Beatrice Frock, NADA

2D1212 NumProg, Kompletterande material

Kap.2. Lösning av ett system av icke-linjära ekvationer. Newtons metod.

Newtons (Newton-Raphsons) metod för lösning av en skalär icke-linjär ekvation $f(x) = 0$ är ett specialfall av flerdimensionell Newton för system av icke-linjära ekvationer. Vi betraktar ett $n \times n$ icke-linjärt system $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, eller

$$\begin{aligned} f_1(x_1, \ldots, x_n) &= 0 \\ f_2(x_1, \ldots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, \ldots, x_n) &= 0 \end{aligned}$$

Flerdimensionell Newton

$$\mathbf{x_{n+1}} = \mathbf{x_n} - \mathbf{J}(\mathbf{x_n})^{-\mathbf{1}} \cdot \mathbf{f}(\mathbf{x_n})$$

med en lämplig startvektor $\mathbf{x_0}$ för iterationerna.
Denna iterationsformel kan lätt härledas m.h.a. Taylor utveckling av en vektorvärd funktion,

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x_0}) + \mathbf{J}(\mathbf{x_0})(\mathbf{x} - \mathbf{x_0}) + \ldots$$

Metoden är normalt kvadratiskt konvergent i en omgivning av roten, under förutsättningen att Jacobianmatrisen är icke-singulär.
Jacobianen

$$J(\mathbf{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

Explicit invertering av Jacobianmatrisen bör undvikas, eftersom operationen matrisinvertering är kostsam och ofta känsligt för störningar, såsom avrundningsfel.
Korrektionen $\mathbf{h}(\mathbf{x_n}) = -\mathbf{J}(\mathbf{x_n})^{-\mathbf{1}} \cdot \mathbf{f}(\mathbf{x_n})$ beräknas i stället som lösningen till det linjära ekvationssystemet

$$J(\mathbf{x_n}) \cdot \mathbf{h}(\mathbf{x_n}) = -\mathbf{f}(\mathbf{x_n})$$

i varje iterationssteg.
Därför övergår formeln i **Matlab** till följande:

$$\mathbf{x_{n+1}} = \mathbf{x_n} - \mathbf{J}(\mathbf{x_n}) \backslash \mathbf{f}(\mathbf{x_n}),$$

med startapproximationen $\mathbf{x_0}$.
De partiella derivatorna som ingår i Jacobianen kan approximeras med en lämplig differenskvot vid behov.
Det är inte alls trivialt att bestämma en bra startvektor för iterationerna. Ibland

erhålls dessutom konvergens till en annan rot än den sökta. Man kan försöka försumma de icke-linjära termerna för att bestämma ett startvärde, eller prova med t.ex. $(0,\ldots,0)^T$ eller $(1,\ldots,1)^T$, om man inte kan utnyttja den fysikaliska bakgrunden till problemet.
Det finns flera andra metoder för lösning av ett icke-linjärt system (t.ex. fixpunktsmetoden), men dessa ingår inte i denna kurs.

Exempel. Handräkning ett steg, samt Matlab-program.

Formulera en algoritm (Matlab-program) för lösning av ekvationssystemet

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} 15x_1 + x_2 - x_3^2 - 30 \\ -x_1 + 30x_2 - x_3 - 30 \\ -x_1^2 + x_2 + 100x_3 - 20 \end{pmatrix} = \mathbf{0}$$

med Newtons metod.
Bestäm en grov startapproximation, och ange explicit det linjära ekvationssystem som skall lösas i den första iterationen. Beräkna även första iteratet.

En startapproximation kan för detta exempel erhållas genom att termer med små koefficienter försummas:

$$15x_1 = 30, \qquad 30x_2 = 30, \qquad 100x_3 = 20 \qquad \Rightarrow x_1 = 2,\ x_2 = 1,\ x_3 = 0.2$$

Vi får:

$$\mathbf{x_0} = \begin{pmatrix} 2 \\ 1 \\ 0.2 \end{pmatrix} \Rightarrow \mathbf{f_0} = \begin{pmatrix} 15\cdot 2 + 1 - 0.2^2 - 30 \\ -2 + 30 - 0.2 - 30 \\ -2^2 + 1 + 100\cdot 0.2 - 20 \end{pmatrix} = \begin{pmatrix} 0.96 \\ -2.2 \\ -3 \end{pmatrix}$$

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{pmatrix} = \begin{pmatrix} 15 & 1 & -2x_3 \\ -1 & 30 & -1 \\ -2x_1 & 1 & 100 \end{pmatrix} \Rightarrow \mathbf{J_0} = \begin{pmatrix} 15 & 1 & -0.4 \\ -1 & 30 & -1 \\ -4 & 1 & 100 \end{pmatrix}$$

Det linjära ekvationssystemet som ska lösas i första iterationen blir

$$\begin{pmatrix} 15 & 1 & -0.4 \\ -1 & 30 & -1 \\ -4 & 1 & 100 \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \end{pmatrix} = -\begin{pmatrix} 0.96 \\ -2.2 \\ -3 \end{pmatrix} \Rightarrow \mathbf{h_0} = \begin{pmatrix} -0.0681 \\ 0.0719 \\ 0.0266 \end{pmatrix}$$

och $\mathbf{x_1} = \mathbf{x_0} + \mathbf{h_0} = \begin{pmatrix} 1.9319 \\ 1.0719 \\ 0.2266 \end{pmatrix}$

Matlab-kod:

```
clear
disp('mata in startvektorns 3 komponenter  ')
x(1)=input('x(1) '), x(2)=input('x(2) '), x(3)=input('x(3) ')
x=x'; % transponera till kolumnvektor
iter=0;
hnorm=1; %initiering
while hnorm>0.5e-6 & iter<8,
   f=[15*x(1)+x(2)-x(3)^2-30
      -x(1)+30*x(2)-x(3)-30
      -x(1)^2+x(2)+100*x(3)-20];
   J=[15  1  -2*x(3)
      -1  30  -1
      -2*x(1)  1  100];
   h=-J\f;
   x=x+h;
   hnorm=norm(h,inf);
   iter=iter+1;
   disp([x'])
   % disp([iter, hnorm])
end, %while
x, iter
```

Efter tre iterationer erhålls resultatet $\mathbf{x} = \begin{pmatrix} 1.9251 \\ 1.0717 \\ 0.2263 \end{pmatrix}$