Newtons metode for system

1 Newton-Raphsons metode for et system av ikke-lineære ligninger

Newton-Raphsons metode er en numerisk metode for å finne løsninger av et system av ligninger

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

For enkelthets skyld ser vi her bare på et system av to ligninger, dvs. $n = 2$.

Anta iterat $k-1$ (x_1^{k-1}, x_2^{k-1}) er gitt, og vi ønsker å bestemme $(\Delta x_1, \Delta x_2)$ slik at

$$f_i(x_1^{k-1} + \Delta x_1, x_2^{k-1} + \Delta x_2) = 0, \qquad i = 1, 2. \tag{1}$$

Ved bruk av Taylor utvikling (avsnitt 2) får vi

$$\begin{bmatrix} f_1\left(x_1^{k-1} + \Delta x_1, x_2^{k-1} + \Delta x_2\right) \\ f_2\left(x_1^{k-1} + \Delta x_1, x_2^{k-1} + \Delta x_2\right) \end{bmatrix} = \begin{bmatrix} f_1\left(x_1^{k-1}, x_2^{k-1}\right) \\ f_2\left(x_1^{k-1}, x_2^{k-1}\right) \end{bmatrix} + J^{k-1} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \vec{R}, \tag{2}$$

hvor matrisa J^{k-1} er Jacobimatrisa beregnet i punktet $\vec{x}^{k-1} = \left[x_1^{k-1}, x_2^{k-1}\right]^T$ det vil si

$$J^{k-1} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}\left(x_1^{k-1}, x_2^{k-1}\right) & \frac{\partial f_1}{\partial x_2}\left(x_1^{k-1}, x_2^{k-1}\right) \\ \frac{\partial f_2}{\partial x_1}\left(x_1^{k-1}, x_2^{k-1}\right) & \frac{\partial f_2}{\partial x_2}\left(x_1^{k-1}, x_2^{k-1}\right) \end{bmatrix},$$

og residualet $\vec{R}$ er av orden 2 eller høyere i $\max\{|\Delta x_1|, |\Delta x_2|\}$. Ved å sette venstre side lik $\vec{0}$ i (2), droppe residualet $\vec{R}$, og erstatte Δx_1 og Δx_2 med approksimasjonene Δx_1^{k-1} og Δx_2^{k-1}, får vi

$$\begin{bmatrix} f_1\left(x_1^{k-1}, x_2^{k-1}\right) \\ f_2\left(x_1^{k-1}, x_2^{k-1}\right) \end{bmatrix} + J^{k-1} \begin{bmatrix} \Delta x_1^{k-1} \\ \Delta x_2^{k-1} \end{bmatrix} = \vec{0}, \tag{3}$$

som også kan skrives

$$J^{k-1} \Delta \vec{x}^{k-1} = -\vec{f}^{k-1}, \tag{4}$$

hvor $\vec{f}^{k-1} = \left[f_1^{k-1}, f_2^{k-1}\right]^T$ og $\Delta \vec{x}^{k-1} = \left[\Delta x_1^{k-1}, \Delta x_2^{k-1}\right]^T$. For at systemet skal ha en løsning må determinanten til J^{k-1} være forskjellig fra 0.

Vi bruker $\Delta \vec{x}^{k-1}$ beregnet som løsning av ligningssystemet (4) for å beregne den nye tilnærmelsen $\vec{x}^k$ som

$$\vec{x}^k = \vec{x}^{k-1} + \Delta \vec{x}^{k-1}. \tag{5}$$

Formlene (5) og (4) definerer Newton-Raphsons iterasjonen for et system av ikke-lineære ligninger.

Det ligger utenfor rammen av denne fremstillingen å diskutere under hvilke betingelser Newton-Raphsons metode konvergerer. Vi ser imidlertid at under forutsetning om konvergens vil alle elementene i $\vec{f}$-vektoren gå mot 0, noe som igjen medfører at elementene i de beregnede $\Delta\vec{x}$-vektorene også går mot 0.

Eks 1.1 For ligningssystemet

$$x_1^2 + x_2^2 - 9 = 0, \qquad x_1 x_2 - 1 = 0$$

har vi Jacobi matrisen

$$J = \begin{bmatrix} 2x_1 & 2x_2 \\ x_2 & x_1 \end{bmatrix}$$

og $\vec{f} = [x_1^2 + x_2^2 - 9, x_1 x_2 - 1]^T$. Systemet har en rot i nærheten av $[0.5,\ 2.5]^T$. Beregningen for Newton Raphson metode er vist i tabell 1.1. Beregningene er kjørt v.h.a Matlab funksjon 1.

k	x_1^k	x_2^k	Δx_1^k	Δx_2^k
0	0.50000000	2.50000000	-0.20833333	0.54166667
1	0.29166667	3.04166667	0.04280303	-0.05946970
2	0.33446970	2.98219697	0.00096667	-0.00100855
3	0.33543637	2.98118842	0.00000037	-0.00000037
4	0.33543674	2.98118805	–	–

Table 1: Beregning fra eksempel 1.1

Matlabrutinen er **IKKE** pensum. Den er bare ment som som liten demonstrasjon av hvordan det kan gjøres. Koden er mere pedagogisk lagt opp, enn et eksempel på bra matlabkode.

```
%
% Matlab-rutine for  beregne losningen av et ikke-lineaert
% system med 2 ukjente v.h.a Newton-Raphons metode.
%
% Systemet som loses er:
%
%    x_1^2+x_2^2-9=0,  x_1*x_2-1=0
%
% Programmet itererer 4 ganger og skriver ut resultatet til skjerm

x=[0.5;2.5];                       % Initialiserer en startvektor (vaar beste gjetning)
J=[2*x(1) 2*x(2); x(2) x(1)];      % Lager J-matrisen
f=[x(1)^2+x(2)^2-9;x(1)*x(2)-1];   % Lager f-vektoren
delta_x=[0;0];                     % Initialiserer delta_x vektoren
fprintf('k  x_1^k       x_2^k       delta_x_1^k     delta_x_2^k\n');
                                                                    % Skriver
                                                                    % ut
                                                                    % forklaring
                                                                    % til tabellen

for i=0:3                                    % Itererer 4 ganger
    fprintf('%1g  ',i);                      % Skriver ut iterasjonsnummeret
    fprintf('%6.8f  ',x);                    % Skriver ut x-vektoren
    delta_x=J\-f;                            % Beregner delta_x
```

```
    x=x+delta_x;                           % Oppdaterer x-vektor
    J=[2*x(1) 2*x(2); x(2) x(1)];          % Oppdaterer J-matrisen
                                           % (avhengig av x)
    f=[x(1)^2+x(2)^2-9;x(1)*x(2)-1];       % Oppdaterer f-vektor (ogsaa
                                           % avhengig av x)
    fprintf('%6.8f     ',delta_x);         % Skriver ut delta_x
    fprintf('\n');                         % Skriver ut et linjeskift
end
fprintf('%1g  ',i+1);                      % Skriver ut siste iterasjonsnummer
fprintf('%6.8f  ',x);                      % Skriver ut siste beregnede x-vektor
fprintf('    --             --   ',delta_x); % Indikerer at vi ikke har
                                             % beregnet en ny delta_x
fprintf('\n');                               % Skriver ut linkjeskift
```

Matlab funksjon 1 Newton-Raphons

2 Taylor utvikling

For hver komponent f_i i systemet har vi

$$\begin{aligned}
f_i(x_1^{k-1}+\Delta x_1, x_2^{k-1}+\Delta x_2) &= f_i(x_1^{k-1}, x_2^{k-1}+\Delta x_2) + \Delta x_1 \frac{\partial f_i}{\partial x_1}(x_1^{k-1}, x_2^{k-1}+\Delta x_2) + \mathcal{O}(\Delta x_1^2) \\
&= f_i(x_1^{k-1}, x_2^{k-1}) + \Delta x_1 \frac{\partial f_i}{\partial x_1}(x_1^{k-1}, x_2^{k-1}) + \Delta x_2 \frac{\partial f_i}{\partial x_2}(x_1^{k-1}, x_2^{k-1}) + \\
&\quad + \mathcal{O}(\Delta x_1^2) + \mathcal{O}(\Delta x_2^2) + \mathcal{O}(\Delta x_1 \cdot \Delta x_2),
\end{aligned}$$

med $i = 1, 2$. Vi tar bare hensyn til de ledde av orden 1 i Δx_1 og Δx_2 og ledd av høyere orden inngå i residualet $\vec{R}$. Vi får

$$\begin{bmatrix} f_1\left(x_1^{k-1}+\Delta x_1, x_2^{k-1}+\Delta x_2\right) \\ f_2\left(x_1^{k-1}+\Delta x_1, x_2^{k-1}+\Delta x_2\right) \end{bmatrix} = \begin{bmatrix} f_1\left(x_1^{k-1}, x_2^{k-1}\right) \\ f_2\left(x_1^{k-1}, x_2^{k-1}\right) \end{bmatrix} + J^{k-1} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix} + \vec{R},$$

med J^{k-1} Jacobimatrisa beregnet i punktet (x_1^{k-1}, x_2^{k-1}).