i de punkter $\{x_j\}$ där lösningen till randvärdesproblemet beräknas, och speciellt få fram $u(1)$.

Felgränsen kan nu bestämmas med experimentell störningsräkning: Genomför lösningen med $\{Y_j\}$ som ovan. Det ger $U = u(1)$. Stör sedan ett Y_j i sänder med 0.1 och beräkna motsvarande lösning U_j . Det ger slutligen felgränsen

$$E_U \approx |U - U_0| + |U - U_1| + \dots + |U - U_5|.$$

En Matlab-implementation kan se ut som följer:

```
n = 81; % Välj n udda så att x=1 finns med bland gridpunkterna

X = [0 0.4 0.8 1.2 1.6 2]';
V = [ones(size(X)) X X.^2 X.^3 X.^4 X.^5]; % Vandermonde-matrisen
Y = [0.1 1.3 1.4 1 0.3 -0.4]'; % Mätvärden

h = 2/(n+1);
x = h:h:2-h; % Inre punkter där lösningen beräknas

c = V\Y;
f = c(1) + c(2)*x + c(3)*x.^2+ c(4)*x.^3+ c(5)*x.^4+ c(6)*x.^5;
u = RVP(n,f');
U = u((n+1)/2); % Värdet i x=1

E=0; % Nollställ felgränsen
for k=1:length(Y)
    YY = Y; YY(k)=YY(k)+0.1; % Stör mätvärde k
    c = V\YY;
    f = c(1) + c(2)*x + c(3)*x.^2+ c(4)*x.^3+ c(5)*x.^4+ c(6)*x.^5;
    u = RVP(n,f');
    Us = u((n+1)/2); % Störda värdet i x=1
    E = E + abs(Us-U); % Addera bidraget till felgränsen
end
U,E
```

P3. Betrakta differensformeln

(5 p)

$$\frac{\alpha y(x - 2h) - 4y(x - h) + \beta y(x + h) + y(x + 2h)}{6h^2},$$

där $h \ll 1$ och y kan deriveras oändligt många gånger. För ett val av α och β approximerar formeln andraderivatan $y''(x)$ med noggrannhetsordning minst ett. Bestäm dessa α och β .

Lösning: Låt

$$T(h) = \frac{\alpha y(x - 2h) - 4y(x - h) + \beta y(x + h) + y(x + 2h)}{6h^2}.$$

Taylorutveckling ger

$$\begin{aligned} T(h) &= \alpha \frac{y(x) - 2hy'(x) + \frac{4h^2}{2}y''(x) + O(h^3)}{6h^2} \\ &\quad - 4 \frac{y(x) - hy'(x) + \frac{h^2}{2}y''(x) + O(h^3)}{6h^2} \\ &\quad + \beta \frac{y(x) + hy'(x) + \frac{h^2}{2}y''(x) + O(h^3)}{6h^2} \\ &\quad + \frac{y(x) + 2hy'(x) + \frac{4h^2}{2}y''(x) + O(h^3)}{6h^2} \\ &= y(x) \frac{\alpha - 4 + \beta + 1}{6h^2} + y'(x) \frac{-2\alpha + 4 + \beta + 2}{6h} + y''(x) \frac{4\alpha - 4 + \beta + 4}{12} + O(h) \\ &= y(x) \frac{\alpha + \beta - 3}{6h^2} + y'(x) \frac{-2\alpha + \beta + 6}{6h} + y''(x) \frac{4\alpha + \beta}{12} + O(h). \end{aligned}$$

För att detta ska approximera $y''(x)$ med första ordningens noggrannhet måste vi därför ha

$$\begin{aligned} \alpha + \beta - 3 &= 0, \\ -2\alpha + \beta + 6 &= 0, \\ 4\alpha + \beta &= 12. \end{aligned}$$

Subtraktion av de två första ekvationerna ger

$$3\alpha - 9 = 0,$$

från vilket följer att $\alpha = 3$. Detta insatt i första ekvationen ger $\beta = 0$. Dessa val uppfyller även den tredje ekvationen.

Alternativ lösning: Ett nödvändigt villkor för att formeln ska ha noggrannhetsordning ett är att den har noggrannhetsordning ett för de specifika valen $y(x) = 1$ och $y(x) = x$. För dessa får vi villkoren

$$\frac{\alpha - 4 + \beta + 1}{6h^2} = 0 + O(h), \quad \frac{-2\alpha + 4 + \beta + 2}{6h} = 0 + O(h)$$

Genom att låta $h \rightarrow 0$ ser vi att det måste gälla att

$$\alpha - 4 + \beta + 1 = 0, \quad -2\alpha + 4 + \beta + 2 = 0,$$

vilket ger samma α och β som ovan. I allmänhet är detta inte ett tillräckligt villkor, men eftersom det står givet i uppgiften att det existerar en kombination av α, β som gör att formeln approximerar $y''(x)$ med noggrannhetsordning ett, måste det vara denna.