1DI2235:A - Metody optymalizacji, lab.
Metody quasi-newtonowskie.

(opracował: M.T., ostatnia modyfikacja: 17 marca 2024)

1 Preliminaria

1.1 Warunek siecznej

Jeśli funkcja celu jest funkcją jednej zmiennej $f_0 : \mathbb{R} \to \mathbb{R}$ to rekurencyjny wzór Newtona ma postać

$$x_{k+1} = x_k - \frac{f_0^{'}(x_k)}{f_0^{''}(x_k)}, \qquad k = 0, 1, \ldots \tag{1}$$

Zamiast drugiej pochodnej $f_0^{''}(x_k)$ możemy spróbować użyć przybliżenia

$$f_0^{''}(x_k) \approx \frac{f_0^{'}(x_k) - f_0^{'}(x_{k-1})}{x_k - x_{k-1}}, \tag{2}$$

które prowadzi do tzw. wzoru siecznych (ang. *secant method*)

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f_0^{'}(x_k) - f_0^{'}(x_{k-1})} f_0^{'}(x_k), \tag{3}$$

gdzie $k = -1, 0, 1, \ldots$.
Zauważmy, że metoda siecznych

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f_0^{'}(x_k) - f_0^{'}(x_{k-1})} f_0^{'}(x_k), \tag{4}$$

gdzie $k = -1, 0, 1, \ldots$, wymaga *dwóch* punktów startowych, tzn. x_{-1} i x_0. Zauważmy również, że w powyższym wzorze nie występuje f_0.

Jeśli przyjąć, że $g(x) = f_0^{'}(x)$, to otrzymujemy wzór

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{g(x_k) - g(x_{k-1})} g(x_k), \tag{5}$$

gdzie $k = -1, 0, 1, \ldots$, który możemy użyć do rozwiązania równania

$$g(x) = 0. \tag{6}$$

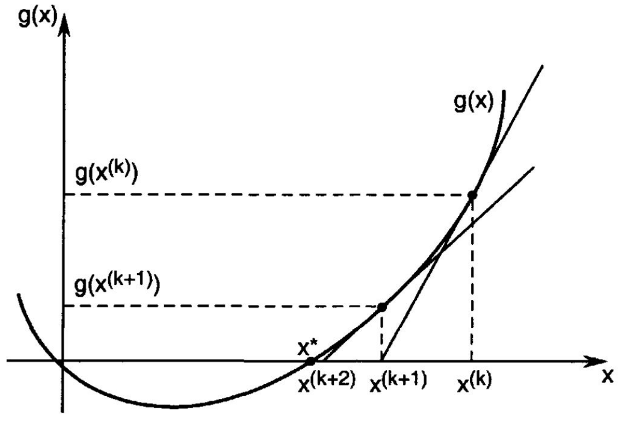

Figure 7.8 **Newton's method of tangents**

Rysunek 1: Ilustracja metody stycznych (Newton) [1].

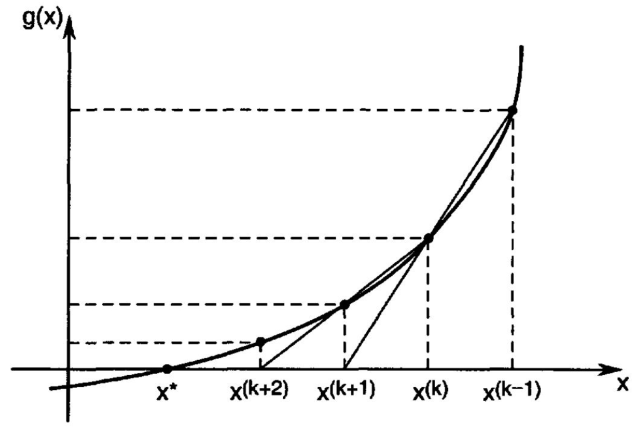

Figure 7.10 **Secant method for root finding**

Rysunek 2: Ilustracja metody siecznych (quasi-Newton) [1].

Wadą metody Newtona dla funkcji wielu zmiennych

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - s_k \left[\nabla^2 f_0(\boldsymbol{x}_k)\right]^{-1} \nabla f_0(\boldsymbol{x}_k) \tag{7}$$

jest koniecznoćś wyznaczenia i odwrócenie w każdej iteracji macierzy hesjanu $\nabla^2 f_0(\boldsymbol{x}_k)$, czy ściślej mówiąc, wyznaczenia macierzy hesjanu $\nabla^2 f_0(\boldsymbol{x}_k)$, rozwiązaniu względem Δ równania

$$\nabla^2 f_0(\boldsymbol{x}_k)\Delta = -\nabla f_0(\boldsymbol{x}_k) \tag{8}$$

a następnie aktualizacji

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + s_k\Delta. \tag{9}$$

W metodach quasi-newtonowskich, zwanych też *metodami zmiennej metryki*, zamiast hesjanu $\nabla^2 f_0(\boldsymbol{x}_k)$ używa się pewnej macierzy $\boldsymbol{H}_k$, aktualizowana w każdej iteracji zgodnie z pewną regułą. Punktem wyjścia metod quasi-newtonowskich jest spostrzeżenie, że dla funkcji kwadratowej

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x}, \tag{10}$$

zachodzi

$$\nabla f(\boldsymbol{x}) = \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{b}, \qquad \nabla^2 f(\boldsymbol{x}) = \boldsymbol{Q}, \tag{11}$$

skąd

$$\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}) = \boldsymbol{Q}(\boldsymbol{x} - \boldsymbol{y}). \tag{12}$$

Mnożąc równanie (12) lewostronnie przez $\boldsymbol{H} = \boldsymbol{Q}^{-1}$ otrzymujemy tzw. warunek siecznej (ang. *secant condition*) dla odwrotności $\boldsymbol{H}$ hesjanu $\boldsymbol{Q}$

$$\boldsymbol{H}\left[\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\right] = \boldsymbol{x} - \boldsymbol{y}. \tag{13}$$

Intuicjnie przyjmujemy, że jeśli $\boldsymbol{H}$ spełnia warunek siecznej a funkcja f_0 może być przybliżona funkcją kwadratową, to odwrotność jej hesjanu $\boldsymbol{Q}^{-1}$ można przybliżyć macierzą $\boldsymbol{H}$.

1.2 Reguły aktualizacji macierzy $\boldsymbol{H}$

W metodach quasi-newtonowskich kładziemy $\boldsymbol{H}_0 = \boldsymbol{I}$ a następnie aktualizujemy wybranym sposobem (w zależności od konkretnej metody), przy spełnieniu warunku siecznej

$$\boldsymbol{H}_{k+1}\left[\nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k)\right] = \boldsymbol{x_{k+1}} - \boldsymbol{x}_k. \tag{14}$$

Przyjmijmy oznaczenia

$$\Delta \boldsymbol{x}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k, \tag{15}$$

oraz

$$\Delta \boldsymbol{g}_k = \nabla f_0(\boldsymbol{x}_{k+1}) - \nabla f_0(\boldsymbol{x}_k). \tag{16}$$

Do najczęściej stosowanych reguł aktualizacji macierzy H, należą tzw. symetryczna poprawka pierwszego rzędu (SR1, ang. *symmetric-rank-one*), poprawka Davidona-Fletchera-Powella (DFP) oraz poprawka Broydena-Fletchera-Goldfarba-Shanno (BFGS).

1.2.1 Symetryczna poprawka pierwszego rzędu (SR1)

$$\boldsymbol{H}_{k+1} = \boldsymbol{H}_k + \frac{(\Delta \boldsymbol{x}_k - \boldsymbol{H}_k \Delta \boldsymbol{g}_k)(\Delta_k \boldsymbol{x} - \boldsymbol{H}_k \Delta \boldsymbol{g}_k)^{\mathrm{T}}}{(\Delta \boldsymbol{x}_k - \boldsymbol{H}_k \Delta \boldsymbol{g}_k)^{\mathrm{T}} \Delta \boldsymbol{g}_k} \tag{17}$$

1.2.2 Poprawka Davidona-Fletchera-Powella (DFP)

$$\boldsymbol{H}_{k+1} = \boldsymbol{H}_k + \frac{\Delta \boldsymbol{x}_k \Delta \boldsymbol{x}_k^{\mathrm{T}}}{\Delta \boldsymbol{g}_k^{\mathrm{T}} \Delta \boldsymbol{x}_k} - \frac{\boldsymbol{H}_k \Delta \boldsymbol{g}_k \left[\boldsymbol{H}_k \Delta \boldsymbol{g}_k\right]^{\mathrm{T}}}{\Delta \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{H}_k \Delta \boldsymbol{g}_k} \tag{18}$$

1.2.3 Poprawka Broydena-Fletchera-Goldfarba-Shanno (BFGS)

$$\begin{aligned} \boldsymbol{H}_{k+1} = \boldsymbol{H}_k &- \frac{\boldsymbol{H}_k \Delta \boldsymbol{g}_k \left[\boldsymbol{H}_k \Delta \boldsymbol{g}_k\right]^{\mathrm{T}}}{\Delta \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{H}_k \Delta \boldsymbol{g}_k} \\ &- \frac{\Delta \boldsymbol{g}_k^{\mathrm{T}} \Delta \boldsymbol{x}_k}{\Delta \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{H}_k \Delta \boldsymbol{g}_k} \frac{\boldsymbol{H}_k \Delta \boldsymbol{g}_k \left[\boldsymbol{H}_k \Delta \boldsymbol{g}_k\right]^{\mathrm{T}}}{\Delta \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{H}_k \Delta \boldsymbol{g}_k} \\ &+ \frac{\boldsymbol{H}_k \Delta \boldsymbol{g}_k \Delta \boldsymbol{x}_k^{\mathrm{T}} + (\boldsymbol{H}_k \Delta \boldsymbol{g}_k \Delta \boldsymbol{x}_k^{\mathrm{T}})^{\mathrm{T}}}{\Delta \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{H}_k \Delta \boldsymbol{g}_k} \end{aligned} \tag{19}$$

Bezpośrednim rachunkiem można wykazać, że reguły SR1, DFP i BFGS zachowują warunek siecznej (14). Wprowadzając oznaczenia

$$\boldsymbol{\delta}_k = \boldsymbol{H}_k \Delta \boldsymbol{g}_k, \qquad \boldsymbol{\Delta}_k = \boldsymbol{H}_{k+1} - \boldsymbol{H}_k \tag{20}$$

możemy powyższe wzory napisać nieco zwięźlej, tzn. dla symetrycznej poprawki pierwszego rzędu (SR1) mamy

$$\boxed{\boldsymbol{\Delta}_k = \frac{(\Delta \boldsymbol{x}_k - \boldsymbol{\delta}_k)(\Delta \boldsymbol{x}_k - \boldsymbol{\delta}_k)^{\mathrm{T}}}{(\Delta \boldsymbol{x}_k - \boldsymbol{\delta}_k)^{\mathrm{T}} \Delta \boldsymbol{g}_k},} \tag{21}$$

dla poprawki Davidona-Fletchera-Powella (DFP)

$$\boxed{\boldsymbol{\Delta}_k = \frac{\Delta \boldsymbol{x}_k \Delta \boldsymbol{x}_k^{\mathrm{T}}}{\Delta \boldsymbol{g}_k^{\mathrm{T}} \Delta \boldsymbol{x}_k} - \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^{\mathrm{T}}}{\Delta \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{\delta}_k},} \tag{22}$$

dla poprawki Broydena-Fletchera-Goldfarba-Shanno (BFGS) mamy

$$\begin{aligned} \boldsymbol{H}_{k+1} = \boldsymbol{H}_k &- \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^{\mathrm{T}}}{\Delta \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{\delta}_k} - \frac{\Delta \boldsymbol{g}_k^{\mathrm{T}} \Delta \boldsymbol{x}_k}{\Delta \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{\delta}_k} \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^{\mathrm{T}}}{\Delta \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{\delta}_k} \\ &+ \frac{\boldsymbol{\delta}_k \Delta \boldsymbol{x}_k^{\mathrm{T}} + (\boldsymbol{\delta}_k \Delta \boldsymbol{x}_k^{\mathrm{T}})^{\mathrm{T}}}{\Delta \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{\delta}_k} \end{aligned} \tag{23}$$

czyli

$$\boxed{\boldsymbol{\Delta}_k = -\left(1 + \frac{\Delta \boldsymbol{g}_k^{\mathrm{T}} \Delta \boldsymbol{x}_k}{\Delta \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{\delta}_k}\right) \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^{\mathrm{T}}}{\Delta \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{\delta}_k} + \frac{\boldsymbol{\delta}_k \Delta \boldsymbol{x}_k^{\mathrm{T}} + (\boldsymbol{\delta}_k \Delta \boldsymbol{x}_k^{\mathrm{T}})^{\mathrm{T}}}{\Delta \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{\delta}_k}.} \tag{24}$$

1.3 Podsumowanie

Posumowując, w metodach quasi-newtonowskich

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + s_k \boldsymbol{H}_k \nabla f_0(\boldsymbol{x}_k) \tag{25}$$

gdzie długość kroku s_k wyznaczamy metodą *backtracking* natomiast macierz $\boldsymbol{H}_k$ ze wzoru na poprawkę SR1, DFP lub BFGS, przyjmując najczęściej $\boldsymbol{H}_0 = \boldsymbol{I}$.

Metoda quasi-newtonowska SK1/DFP/BFGS

dane: punkt startowy $\boldsymbol{x}_0$, $\epsilon > 0$, $\boldsymbol{H}_0$.

```
x ← x_0
while 1
    g ← ∇f_0(x)
    v ← −Hg
    ξ ← −g^T v
    if ξ < ε
        break
    else
    s ← 1
    while  f_0(x + sv) > f_0(x) + sαg^T v
        s ← βs
    end
        x ← x + sv
    end
    Δx ← sv
    Δg ← ∇f_0(x) − g
    Δδ ← HΔg
    Δ ← SK1/DFP/BFGS(Δx, Δg, δ)
    H ← H + Δ
end
x_optimal ← x
```

2 Zadania

Zadanie 1. Napisać skrypt w środowisku Matlab do szukania minimum funkcji metodą SR1, DFP i BFGS. Zakładamy, że znana z góry funkcja $f_0 : \mathbb{R}^2 \to \mathbb{R}$ jest postaci

$$f_0(\boldsymbol{x}) = \mathrm{e}^{x_1 + 3x_2 - 0.1} + \mathrm{e}^{-x_1 - 0.1} + (\boldsymbol{x} - \boldsymbol{x}_{\mathrm{c}})^{\mathrm{T}} \boldsymbol{P} (\boldsymbol{x} - \boldsymbol{x}_{\mathrm{c}}) \tag{26}$$

gdzie

$$\boldsymbol{P} = \frac{1}{8} \begin{bmatrix} 7 & \sqrt{3} \\ \sqrt{3} & 5 \end{bmatrix}, \qquad \boldsymbol{x}_{\mathrm{c}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{27}$$

Jako punkt startowy przyjąć

$$\boldsymbol{x}_0 = \begin{bmatrix} 2 & -2 \end{bmatrix}^{\mathrm{T}}, \tag{28}$$

dokładność rozwiązania $\epsilon =$ `1e-4`. W procedurze *backtracking line search* przyjąć $\alpha = 0.5$ oraz $\beta = 0.5$.

Wskazówka 1: Gradient funkcji (26) jest postaci

$$\nabla f_0(\boldsymbol{x}) = \begin{bmatrix} \mathrm{e}^{x_1 + 3x_2 - 0.1} - \mathrm{e}^{-x_1 - 0.1} \\ 3\mathrm{e}^{x_1 + 3x_2 - 0.1} \end{bmatrix} + 2\boldsymbol{P}(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{c}}). \tag{29}$$

Zadanie 2. Napisz skrypt w środowisku Matlab do szukania minimum funkcji metodą SR1, DFP i BFGS i użyj go do znalezienia minimum funkcji Rosenbrocka

$$f(\boldsymbol{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2. \tag{30}$$

Jako punkt startowy przyjmij $\boldsymbol{x}_0 = [0\ 0]^{\mathrm{T}}$. (Funkcja Rosenbrocka ma minimum w punkcie $\boldsymbol{x}^* = [1\ 1]^{\mathrm{T}}$).

3 Uwaga historyczna

Poniższy cytat pochodzi z [2].

> In the mid 1950s, W.C. Davidon, a physicist working at Argonne National Laboratory, was using the coordinate descent method [...] to perform a long optimization calculation. At that time computers were not very stable, and to Davidon's frustration, the computer system would always crash before the calculation was finished. So Davidon decided to find a way of accelerating the iteration. The algorithm he developed — the first quasi-Newton algorithm — turned out to be one of the most creative ideas in nonlinear optimization. It was soon demonstrated by Fletcher and Powell that the new algorithm was much faster and more reliable than the other existing methods, and this dramatic advance transformed nonlinear optimization overnight. During the following twenty years, numerous variants were proposed and hundreds of papers were devoted to their study. An interesting historical irony is that Davidon's paper [3] was not accepted for publication; it remained as a technical report for more than thirty years until it appeared in the first issue of the *SIAM Journal on Optimization* in 1991 [4].
>
> Quasi-Newton methods, like steepest descent, require only the gradient of the objective function to be supplied at each iterate. By measuring the changes in gradients, they construct a model of the objective function that is good enough to produce superlinear convergence. The improvement over steepest descent is dramatic, especially on difficult problems. Moreover, since second derivatives are not required, quasi-Newton methods are sometimes more efficient than Newton's method. Today, optimization software libraries contain a variety of quasi-Newton algorithms for solving unconstrained, constrained, and large-scale optimization problems.

Literatura

[1] E.K.P. Chong and S.H. Zak. *An Introduction to Optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimi. Wiley, 2004.

[2] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.

[3] William C. Davidon. *Variable metric method for minimization*. Technical ReportANL–5990 (revised), Argonne, IL, 1959.

[4] William C. Davidon. Variable metric method for minimization. *SIAM journal on optimization*, 1(1):1–17, 1991.