

Metody gradientowe poszukiwania ekstremum

Korzystając z informacji o wartości funkcji oraz jej gradientu. Wykazując ich zbieżność zakłada się, że funkcja celu jest ograniczona od dołu funkcją wypukłą klasy C^2 taką, że można ją aproksymować formą kwadratową o postaci :

$$f(\mathbf{x}) = c + \mathbf{b}^T \mathbf{x} + 1/2 \mathbf{x}^T \mathbf{A} \mathbf{x}$$

gdzie: \mathbf{A} - oznacza macierz symetryczną dodatnio określoną, której elementami są drugie pochodne cząstkowe funkcji $f(\mathbf{x})$.

Na przykład, dla funkcji celu zdefiniowanej jako błąd średniokwadratowy dwóch napięć w układzie:

$$f(\mathbf{x}) = \frac{1}{2} W_1 (U_1 - \underline{U}_1)^2 + \frac{1}{2} W_2 (U_2 - \underline{U}_2)^2, \quad \mathbf{x} = \begin{Bmatrix} U_1 \\ U_2 \end{Bmatrix}, \quad \underline{U} - \text{żądana wartość napięcia},$$

otrzymuje się:

$$f(\mathbf{x}) = \frac{1}{2} \{U_1, U_2\} \begin{bmatrix} W_1 & 0 \\ 0 & W_2 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \end{Bmatrix} - \{W_1 \cdot \underline{U}_1, W_2 \cdot \underline{U}_2\} \begin{Bmatrix} U_1 \\ U_2 \end{Bmatrix} + \frac{1}{2} (W_1 \cdot \underline{U}_1^2 + W_2 \cdot \underline{U}_2^2)$$

czyli

$$\mathbf{A} = \begin{bmatrix} W_1 & 0 \\ 0 & W_2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} W_1 \cdot \underline{U}_1 \\ W_2 \cdot \underline{U}_2 \end{bmatrix}, \quad c = \frac{1}{2} (W_1 \cdot \underline{U}_1^2 + W_2 \cdot \underline{U}_2^2).$$

Podana forma kwadratowa może zostać zminimalizowana poprzez rozwiązanie układu równań $\mathbf{A}\mathbf{x} - \mathbf{b} = 0$. Jeżeli \mathbf{x}_0 jest rozwiązaniem tego równania i wprowadzimy odchyłkę \mathbf{e} , to

$$\begin{aligned} f(\mathbf{x}_0 + \mathbf{e}) &= \frac{1}{2} (\mathbf{x}_0 + \mathbf{e})^T \mathbf{A} (\mathbf{x}_0 + \mathbf{e}) + \mathbf{b}^T (\mathbf{x}_0 + \mathbf{e}) + c \quad (\text{z definicji formy kwadratowej}), \\ &= \frac{1}{2} \mathbf{x}_0^T \mathbf{A} \mathbf{x}_0 + \mathbf{e}^T \mathbf{A} \mathbf{x}_0 + \frac{1}{2} \mathbf{e}^T \mathbf{A} \mathbf{e} + \mathbf{b}^T \mathbf{x}_0 + \mathbf{b}^T \mathbf{e} + c \quad (\text{na podstawie symetrii } \mathbf{A}), \\ &= \frac{1}{2} \mathbf{x}_0^T \mathbf{A} \mathbf{x}_0 + \mathbf{b}^T \mathbf{x}_0 + c - \mathbf{e}^T \mathbf{b} + \frac{1}{2} \mathbf{e}^T \mathbf{A} \mathbf{e} + \mathbf{b}^T \mathbf{e} = f(\mathbf{x}_0) + \frac{1}{2} \mathbf{e}^T \mathbf{A} \mathbf{e}. \end{aligned}$$

Gdy \mathbf{A} jest dodatnio określona, ostatni składnik jest dodatni dla $\mathbf{e} \neq 0$.

Ogólny algorytm:

- 1" wyznaczenie kierunku poszukiwań,
- 2" określenie minimum w tym kierunku (np. interpolacja kwadratowa lub sześcienna).

Metoda gradientu prostego

Jak powyżej, jednak zamiast szukać minimum wykonywany jest krok o długości e .

Informacje wejściowe :

- \mathbf{x}_0 - arbitralnie wybrany punkt startowy,
- e - początkowa długość kroku,
- β - współczynnik zmniejszenia kroku,
- ε - wymagana dokładność obliczeń minimum,
- n - liczba zmiennych niezależnych.

Algorytm obliczeń :

1" oblicz w punkcie startowym \mathbf{x}_0 wartość funkcji celu $F_0 = f(\mathbf{x}_0)$
oraz jej gradient $\mathbf{g}_0 = \mathbf{g}(\mathbf{x}_0)$;

2" wyznacz kierunek poszukiwań $\boldsymbol{\xi} = -\mathbf{g}_0$;

3" wzdłuż kierunku $\boldsymbol{\xi}$ wykonaj krok o długości e oraz określ współrzędne nowego punktu :

$$\mathbf{x}_{i+1} = \mathbf{x}_i + e \boldsymbol{\xi},$$

przy czym dla pierwszej iteracji $\mathbf{x}_i = \mathbf{x}_0$;

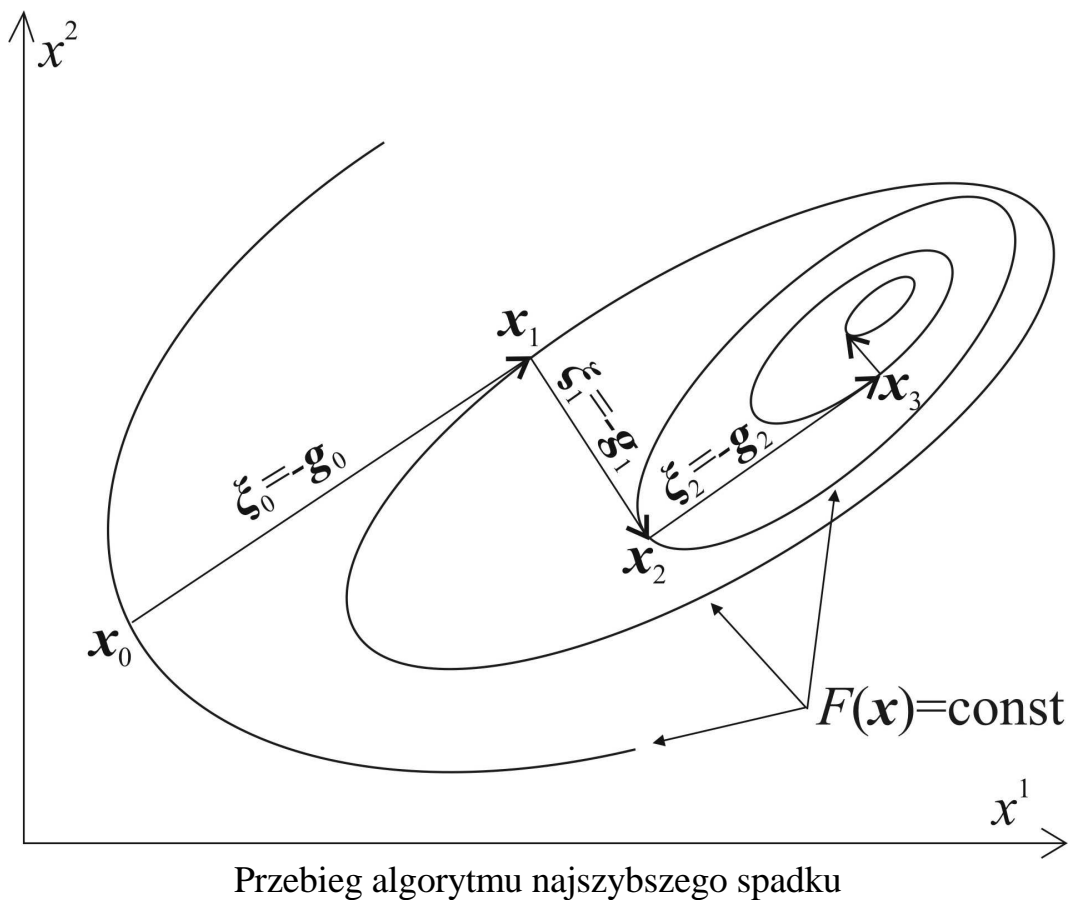
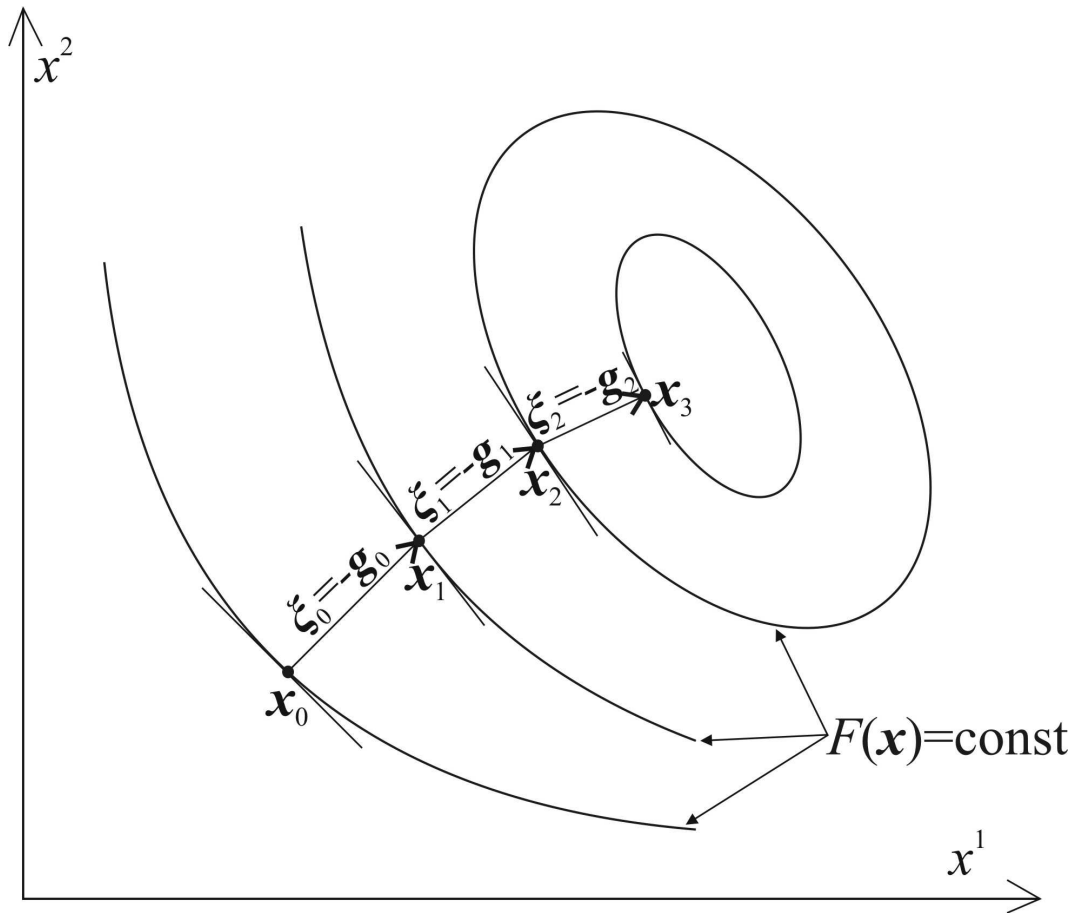
4" oblicz w nowym punkcie wartość funkcji $F = f(\mathbf{x}_{i+1})$
oraz gradientu $\mathbf{g} = \mathbf{g}(\mathbf{x}_{i+1})$.

Jeśli krok był pomyślny $F < F_0$, to powtarzaj od (2") podstawiając \mathbf{g} w miejsce \mathbf{g}_0 .
Inaczej :

5" sprawdź, czy osiągnięto minimum. Jeśli nie, wróć do (4") podstawiając

$$\mathbf{x}_i = \mathbf{x}_{i+1} - e \boldsymbol{\xi}$$

oraz zmniejsz krok o β . Przejdź do (3").



Metoda najszybszego spadku

Zastosowano tu dodatkowo minimalizację funkcji wzdłuż wyznaczonego kierunku.

Zbieżność metody

Założmy, że funkcja celu w pobliżu ekstremum wyraża się przez :

$$f(\mathbf{x}) = 1/2 \mathbf{x}^T \mathbf{A} \mathbf{x} ,$$

wtedy gradient funkcji :

$$\Delta f(\mathbf{x}) = \mathbf{A} \mathbf{x} .$$

Przesunięcie od i do $i+1$ można zapisać jako

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \gamma \mathbf{A} \mathbf{x}_i \quad \text{lub} \quad \mathbf{x}_{i+1} = (\mathbf{1} - \gamma \mathbf{A}) \mathbf{x}_i = \frac{P_1(\mathbf{A})}{P_1(\mathbf{0})} \cdot \mathbf{x}_i = \frac{P_1(\lambda)}{P_1(0)} \cdot \mathbf{x}_i , \quad \text{gd}y\acute{z}: P_1(\mathbf{A}) \cdot \mathbf{v} = P_1(\lambda) \cdot \mathbf{v} ,$$

gdzie: $P_1(\lambda) = 1 - \gamma \lambda$, λ , \mathbf{v} - wartość własna i wektor własny macierzy \mathbf{A} ($\det(\mathbf{A} - \lambda \mathbf{I}) = 0$).

Utwórzmy wielomian pierwszego stopnia macierzy \mathbf{A} , $Q_1(\lambda)$, taki, że:

$$Q_1(\lambda_1) = \alpha_0 + \alpha_1 \lambda_1 = -1 ,$$

$$Q_1(\lambda_n) = \alpha_0 + \alpha_1 \lambda_n = 1 ,$$

gdzie : λ_1 i λ_n są najmniejszą i największą wartością własną macierzy \mathbf{A} . Dodatkowo określona macierz \mathbf{A} ma dodatnie wartości własne. Rozwiązując układ równań otrzymuje się α_0 i α_1 :

$$\alpha_0 = \frac{-\lambda_n - \lambda_1}{\lambda_n - \lambda_1} \quad \text{oraz} \quad \alpha_1 = \frac{2}{\lambda_n - \lambda_1} .$$

Stąd

$$Q_1(\lambda) = \frac{2\lambda - (\lambda_1 + \lambda_n)}{\lambda_n - \lambda_1} , \quad f(\mathbf{x}_{i+1}) = f\left(\frac{P_1(\mathbf{A})}{P_1(0)} \cdot \mathbf{x}_i\right) \leq f\left(\frac{Q_1(\mathbf{A})}{Q_1(0)} \cdot \mathbf{x}_i\right) ,$$

a rozkładając \mathbf{x}_i wzdłuż wektorów własnych \mathbf{v}_j odpowiadających wartościom wł. λ_j dla $j = 1, \dots, n$

$$\mathbf{x}_i = \sum_{j=1}^n a_{ij} \mathbf{v}_j .$$

Korzystając z formy kwadratowej (funkcji celu) :

$$f(\mathbf{x}_{i+1}) \leq f\left(\frac{Q_1(\mathbf{A})}{Q_1(0)} \cdot \mathbf{x}_i\right) = \frac{1}{2} \left[\frac{Q_1(\mathbf{A})}{Q_1(0)} \cdot \mathbf{x}_i \right]^T \mathbf{A} \frac{Q_1(\mathbf{A})}{Q_1(0)} \mathbf{x}_i ,$$

skąd

$$f(\mathbf{x}_{i+1}) \leq \frac{1}{2} \frac{1}{Q_1(0)^2} \sum_j a_{ij}^2 Q_1(\lambda_j)^2 \lambda_j \leq \frac{1}{2} \frac{1}{Q_1(0)^2} \sum_j a_{ij}^2 \lambda_j .$$

Ponieważ z definicji $|Q_1(\lambda_j)| \neq 0$, $j = 1, 2, \dots, n$ więc ostatecznie:

$$f(\mathbf{x}_{i+1}) \leq \frac{1}{Q_1(0)^2} f(\mathbf{x}_i) \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \right)^2 f(\mathbf{x}_i) .$$

Informacje wejściowe

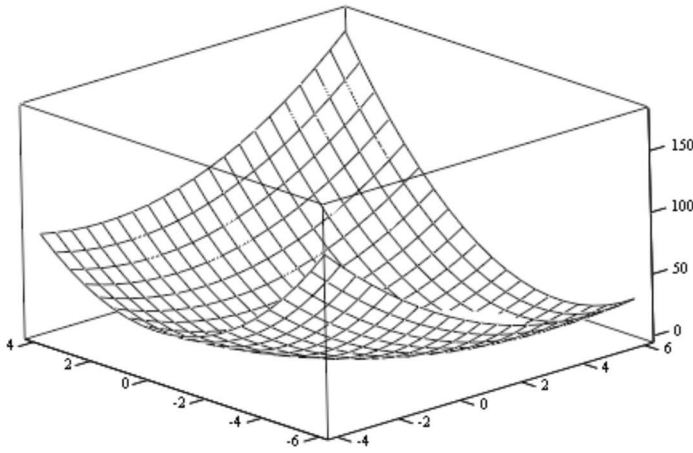
- \mathbf{x}_0 - arbitralnie wybrany punkt startowy,
 e - początkowa długość kroku,
 ε_j - wymagana dokładność obliczeń w aktualnie występującym kierunku poszukiwań,
 ε_0 - wymagana dokładność obliczeń minimum globalnego,
 n - liczba zmiennych niezależnych.

Algorytm obliczeń :

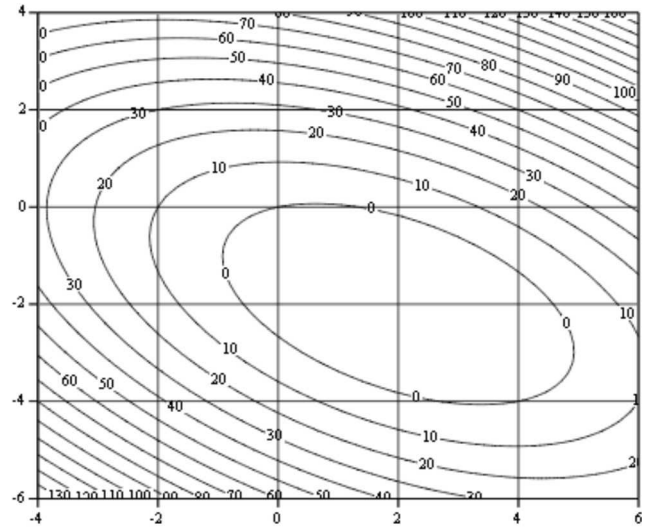
- 1" oblicz w punkcie startowym \mathbf{x}_0 wartość funkcji celu $F_0 = f(\mathbf{x}_0)$
oraz jej gradient $\mathbf{g}_0 = \mathbf{g}(\mathbf{x}_0)$;
- 2" wyznacz kierunek poszukiwań $\xi_i = -\mathbf{g}$;
- 3" wzdłuż kierunku ξ_i określ λ_i minimalizujące $f(\mathbf{x}_{i-1} + \lambda_i \xi_i)$
oraz współrzędne nowego punktu $\mathbf{x}_i = \mathbf{x}_{i-1} + \lambda_i \xi_i$
- 4" oblicz w nowym punkcie wartość gradientu $\mathbf{g} = \mathbf{g}(\mathbf{x}_{i+1})$.
- 5" sprawdź, czy osiągnięto minimum. Jeśli nie, wróć do (2")

Przykład

Forma kwadratowa: $f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \cdot \mathbf{A} \cdot \mathbf{x} - \mathbf{b}^T \cdot \mathbf{x} + c$, $\mathbf{A} = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}$, $\mathbf{b} = \begin{Bmatrix} 2 \\ -8 \end{Bmatrix}$, $c=0$.



f



f

- Gradient formy kwadratowej: $\text{grad } f(\mathbf{x}) = \mathbf{A} \mathbf{x} - \mathbf{b}$ (reszta równania).
- Kierunek poszukiwań $\xi_i = -\text{grad } f(\mathbf{x})$
- Wyznaczanie minimum na kierunku:

$$\mathbf{x}_{(1)} = \mathbf{x}_{(0)} + \alpha \cdot \xi_{(0)}$$

W punkcie minimum na kierunku gradient musi być ortogonalny do tego kierunku:

$$\xi_{(1)}^T \xi_{(0)} = 0$$

$$(\mathbf{b} - \mathbf{A} \cdot \mathbf{x}_{(1)})^T \cdot \xi_{(0)} = 0$$

$$(\mathbf{b} - \mathbf{A} \cdot (\mathbf{x}_{(0)} + \alpha \xi_{(0)}))^T \cdot \xi_{(0)} = 0$$

$$(\mathbf{b} - \mathbf{A} \cdot \mathbf{x}_{(0)})^T \cdot \xi_{(0)} - \alpha (\mathbf{A} \cdot \xi_{(0)})^T \cdot \xi_{(0)} = 0$$

$$(\mathbf{b} - \mathbf{A} \cdot \mathbf{x}_{(0)})^T \cdot \xi_{(0)} = \alpha (\mathbf{A} \cdot \xi_{(0)})^T \cdot \xi_{(0)}$$

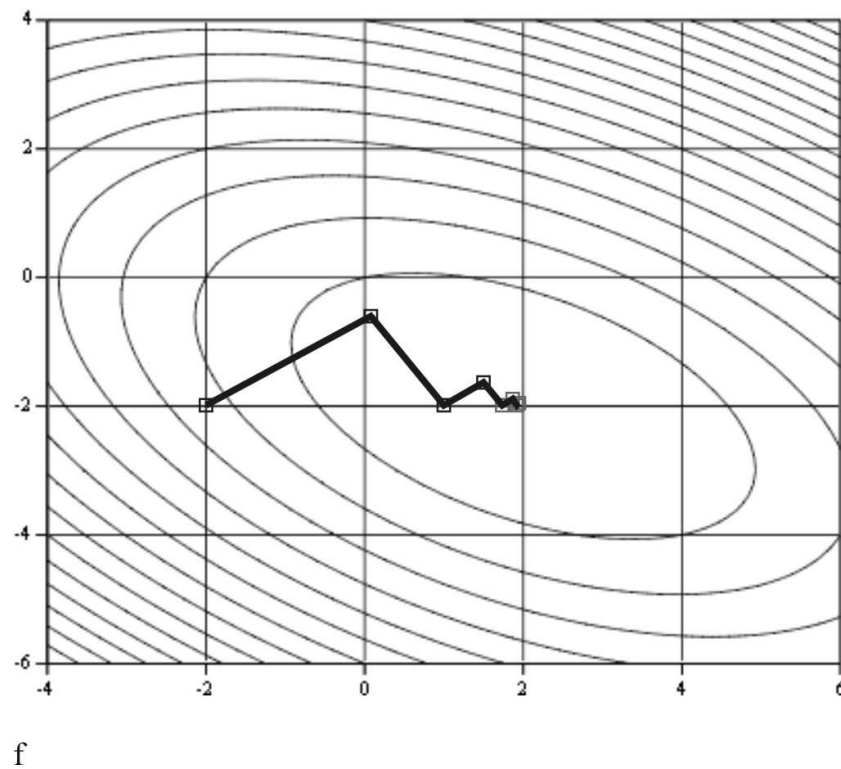
$$\xi_{(0)}^T \cdot \xi_{(0)} = \alpha \xi_{(0)}^T \cdot \mathbf{A} \cdot \xi_{(0)}$$

$$\alpha = \frac{\xi_{(0)}^T \cdot \xi_{(0)}}{\xi_{(0)}^T \cdot \mathbf{A} \cdot \xi_{(0)}}$$

Procedura programu Mathcad:

$$S_D := \left| \begin{array}{l} x_1 \leftarrow x_0 \\ \text{for } i \in 1..I \\ \quad r_i \leftarrow b - A \cdot x_i \\ \quad \text{grad}_f \leftarrow -r_i \\ \quad \alpha \leftarrow \frac{r_i^T \cdot r_i}{r_i^T \cdot A \cdot r_i} \\ \quad x_{i+1} \leftarrow x_i + \alpha \cdot r_i \end{array} \right|_x$$

Przebieg algorytmu:



ITERACJE JACOBIEGO

Metoda Jacobiego służy do rozwiązywania zadań równoważnych do $Ax = b$. Polega ona na rozkładzie macierzy A na macierz diagonalną D utworzoną z jej przekątnej głównej oraz na pozostałą część E :

$$A = D + E$$

Wyprowadzenie metody:

$$A \cdot x = b$$

$$D \cdot x = -E \cdot x + b$$

$$x = -D^{-1} \cdot E \cdot x + D^{-1} \cdot b$$

Wzór ten można zapisać jako:

$$x = B \cdot x + z, \text{ gdzie: } B = -D^{-1} \cdot E, z = D^{-1} \cdot b.$$

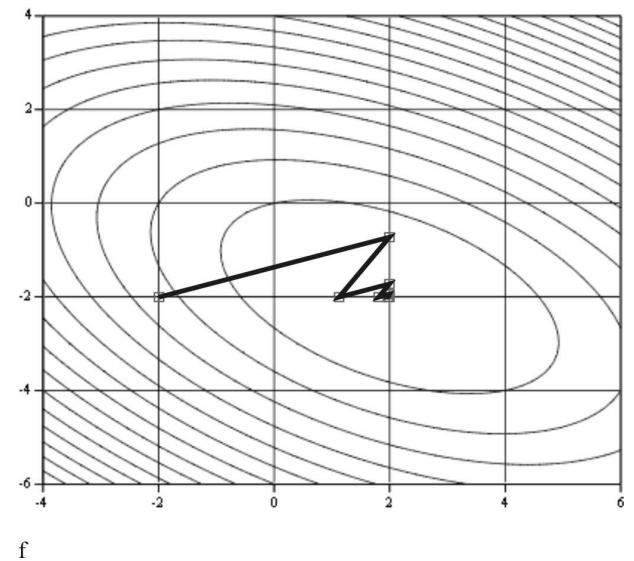
Macierz D jest diagonalna i łatwa do odwrócenia. Można podać wzór iteracyjny:

$$x_{i+1} = B \cdot x_i + z$$

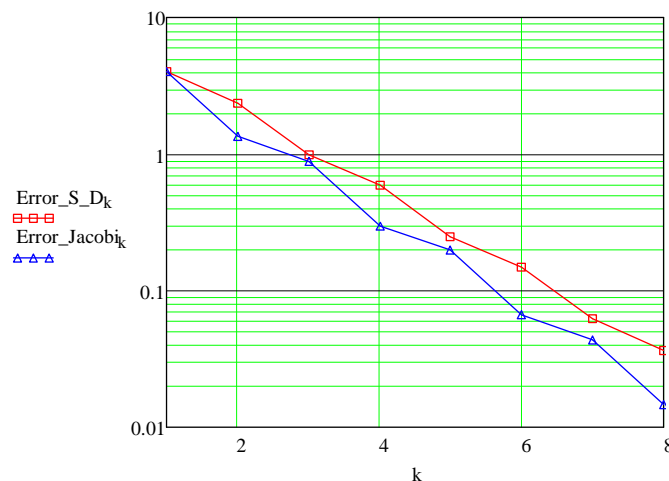
Procedura Mathcad'a:

```
Jacobi :=
| x1 ← x0
| D ← diag(diag(A))
| E ← A - D
| B ← -D-1 · E
| z ← D-1 · b
| for i ∈ 1..I
|   xi+1 ← B xi + z
| x
```

Przebieg algorytmu:



Porównanie zbieżności metody najszybszego spadku i metody Jacobiego



METODA NEWTONA-RAPHSONA

Wprowadzenie

W wielu przypadkach równanie opisujące obwód nieliniowy jest jednorodne, tzn. ma postać (1). Metoda Newtona-Raphsona, która należy do klasy metod iteracyjnych, pozwala rozwiązać równanie o postaci (1) przekształcając je do postaci (2)

$$h(x) = 0, \quad (1)$$

$$x = x - g(x) h(x). \quad (2)$$

Funkcje $g(x)$ wybiera się tak, aby proces iteracyjny był zbieżny. Spośród wielu możliwych postaci funkcji $g(x)$ najdogodniej jest wybrać ją jako odwrotność pochodnej funkcji $h'(x)$, tzn. przyjmując:

$$g(x) = \frac{1}{h'(x)}. \quad (3)$$

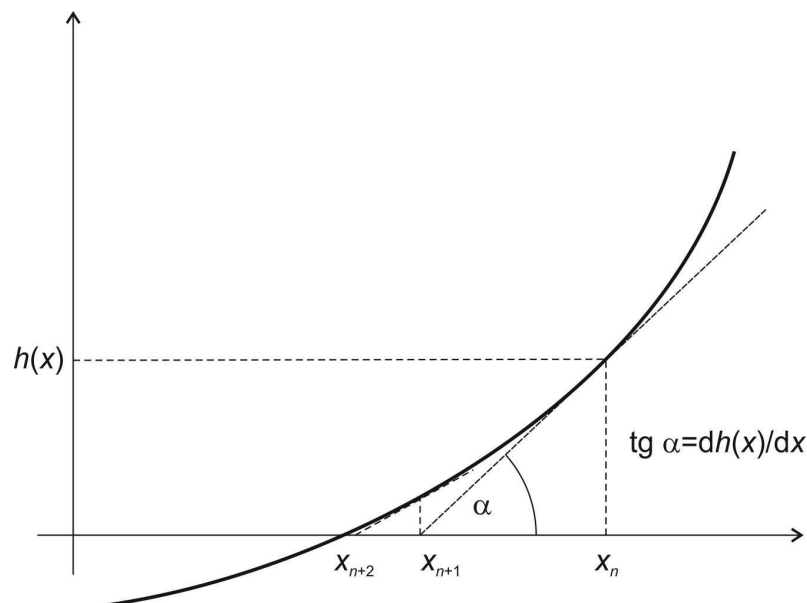
Wówczas równanie (2) przyjmuje postać:

$$x = x - \frac{h(x)}{h'(x)}, \quad (4)$$

a algorytm jego rozwiązania ma postać

$$x_{n+1} = x_n - \frac{h(x_n)}{h'(x_n)}. \quad (5)$$

Proces rozwiązania dany algorytmem (5) można prosto zinterpretować graficznie. Przykład takiej interpretacji jest przedstawiony na rys.1.



Rys.1

Metoda Newtona-Raphsona prowadzi do procesu iteracyjnego, który jest tym szybciej zbieżny, im wartość początkowa x_0 poszukiwanego rozwiązania jest bliższa wartości rzeczywistej \underline{x} . Zależnie od postaci funkcji $h(x)$ proces przy określonej wartości x_0 może być rozbieżny. Jednakże podkreślić należy, że przez odpowiedni dobór wartości początkowej x_0 zbieżność procesu iteracji według metody Newtona-Raphsona można zawsze uzyskać.

Przykład rozwiązania równania

W celu zilustrowania tej metody można rozwiązać równanie o postaci

$$x+2x^{1/3}-4=0 \quad (6)$$

Stąd mamy

$$h(x)=x+2x^{1/3}-4 \quad (7)$$

a:

$$h'(x)=1+\frac{2}{3}x^{-2/3} \quad (8)$$

Zatem algorytm obliczeń można zapisać, zgodnie z zależnością (5), w następującej postaci:

$$x_{n+1}=x_n-\frac{x_n+2x_n^{1/3}-4}{1+\frac{2}{3}x_n^{-2/3}} \quad (9)$$

Przyjmując wartość początkową $x_0=27$ otrzymujemy po ośmiu krokach iteracyjnych wartość: $x = 1.641$. Poniżej przedstawione są wyniki uzyskane za pomocą programu Mathcad:

	0		0
$x =$	27	$h =$	29
1	$3.553 \cdot 10^{-15}$	1	-4
2	$1.397 \cdot 10^{-9}$	2	-3.9978
3	$7.495 \cdot 10^{-6}$	3	-3.9609
4	$2.282 \cdot 10^{-3}$	4	-3.7344
5	0.097	5	-2.9845
6	0.814	6	-1.318
7	1.561	7	-0.1185
8	1.641	8	-0.0006

Tok postępowania w metodzie Newtona-Raphsona można usystematyzować:

1. Wybór w miarę dobrej wartości początkowej x_0 do rozwiązania równania (1). Jeśli $h(x_0)=0$, to przyjęta wartość stanowi rozwiązanie, czyli $x_0=\underline{x}$. Jeżeli $h(x_0)$ różne od 0, to przechodzimy do drugiego kroku.

2. Przyjmujemy następną wartość; zgodnie z zależnością: $x_{n+1}=x_n-\frac{h(x_n)}{h'(x_n)}$.

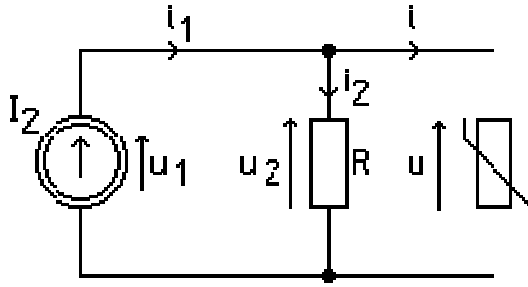
W przypadku, gdy $h(x_1)$ jest różne od 0, proces powtarzamy n razy. W praktyce proces powtarzamy tak długo, aż wartość $h(x_n)$ będzie dostatecznie mała, tzn. aż do spełnienia nierówności:

$$|x_{n+1}-x_n| \cong 0 \quad \text{lub} \quad (10)$$

$$|h(x_n)| \leq \varepsilon. \quad (11)$$

Przykład rozwiązania obwodu nieliniowego

Rozpatrzmy teraz przykład rozwiązania metoda Newtona-Raphsona obwodu nieliniowego przedstawionego na rys.3:



Niech: $R=1[\Omega]$, $I_2=1[A]$,

a charakterystyka opornika nieliniowego jest dana wyrażeniem:

$$i=2(e^{0.1u}-1). \quad (12)$$

Równania opisujące elementy obwodu oraz wyznaczone na podstawie praw Kirchhoffa przyjmują postać:

$$\begin{aligned} u_2 &= Ri_2 \\ i &= 2(e^{0.1u} - 1) \\ i_1 &= I_2 = 1A \\ i_1 - i - i_2 &= 0, \\ u_1 - u_2 &= 0, \\ u - u_2 &= 0, \end{aligned} \quad (13)$$

Po podstawieniu otrzymujemy równanie dla poszukiwanego napięcia, które w postaci dogodnej do zastosowania metody Newtona-Raphsona przedstawiamy jako $h(u)=0$:

$$\begin{aligned} h(u) &= 1 - 2(e^{0.1u} - 1) - u = 0, \\ h'(u) &= -0.2e^{0.1u} - 1. \end{aligned} \quad (14)$$

Ze względu na występowanie funkcji wykładniczej w wyrażeniu $h(u)$ możemy stwierdzić, że funkcja $h(u)$ jest monotoniczna, a zatem proces iteracji jest zbieżny. Przyjmujemy wartość początkową $u_0=3V$ i żądamy, by obliczenia zakończyć dla wartości bezwzględnej $h(u)$ mniejszej od 10^{-8} . Wynik jest następujący:

	0		0
0	3	0	-2.6997
1	0.87419	1	-0.0569
2	0.82749	2	$-2.3769 \cdot 10^{-5}$
3	0.82747	3	$-4.142 \cdot 10^{-12}$
4	0.82747	4	0
5	0.82747	5	0
6	0.82747	6	0
7	0.82747	7	0
8	0.82747	8	0

Żadaną dokładność uzyskujemy po czterech iteracjach.

Układy równań nieliniowych

Metoda Newtona-Raphsona ma również zastosowanie w przypadku algebraicznych równań nieliniowych, które opisują obwody z elementami wielokońcówkowymi i źródłami sterowanymi. W przypadku układu n równań z n niewiadomymi zapisanymi w postaci wektora

$$\mathbf{x} = [x_1, x_2, x_3, \dots, x_n]^T \quad (15)$$

równanie opisujące obwód przedstawiamy w postaci

$$\mathbf{h}(\mathbf{x}) = \mathbf{0} \quad (16)$$

gdzie

$$\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), h_3(\mathbf{x}), \dots, h_n(\mathbf{x})]^T \quad (17)$$

Algorytm obliczenia rozwiązania równania (17) jest następujący

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{J}^{-1}(\mathbf{x}) \mathbf{h}(\mathbf{x}) \quad (18)$$

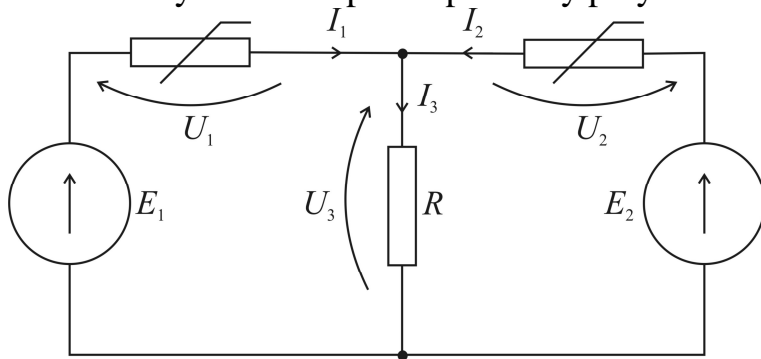
gdzie $\mathbf{J}(\mathbf{x})$ oznacza macierz Jacobiego i ma postać:

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \dots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \dots & \frac{\partial h_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial h_n}{\partial x_1} & \frac{\partial h_n}{\partial x_2} & \dots & \frac{\partial h_n}{\partial x_n} \end{bmatrix} \quad (19)$$

gdzie: h_n i x_n oznaczają n -te elementy wektorów \mathbf{h} i \mathbf{x} .

Przykład rozwiązania układu równań nieliniowych

Jako ilustrację zastosowania metody Newtona-Raphsona w przypadku wyznaczania wielu niewiadomych można podać poniższy przykład:



Dane:

$$E_1 = 10\text{V},$$

$$E_2 = 10\text{V},$$

$$R = 1\Omega,$$

$$I_1 = 20(e^{0,1U_1} - 1)[\text{A}],$$

$$I_2 = 0,5U_2^2[\text{A}].$$

Na podstawie drugiego i pierwszego prawa Kirchhoffa otrzymujemy równania:

$$U_1 + U_3 = E_1,$$

$$U_2 + U_3 = E_2,$$

$$U_3 = R \cdot I_3 = R(I_1 + I_2) = R[20(e^{0,1U_1} - 1) + 0,5U_2^2].$$

Stąd otrzymuje się dwa równania:

$$U_1 + R[20(e^{0,1U_1} - 1) + 0,5U_2^2] = E_1,$$

$$U_2 + R[20(e^{0,1U_1} - 1) + 0,5U_2^2] = E_2 \text{ oraz}$$

$$h(\mathbf{U}) = \begin{cases} U_1 + R[20(e^{0,1U_1} - 1) + 0,5U_2^2] - E_1 \\ U_2 + R[20(e^{0,1U_1} - 1) + 0,5U_2^2] - E_2 \end{cases}.$$

Zatem macierz Jacobiego dla danego układu ma postać:

$$\mathbf{J}(\mathbf{U}) = \begin{bmatrix} 1 + 2 \cdot R \cdot e^{0,1U_1} & U_2 \\ 2 \cdot R \cdot e^{0,1U_1} & 1 + U_2 \end{bmatrix} \quad (21)$$

Korzystając z równania (21) piszemy:

$$\begin{Bmatrix} U_{1,n+1} \\ U_{2,n+1} \end{Bmatrix} = \begin{Bmatrix} U_{1,n} \\ U_{2,n} \end{Bmatrix} - \begin{bmatrix} 1 + 2 \cdot R \cdot e^{0,1U_1} & U_2 \\ 2 \cdot R \cdot e^{0,1U_1} & 1 + U_2 \end{bmatrix}^{-1} \begin{Bmatrix} U_1 + R[20(e^{0,1U_1} - 1) + 0,5 \cdot U_2^2] - E_1 \\ U_2 + R[20(e^{0,1U_1} - 1) + 0,5 \cdot U_2^2] - E_2 \end{Bmatrix}. \quad (22)$$

Przy użyciu programu Mathcad uzyskujemy następujące wyniki:

$(U_n)_1 =$	$(U_n)_2 =$	$(h_n)_1 =$	$(h_n)_2 =$	$I1_n =$	$I2_n =$	$I3_n =$
0	0	-10	-10	0	0	0
3.333	3.333	6.801	6.801	7.912	5.556	13.468
2.379	2.379	0.579	0.579	5.371	2.829	8.2
2.281	2.281	$5.998 \cdot 10^{-3}$	$5.998 \cdot 10^{-3}$	5.124	2.601	7.725
2.28	2.28	$6.706 \cdot 10^{-7}$	$6.706 \cdot 10^{-7}$	5.121	2.599	7.72
2.28	2.28	$8.882 \cdot 10^{-15}$	$1.066 \cdot 10^{-14}$	5.121	2.599	7.72
2.28	2.28	0	$-3.553 \cdot 10^{-15}$	5.121	2.599	7.72

czyli przybliżonymi wartościami prądów są: $I_1 = 5,121\text{A}$ oraz $I_2 = 2,599\text{A}$.

Metoda gradientów sprzężonych

Opracowana przez Hestensa i Stiefela w 1952 do rozwiązywania układów równań liniowych.

Kierunki poszukiwań tworzone są tak, aby każdy kolejny był sprzężony do wszystkich poprzednich.

Dwa kierunki ξ_i oraz ξ_j są wzajemnie sprzężone względem dodatnio określonej macierzy A , jeśli

$$\xi_i^T A \xi_j = 0 \quad \text{dla } i \neq j.$$

Kierunki wzajemnie sprzężone są liniowo niezależne.

Lemat: Jeśli punkt \mathbf{x}_{i+1} został osiągnięty w rezultacie minimalizacji funkcji

$$f(\mathbf{x}) = a + \mathbf{b}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T A \mathbf{x}$$

wzdłuż wzajemnie sprzężonych kierunków $\xi_1, \xi_2, \dots, \xi_i$ to

$$\xi_s^T \nabla f(\mathbf{x}_{i+1}) = 0, \quad \text{dla } s = 1, 2, \dots, i.$$

Dowód

$$\begin{aligned} \nabla f(\mathbf{x}_{i+1}) &= A \mathbf{x}_{i+1} + \mathbf{b} = A \left(\mathbf{x}_{s+1} + \sum_{j=s+1}^i \lambda_j \xi_j \right) + \mathbf{b} \\ &= \nabla f(\mathbf{x}_{s+1}) + \sum_{j=s+1}^i \lambda_j A \xi_j \quad \text{dla } s = 0, 1, 2, \dots, i-1 \end{aligned}$$

ale

$$\xi_s^T \nabla f(\mathbf{x}_{i+1}) = \xi_s^T \nabla f(\mathbf{x}_{s+1}) + \sum_{j=s+1}^i \lambda_j \xi_s^T A \xi_j = 0 \quad \text{c.n.d.}$$

gdyż:

$$(\diamond) \quad \xi_s^T \nabla f(\mathbf{x}_{i+1}) = 0 \quad \text{z warunku minimalizacji funkcji wzdłuż kierunku } \xi_s,$$

$$(\diamond\diamond) \quad \sum_{j=s+1}^i \lambda_j \xi_s^T A \xi_j = 0 \quad \text{z warunku sprzężenia kierunków.}$$

Po n iteracjach $\nabla f(\mathbf{x}_{n+1})$ będzie ortogonalny do n liniowo niezależnych wektorów, musi więc się równać zeru.

Tworzenie kierunków sprzężonych

pierwszy kierunek - minus gradient : $\xi_1 = -\nabla f(\mathbf{x}_1) = -\mathbf{A} \mathbf{x}_1 - \mathbf{b}$,
 minimalizacja $f(x)$ wzdłuż tego kierunku : $\xi_1^T [\mathbf{A} (\mathbf{x}_1 + \lambda_1 \xi_1) + \mathbf{b}] = 0$,
 daje wartość λ_1 i można określić punkt \mathbf{x}_2 : $\mathbf{x}_2 = \mathbf{x}_1 + \lambda_1 \xi_1$.

W drugim kroku nowy kierunek w punkcie \mathbf{x}_2 jest wyznaczany wg reguły:

$\xi_2 = -\nabla f(\mathbf{x}_2) + \beta_2 \xi_1$, a współczynnik β_2 jest tak dobierany,
 aby kierunki ξ_2, ξ_1 były sprzężone :

$$0 = \xi_1^T \mathbf{A} \xi_2 = -\xi_1^T \mathbf{A} \nabla f(\mathbf{x}_2) + \beta_2 \xi_1^T \mathbf{A} \xi_1 = -\xi_1^T \mathbf{A} [\nabla f(\mathbf{x}_2) - \beta_2 \xi_1] =$$

$$= -(\mathbf{x}_2 - \mathbf{x}_1)^T \mathbf{A} [\nabla f(\mathbf{x}_2) - \beta_2 \xi_1] = -[\nabla f(\mathbf{x}_2) - \nabla f(\mathbf{x}_1)]^T [\nabla f(\mathbf{x}_2) - \beta_2 \xi_1],$$

$$\beta_2 = \frac{\nabla f(\mathbf{x}_2)^T \nabla f(\mathbf{x}_2)}{\nabla f(\mathbf{x}_1)^T \nabla f(\mathbf{x}_1)}, \text{ gdyż : } \nabla f(\mathbf{x}_1)^T \nabla f(\mathbf{x}_2) = 0.$$

Podobnie w i -tym kroku : $\xi_i = -\nabla f(\mathbf{x}_i) + \beta_i \xi_{i-1}$, a :

$$\beta_i = \frac{\nabla f(\mathbf{x}_i)^T \nabla f(\mathbf{x}_i)}{\nabla f(\mathbf{x}_{i-1})^T \nabla f(\mathbf{x}_{i-1})}.$$

Informacje wejściowe :

- \mathbf{x}_0 - punkt startowy,
- e - początkowa długość kroku,
- ε_j - dokładność obliczeń w aktualnym kierunku poszukiwań
- ε_0 - dokładność obliczeń minimum globalnego
- n - liczba zmiennych niezależnych.

Algorytm :

- (1) obliczyć $f(\mathbf{x}_0)$ oraz $\nabla f(\mathbf{x}_0)$,
- (2) wyznaczyć kierunek $\xi_i = -\nabla f$,
- (3) wzdłuż ξ_i określić λ_i minimalizujące $f(\mathbf{x}_{i-1} + \lambda_i \xi_i)$ oraz $\mathbf{x}_i = \mathbf{x}_{i-1} + \lambda_i \xi_i$;
- (4) obliczyć $\nabla f(\mathbf{x}_i)$,
- (5) zbadać, czy jest to minimum, jeżeli tak to STOP .
- (6)

$$\beta_i = \frac{\nabla f(\mathbf{x}_i)^T \nabla f(\mathbf{x}_i)}{\nabla f(\mathbf{x}_{i-1})^T \nabla f(\mathbf{x}_{i-1})}, \xi_i = -\nabla f(\mathbf{x}_i) + \beta_i \xi_{i-1}$$

- (7) podstawić \mathbf{x}_i w miejsce \mathbf{x}_{i-1} oraz sprawdzenie, czy wykonano n iteracji.
 Jeśli nie, to powrót do kroku (3) i zwiększenie i o 1.
 Jeżeli wykonano n iteracji, podstawić $\nabla f(\mathbf{x}_i)$ w miejsce $\nabla f(\mathbf{x}_0)$ i wrócić do (2).

Kryterium zbieżności

Punkt \mathbf{x} jest ekstremum, gdy $\nabla f(\mathbf{x})^T \nabla f(\mathbf{x}) = 0$ (z błędem zaokrągleń).