1DI2235:A - Metody optymalizacji, lab.
Metoda Newtona

(opracował: M.T., ostatnia modyfikacja: 21 marca 2023)

1 Preliminaria

Metoda Newtona jest jedną z najważniejszych metod optymalizacji, bez dużej przesady można ją nazwać matką wszystkich metod optymalizacji (Rys. 1). Szczegółowe omówienie tej metody można znaleźć np. w [1].

Rysunek 1: The GBU-43/B Massive Ordnance Air Blast (MOAB, commonly known as "Mother of All Bombs") is a large-yield bomb, developed for the United States military by Albert L. Weimorts, Jr. of the Air Force Research Laboratory. At the time of development, it was said to be the most powerful non-nuclear weapon in the American arsenal. The bomb is designed to be delivered by a C-130 Hercules, primarily the MC-130E Combat Talon I or MC-130H Combat Talon II variants. The MOAB was first dropped in combat in the 13 April 2017 airstrike against an Islamic State of Iraq and the Levant – Khorasan Province (ISIS) tunnel complex in Achin District, Afghanistan. [`https://en.wikipedia.org/wiki/GBU-43/B_MOAB`]

1.1 Rozwiązywanie układów równań nieliniowych metodą Newtona

Metoda Newtona polega na wyznaczeniu rozwiązania układu nieliniowych równań

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}, \tag{1}$$

gdzie

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^n, \quad \boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_n(\boldsymbol{x}) \end{bmatrix} \in \mathbb{R}^n, \tag{2}$$

przez rozwiązanie sekwencji układów równań liniowych

$$\boldsymbol{f}(\boldsymbol{x}_k) + \mathrm{D}\boldsymbol{f}(\boldsymbol{x}_k)(\boldsymbol{x} - \boldsymbol{x}_k) = \boldsymbol{0}, \quad k = 0, 1, \ldots \tag{3}$$

gdzie

$$\mathrm{D}\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \ldots & \frac{\partial f_1(\boldsymbol{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(\boldsymbol{x})}{\partial x_1} & \ldots & \frac{\partial f_n(\boldsymbol{x})}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{n \times n}. \tag{4}$$

Jeśli $\det[\mathrm{D}\boldsymbol{f}(\boldsymbol{x}_k)] \neq 0$ to rozwiązaniem układu równań (3) jest

$$\boldsymbol{x} = \boldsymbol{x}_k - [\mathrm{D}\boldsymbol{f}(\boldsymbol{x}_k)]^{-1} \boldsymbol{f}(\boldsymbol{x}_k), \quad k = 0, 1, \ldots \tag{5}$$

skąd otrzymuje się wzór rekurencyjny metody Newtona

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - [\mathrm{D}\boldsymbol{f}(\boldsymbol{x}_k)]^{-1} \boldsymbol{f}(\boldsymbol{x}_k), \quad k = 0, 1, \ldots \tag{6}$$

Dysponując punktem startowym $\boldsymbol{x}_0$ i korzystając ze wzoru (6) możemy wyznaczyć ciąg $\boldsymbol{x}_0$, $\boldsymbol{x}_1$, $\boldsymbol{x}_2, \ldots$, który pod pewnymi warunkami jest zbieżmy do rozwiązania układu (1).

Przykład 1. Weźmy pod uwagę zadanie wyznaczenia rzeczywistego pierwiastka kwadratowego pewnej dodatniej liczby rzeczywistej $\varrho > 0$. Pierwiastkiem kwadratowym liczby $\varrho > 0$, oznaczanym $\sqrt{\varrho}$ nazywamy liczbę $\xi > 0$ taką, że $\xi^2 = \varrho$. Łatwo zauważyć, że ξ jest dodatnim rozwiązaniem równania $f(x) = 0$ dla $f(x) = x^2 - \varrho$. Uwzględniając, że $f'(x) = 2x$ i korzystając ze wzoru (6) otrzymujemy

$$x_{k+1} = x_k - \frac{x_k^2 - \varrho}{2x_k}, \quad k = 0, 1, \ldots \tag{7}$$

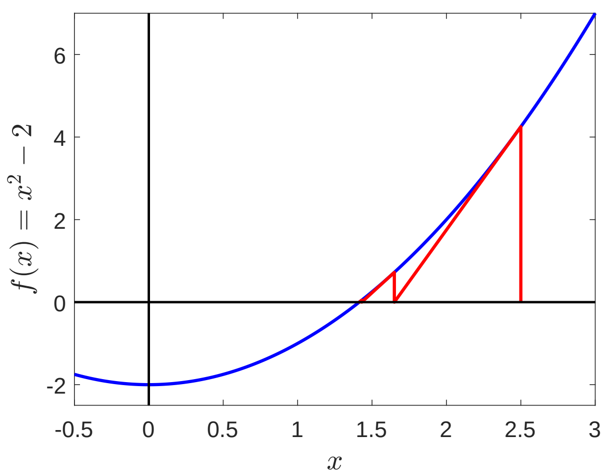
czyli

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{\varrho}{x_k}\right), \quad k = 0, 1, \ldots \tag{8}$$

Jako punkt początkowy x_0, w rozpatrywanym przypadku, możemy przyjąć dowolną liczbą dodatnią.
Rozpatrzmy, przykładowo, problem wyznaczenia $\sqrt{2}$, tzn. sytuację gdy $\varrho = 2$. Jako punkt startowy przyjmijmy $x_0 = 2.5$. Korzystając z (8) otrzymujemy

$$\begin{aligned} x_0 &= 2.500000000000000 \\ x_1 &= 1.650000000000000 \\ x_2 &= 1.431060606060606 \\ x_3 &= 1.414312727593564 \\ x_4 &= 1.414213565849604 \\ x_5 &= 1.414213562373095 \end{aligned}$$

Wartość $\sqrt{2}$ z dokładnością do 15 miejsc po przecinku wynosi 1.414213562373095, jak można zauważyć, dla przyjętego punktu startowego otrzymuje się ją już w piątej iteracji. Interpretację geometryczną pierwszych dwóch iteracji przedstawia Rys. 2.



Rysunek 2: Interpretacja geometryczna pierwszych dwóch iteracji metody Newtona przy wyznaczaniu $\sqrt{2}$. Punkt startowy $x_0 = 2.5$.

1.2 Metoda Newtona w optymalizacji

Często spotykanym w optymalizacji problemem jest wyznaczenie rozwiązania układu równań nieliniowych

$$\nabla f_0(\boldsymbol{x}) = \boldsymbol{0}. \tag{9}$$

Podstawiając $f(\boldsymbol{x}) \equiv \nabla f_0(\boldsymbol{x})$, oraz uwzględniając, że

$$\begin{aligned}
\mathrm{D}\left\{\nabla f_0(\boldsymbol{x})\right\} &= \mathrm{D}\begin{bmatrix} \frac{\partial f_0}{\partial x_1} \\ \vdots \\ \frac{\partial f_0}{\partial x_n} \end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial}{\partial x_1}\left(\frac{\partial f_0}{\partial x_1}\right) & \cdots & \frac{\partial}{\partial x_n}\left(\frac{\partial f_0}{\partial x_1}\right) \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x_1}\left(\frac{\partial f_0}{\partial x_n}\right) & \cdots & \frac{\partial}{\partial x_n}\left(\frac{\partial f_0}{\partial x_n}\right) \end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial^2 f_0}{\partial x_1^2} & \cdots & \frac{\partial^2 f_0}{\partial x_n \partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f_0}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f_0}{\partial x_n^2} \end{bmatrix}
\end{aligned}$$

tzn.

$$\mathrm{D}\left\{\nabla f_0(\boldsymbol{x})\right\} = \nabla^2 f_0(\boldsymbol{x}), \tag{10}$$

otrzymujemy z zależności (6) rekurencyjny wzór metody Newtona.

$$\boxed{\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \left[\nabla^2 f_0(\boldsymbol{x}_k)\right]^{-1} \nabla f_0(\boldsymbol{x}_k).} \tag{11}$$

Metoda Newtona (wersja A)

dane: punkt startowy $\boldsymbol{x}_0$, liczba iteracji N

$\boldsymbol{x} \leftarrow \boldsymbol{x}_0$
for $k = 1 : N$
  $\boldsymbol{x} \leftarrow \boldsymbol{x} - \left[\nabla^2 f_0(\boldsymbol{x})\right]^{-1} \nabla f_0(\boldsymbol{x})$
end
$\boldsymbol{x}_{\text{optimal}} \leftarrow \boldsymbol{x}$

Metoda Newtona (wersja B)

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

$\boldsymbol{x} \leftarrow \boldsymbol{x}_0$
$\boldsymbol{g} \leftarrow \nabla f_0(\boldsymbol{x})$
$\boldsymbol{v} \leftarrow -\left[\nabla^2 f_0(\boldsymbol{x})\right]^{-1} \boldsymbol{g}$
$\Delta \leftarrow -\boldsymbol{g}^{\mathrm{T}}\boldsymbol{v}$
while $\Delta > \epsilon$
  $\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{v}$
  $\Delta \leftarrow -\boldsymbol{g}^{\mathrm{T}}\boldsymbol{v}$
  $\boldsymbol{g} \leftarrow \nabla f_0(\boldsymbol{x})$
  $\boldsymbol{v} \leftarrow -\left[\nabla^2 f_0(\boldsymbol{x})\right]^{-1} \boldsymbol{g}$
end
$\boldsymbol{x}_{\text{optimal}} \leftarrow \boldsymbol{x}$

Wielkość $\sqrt{\Delta}$, gdzie $\Delta \equiv \left[\nabla f_0(\boldsymbol{x})\right]^{\mathrm{T}} \left[\nabla^2 f_0(\boldsymbol{x})\right]^{-1} \nabla f_0(\boldsymbol{x})$, nazywamy dekrementem Newtona (ang. *Newton's decrement*). Algorytm Newtona kończy działanie, kiedy kwadrat dekrementu spada poniżej pewnej ustalonej wartości ϵ.

Metoda Newtona (wersja C)

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

$\boldsymbol{x} \leftarrow \boldsymbol{x}_0$
while 1
  $\boldsymbol{g} \leftarrow \nabla f_0(\boldsymbol{x})$
  $\boldsymbol{v} \leftarrow -\left[\nabla^2 f_0(\boldsymbol{x})\right]^{-1} \boldsymbol{g}$
  $\Delta \leftarrow -\boldsymbol{g}^{\mathrm{T}}\boldsymbol{v}$
  if $\Delta < \epsilon$
    break
  else
    $\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{v}$
  end
end
$\boldsymbol{x}_{\text{optimal}} \leftarrow \boldsymbol{x}$

W praktyce zamiast klasycznego wzoru Newtona (11) używa się tzw. metody Newtona z tłumieniem, w której

$$\boxed{\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - s_k \left[\nabla^2 f_0(\boldsymbol{x}_k)\right]^{-1} \nabla f_0(\boldsymbol{x}_k),} \tag{12}$$

gdzie parametr $s_k > 0$ jest tzw. czynnikiem skalującym (czasami zwanym też, niezbyt ścieśle, długością kroku), którego wartość najczęściej wyznacza się tzw. metodą *backtracking line search*, tzn. dla przyjętych wartości parametrów $\alpha \in (0,\, 1)$, $\beta \in (0,\, 1)$ oraz kierunku poszukiwań

$$\boldsymbol{v}_k = -\left[\nabla^2 f_0(\boldsymbol{x}_k)\right]^{-1} \nabla f_0(\boldsymbol{x}_k) \tag{13}$$

iterujemy

$s \leftarrow 1$
while $f_0(\boldsymbol{x}_k + s\boldsymbol{v}_k) > f_0(\boldsymbol{x}_k) + s\alpha \nabla f_0(\boldsymbol{x}_k)^{\mathrm{T}}\boldsymbol{v}_k$
  $s \leftarrow \beta s$
end
$s_k \leftarrow s$

Metoda Newtona z tłumieniem (wersja A)

dane: punkt startowy $\boldsymbol{x}_0$, liczba iteracji N

$\boldsymbol{x} \leftarrow \boldsymbol{x}_0$
for $k = 1 : N$
  $\boldsymbol{g} \leftarrow \nabla f_0(\boldsymbol{x})$
  $\boldsymbol{v} \leftarrow -\left[\nabla^2 f_0(\boldsymbol{x})\right]^{-1} \boldsymbol{g}$
  $s \leftarrow 1$
  while $f_0(\boldsymbol{x} + s\boldsymbol{v}) > f_0(\boldsymbol{x}) + s\alpha\boldsymbol{g}^{\mathrm{T}}\boldsymbol{v}$
    $s \leftarrow \beta s$
  end
  $\boldsymbol{x} \leftarrow \boldsymbol{x} + s\boldsymbol{v}$
end
$\boldsymbol{x}_{\text{optimal}} \leftarrow \boldsymbol{x}$

Metoda Newtona z tłumieniem (wersja B)

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

$\boldsymbol{x} \leftarrow \boldsymbol{x}_0$
$\boldsymbol{g} \leftarrow \nabla f_0(\boldsymbol{x})$
$\boldsymbol{v} \leftarrow -\left[\nabla^2 f_0(\boldsymbol{x})\right]^{-1} \boldsymbol{g}$
$\Delta \leftarrow -\boldsymbol{g}^{\mathrm{T}}\boldsymbol{v}$
while $\Delta > \epsilon$
  $s \leftarrow 1$
  while $f_0(\boldsymbol{x} + s\boldsymbol{v}) > f_0(\boldsymbol{x}) + s\alpha \boldsymbol{g}^{\mathrm{T}}\boldsymbol{v}$
    $s \leftarrow \beta s$
  end
  $\boldsymbol{x} \leftarrow \boldsymbol{x} + s\boldsymbol{v}$
  $\Delta \leftarrow -\boldsymbol{g}^{\mathrm{T}}\boldsymbol{v}$
  $\boldsymbol{g} \leftarrow \nabla f_0(\boldsymbol{x})$
  $\boldsymbol{v} \leftarrow -\left[\nabla^2 f_0(\boldsymbol{x})\right]^{-1} \boldsymbol{g}$
end
$\boldsymbol{x}_{\mathrm{optimal}} \leftarrow \boldsymbol{x}$

Metoda Newtona z tłumieniem (wersja C)

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

$\boldsymbol{x} \leftarrow \boldsymbol{x}_0$
while 1
  $\boldsymbol{g} \leftarrow \nabla f_0(\boldsymbol{x})$
  $\boldsymbol{v} \leftarrow -\left[\nabla^2 f_0(\boldsymbol{x})\right]^{-1} \boldsymbol{g}$
  $\Delta \leftarrow -\boldsymbol{g}^{\mathrm{T}}\boldsymbol{v}$
  if $\Delta < \epsilon$
    break
  else
  $s \leftarrow 1$
  while $f_0(\boldsymbol{x} + s\boldsymbol{v}) > f_0(\boldsymbol{x}) + s\alpha \boldsymbol{g}^{\mathrm{T}}\boldsymbol{v}$
    $s \leftarrow \beta s$
  end
    $\boldsymbol{x} \leftarrow \boldsymbol{x} + s\boldsymbol{v}$
  end
end
$\boldsymbol{x}_{\mathrm{optimal}} \leftarrow \boldsymbol{x}$

2 Zadania

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

$$f_0(\boldsymbol{x}) = \mathrm{e}^{x_1+3x_2-0.1} + \mathrm{e}^{-x_1-0.1} + (\boldsymbol{x} - \boldsymbol{x}_{\mathrm{c}})^{\mathrm{T}} \boldsymbol{P} (\boldsymbol{x} - \boldsymbol{x}_{\mathrm{c}}) \tag{14}$$

gdzie

$$\boldsymbol{P} = \frac{1}{8}\begin{bmatrix} 7 & \sqrt{3} \\ \sqrt{3} & 5 \end{bmatrix}, \qquad \boldsymbol{x}_{\mathrm{c}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{15}$$

Jako punkt startowy przyjąć

$$\boldsymbol{x}_0 = \begin{bmatrix} 2 \\ -2 \end{bmatrix}, \tag{16}$$

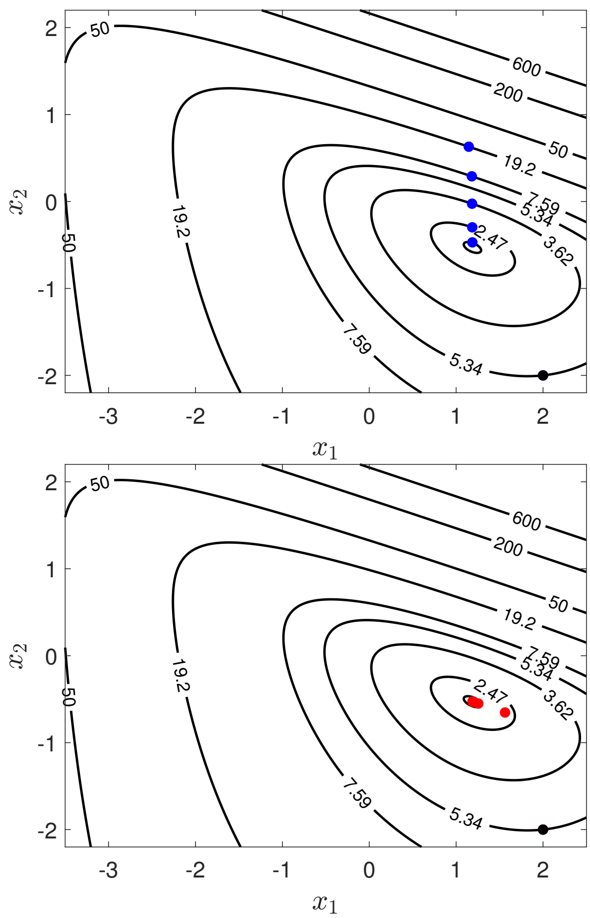
dokładność rozwiązania $\epsilon =$ `1e-4`. W procedurze *backtracking line search* przyjąć $\alpha = 0.5$ oraz $\beta = 0.5$. Wygenerować wykres poziomic i kolejnych punktów iteracyjnych tak jak na Rys. 3, skorzystać w tym celu z funkcji `meshgrid`, `arrayfun` oraz `contour` środowiska Matlab. Dla porównania końcowego wyniku rozwiązać to samo zadanie korzystając z (a) pakietu `CVX` (b) funkcji `fminsearch` środowiska Matlab.

Wskazówka 1: Gradient funkcji (14) jest postaci

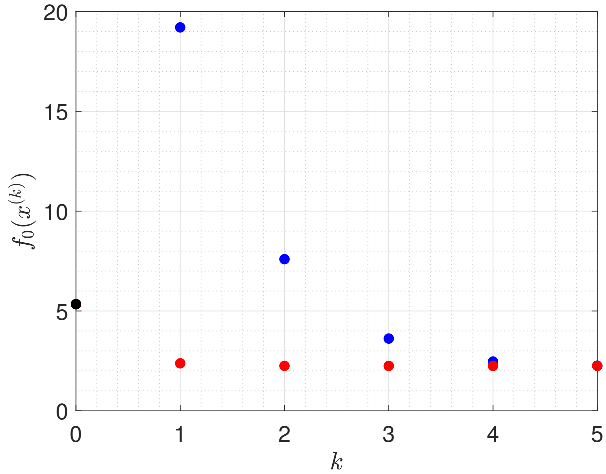
$$\nabla f_0(\boldsymbol{x}) = \begin{bmatrix} \mathrm{e}^{x_1+3x_2-0.1} - \mathrm{e}^{-x_1-0.1} \\ 3\mathrm{e}^{x_1+3x_2-0.1} \end{bmatrix} + 2\boldsymbol{P}(\boldsymbol{x} - \boldsymbol{x}_{\mathrm{c}}). \tag{17}$$

Hesjan funkcji (14) jest postaci

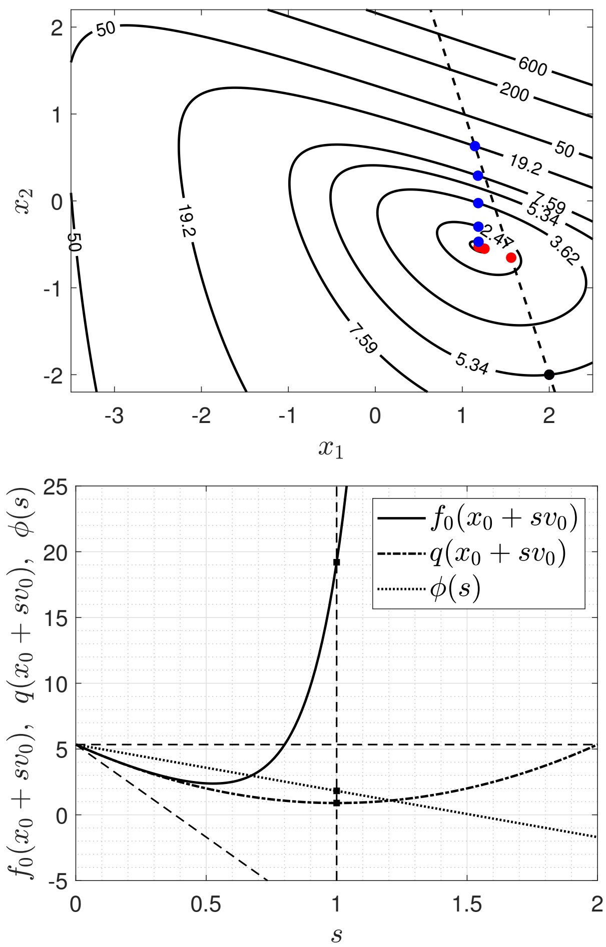
$$\nabla^2 f_0(\boldsymbol{x}) = \begin{bmatrix} \mathrm{e}^{x_1+3x_2-0.1} + \mathrm{e}^{-x_1-0.1} & 3\mathrm{e}^{x_1+3x_2-0.1} \\ 3\mathrm{e}^{x_1+3x_2-0.1} & 9\mathrm{e}^{x_1+3x_2-0.1} \end{bmatrix} + 2\boldsymbol{P}. \tag{18}$$



Rysunek 3: Poziomice funkcji celu $f_0(\boldsymbol{x}) = \mathrm{e}^{x_1+3x_2-0.1} + \mathrm{e}^{-x_1-0.1} + (\boldsymbol{x} - \boldsymbol{x}_{\mathrm{c}})^{\mathrm{T}} \boldsymbol{P} (\boldsymbol{x} - \boldsymbol{x}_{\mathrm{c}})$ wraz z kolejnymi iteracjami dla punktu startowego $\boldsymbol{x}_0 = [2,\ -2]^{\mathrm{T}}$. Górny panel – klasyczna metoda Newtona. Dolny panel – metoda Newtona z tłumieniem.



Rysunek 4: Wartości funkcji celu $f_0(\boldsymbol{x}_k)$ dla $k = 0, 1, \ldots$. Wartość $k = 0$ odpowiada punktowi startowemu. Kolor niebieski – metoda Newtona, kolor czerwony – metoda Newtona z tłumieniem.



Rysunek 5: Schemat metody *backtracking line search* w pierwszej iteracji dla $\alpha = 0.5$, $\beta = 0.5$. Panel górny – linia przekroju. Panel dolny – płaszczyzna przekroju. Funkcja q jest przybliżeniem kwadratowym funkcji celu f_0 w punkcie $\boldsymbol{x}_0$, tzn. $q(\boldsymbol{x}_0 + s\boldsymbol{v}_0) = f_0(\boldsymbol{x}_0) + s\left[\nabla f_0(\boldsymbol{x}_0)\right]^{\mathrm{T}} \boldsymbol{v}_0 + \frac{s^2}{2}\boldsymbol{v}_0^{\mathrm{T}}\left[\nabla^2 f_0(\boldsymbol{x}_0)\right]\boldsymbol{v}_0$. Funkcja ϕ jest określona wzorem $\phi(s) = f_0(\boldsymbol{x}_0) + \alpha s\left[\nabla f_0(\boldsymbol{x}_0)\right]^{\mathrm{T}} \boldsymbol{v}_0$.

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

$$f_0(\boldsymbol{x}) = t\left[\mathrm{e}^{x_1+3x_2-0.1} + \mathrm{e}^{-x_1-0.1}\right] - \log\left[1 - (\boldsymbol{x}-\boldsymbol{x}_{\mathrm{c}})^{\mathrm{T}}\boldsymbol{P}(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{c}})\right] \qquad (19)$$

gdzie $\boldsymbol{P}$ i $\boldsymbol{x}_{\mathrm{c}}$ są dane wzorem (15). Jako punkt startowy przyjąć $\boldsymbol{x}_0 = \boldsymbol{x}_{\mathrm{c}}$, dokładność rozwiązania $\epsilon =$ `1e-4`. Wartość parametru t przyjąć, kolejno, $t = 0.1$, $t = 1.0$ i $t = 10.0$. W procedurze *backtracking line search* przyjąć $\alpha = 0.3$ oraz $\beta = 0.8$. Dla porównania końcowego wyniku rozwiązać to samo zadanie korzystając z (a) pakietu `CVX` (b) funkcji `fminsearch` środowiska Matlab.

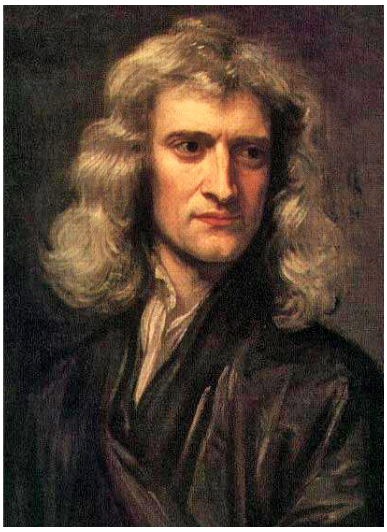
Wskazówka 1: Należy pamiętać, że rzeczywista funkcja `log` nie jest określona dla argumentów ujemnych. W środowisku Matlab, użycie funkcji `log` z ujemnym argumentem spowoduje potraktowanie jej jako funkcji zespolonej która dla ujemnych argumentów przyjmuje wartości zespolone, co w rozpatrywanym kontekście optymalizacji funkcji celu (19) prowadzi do wyników pozbawionych sensu. Dlatego kodując należy zmodyfikować funkcję `log` w taki sposób, aby dla ujemnych argumentów był zwracany obiekt inf.

Wskazówka 2: Gradient funkcji (14) jest postaci

$$\nabla f_0(\boldsymbol{x}) = t\begin{bmatrix} \mathrm{e}^{x_1+3x_2-0.1} - \mathrm{e}^{-x_1-0.1} \\ 3\mathrm{e}^{x_1+3x_2-0.1} \end{bmatrix} + \frac{2\boldsymbol{P}(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{c}})}{1 - (\boldsymbol{x}-\boldsymbol{x}_{\mathrm{c}})^{\mathrm{T}}\boldsymbol{P}(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{c}})}. \qquad (20)$$

Hesjan funkcji (14) jest postaci

$$\nabla^2 f_0(\boldsymbol{x}) = t\begin{bmatrix} \mathrm{e}^{x_1+3x_2-0.1} + \mathrm{e}^{-x_1-0.1} & 3\mathrm{e}^{x_1+3x_2-0.1} \\ 3\mathrm{e}^{x_1+3x_2-0.1} & 9\mathrm{e}^{x_1+3x_2-0.1} \end{bmatrix} + \frac{4\boldsymbol{P}(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{c}})(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{c}})^{\mathrm{T}}\boldsymbol{P}}{\left[1 - (\boldsymbol{x}-\boldsymbol{x}_{\mathrm{c}})^{\mathrm{T}}\boldsymbol{P}(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{c}})\right]^2} + \frac{2\boldsymbol{P}}{1 - (\boldsymbol{x}-\boldsymbol{x}_{\mathrm{c}})^{\mathrm{T}}\boldsymbol{P}(\boldsymbol{x}-\boldsymbol{x}_{\mathrm{c}})}. \qquad (21)$$



Rysunek 6: Sir Isaac Newton (1642 – 1726) was an English mathematician, physicist, astronomer, theologian, and author (...) who is widely recognised as one of the most influential scientists of all time, and a key figure in the scientific revolution. His book Philosophia Naturalis Principia Mathematica (*Mathematical Principles of Natural Philosophy*), first published in 1687, laid the foundations of classical mechanics. [`https://en.wikipedia.org/wiki/Isaac_Newton`]

Literatura

[1] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. `http://web.stanford.edu/~boyd/cvxbook/`.