

## Układy równań nieliniowych – metoda linearyzacji (Newtona)

Rozwiązujemy układ równań postaci

$$(UN) \quad \begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases}$$

lub w zapisie wektorowym  $\mathbf{F}(\mathbf{X}) = \mathbf{0}$ , gdzie

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \mathbf{F}(\mathbf{X}) = \begin{bmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ f_n(x_1, x_2, \dots, x_n) \end{bmatrix}.$$

Oznaczmy przez  $\mathbf{X}^*$  dokładne rozwiązanie układu (UN), a przez  $\mathbf{X}^0$  początkowe przybliżenie rozwiązania tego układu. Niech  $\mathbf{h} = \mathbf{X}^* - \mathbf{X}^0$ , skąd po rozpisaniu wektorów na współrzędne otrzymamy

$$\begin{aligned} x_1^* &= x_1^0 + h_1 \\ &\dots\dots\dots \\ x_j^* &= x_j^0 + h_j \\ &\dots\dots\dots \\ x_n^* &= x_n^0 + h_n. \end{aligned}$$

Po rozwinięciu w szereg Taylora każdej z funkcji składowej funkcji wektorowej  $\mathbf{F}$ , ograniczając się tylko do pierwszych pochodnych otrzymamy

$$0 = f_i(x_1^*, \dots, x_n^*) \approx f_i(x_1^0, \dots, x_n^0) + \sum_{j=1}^n \frac{\partial f_i}{\partial x_j}(x_1^0, \dots, x_n^0) h_j$$

dla  $i = 1, \dots, n$ . Znając  $\mathbf{X}^0$  oraz wektor poprawek  $\mathbf{h}$  możemy wyznaczyć wektor  $\mathbf{X}^*$  rozwiązań układu (UN) (a właściwie jego dokładniejsze kolejne przybliżenie). Ponieważ  $\mathbf{X}^0$  jest dane, wystarczy w tym celu wyznaczyć składowe wektora poprawek  $\mathbf{h} = [h_1, h_2, \dots, h_n]^T$  rozwiązując następujący układ równań liniowych otrzymany z powyższego częściowego rozwinięcia w szereg Taylora:  $\mathbf{A} \cdot \mathbf{h} = -\mathbf{F}$ , gdzie

$$\mathbf{h} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_n \end{bmatrix}, \quad \mathbf{F} = \mathbf{F}(\mathbf{X}^0) = \begin{bmatrix} f_1(x_1^0, \dots, x_n^0) \\ f_2(x_1^0, \dots, x_n^0) \\ \vdots \\ f_n(x_1^0, \dots, x_n^0) \end{bmatrix},$$
$$\mathbf{A} = \frac{\partial \mathbf{F}(\mathbf{X}^0)}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x_1^0, \dots, x_n^0) & \dots & \frac{\partial f_1}{\partial x_n}(x_1^0, \dots, x_n^0) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1}(x_1^0, \dots, x_n^0) & \dots & \frac{\partial f_n}{\partial x_n}(x_1^0, \dots, x_n^0) \end{bmatrix}.$$

Macierz  $\mathbf{A}$  jest nazywana macierzą gradientów.

Znając teraz  $\mathbf{h}$  możemy obliczyć pierwsze przybliżenie rozwiązania układu (UN) ze wzoru  $\mathbf{X}^1 = \mathbf{X}^0 + \mathbf{h}$ . Kolejne przybliżenia rozwiązania dokładnego oblicza się podobnie:

$$\mathbf{X}^{r+1} = \mathbf{X}^r + \mathbf{h}^{r+1},$$

gdzie kolejne  $\mathbf{h}^{r+1}$  obliczamy z kolejnych układów równań  $\mathbf{A}^r \cdot \mathbf{h}^{r+1} = -\mathbf{F}^r$ , w których  $\mathbf{A}^r = \frac{\partial \mathbf{F}(\mathbf{X}^r)}{\partial \mathbf{X}}$ ,  $\mathbf{F}^r = \mathbf{F}(\mathbf{X}^r)$ .

Trudnością może być problem obliczania elementów macierzy gradientów  $\mathbf{A}$ . Można np. dla danej funkcji wektorowej  $\mathbf{F}$  obliczyć wszystkie  $n^2$  wzorów na jej pierwsze pochodne cząstkowe występujące w  $\mathbf{A}$  i zadeklarować (zdefiniować oczywiście też) je wszystkie w naszym programie. Jednak wtedy program nie będzie wystarczająco elastyczny, zmiana  $n$  składowych funkcji pociągnie za sobą konieczność analitycznego obliczenia wzorów na  $n^2$  pochodnych i wpisania ich w kodzie programu, co jest nieopłacalne już dla niewielkich wartości  $n$ .

Lepszym rozwiązaniem jest numeryczne obliczanie pochodnych np. metodą różnic skończonych

$$\frac{\partial f_i}{\partial x_j} = \frac{f_i(x_1^r, \dots, x_j^r + \Delta, \dots, x_n^r) - f_i(x_1^r, \dots, x_j^r - \Delta, \dots, x_n^r)}{2\Delta}$$

lub metodą Steffensena

$$\frac{\partial f_i}{\partial x_j} = \frac{f_i(x_1^r, \dots, x_j^r + \bar{\Delta}, \dots, x_n^r) - f_i(x_1^r, \dots, x_j^r, \dots, x_n^r)}{\bar{\Delta}},$$

$$\text{gdzie } \bar{\Delta} = f_j(x_1^r, \dots, x_j^r, \dots, x_n^r).$$

W zadaniu mamy dane: funkcję wektorową  $\mathbf{F}$  (przyrównaną do zera, a więc mamy dany układ równań nieliniowych), wektor początkowy  $\mathbf{X}^0$  (może być zadawany przez użytkownika) oraz „dokładność” obliczeń  $\varepsilon_X$  lub  $\varepsilon_Y$  (potrzebną do warunku zakończenia obliczeń, zadawaną przez użytkownika).

W pętli programu obliczamy kolejno: wartości funkcji  $\mathbf{F}$ , elementy macierzy  $\mathbf{A}$ , wartości poprawek  $\mathbf{h}$  z układu równań i kolejne przybliżenie rozwiązania. Na końcu sprawdzamy warunek zakończenia pętli, który może mieć postać

$$\max_{1 \leq i \leq n} |x_i^r - x_i^{r-1}| = \max_{1 \leq i \leq n} |h_i^r| < \varepsilon_X \quad \text{„dokładność rozwiązania”}$$

lub (oraz)

$$\max_{1 \leq i \leq n} |f_i(x_1^r, \dots, x_n^r)| < \varepsilon_Y \quad \text{„dokładność osiągnięcia zera”}.$$

Wynikiem jest wektor rozwiązań  $\mathbf{X}$  otrzymany po zakończeniu pętli, wypisana powinna być też zadana dokładność obliczeń.

## Przykład

Rozwiąż układ równań nieliniowych:

$$\begin{cases} f(x,y) = 0 \\ g(x,y) = 0 \end{cases} \quad \text{gdzie} \quad \begin{cases} f(x,y) = x^3y^2 + xy + xy^3 - 3, \\ g(x,y) = x^2 + x^2y^2 - 2xy. \end{cases} \quad \text{Punkt początkowy: } \begin{cases} x_0 = 1,1 \\ y_0 = 1,05. \end{cases}$$

Tu:

$$X = \begin{bmatrix} x \\ y \end{bmatrix}, h = \begin{bmatrix} h_x \\ h_y \end{bmatrix}, F = F(X) = \begin{bmatrix} f(x,y) \\ g(x,y) \end{bmatrix}, A = \begin{bmatrix} f_x & f_y \\ g_x & g_y \end{bmatrix}. \quad \left( \text{Rozwiązanie: } \begin{cases} x = 1 \\ y = 1 \end{cases} \right)$$

---

Opis kolejnych czynności:

- Czytaj  $x, y$  (początkowe),  $epsX$ ,  $epsF$ ,
- Rozwiązuj (w pętli do .. while) układ równań liniowych  $A \cdot h = -F$  z niewiadomą  $h$  (tu, dla  $n = 2$  można się obejść bez użycia tablic, stosując metodę wyznaczników):  
obliczaj w pętli kolejno:

$$\begin{aligned} w &= f_x \cdot g_y - g_x \cdot f_y \\ w_x &= g \cdot f_y - f \cdot g_y \\ w_y &= f \cdot g_x - g \cdot f_x \\ h_x &= w_x/w ; \quad h_y = w_y/w \\ x_p &= x ; \quad y_p = y \\ x &= x_p + h_x \\ y &= y_p + h_y \end{aligned}$$

aż do spełnienia warunku:

$$\max(|h_x|, |h_y|) < epsX \quad \text{oraz} \quad \max(|f(x,y)|, |g(x,y)|) < epsF.$$

- Pisz  $x, y$ , liczbę iteracji.

Przyjąć np.  $epsX = epsF = 10^{-8}$  (1e-8).

Ponadto spróbuj znaleźć drugie rozwiązanie układu (zob. wykres) startując np. z punktu początkowego  $x_0 = -1$ ,  $y_0 = -1,5$ .