KTH Matematik

Tentamen del 2
Numeriska beräkningar SF1522
2020-01-10, 08.00-11.00.

Inga hjälpmedel är tillåtna (ej heller miniräknare).

Skriv lösningar med fullständiga meningar och utförliga motiveringar för att undvika poängavdrag.
Del 2 rättas om del 1 är godkänd. Max antal poäng är 40.
Betygsgränser: D 8, C 16, B 24 och A 32 poäng.

1. (12p) Om man har tre punkter i (x, y)-planet som inte ligger på en rät linje så kan man alltid anpassa en cirkel som går exakt igenom punkterna genom att lösa ekvationssystemet

$$(x_i - X)^2 + (y_i - Y)^2 = R^2, \qquad i = 1, 2, 3,$$

där X är x-koordinaten för cirkelns mittpunkt, Y är y-koordinaten för cirkelns mittpunkt och R är cirkelns radie.

a) (4p) Skriv om ekvationssystemet på formen

$$c_1 + c_2 \cdot x_i + c_3 \cdot y_i = x_i^2 + y_i^2, \qquad i = 1, 2, 3,$$

så att systemet kan lösas som ett linjärt ekvationssystem. Vad blir uttrycken för X, Y och R i de nya parametrarna c_1, c_2 och c_3?

(Ledning: $c_2 = 2 \cdot X$)

b) (4p) Skriv ett Matlab-program som beräknar cirkeln som går genom de tre punkterna $(x, y) = (3, 5)$, $(x, y) = (7, 8)$ och $(x, y) = (4, 9)$. Värdena för koordinaterna för mittpunkten och radien skall skrivas ut.

c) (4p) Om man har fler punkter än tre är det inte troligt att man kan hitta en cirkel som går perfekt igenom samtliga punkter. Skriv ett Matlab-program som med minstakvadratmetoden anpassar bästa möjliga cirkel till de fem punkterna i tabellen nedan.

x	3	7	4	6	7
y	5	8	9	5	5

Värdena för koordinaterna för mittpunkten och radien skall skrivas ut.

Lösning

a) Omskrivning av cirkelns ekvation:

$$\begin{aligned} &(x_i - X)^2 + (y_i - Y)^2 = R^2 \\ \Rightarrow &x_i^2 - 2x_iX + X^2 + y_i^2 - 2y_iY + Y^2 = R^2 \\ \Rightarrow &\underbrace{R^2 - X^2 - Y^2}_{c_1} + \underbrace{2X}_{c_2} \cdot x_i + \underbrace{2Y}_{c_3} \cdot y_i = x_i^2 + y_i^2 \end{aligned}$$

Alltså får vi

$$X = \frac{c_2}{2}$$
$$Y = \frac{c_3}{2}$$
$$R = \sqrt{c_1 + X^2 + Y^2} = \sqrt{c_1 + \frac{c_2^2 + c_3^2}{4}}$$

b) Lösning av det linjära systemet

$$c_1 + c_2 \cdot x_i + c_3 \cdot y_i = x_i^2 + y_i^2, \qquad i = 1, 2, 3,$$

bestämmer c_1, c_2, c_3, vilket ger oss X, Y, R. Matlab-kod:

```
x = [3; 7; 4];
y = [5; 8; 9];
A = [ones(3,1) x y];
b = x.^2 + y.^2;
c = A\b;
disp('Koordinater mittpunkt:')
X = c(2)/2
Y = c(3)/2
disp('Radie:')
R = sqrt(c(1) + (c(2)^2+c(3)^2)/4)
```

c) Lösningen till detta problem blir samma som i (b)-uppgiften, men skillnaden att matrisen A är överbestämd, och måste lösas med minstakvadratmetoden:

```
x = [3; 7; 4; 6; 7];
y = [5; 8; 9; 5; 5];
A = [ones(5,1) x y];
b = x.^2 + y.^2;
c = A\b; % MKM
disp('Koordinater mittpunkt:')
X = c(2)/2
Y = c(3)/2
disp('Radie:')
R = sqrt(c(1) + (c(2)^2+c(3)^2)/4)
```

2. (14p) En funktion $f(x)$ är känd i följande punkter:

x	-1	1	3	5
$f(x)$	10	14	18	-2

a) (1p) Beräkna en approximation till $f'(2)$ med hjälp av uttrycket som är givet i Uppgift 1 i Del 1 av denna tentamen, med steglängder $h = 1$ och $h = 3$.

b) (6p) Beräkna en förbättrad approximation till $f'(2)$ med hjälp av Richardsonextrapolation och dina två approximationer från föregående deluppgift.
Obs: Tänk på förhållandet mellan steglängderna!

c) (4p) Hitta det kubiska polynom $p(x)$ som passerar genom datapunkterna ovan, t.ex. genom att använda Newtons ansats. Beräkna $p(2)$.

d) (3p) Beräkna $p'(2)$. Detta värde är en approximation till $f'(2)$. Är det en bättre eller sämre approximation än den som du beräknade i (b)-uppgiften?

Lösning

a) Approximationen är $f'(x) \approx (f(x+h) - f(x-h))/2h$. Vi skriver

$$f'(2) \approx D(h) = \frac{f(2+h) - f(2-h)}{2h}.$$

För $h = 1$ och $h = 3$ fås då

$$\begin{aligned} D(3) &= (f(5) - f(-1))/6 = (-2 - 10)/6 = -2, \\ D(1) &= (f(3) - f(1))/2 = (18 - 14)/2 = 2 \end{aligned}$$

b) För approximationen gäller att (vilket kan ses i den givna formeln) det finns en konstant c så att

$$f'(2) = D(h) + ch^2 + \mathcal{O}(h^4).$$

Med $h = 1$ och $h = 3$ kan vi då ställa upp systemet

$$\begin{aligned} f'(2) &= D(1) + c \cdot 1^2 + \mathcal{O}(h^4), \\ f'(2) &= D(3) + c \cdot 3^2 + \mathcal{O}(h^4). \end{aligned}$$

Eliminering av första feltermen ger

$$\begin{aligned} 9f'(2) - f'(2) &= 9D(1) - D(3) + \mathcal{O}(h^4), \\ \Rightarrow f(2) &= \frac{9D(1) - D(3)}{9-1} + \mathcal{O}(h^4) \approx \frac{9 \cdot 2 - (-2)}{8} = \frac{20}{8} = \frac{5}{2} = 2.5 \end{aligned}$$

Generell formel för Richardsonextrapolation som kan användas är

$$R(h) = \frac{a^p D(h) - D(ah)}{a^p - 1}.$$

c) Newtons ansats kan se olika ut beroende på hur punkterna ordnas. En variant är:

$$p(x) = c_1 + c_2(x+1) + c_3(x+1)(x-1) + c_4(x+1)(x-1)(x-3)$$

Insättning av punktvis data:

$$\begin{aligned} & p(-1) = c_1 = 10 \\ \Rightarrow & \mathbf{c_1 = 10} \end{aligned}$$

$$\begin{aligned} & p(1) = c_1 + c_2 \cdot 2 = 14 \\ \Rightarrow & 10 + c_2 \cdot 2 = 14 \\ \Rightarrow & \mathbf{c_2 = 2} \end{aligned}$$

$$\begin{aligned} & p(3) = c_1 + c_2 \cdot 4 + c_3 \cdot 4 \cdot 2 = 18 \\ \Rightarrow & 0 + 8 + c_3 \cdot 8 = 18 \\ \Rightarrow & \mathbf{c_3 = 0} \end{aligned}$$

$$\begin{aligned} & p(5) = c_1 + c_2 \cdot 6 + c_4 \cdot 6 \cdot 4 \cdot 2 = -2 \Rightarrow \qquad 10 + 12 + c_4 \cdot 48 = -2 \\ \Rightarrow & \mathbf{c_4 = -\frac{1}{2}} \end{aligned}$$

Så polynomet genom punkterna är

$$p(x) = 10 + 2(x+1) - \frac{1}{2}(x+1)(x-1)(x-3).$$

och

$$p(2) = 10 + 2 \cdot 3 - \frac{1}{2} \cdot 3 \cdot 1 \cdot (-1) = \frac{20 + 12 + 3}{2} = \frac{35}{2}$$

d) För att ta fram $p'(x)$ kan man antingen derivera $p(x)$ direkt, eller först skriva om på naiv form och sen differentiera. Vi gör det senare:

$$\begin{aligned} p(x) &= \frac{1}{2}\left(-x^3 + 3x^2 + 5x + 21\right) \\ p'(x) &= \frac{1}{2}\left(-3x^2 + 6x + 5\right) \end{aligned}$$

vilket ger

$$p'(2) = \frac{1}{2}(-3 \cdot 4 + 6 \cdot 2 + 5) = \frac{5}{2} = 2.5$$

Detta är samma värde som vi fick i b-uppgiften, alltså är det varken en bättre eller sämre approximation, utan de två approximationerna är ekvivalenta.

3. (14p) Givet integralen

$$I(k) = \int_{-1}^{1} \frac{\cos \pi x}{x^2 + k^2} dx, \quad k > 0,$$

a) (3p) Beräkna en approximation till $I(0.5)$ med hjälp av trapetsregeln och 4 lika stora delintervall.

b) (4p) Formulera en algoritm, gärna i form av en Matlab-funktion, som givet värden k och N beräknar en approximation till $I(k)$ med hjälp av trapetsregeln och N lika stora delintervall. Du kan anta att $k > 0$ och att N är ett heltal.

c) (7p) Vi vill nu bestämma k så att

$$I(k) = k. \tag{1}$$

Formulera en algoritm, gärna i form av ett Matlab-program, som givet en startgissning för k löser ekvationen (1). Din lösning ska använda sig av algoritmen som du formulerade i (b)-uppgiften, för ett godtyckligt heltalsvärde på N.

För full poäng på uppgift 3 får du inte använda inbyggda Matlab-funktioner för ekvationslösning eller integration.

Lösning

a) Trapetsregelen med $h = 0.5$ ger $x = (-1, -1/2, 0, 1/2, 1)$. Utvärdera integranden $f(x) = \cos(\pi x)/(x^2 + k^2)$ i dessa punkter med $k = 1/2$,

x	-1	-1/2	0	1/2	1
$f(x)$	$-4/5$	0	4	0	$-4/5$

Trapetsregeln ger

$$\begin{aligned} I(0.5) &\approx h(f(-1)/2 + f(0.5) + f(0) + f(1/2) + f(1)/2) \\ &= \frac{1}{2}(-4/10 + 4 - 4/10) = \frac{-4 + 40 - 4}{20} \\ &= \frac{16}{10} = 1.6 \end{aligned}$$

b) Förslag på Matlab-funktion som löser uppgiften

```
function I = trapets(k, N)
    f = @(x) cos(pi*x)/(x^2+k^2);
    h = 2/N;
    x = -1:h:1;
    I = h/2*( f(x(1)) + f(x(N+1)) );
    for i=2:N
        I = I + h*f(x(i));
    end
end
```

c) Vi söker en lösning till problemet

$$F(k) = -k + I(k) = -k + \int_{-1}^{1} \frac{\cos \pi x}{x^2 + k^2} dx = 0.$$

Detta kan lösas med exempelvis Newtons metod: Givet startgissning k_0,

$$k_{n+1} = k_n - \frac{F(k)}{F'(k)}, \quad n = 1, 2, 3, \ldots.$$

Derivatan $F'(k)$ ges av

$$F'(k) = -1 + I'(k) = -1 - \int_{-1}^{1} \frac{2k \cos \pi x}{(x^2 + k^2)^2} dx.$$

Vi kan beräkna $I'(k)$ med trapetsregeln, t.ex. på samma sätt som ovan,

```
function Ip = trapets_prim(k, N)
    fp = @(x) -2*k*cos(pi*x)/(x^2+k^2)^2;
    h = 2/N;
    x = -1:h:1;
    Ip = h/2*( fp(x(1)) + fp(x(N+1)) );
    for i=2:N
        Ip = I + h*fp(x(i));
    end
end
```

Givet en startgissning `k0`, ett antal intervall `N` och en tolerans `tol` kan en Matlabkod för att lösa detta problem se ut som följande:

```
k = k0;
dk = 1;
while abs(dk)>tol
    F = -k + trapets(k, N);
    Fp = -1 + trapets_prim(k, N);
    dk = -F/Fp;
    k = k + dk;
end
```