Optymalizacja ciągła

1. Optymalizacja funkcji jednej zmiennej

Wojciech Kotłowski

Instytut Informatyki PP
http://www.cs.put.poznan.pl/wkotlowski/

28.02.2019

Plan wykładu

Optymalizacja funkcji jednej zmiennej ($x \in \mathbb{R}$):

$$\begin{aligned} \text{min} \quad & f(x) \\ \text{p.o.} \quad & x \in \mathcal{X} \subseteq \mathbb{R} \end{aligned}$$

1. Rozwiązanie analityczne
2. Przeszukiwanie jednostajne
3. Optymalizacja funkcji jednomodalnych: przeszukiwanie dychotomiczna i metoda złotego podziału
4. Optymalizacja funkcji różniczkowalnych: metoda bisekcji
5. Optymalizacja funkcji różniczkowalnych: metoda Newtona

Rozwiązanie analityczne

Rozwiązanie analityczne

W pewnych przypadkach można znaleźć minimum funkcji bez odwoływania się do metod numerycznych

Założenia:

- Funkcja f zdefiniowana jest na dziedzinie $\mathcal{X} = [x_{\min}, x_{\max}]$
- Funkcja f jest **różniczkowalna** na $\mathcal{X}$

Warunek konieczny na istnienie minimum

Jeśli różniczkowalna funkcja f osiąga **minimum lokalne** w punkcie x_0 wewnątrz dziedziny, to $f'(x_0) = 0$.

Warunek konieczny na istnienie minimum

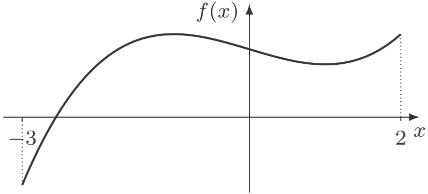
Jeśli różniczkowalna funkcja f osiąga **minimum lokalne** w punkcie x_0 wewnątrz dziedziny, to $f'(x_0) = 0$.

Wniosek: Algorytm znajdywania minimum funkcji f na $\mathcal{X} = [x_{\min}, x_{\max}]$:

1. Wyznacz punkty zerowanie się pochodnej $\mathcal{X}_0 = \{x \in \mathcal{X} \colon f'(x) = 0\}$
2. Sprawdź wartość funkcji dla wszystkich punktów z $\mathcal{X}_0$ oraz na krańcach dziedziny $\{x_{\min}, x_{\max}\}$ i wybierz ten o najmniejszej wartości funkcji.

Przykład

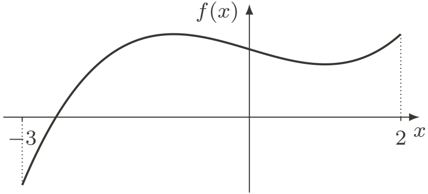
$$\begin{array}{ll} \text{min} & f(x) = x^3 - 3x + 9 \\ \text{p.o.} & x \in [-3, 2] \end{array}$$



Przykład

min $\quad f(x) = x^3 - 3x + 9$

p.o. $\quad x \in [-3, 2]$



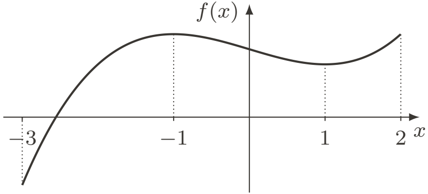
1. Obliczamy pochodną:

$$f'(x) \;=\; 3x^2 - 3$$

Przykład

min $\quad f(x) = x^3 - 3x + 9$

p.o. $\quad x \in [-3, 2]$



1. Obliczamy pochodną:

$$f'(x) \;=\; 3x^2 - 3$$

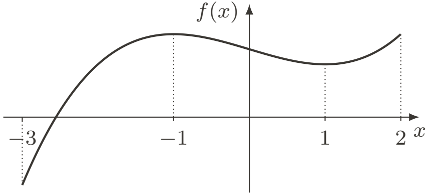
2. Wyznaczamy punkty zerowania się pochodnej:

$$f'(x) = 0 \quad \iff \quad 3x^2 - 3 = 0 \quad \iff \quad x = 1 \text{ lub } x = -1$$

Przykład

min $\quad f(x) = x^3 - 3x + 9$

p.o. $\quad x \in [-3, 2]$



1. Obliczamy pochodną:

$$f'(x) \ = \ 3x^2 - 3$$

2. Wyznaczamy punkty zerowania się pochodnej:

$$f'(x) = 0 \quad \Longleftrightarrow \quad 3x^2 - 3 = 0 \quad \Longleftrightarrow \quad x = 1 \text{ lub } x = -1$$

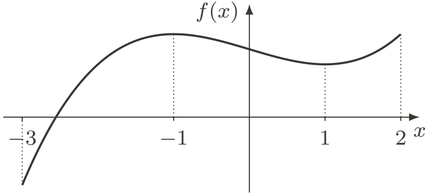
3. Obliczamy wartość funkcji w kandydatach na minimum globalne:

$$f(-3) = -9, \quad f(-1) = 11, \quad f(1) = 7, \quad f(2) = 11$$

Przykład

min $\quad f(x) = x^3 - 3x + 9$

p.o. $\quad x \in [-3, 2]$



1. Obliczamy pochodną:

$$f'(x) \; = \; 3x^2 - 3$$

2. Wyznaczamy punkty zerowania się pochodnej:

$$f'(x) = 0 \quad \Longleftrightarrow \quad 3x^2 - 3 = 0 \quad \Longleftrightarrow \quad x = 1 \text{ lub } x = -1$$

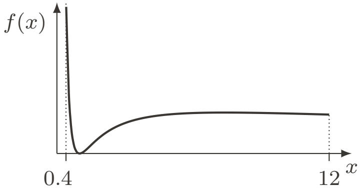
3. Obliczamy wartość funkcji w kandydatach na minimum globalne:

$$f(-3) = -9, \quad f(-1) = 11, \quad f(1) = 7, \quad f(2) = 11$$

4. **Wniosek**: minimum globalne w $x^* = -3$

Przykład

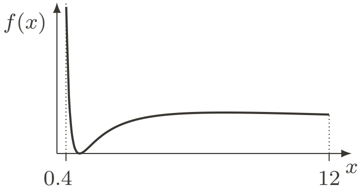
$$\text{min} \quad f(x) = \frac{\ln^2(x)}{x}$$
$$\text{p.o.} \quad x \in [0.4, 12]$$



Przykład

min $\quad f(x) = \dfrac{\ln^2(x)}{x}$

p.o. $\quad x \in [0.4, 12]$

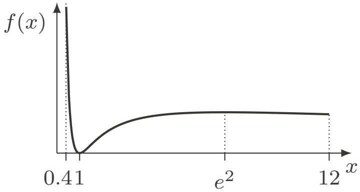


1. Obliczamy pochodną:

$$f'(x) \;=\; \frac{2\ln(x) - \ln^2(x)}{x^2} \;=\; \frac{\ln(x)\,(2 - \ln(x))}{x^2}$$

Przykład

$$\begin{aligned} \text{min} \quad & f(x) = \frac{\ln^2(x)}{x} \\ \text{p.o.} \quad & x \in [0.4, 12] \end{aligned}$$



1. Obliczamy pochodną:

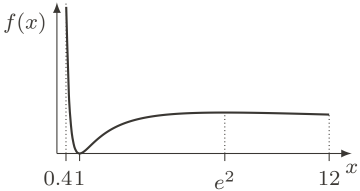
$$f'(x) \;=\; \frac{2\ln(x) - \ln^2(x)}{x^2} \;=\; \frac{\ln(x)\,(2 - \ln(x))}{x^2}$$

2. Wyznaczamy punkty zerowania się pochodnej:

$$f'(x) = 0 \quad \Longleftrightarrow \quad \ln(x)\,(2 - \ln(x)) = 0 \quad \Longleftrightarrow \quad x = 1 \text{ lub } x = e^2$$

Przykład

$$\min \quad f(x) = \frac{\ln^2(x)}{x}$$
$$\text{p.o.} \quad x \in [0.4, 12]$$



1. Obliczamy pochodną:

$$f'(x) = \frac{2\ln(x) - \ln^2(x)}{x^2} = \frac{\ln(x)\,(2 - \ln(x))}{x^2}$$

2. Wyznaczamy punkty zerowania się pochodnej:

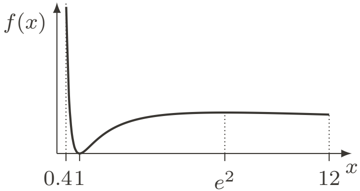
$$f'(x) = 0 \iff \ln(x)\,(2 - \ln(x)) = 0 \iff x = 1 \text{ lub } x = e^2$$

3. Obliczamy wartość funkcji w kandydatach na minimum globalne:

$$f(1) = 0, \quad f(e^2) = 0.541, \quad f(0.4) = 2.1, \quad f(12) = 0.515$$

Przykład

$$\begin{array}{ll} \text{min} & f(x) = \dfrac{\ln^2(x)}{x} \\ \text{p.o.} & x \in [0.4, 12] \end{array}$$



1. Obliczamy pochodną:

$$f'(x) \;=\; \frac{2\ln(x) - \ln^2(x)}{x^2} \;=\; \frac{\ln(x)\,(2 - \ln(x))}{x^2}$$

2. Wyznaczamy punkty zerowania się pochodnej:

$$f'(x) = 0 \quad \iff \quad \ln(x)\,(2 - \ln(x)) = 0 \quad \iff \quad x = 1 \text{ lub } x = e^2$$

3. Obliczamy wartość funkcji w kandydatach na minimum globalne:

$$f(1) = 0, \quad f(e^2) = 0.541, \quad f(0.4) = 2.1, \quad f(12) = 0.515$$

4. **Wniosek**: minimum globalne w $x^* = 1$

Czy to wystarcza?

Czy to wystarcza?

Oczywiście **nie wystarcza**:

- Rzadko kiedy potrafimy analitycznie wyznaczyć punkty zerowania się pochodnej, tzn. rozwiązać równanie $f'(x) = 0$
- Funkcja może nie być różniczkowalna

Czy to wystarcza?

Oczywiście **nie wystarcza**:

- Rzadko kiedy potrafimy analitycznie wyznaczyć punkty zerowania się pochodnej, tzn. rozwiązać równanie $f'(x) = 0$
- Funkcja może nie być różniczkowalna

Musimy odwołać się do **numerycznych metod optymalizacji**

Przeszukiwanie jednostajne

Opis problemu

$$\begin{aligned} \text{min} \quad & f(x) \\ \text{p.o.} \quad & x_{\min} \leqslant x \leqslant x_{\max} \end{aligned}$$

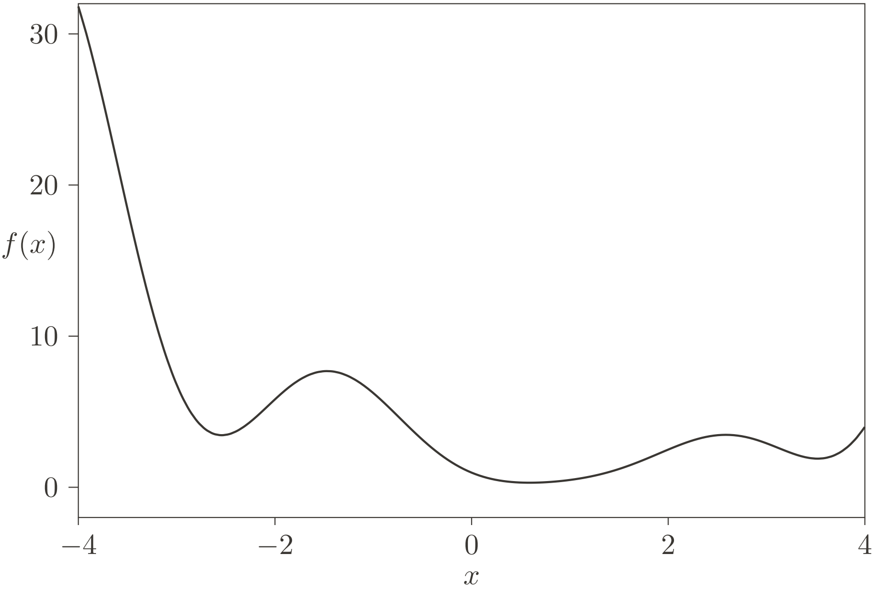
- Nie zakładamy nic o różniczkowalności
- Kształt funkcji zupełnie nieznany

Metoda przeszukiwania jednostajnego

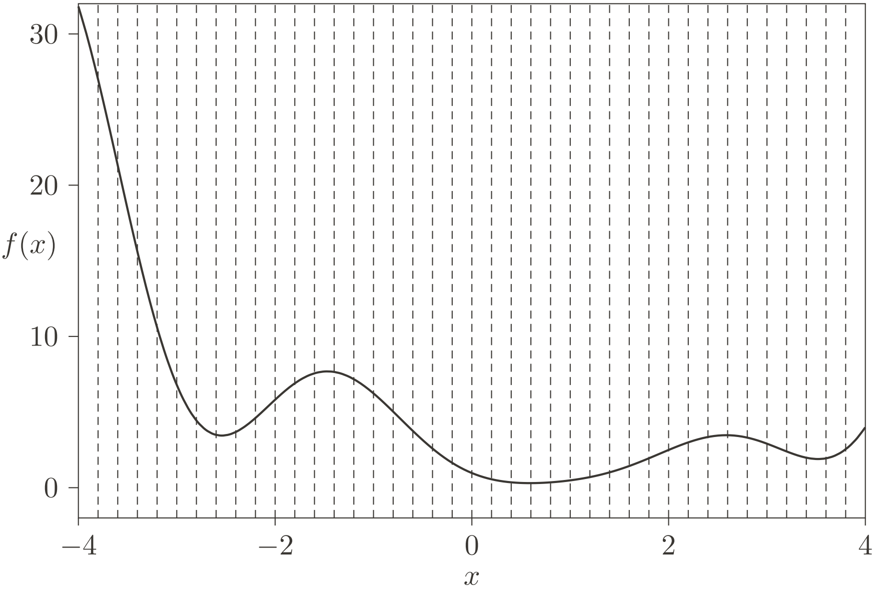
Podziel dziedzinę funkcji $\mathcal{X} = [x_{\min}, x_{\max}]$ na bardzo wiele krótkich odcinków i sprawdź wartość funkcji w każdym z odcinków (np. w punkcie środkowym odcinka)

Zwracamy punkt $\widehat{x}$ o najmniejszej wartości funkcji

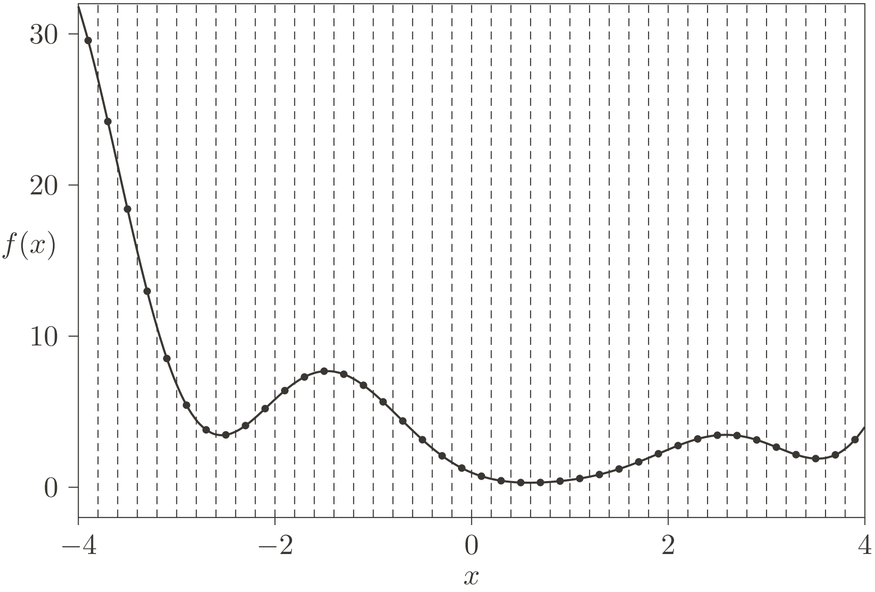
Przeszukiwanie jednostajne – przykład



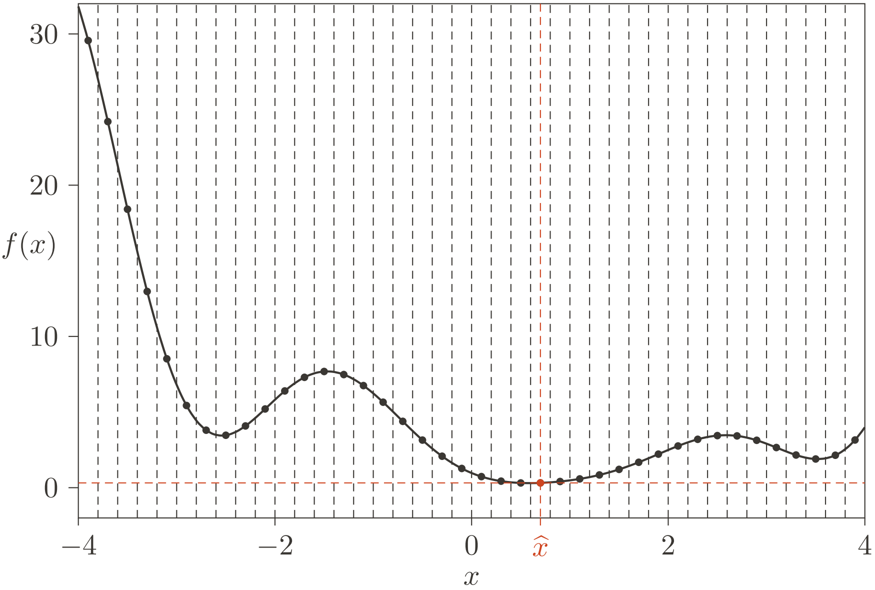
Przeszukiwanie jednostajne – przykład



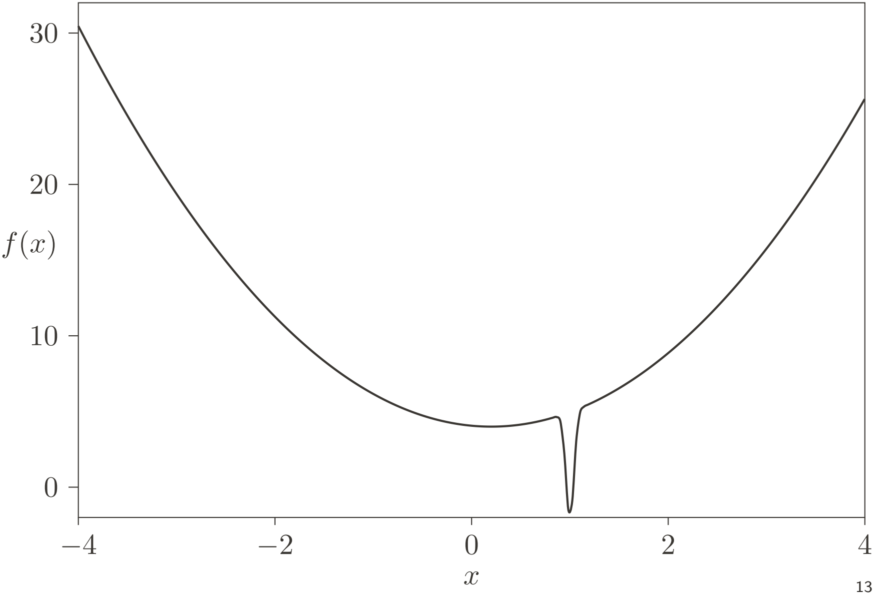
Przeszukiwanie jednostajne – przykład



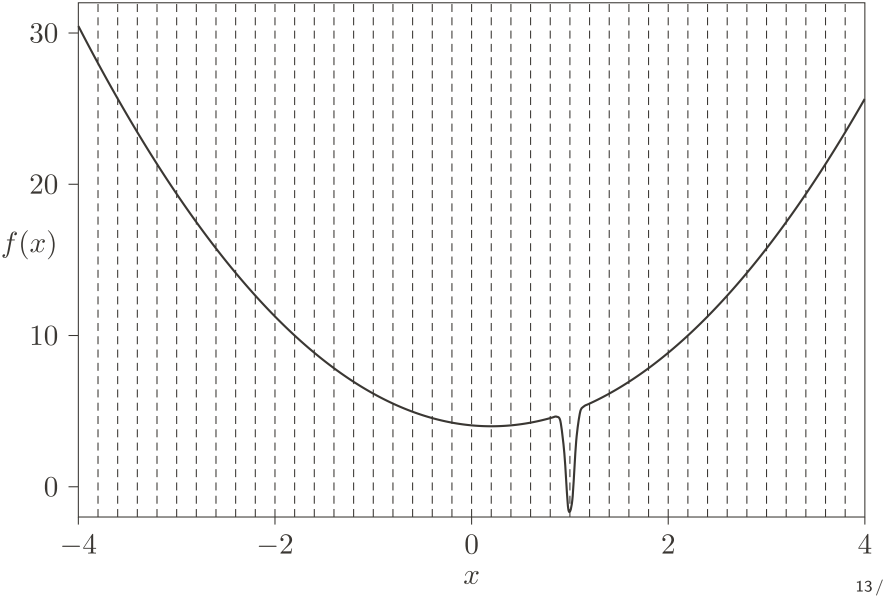
Przeszukiwanie jednostajne – przykład



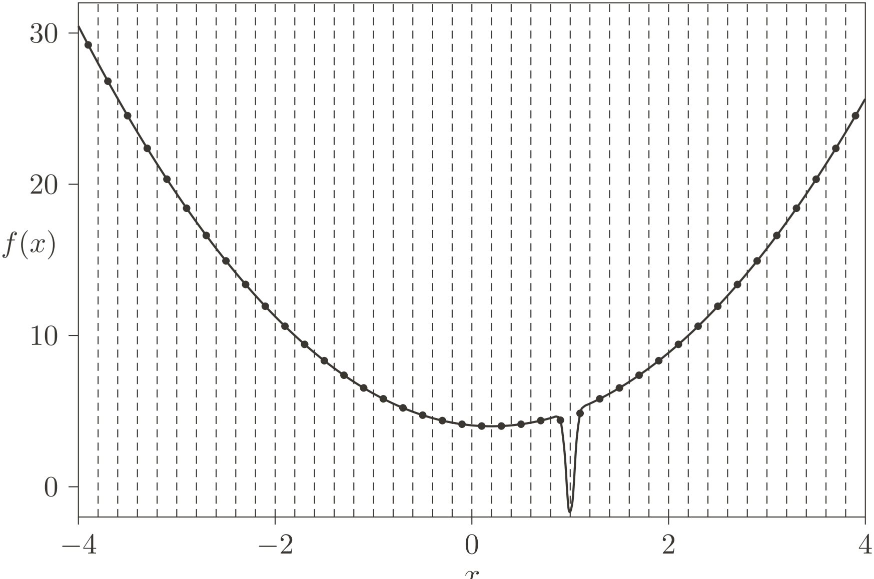
Czy przeszukiwanie jednostajne gwarantuje znalezienie minimum?



Czy przeszukiwanie jednostajne gwarantuje znalezienie minimum?



Czy przeszukiwanie jednostajne gwarantuje znalezienie minimum?



Czy przeszukiwanie jednostajne gwarantuje znalezienie minimum?

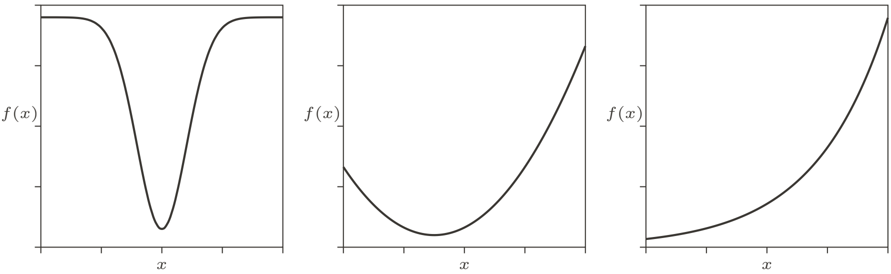
Przeszukiwanie jednostajne

- Metoda o złożoności **liniowej** z liczbą odcinków (bardzo wolno!)
- Nie gwarantuje znalezienia minimum globalnego, nawet w przybliżeniu (choć dla większości funkcji w praktyce działa dobrze)
- Jakość rozwiązania silnie zależy od regularności funkcji
- Alternatywa: **przeszukiwanie losowe** (losuj punkty z dziedziny i sprawdzaj wartość funkcji), posiada podobne wady
- Bez dodatkowych założeń niewiele więcej można zrobić

Optymalizacja funkcji jednomodalnych (bez informacji o pochodnej)

Funkcja jednomodalna

Funkcja jednomodalna to funkcja posiadająca tylko jedno (globalne) minimum (być może na granicy dziedziny)



Własność funkcji jednomodalnej

Niech funkcja $f(x)$ posiada jedno (ścisłe) minimum na przedziale $[a, b]$. Wybierzmy dowolne $x_\ell, x_p \in [a, b]$ takie, że $x_\ell < x_p$.

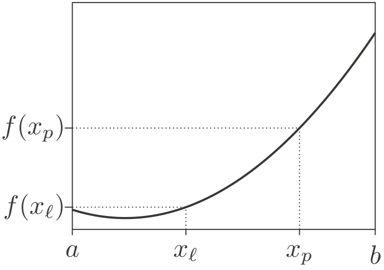
- Jeśli $f(x_\ell) \geqslant f(x_p)$ to minimum na pewno nie ma w $[a, x_\ell)$
- Jeśli $f(x_\ell) \leqslant f(x_p)$ to minimum na pewno nie ma w $(x_p, b]$

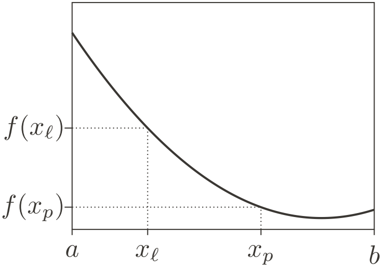
Własność funkcji jednomodalnej

Niech funkcja $f(x)$ posiada jedno (ścisłe) minimum na przedziale $[a, b]$. Wybierzmy dowolne $x_\ell, x_p \in [a, b]$ takie, że $x_\ell < x_p$.

- Jeśli $f(x_\ell) \geqslant f(x_p)$ to minimum na pewno nie ma w $[a, x_\ell)$
- Jeśli $f(x_\ell) \leqslant f(x_p)$ to minimum na pewno nie ma w $(x_p, b]$

Dowód: obrazkowy, nie wprost



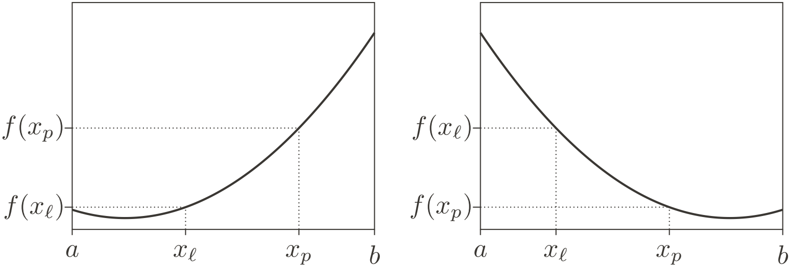


Własność funkcji jednomodalnej

Niech funkcja $f(x)$ posiada jedno (ścisłe) minimum na przedziale $[a, b]$.
Wybierzmy dowolne $x_\ell, x_p \in [a, b]$ takie, że $x_\ell < x_p$.

- Jeśli $f(x_\ell) \geqslant f(x_p)$ to minimum na pewno nie ma w $[a, x_\ell)$
- Jeśli $f(x_\ell) \leqslant f(x_p)$ to minimum na pewno nie ma w $(x_p, b]$

Dowód: obrazkowy, nie wprost



Wniosek: próbkując funkcję w dwóch punktach x_ℓ, x_p zawsze możemy odrzucić jeden z obszarów $[a, x_\ell)$ lub $(x_p, b]$!

Algorytm znajdywania minimum funkcji jednomodalnej

Rozpocznij od pełnej dziedziny $a = x_{\min}$, $b = x_{\max}$

W kolejnych iteracjach:

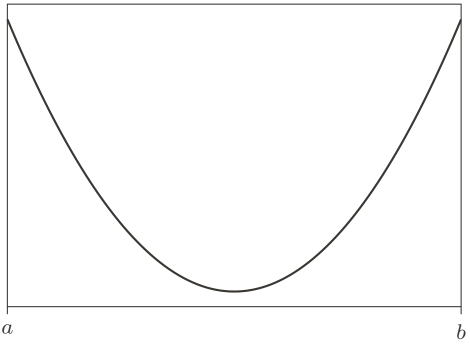
1. Wybierz $x_\ell, x_p \in [a, b]$ takie, że $x_\ell < x_p$
2. Jeśli $f(x_\ell) \leqslant f(x_p)$, przypisz $b = x_p$, w przeciwny przypadku przypisz $a = x_\ell$

Algorytm znajdywania minimum funkcji jednomodalnej

Rozpocznij od pełnej dziedziny $a = x_{\min}$, $b = x_{\max}$

W kolejnych iteracjach:

1. Wybierz $x_\ell, x_p \in [a, b]$ takie, że $x_\ell < x_p$
2. Jeśli $f(x_\ell) \leqslant f(x_p)$, przypisz $b = x_p$, w przeciwny przypadku przypisz $a = x_\ell$

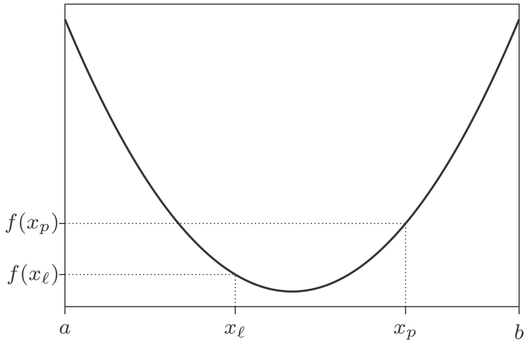


Algorytm znajdywania minimum funkcji jednomodalnej

Rozpocznij od pełnej dziedziny $a = x_{\min}$, $b = x_{\max}$

W kolejnych iteracjach:

1. Wybierz $x_\ell, x_p \in [a, b]$ takie, że $x_\ell < x_p$
2. Jeśli $f(x_\ell) \leqslant f(x_p)$, przypisz $b = x_p$, w przeciwny przypadku przypisz $a = x_\ell$

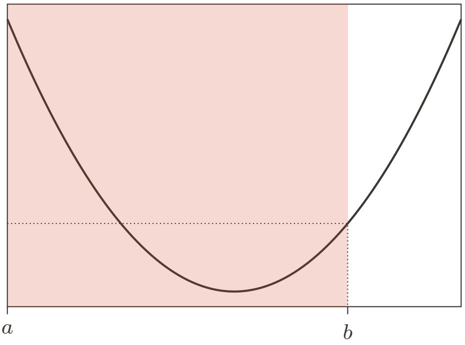


Algorytm znajdywania minimum funkcji jednomodalnej

Rozpocznij od pełnej dziedziny $a = x_{\min}$, $b = x_{\max}$

W kolejnych iteracjach:

1. Wybierz $x_\ell, x_p \in [a, b]$ takie, że $x_\ell < x_p$
2. Jeśli $f(x_\ell) \leqslant f(x_p)$, przypisz $b = x_p$, w przeciwny przypadku przypisz $a = x_\ell$

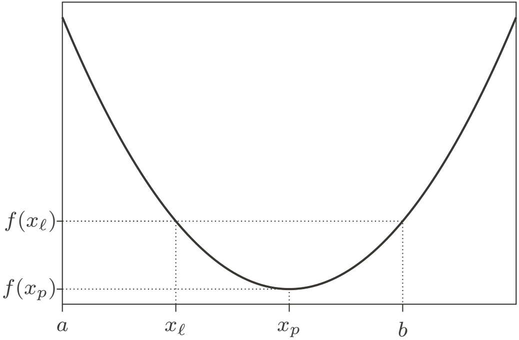


Algorytm znajdywania minimum funkcji jednomodalnej

Rozpocznij od pełnej dziedziny $a = x_{\min}$, $b = x_{\max}$

W kolejnych iteracjach:

1. Wybierz $x_\ell, x_p \in [a, b]$ takie, że $x_\ell < x_p$
2. Jeśli $f(x_\ell) \leqslant f(x_p)$, przypisz $b = x_p$, w przeciwny przypadku przypisz $a = x_\ell$

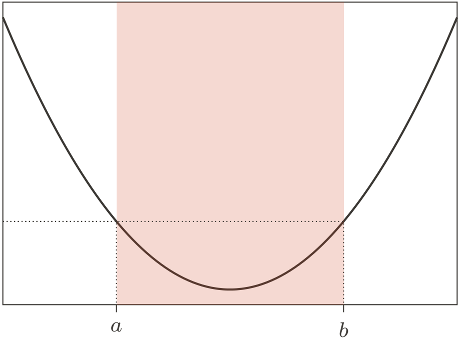


Algorytm znajdywania minimum funkcji jednomodalnej

Rozpocznij od pełnej dziedziny $a = x_{\min}$, $b = x_{\max}$

W kolejnych iteracjach:

1. Wybierz $x_\ell, x_p \in [a, b]$ takie, że $x_\ell < x_p$
2. Jeśli $f(x_\ell) \leqslant f(x_p)$, przypisz $b = x_p$, w przeciwny przypadku przypisz $a = x_\ell$

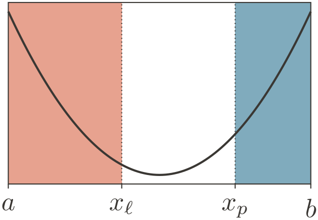


Algorytm znajdywania minimum funkcji jednomodalnej

Jak dobrać punkty x_ℓ, x_p, aby maksymalnie zawęzić obszar przeszukiwań?

Algorytm znajdywania minimum funkcji jednomodalnej

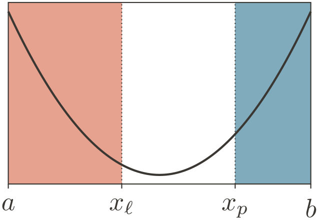
Jak dobrać punkty x_ℓ, x_p, aby maksymalnie zawęzić obszar przeszukiwań?



W każdej iteracji odrzucamy jeden z kolorowych obszarów (nie wiemy który)

Algorytm znajdywania minimum funkcji jednomodalnej

Jak dobrać punkty x_ℓ, x_p, aby maksymalnie zawęzić obszar przeszukiwań?

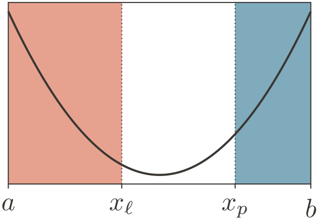


W każdej iteracji odrzucamy jeden z kolorowych obszarów (nie wiemy który)

Oba obszary powinny być **tej samej wielkości**

Algorytm znajdywania minimum funkcji jednomodalnej

Jak dobrać punkty x_ℓ, x_p, aby maksymalnie zawęzić obszar przeszukiwań?



- W każdej iteracji odrzucamy jeden z kolorowych obszarów (nie wiemy który)
- Oba obszary powinny być **tej samej wielkości**
- Punkty x_ℓ, x_p **jak najbliżej środka**

Algorytm przeszukiwania dychotomicznego

Wejście: procedura wyznaczająca wartość funkcji $f(x)$ w dowolnym punkcie dziedziny $[x_{\min}, x_{\max}]$; liczba odwołań do funkcji n; bardzo mała stała δ (np. $\delta = 10^{-8}$)

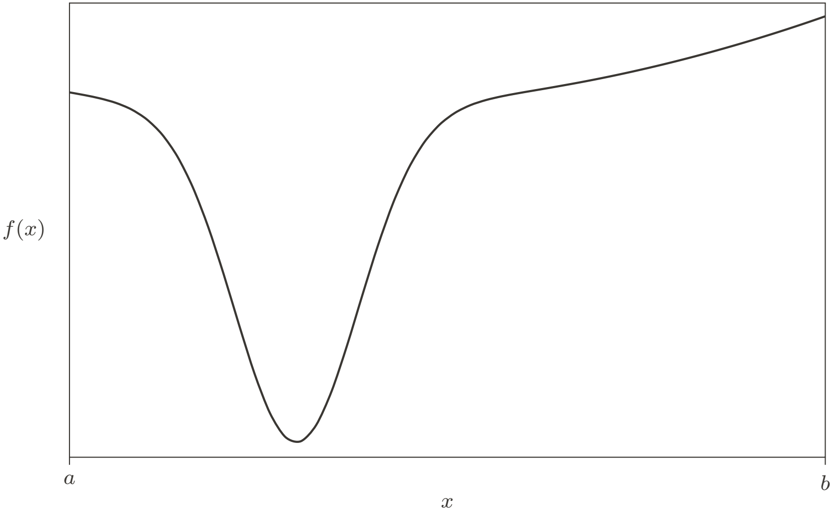
Inicjalizuj: $a = x_{\min}, b = x_{\max}$

Powtarzaj dla $k = 1, 2, \ldots, n/2$:

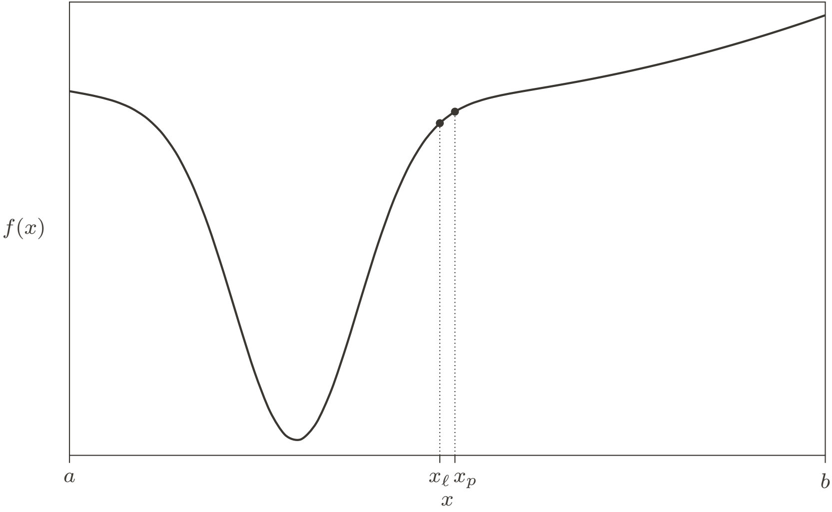
1. Wyznacz środkowy punkt $m = \frac{a+b}{2}$
2. Przypisz $x_\ell = m - \delta$, $x_p = m + \delta$
3. Wyznacz wartości funkcji $f(x_\ell)$ i $f(x_p)$
4. Jeśli $f(x_\ell) \leqslant f(x_p)$, przypisz $b = x_p$, w przeciwnym przypadku przypisz $a = x_\ell$

Zwróć wartość $x_n = \frac{a+b}{2}$

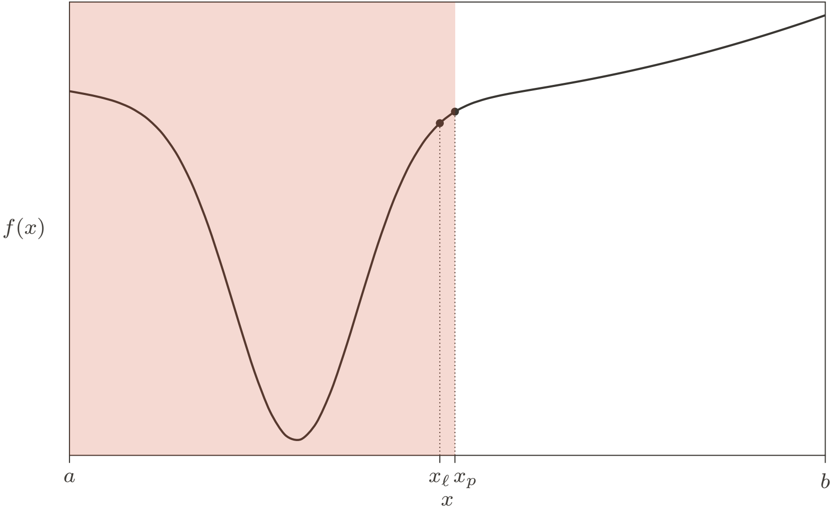
Przeszukiwanie dychotomiczne – przykład



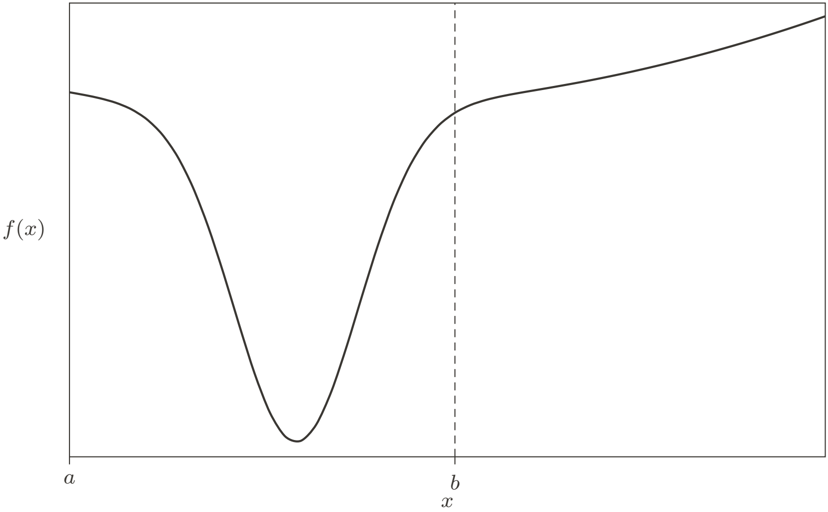
Przeszukiwanie dychotomiczne – przykład



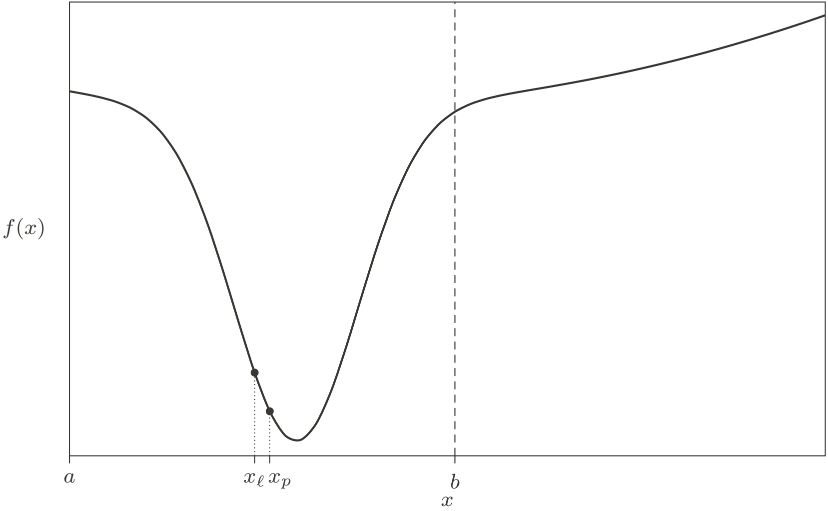
Przeszukiwanie dychotomiczne – przykład



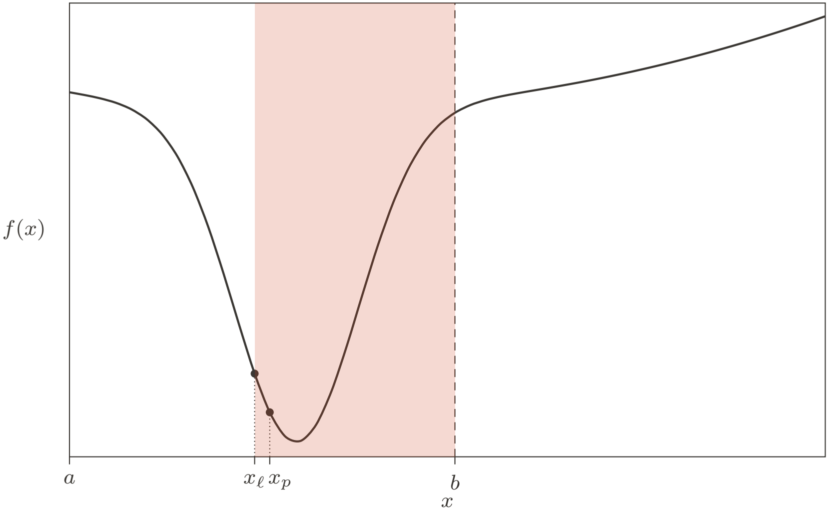
Przeszukiwanie dychotomiczne – przykład



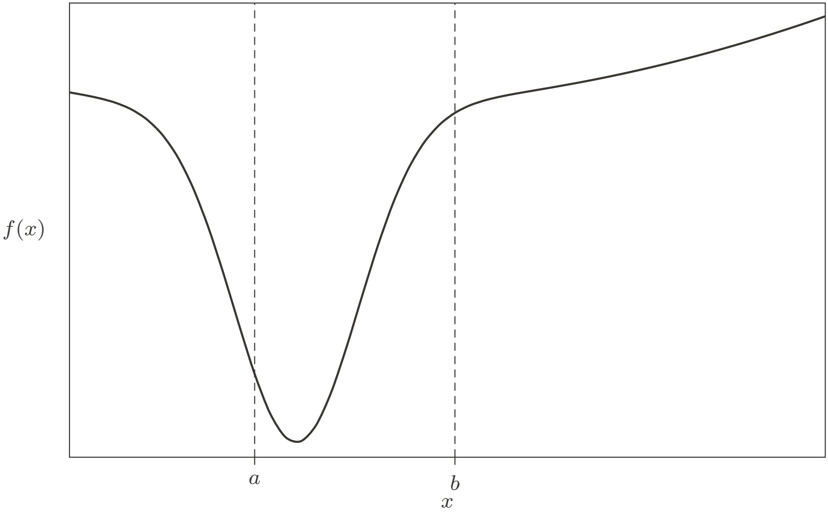
Przeszukiwanie dychotomiczne – przykład



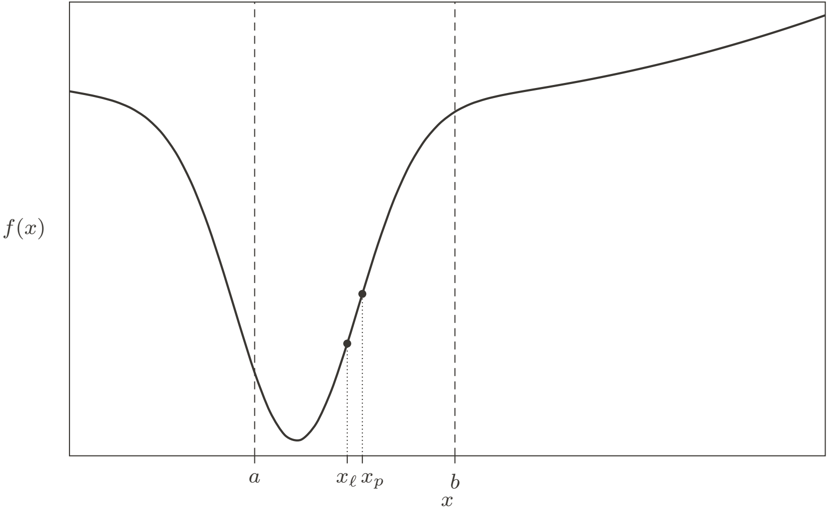
Przeszukiwanie dychotomiczne – przykład



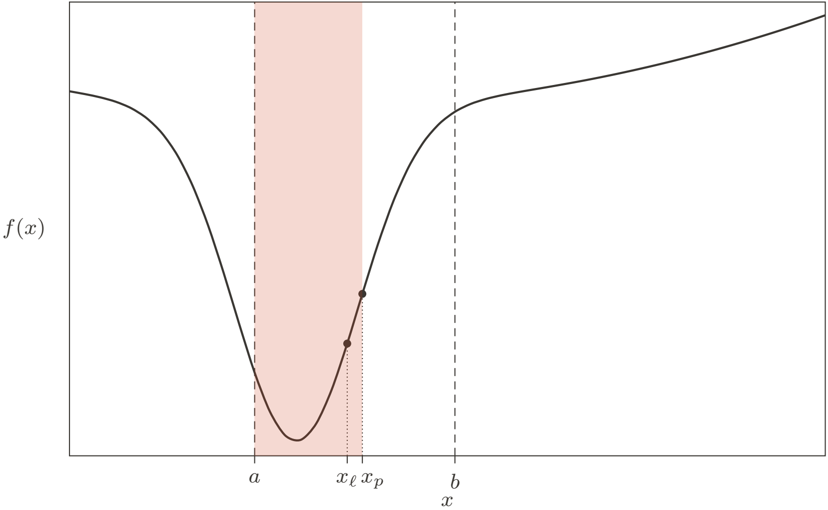
Przeszukiwanie dychotomiczne – przykład



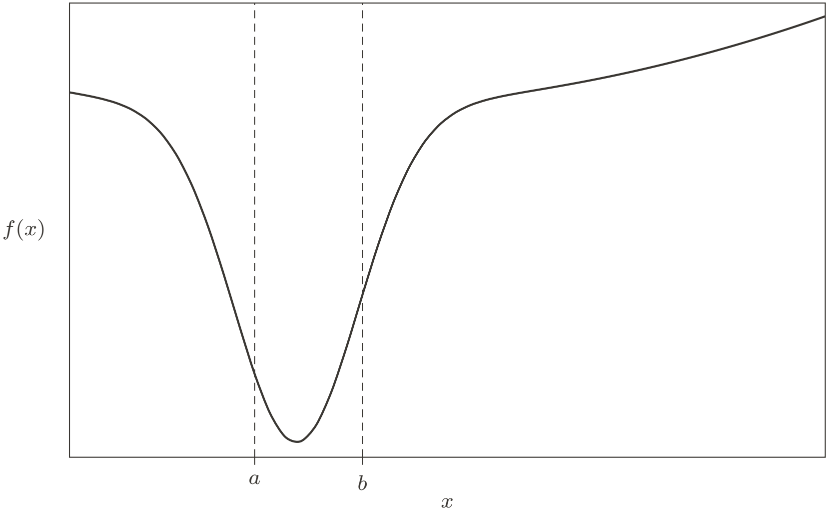
Przeszukiwanie dychotomiczne – przykład



Przeszukiwanie dychotomiczne – przykład



Przeszukiwanie dychotomiczne – przykład



Przeszukiwanie dychotomiczne – zbieżność

- Jeśli na początku iteracji obszar lokalizacji minimum ma długość x, pod koniec iteracji jest zawężony do $\frac{1}{2}x + \delta \simeq \frac{1}{2}x$

Przeszukiwanie dychotomiczne – zbieżność

- Jeśli na początku iteracji obszar lokalizacji minimum ma długość x, pod koniec iteracji jest zawężony do $\frac{1}{2}x + \delta \simeq \frac{1}{2}x$
- **Wniosek:** rozpoczynając z zakresem o długości $|\mathcal{X}|$, po k iteracjach minimum jest zlokalizowane z dokładnością $\left(\frac{1}{2}\right)^k |\mathcal{X}|$

Przeszukiwanie dychotomiczne – zbieżność

- Jeśli na początku iteracji obszar lokalizacji minimum ma długość x, pod koniec iteracji jest zawężony do $\frac{1}{2}x + \delta \simeq \frac{1}{2}x$
- **Wniosek:** rozpoczynając z zakresem o długości $|\mathcal{X}|$, po k iteracjach minimum jest zlokalizowane z dokładnością $\left(\frac{1}{2}\right)^k |\mathcal{X}|$
- **Wniosek:** po n odwołaniach do funkcji:

$$|x_n - x^*| \leqslant \left(\frac{1}{2}\right)^{n/2} |\mathcal{X}| = \left(\frac{1}{\sqrt{2}}\right)^n |\mathcal{X}|,$$

punkt zwracany przez algorytm

punkt optymalny (minimum $f(x)$)

Przeszukiwanie dychotomiczne – zbieżność

n	$\lvert x_n - x^* \rvert$
0	1
1	0.707
2	0.5
5	0.177
10	0.031
20	0.001
50	$3 \cdot 10^{-8}$

Czy można to zrobić szybciej?

Czy można to zrobić szybciej?

Dotychczas w każdej iteracji próbkowaliśmy funkcję w dwóch punktach i zmniejszaliśmy przedział przeszukiwań o połowę.

Jeden z dwóch punktów staje się nową granicą przedziału, po czym próbkujemy funkcję w dwóch nowych punktach.

Czy można to zrobić szybciej?

Dotychczas w każdej iteracji próbkowaliśmy funkcję w dwóch punktach i zmniejszaliśmy przedział przeszukiwań o połowę.

Jeden z dwóch punktów staje się nową granicą przedziału, po czym próbkujemy funkcję w dwóch nowych punktach.

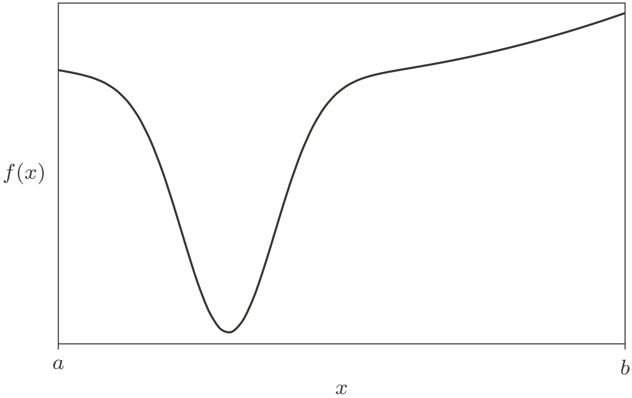
Pomysł: wykorzystać jeden z punktów z poprzedniej iteracji i próbkować funkcję tylko raz na iterację!

Czy można to zrobić szybciej?

Dotychczas w każdej iteracji próbkowaliśmy funkcję w dwóch punktach i zmniejszaliśmy przedział przeszukiwań o połowę.

Jeden z dwóch punktów staje się nową granicą przedziału, po czym próbkujemy funkcję w dwóch nowych punktach.

Pomysł: wykorzystać jeden z punktów z poprzedniej iteracji i próbkować funkcję tylko raz na iterację!

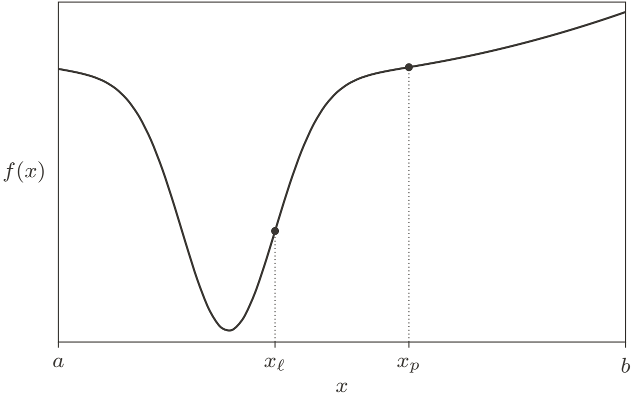


Czy można to zrobić szybciej?

Dotychczas w każdej iteracji próbkowaliśmy funkcję w dwóch punktach i zmniejszaliśmy przedział przeszukiwań o połowę.

Jeden z dwóch punktów staje się nową granicą przedziału, po czym próbkujemy funkcję w dwóch nowych punktach.

Pomysł: wykorzystać jeden z punktów z poprzedniej iteracji i próbkować funkcję tylko raz na iterację!

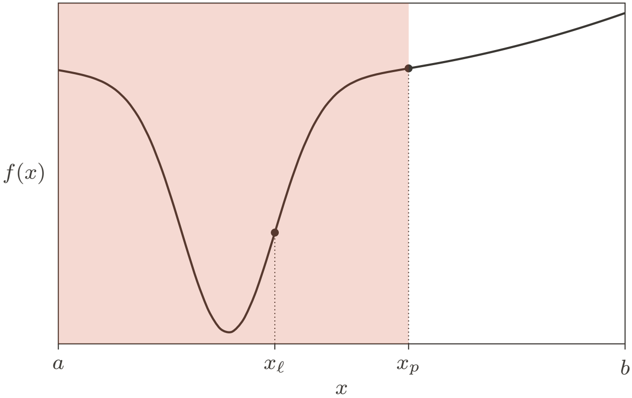


Czy można to zrobić szybciej?

Dotychczas w każdej iteracji próbkowaliśmy funkcję w dwóch punktach i zmniejszaliśmy przedział przeszukiwań o połowę.

Jeden z dwóch punktów staje się nową granicą przedziału, po czym próbkujemy funkcję w dwóch nowych punktach.

Pomysł: wykorzystać jeden z punktów z poprzedniej iteracji i próbkować funkcję tylko raz na iterację!

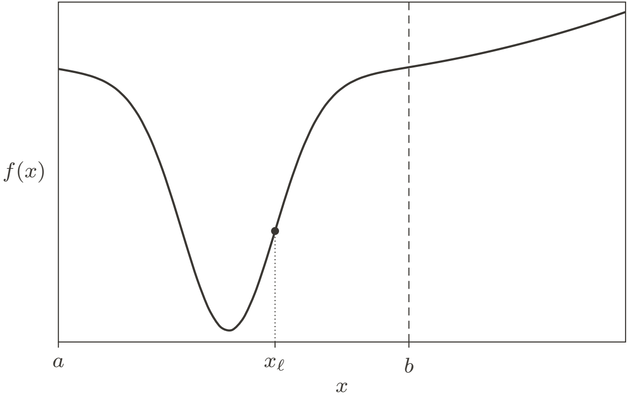


Czy można to zrobić szybciej?

Dotychczas w każdej iteracji próbkowaliśmy funkcję w dwóch punktach i zmniejszaliśmy przedział przeszukiwań o połowę.

Jeden z dwóch punktów staje się nową granicą przedziału, po czym próbkujemy funkcję w dwóch nowych punktach.

Pomysł: wykorzystać jeden z punktów z poprzedniej iteracji i próbkować funkcję tylko raz na iterację!

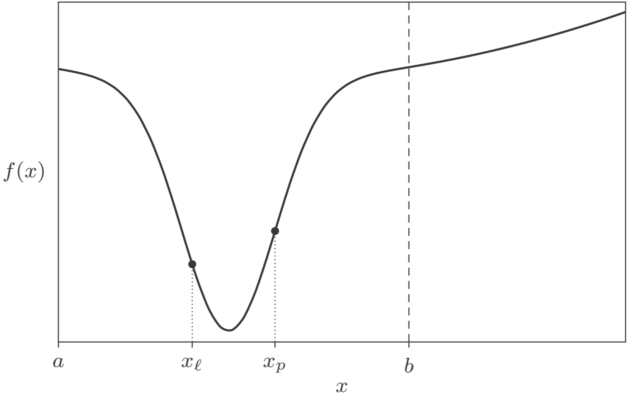


Czy można to zrobić szybciej?

Dotychczas w każdej iteracji próbkowaliśmy funkcję w dwóch punktach i zmniejszaliśmy przedział przeszukiwań o połowę.

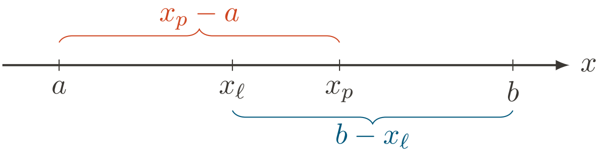
Jeden z dwóch punktów staje się nową granicą przedziału, po czym próbkujemy funkcję w dwóch nowych punktach.

Pomysł: wykorzystać jeden z punktów z poprzedniej iteracji i próbkować funkcję tylko raz na iterację!



Konstrukcja algorytmu

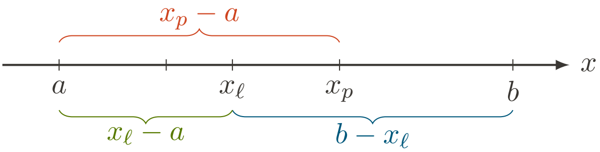
Konstrukcja algorytmu



- W następnej iteracji obszar zwęża się do $x_p - a$ lub $b - x_\ell$. Ponieważ nie wiemy, który przypadek nastąpi, ustalamy:

$$x_p - a = b - x_\ell$$

Konstrukcja algorytmu



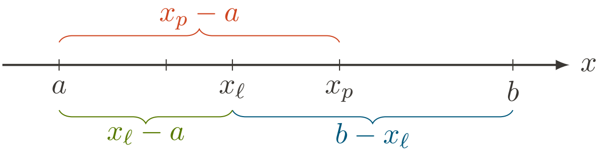
- W następnej iteracji obszar zwęża się do $x_p - a$ lub $b - x_\ell$. Ponieważ nie wiemy, który przypadek nastąpi, ustalamy:

$$x_p - a \; = \; b - x_\ell$$

- Jeśli wybierzemy obszar np. $x_p - a$, punkt x_ℓ zostanie użyty ponownie. Aby zachować proporcje podziału, ustalamy:

$$\frac{x_p - a}{b - a} = \frac{x_\ell - a}{x_p - a}$$

Konstrukcja algorytmu



- W następnej iteracji obszar zwęża się do $x_p - a$ lub $b - x_\ell$. Ponieważ nie wiemy, który przypadek nastąpi, ustalamy:

$$x_p - a \;=\; b - x_\ell \quad\Longrightarrow\quad x_\ell - a \;=\; b - x_p$$

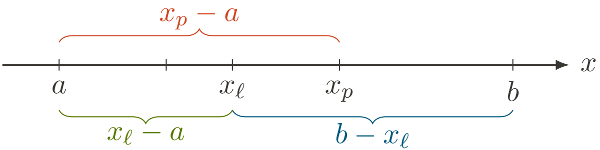
- Jeśli wybierzemy obszar np. $x_p - a$, punkt x_ℓ zostanie użyty ponownie. Aby zachować proporcje podziału, ustalamy:

$$\frac{x_p - a}{b - a} = \frac{x_\ell - a}{x_p - a}$$

- Podstawiając pierwsze z równań do drugiego:

$$\frac{x_p - a}{b - a} = \frac{b - x_p}{x_p - a} = \frac{b - a - (x_p - a)}{x_p - a} = \frac{b - a}{x_p - a} - 1$$

Konstrukcja algorytmu



- W następnej iteracji obszar zwęża się do $x_p - a$ lub $b - x_\ell$. Ponieważ nie wiemy, który przypadek nastąpi, ustalamy:

$$x_p - a \;=\; b - x_\ell \quad\Longrightarrow\quad x_\ell - a \;=\; b - x_p$$

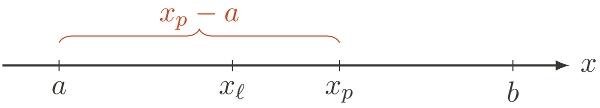
- Jeśli wybierzemy obszar np. $x_p - a$, punkt x_ℓ zostanie użyty ponownie. Aby zachować proporcje podziału, ustalamy:

$$\frac{x_p - a}{b - a} = \frac{x_\ell - a}{x_p - a}$$

- Podstawiając pierwsze z równań do drugiego:

$$\underbrace{\frac{x_p - a}{b - a}}_{q} = \frac{b - x_p}{x_p - a} = \frac{b - a - (x_p - a)}{x_p - a} = \underbrace{\frac{b - a}{x_p - a}}_{1/q} - 1$$

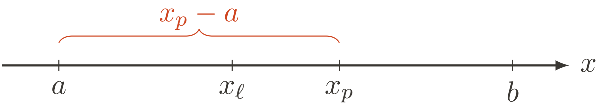
Konstrukcja algorytmu



- Otrzymaliśmy równanie:

$$q = \frac{1}{q} - 1 \quad \Longrightarrow \quad q^2 + q - 1 = 0, \qquad \text{dla} \;\; q = \frac{x_p - a}{b - a}$$

Konstrukcja algorytmu



- Otrzymaliśmy równanie:

$$q = \frac{1}{q} - 1 \quad \Longrightarrow \quad q^2 + q - 1 = 0, \qquad \text{dla} \quad q = \frac{x_p - a}{b - a}$$

- Rozwiązanie:

$$q = \frac{\sqrt{5} - 1}{2} \simeq 0.618.$$

Złoty podział odcinka.

Algorytm złotego podziału

Wejście: procedura wyznaczająca wartość funkcji $f(x)$ w dowolnym punkcie dziedziny $[x_{\min}, x_{\max}]$; liczba odwołań do funkcji n; $\alpha := \frac{\sqrt{5}-1}{2} \simeq 0.618$

Inicjalizuj: $a = x_{\min}, b = x_{\max}$
$x_\ell = \alpha a + (1-\alpha)b$, $x_p = (1-\alpha)a + \alpha b$

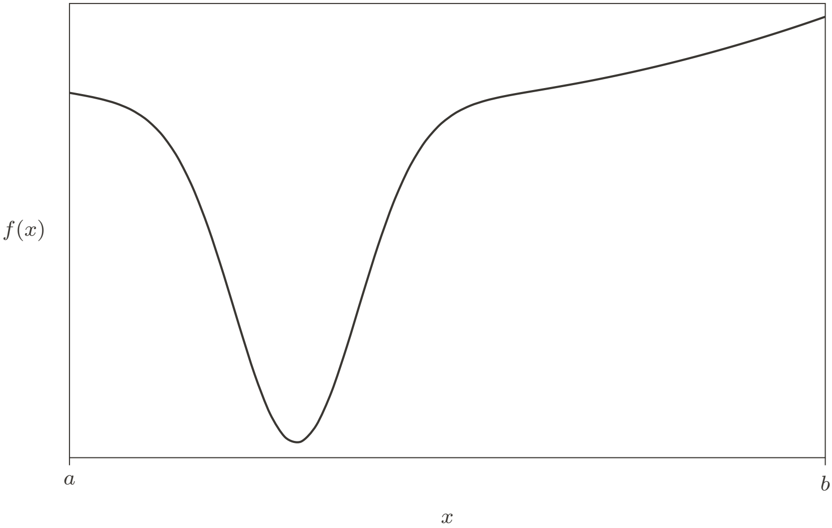
Wyznacz wartości funkcji $f(x_\ell)$, $f(x_p)$

Powtarzaj dla $k = 2, 3, \ldots, n$:

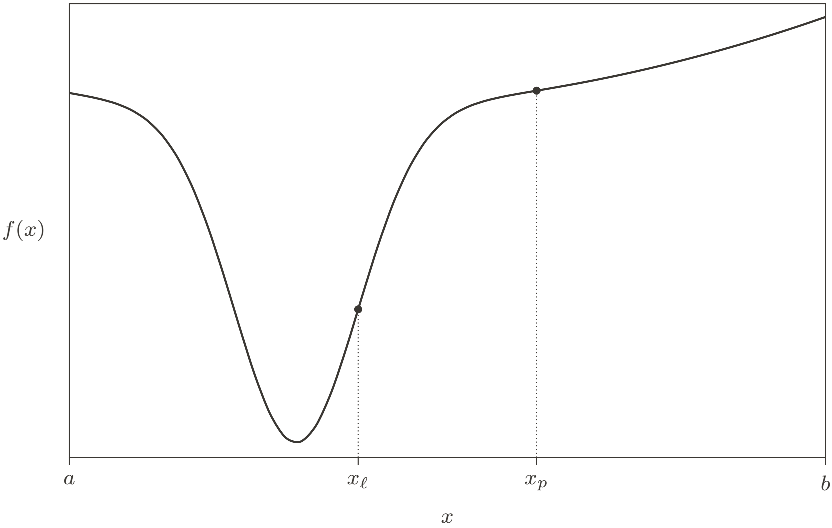
- **Jeśli** $f(x_\ell) < f(x_p)$
  - Przypisz $b = x_p$, $x_p = x_\ell$, $x_\ell = \alpha a + (1-\alpha)b$
  - Jeśli $k \neq n$, wyznacz wartość funkcji $f(x_\ell)$.
- **w przeciwnym przypadku**
  - Przypisz $a = x_\ell$, $x_\ell = x_p$, $x_p = (1-\alpha)a + \alpha b$
  - Jeśli $k \neq n$, wyznacz wartość funkcji $f(x_p)$.

Zwróć wartość $x_n = \frac{a+b}{2}$

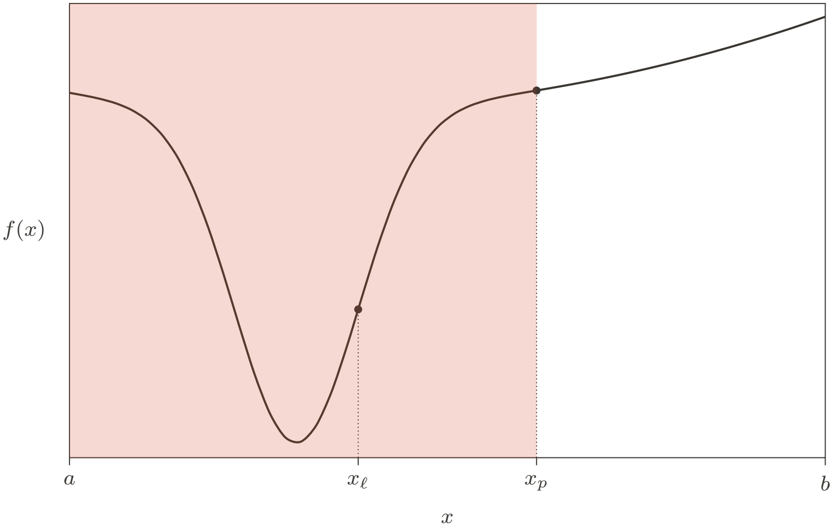
Algorytm złotego podziału – przykład



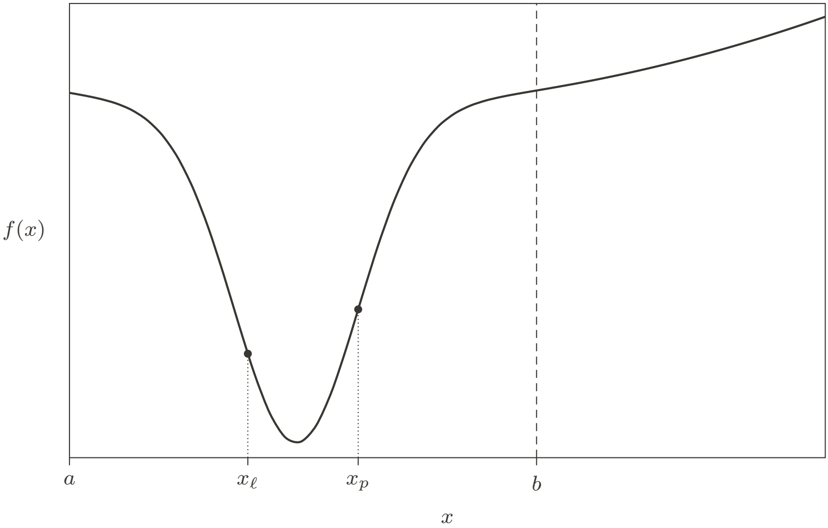
Algorytm złotego podziału – przykład



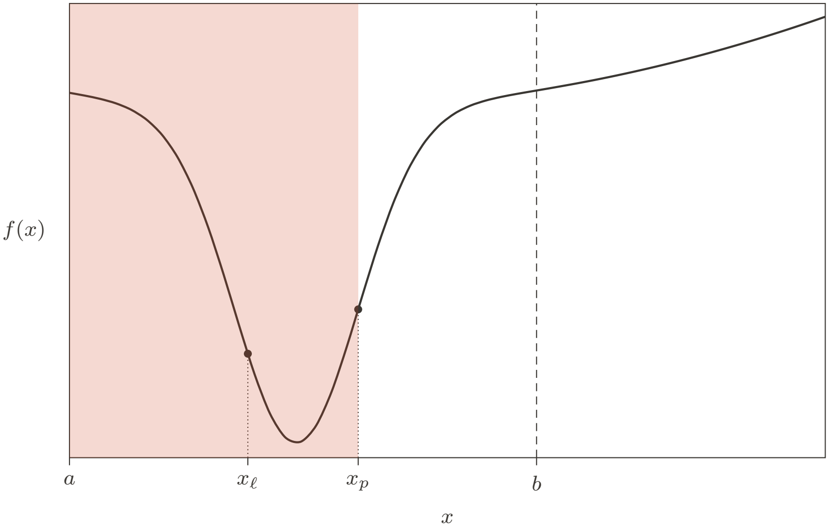
Algorytm złotego podziału – przykład



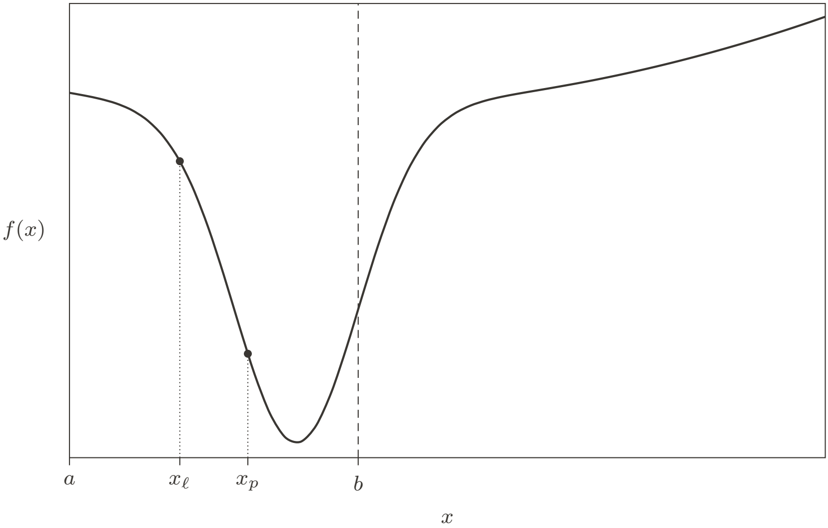
Algorytm złotego podziału – przykład



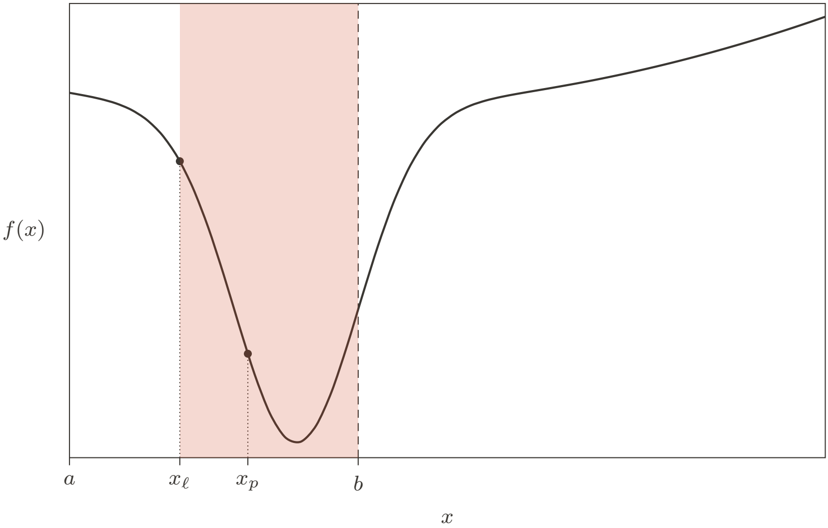
Algorytm złotego podziału – przykład



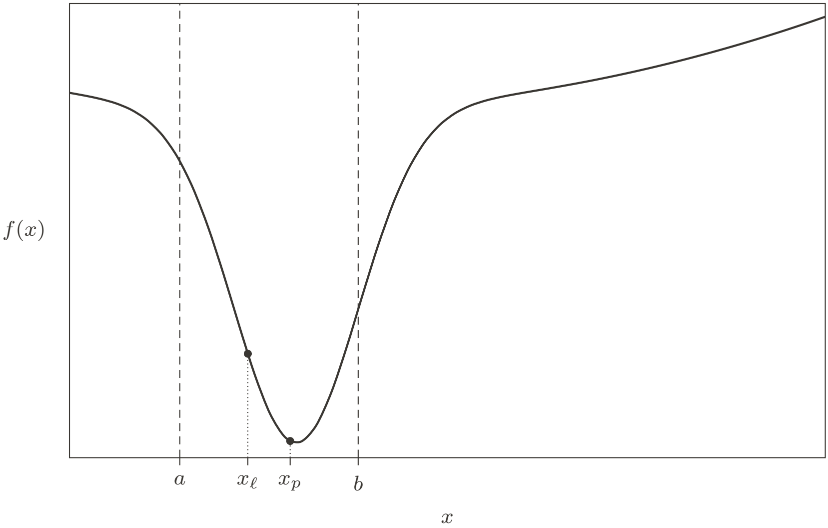
Algorytm złotego podziału – przykład



Algorytm złotego podziału – przykład



Algorytm złotego podziału – przykład



Algorytm złotego podziału – zbieżność

- Jeśli na początku iteracji obszar lokalizacji minimum ma długość x, pod koniec iteracji jest zawężony do αx, gdzie $\alpha = \frac{\sqrt{5}-1}{2}$

Algorytm złotego podziału – zbieżność

- Jeśli na początku iteracji obszar lokalizacji minimum ma długość x, pod koniec iteracji jest zawężony do αx, gdzie $\alpha = \frac{\sqrt{5}-1}{2}$
- **Wniosek:** rozpoczynając z zakresem o długości $|\mathcal{X}|$, po $n-1$ iteracjach minimum jest zlokalizowane z dokładnością $\alpha^{n-1}|\mathcal{X}|$

Algorytm złotego podziału – zbieżność

- Jeśli na początku iteracji obszar lokalizacji minimum ma długość x, pod koniec iteracji jest zawężony do αx, gdzie $\alpha = \frac{\sqrt{5}-1}{2}$
- **Wniosek:** rozpoczynając z zakresem o długości $|\mathcal{X}|$, po $n-1$ iteracjach minimum jest zlokalizowane z dokładnością $\alpha^{n-1}|\mathcal{X}|$
- **Wniosek:** po n odwołaniach do funkcji:

$$|x_n - x^*| \leqslant \alpha^{n-1}|\mathcal{X}| \simeq (0.618)^n|\mathcal{X}|,$$

Porównanie: dychotomiczne vs. złoty podział

algorytm	zbieżność	
przeszukiwania dychotomicznego	$\left(\frac{1}{\sqrt{2}}\right)^n$	$\simeq (0.707)^n$
złotego podziału	$\left(\frac{\sqrt{5}-1}{2}\right)^{n-1}$	$\simeq (0.618)^n$

Porównanie: dychotomiczne vs. złoty podział

algorytm	zbieżność	
przeszukiwania dychotomicznego	$\left(\frac{1}{\sqrt{2}}\right)^n$	$\simeq (0.707)^n$
złotego podziału	$\left(\frac{\sqrt{5}-1}{2}\right)^{n-1}$	$\simeq (0.618)^n$

- Zbieżność $(0.618)^n$ jest **asymptotycznie optymalna**.
- Algorytm optymalny to **metoda Fibonacciego**:
  - Zbieżność bardzo podobna jak dla algorytmu złotego podziału (asymptotycznie taka sama).
  - Bardziej skomplikowany i przez to mniej praktyczny.

Optymalizacja funkcji różniczkowalnych: algorytm bisekcji

Założenia

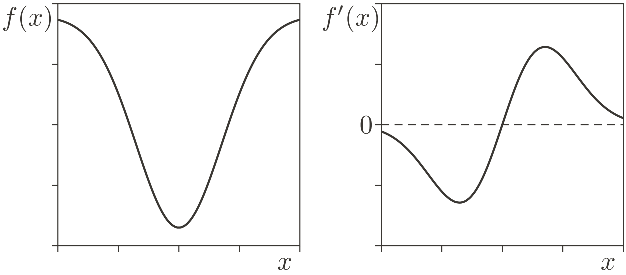
Założenia:

1. Funkcja f ma **ciągłą pochodną** na $\mathcal{X} = [x_{\min}, x_{\max}]$
2. Pochodna zmienia na $\mathcal{X}$ jednokrotnie znak z ujemnego na dodatni

Założenia:

1. Funkcja f ma **ciągłą pochodną** na $\mathcal{X} = [x_{\min}, x_{\max}]$
2. Pochodna zmienia na $\mathcal{X}$ jednokrotnie znak z ujemnego na dodatni

Wniosek: funkcja jest **jednomodalna**: wpierw ściśle malejąca, potem ściśle rosnąca, z minimum w punkcie **zerowania się pochodnej**

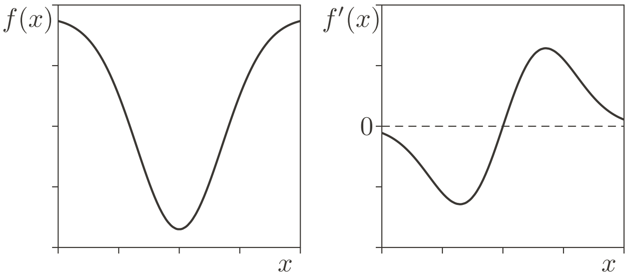


Założenia

Założenia:

1. Funkcja f ma **ciągłą pochodną** na $\mathcal{X} = [x_{\min}, x_{\max}]$
2. Pochodna zmienia na $\mathcal{X}$ jednokrotnie znak z ujemnego na dodatni

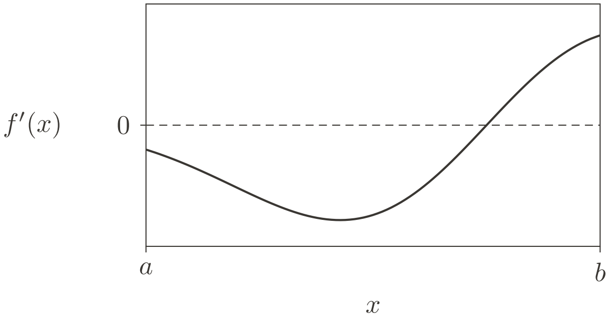
Wniosek: funkcja jest **jednomodalna**: wpierw ściśle malejąca, potem ściśle rosnąca, z minimum w punkcie **zerowania się pochodnej**



Wniosek: Sprowadzamy problem do znalezienia **miejsca zerowego** pochodnej

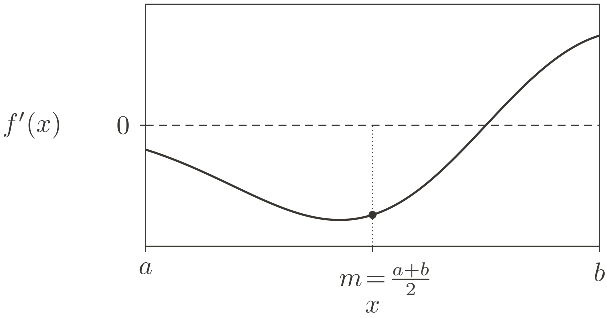
Metoda bisekcji

Rozważmy przedział $[a, b]$ z $f'(a) < 0$ i $f'(b) > 0$, w którym f' ma dokładnie jedno miejsce zerowe



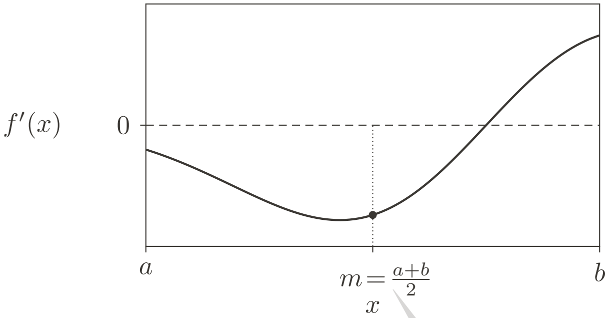
Metoda bisekcji

Rozważmy przedział $[a, b]$ z $f'(a) < 0$ i $f'(b) > 0$, w którym f' ma dokładnie jedno miejsce zerowe



Metoda bisekcji

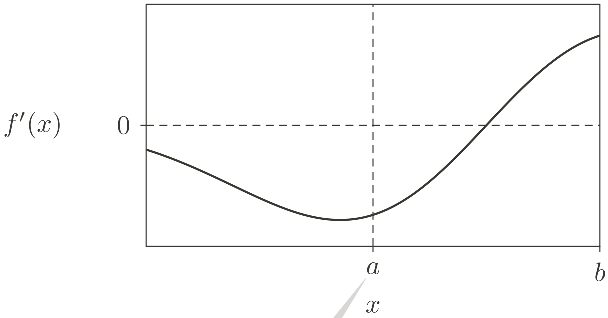
Rozważmy przedział $[a, b]$ z $f'(a) < 0$ i $f'(b) > 0$, w którym f' ma dokładnie jedno miejsce zerowe



Skoro $f'(m) < 0$ to miejsce zerowe musi leżeć po **prawej** stronie!

Metoda bisekcji

Rozważmy przedział $[a, b]$ z $f'(a) < 0$ i $f'(b) > 0$, w którym f' ma dokładnie jedno miejsce zerowe



Zawężamy obszar przyjmując $a = m$
i ponownie mamy $f'(a) < 0$, $f'(b) > 0$

Metoda bisekcji

Wejście: procedura wyznaczająca wartość pochodnej $f'(x)$ w dowolnym punkcie dziedziny $[x_{\min}, x_{\max}]$; liczba odwołań do pochodnej n;

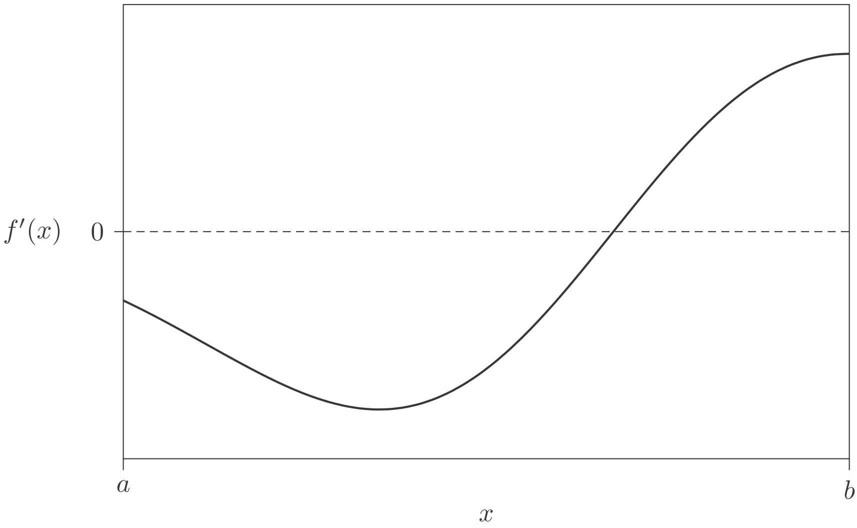
Inicjalizuj: $a = x_{\min}, b = x_{\max}$

Powtarzaj dla $k = 1, 2, \ldots, n$:

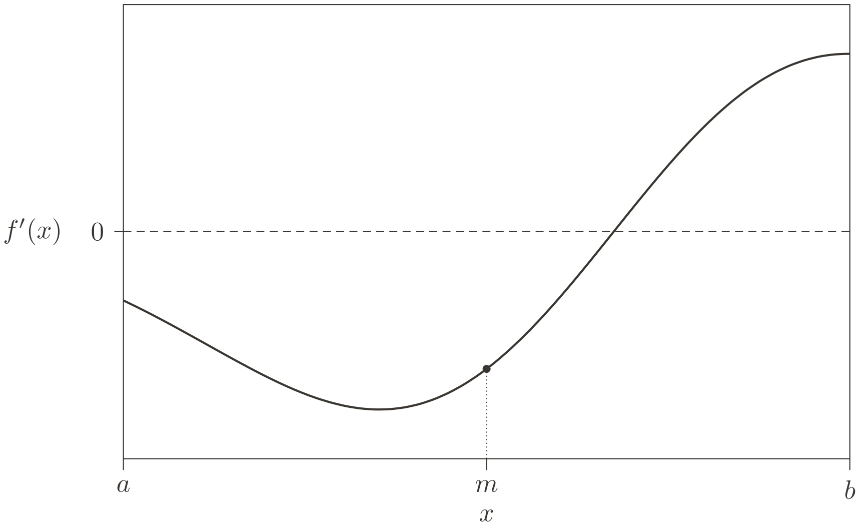
- Wyznacz pochodną $f'(m)$ w punkcie $m = \frac{a+b}{2}$
- Jeśli $f'(m) = 0$, zakończ algorytm (znaleziono minimum)
- W przeciwny przypadku:
  - Jeśli $f'(m) > 0$ przypisz $b = m$
  - Jeśli $f'(m) < 0$ przypisz $a = m$

Zwróć wartość $x_n = \frac{a+b}{2}$

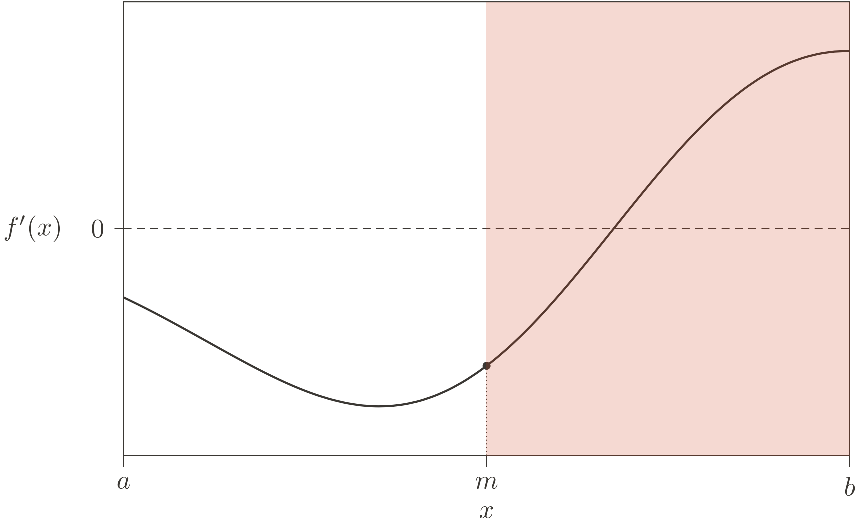
Metoda bisekcji – przykład



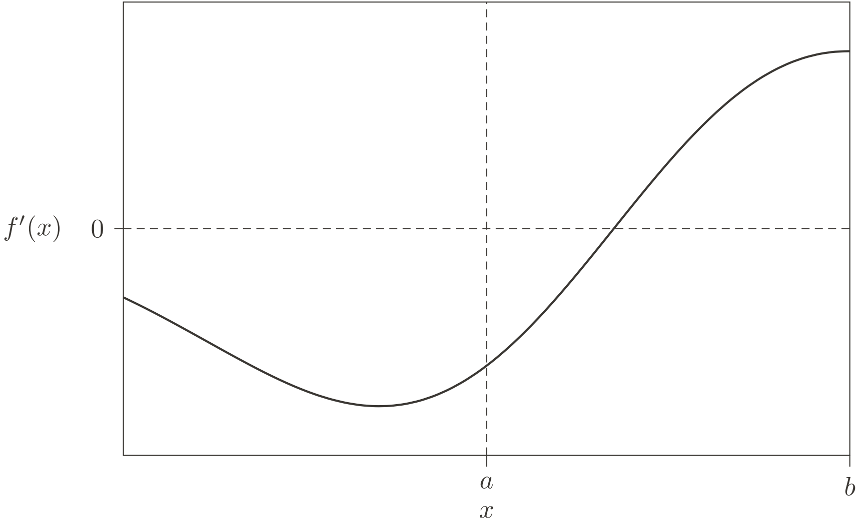
Metoda bisekcji – przykład



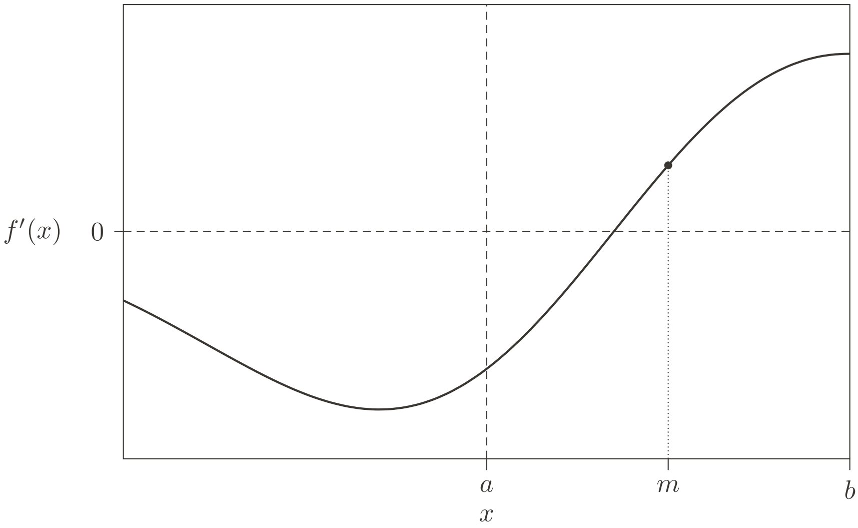
Metoda bisekcji – przykład



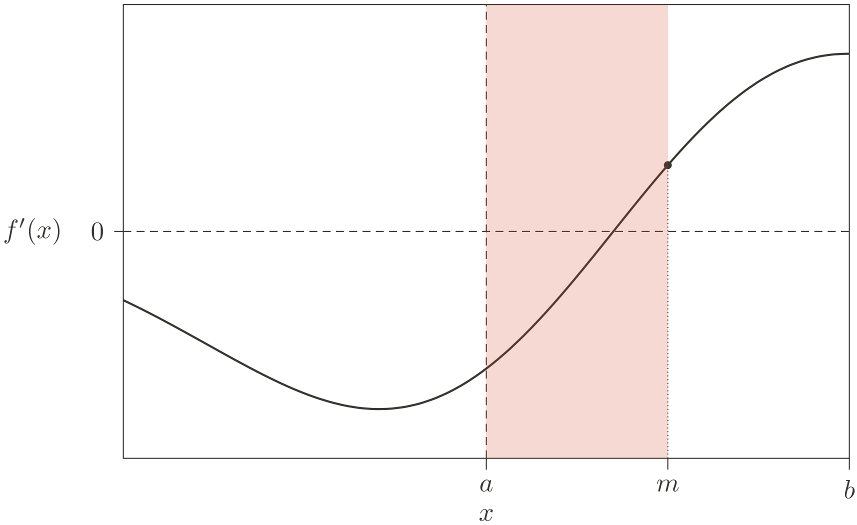
Metoda bisekcji – przykład



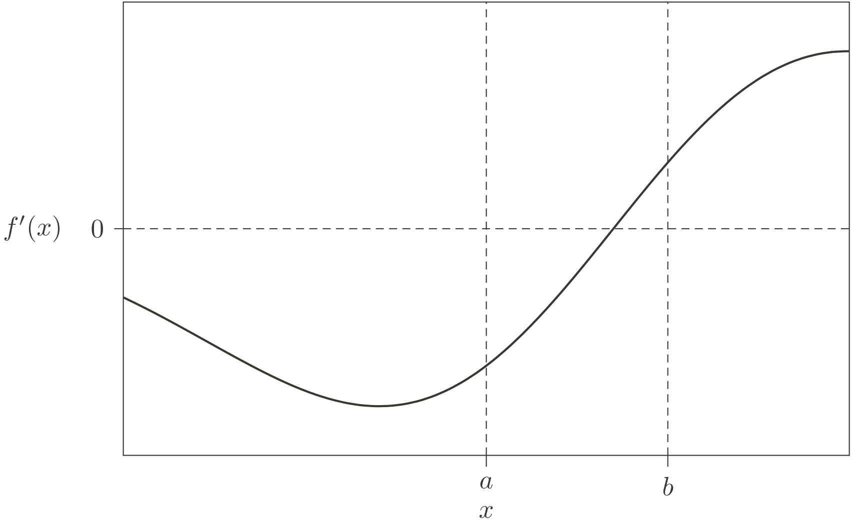
Metoda bisekcji – przykład



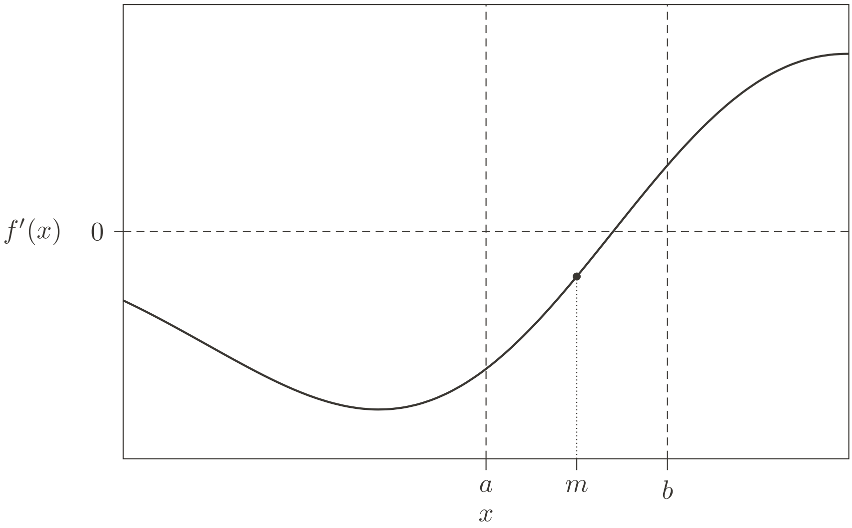
Metoda bisekcji – przykład



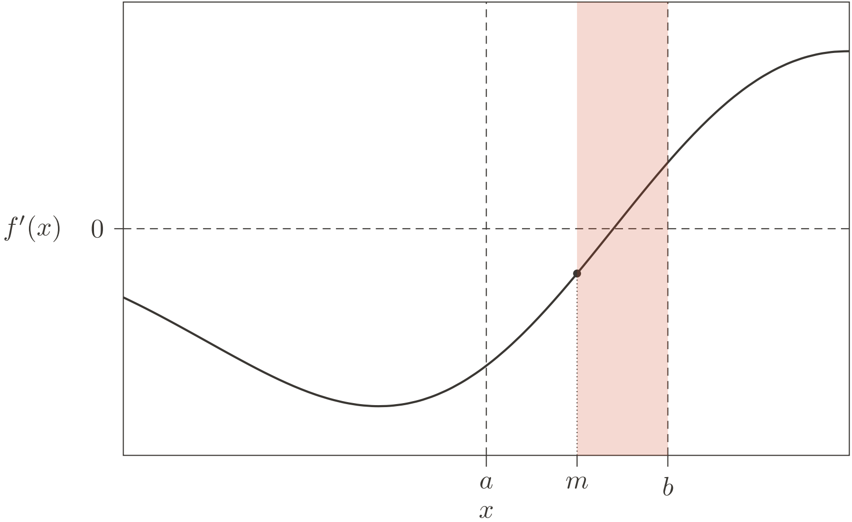
Metoda bisekcji – przykład



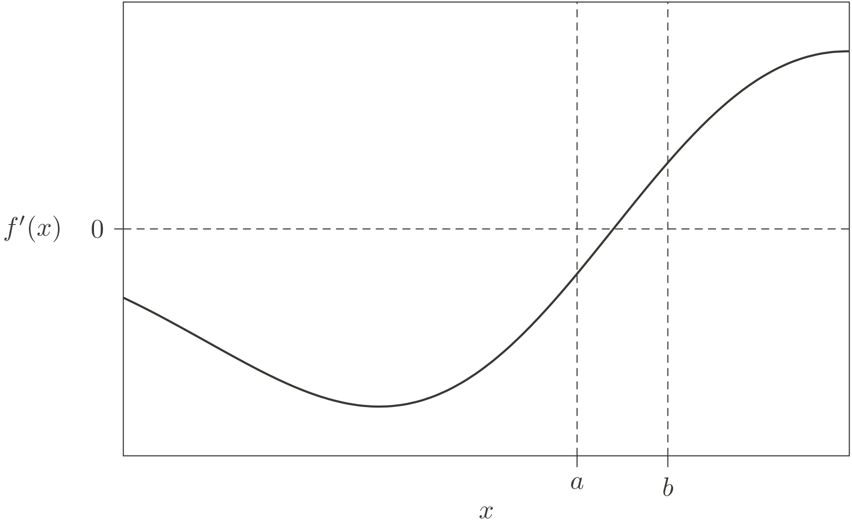
Metoda bisekcji – przykład



Metoda bisekcji – przykład



Metoda bisekcji – przykład



Metoda bisekcji – zbieżność

- Jeśli na początku iteracji obszar lokalizacji minimum ma długość x, pod koniec iteracji jest zawężony do $\frac{1}{2}x$

Metoda bisekcji – zbieżność

- Jeśli na początku iteracji obszar lokalizacji minimum ma długość x, pod koniec iteracji jest zawężony do $\frac{1}{2}x$
- **Wniosek:** po n odwołaniach do pochodnej funkcji:

$$|x_n - x^*| \leqslant \left(\frac{1}{2}\right)^n |\mathcal{X}|$$

Metoda bisekcji – zbieżność

- Jeśli na początku iteracji obszar lokalizacji minimum ma długość x, pod koniec iteracji jest zawężony do $\frac{1}{2}x$
- **Wniosek:** po n odwołaniach do pochodnej funkcji:

$$|x_n - x^*| \leqslant \left(\frac{1}{2}\right)^n |\mathcal{X}|$$

algorytm	zbieżność
przeszukiwania dychotomicznego	$(0.707)^n$
złotego podziału	$(0.618)^n$
bisekcji	$(0.5)^n$

Metoda bisekcji – zbieżność

- Jeśli na początku iteracji obszar lokalizacji minimum ma długość x, pod koniec iteracji jest zawężony do $\frac{1}{2}x$
- **Wniosek:** po n odwołaniach do pochodnej funkcji:

$$|x_n - x^*| \leqslant \left(\frac{1}{2}\right)^n |\mathcal{X}|$$

algorytm	zbieżność
przeszukiwania dychotomicznego	$(0.707)^n$
złotego podziału	$(0.618)^n$
bisekcji	$(0.5)^n$

Znajomość pochodnej pozwala na szybszą zbieżność!

Algorytm przeszukiwania dychotomicznego jest przybliżeniem metody bisekcji, gdzie znak pochodnej szacuje się poprzez $f(m+\delta) - f(m-\delta)$

Optymalizacja funkcji różniczkowalnych: algorytm Newtona

Wielomian Taylora i wzór Taylora

Wielomian:

$$P_m(x;x_0) = f(x_0)+\frac{f'(x_0)}{1!}(x-x_0)+\frac{f''(x_0)}{2!}(x-x_0)^2+\ldots+\frac{f^{(m)}(x_0)}{m!}(x-x_0)^m$$

nazywamy **Wielomianem Taylora** rzędu m funkcji f w punkcie x_0.

Wielomian Taylora coraz lepiej przybliża przebieg funkcji wokół punktu x_0 dla coraz większych m.

Wielomian Taylora i wzór Taylora

Wielomian:

$$P_m(x;x_0) = f(x_0) + \frac{f'(x_0)}{1!}(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \ldots + \frac{f^{(m)}(x_0)}{m!}(x-x_0)^m$$

nazywamy **Wielomianem Taylora** rzędu m funkcji f w punkcie x_0.

Wielomian Taylora coraz lepiej przybliża przebieg funkcji wokół punktu x_0 dla coraz większych m.

Wzór Taylora: Błąd oszacowania funkcji $f(x)$ wielomianem Taylora $P_m(x;x_0)$ wokół punktu x_0:

$$f(x) = P_m(x;x_0) + R_m(x;x_0), \quad R_m(x;x_0) = \underbrace{\frac{f^{(m+1)}(\xi)}{(m+1)!}(x-x_0)^{m+1}}_{\text{reszta w postaci Lagrange'a}},$$

gdzie ξ jest punktem leżącym między x a x_0

Wielomiany Taylora – przykład

Rozwinięcie Taylora funkcji $f(x) = e^x$ w punkcie $x_0 = 0$.

Wielomiany Taylora – przykład

Rozwinięcie Taylora funkcji $f(x) = e^x$ w punkcie $x_0 = 0$.

$$f(x) = e^x \qquad f(x_0) = e^0 = 1 \qquad P_0(x;0) = 1$$

Wielomiany Taylora – przykład

Rozwinięcie Taylora funkcji $f(x) = e^x$ w punkcie $x_0 = 0$.

$f(x) = e^x \qquad f(x_0) = e^0 = 1 \qquad P_0(x;0) = 1$

$f'(x) = e^x \qquad f'(x_0) = e^0 = 1 \qquad P_1(x;0) = 1 + \frac{1}{1!}x = 1 + x$

Wielomiany Taylora – przykład

Rozwinięcie Taylora funkcji $f(x) = e^x$ w punkcie $x_0 = 0$.

$$
\begin{array}{lll}
f(x) = e^x & f(x_0) = e^0 = 1 & P_0(x;0) = 1 \\
f'(x) = e^x & f'(x_0) = e^0 = 1 & P_1(x;0) = 1 + \frac{1}{1!}x = 1 + x \\
f''(x) = e^x & f''(x_0) = e^0 = 1 & P_2(x;0) = 1 + x + \frac{1}{2!}x = 1 + x + \frac{x^2}{2}
\end{array}
$$

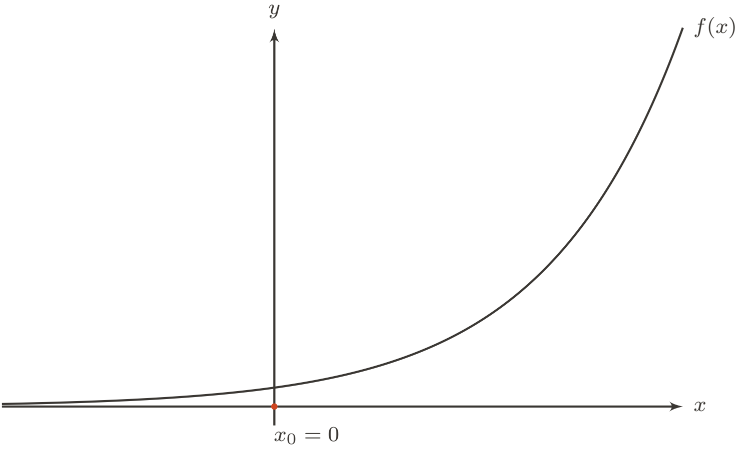
Wielomiany Taylora – przykład

Rozwinięcie Taylora funkcji $f(x) = e^x$ w punkcie $x_0 = 0$.

$$
\begin{array}{lll}
f(x) = e^x & f(x_0) = e^0 = 1 & P_0(x;0) = 1 \\
f'(x) = e^x & f'(x_0) = e^0 = 1 & P_1(x;0) = 1 + \frac{1}{1!}x = 1 + x \\
f''(x) = e^x & f''(x_0) = e^0 = 1 & P_2(x;0) = 1 + x + \frac{1}{2!}x = 1 + x + \frac{x^2}{2} \\
f^{(3)}(x) = e^x & f^{(3)}(x_0) = e^0 = 1 & P_3(x;0) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6}
\end{array}
$$

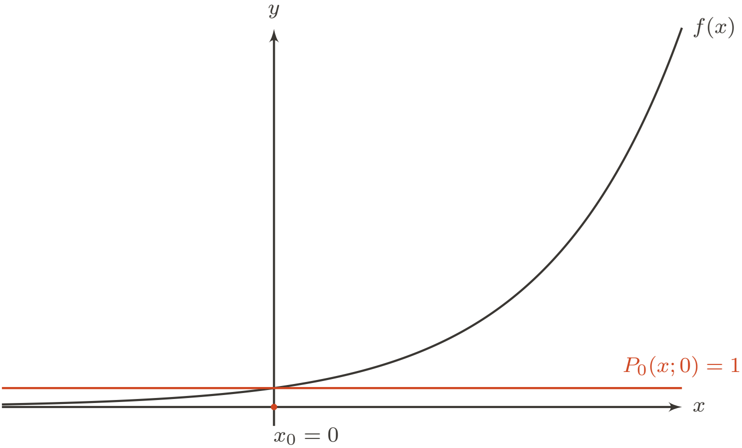
Wielomiany Taylora – przykład

$$f(x) = e^x, \qquad x_0 = 0.$$



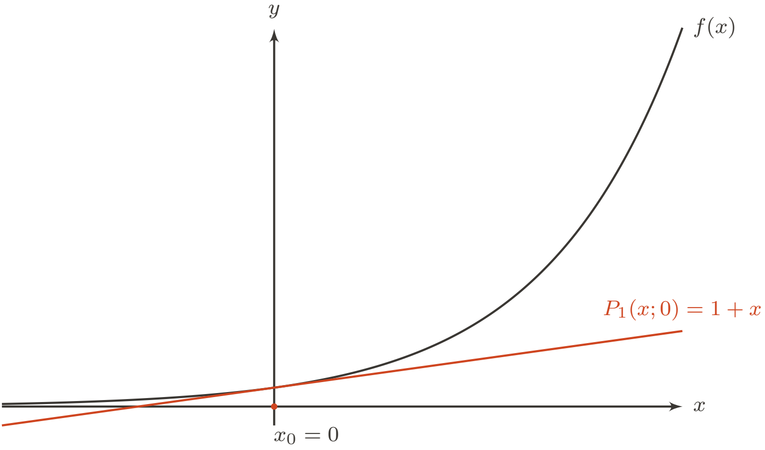
Wielomiany Taylora – przykład

$$f(x) = e^x, \qquad x_0 = 0.$$



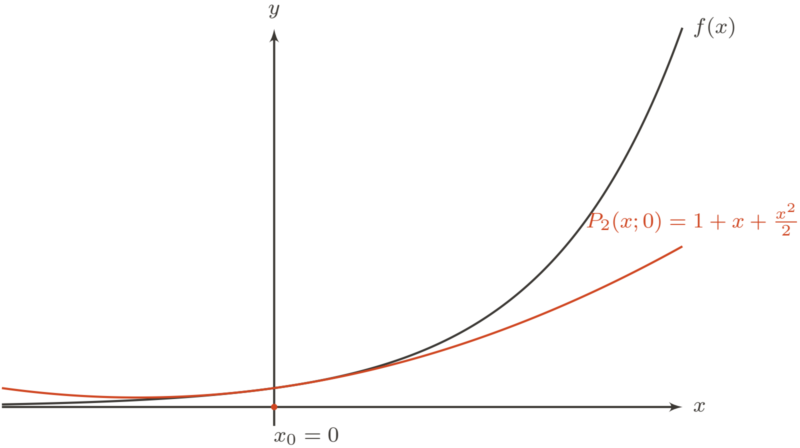
Wielomiany Taylora – przykład

$$f(x) = e^x, \qquad x_0 = 0.$$



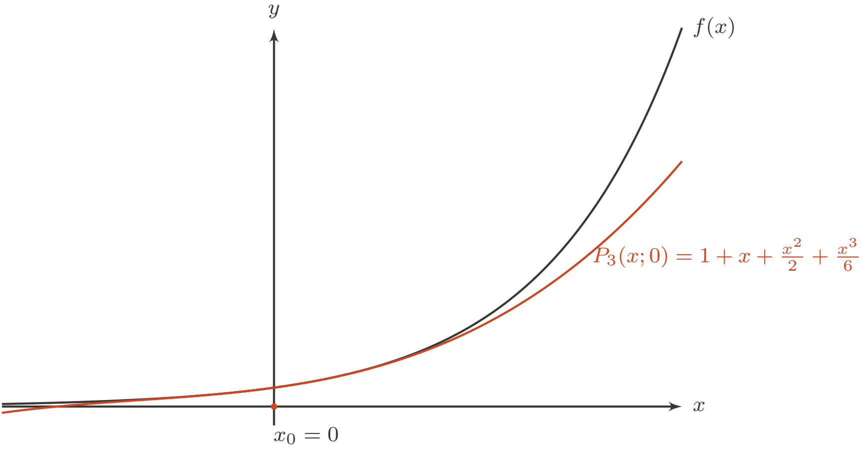
Wielomiany Taylora – przykład

$$f(x) = e^x, \qquad x_0 = 0.$$



Wielomiany Taylora – przykład

$$f(x) = e^x, \qquad x_0 = 0.$$



Założenia metody Newtona

Warunek wystarczający na minimum: Jeśli $f'(x^*) = 0$ i $f''(x^*) > 0$ w punkcie $x^* \in \mathcal{X}$, to funkcja $f(x)$ ma **minimum lokalne** w x^*.

Założenia metody Newtona

Warunek wystarczający na minimum: Jeśli $f'(x^*) = 0$ i $f''(x^*) > 0$ w punkcie $x^* \in \mathcal{X}$, to funkcja $f(x)$ ma **minimum lokalne** w x^*.

Wykorzystując tę własność możemy rozsądnie założyć, że $f''(x^*) > 0$ w punkcie minimum x^* funkcji $f(x)$

Założenia metody Newtona

Warunek wystarczający na minimum: Jeśli $f'(x^*) = 0$ i $f''(x^*) > 0$ w punkcie $x^* \in \mathcal{X}$, to funkcja $f(x)$ ma **minimum lokalne** w x^*.

Wykorzystując tę własność możemy rozsądnie założyć, że $f''(x^*) > 0$ w punkcie minimum x^* funkcji $f(x)$

Przyjmując, że druga pochodna jest **ciągła**, zakładamy dodatkowo, że $f''(x) > 0$ w pewnym **otoczeniu** punktu x^*.

Metoda Newtona

- Metoda Newtona to iteracyjny algorytm minimalizacji **bez ograniczeń** funkcji $f(x)$ o ciągłej pierwszej i drugiej pochodnej

Metoda Newtona

- Metoda Newtona to iteracyjny algorytm minimalizacji **bez ograniczeń** funkcji $f(x)$ o ciągłej pierwszej i drugiej pochodnej
- Wynikiem metody jest ciąg punktów $x_1, x_2, \ldots$ zbiegających do minimum lokalnego x^* funkcji $f(x)$

Metoda Newtona

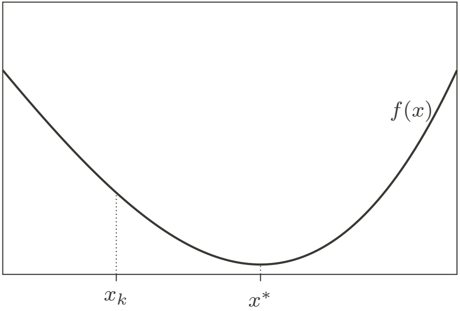
- Metoda Newtona to iteracyjny algorytm minimalizacji **bez ograniczeń** funkcji $f(x)$ o ciągłej pierwszej i drugiej pochodnej
- Wynikiem metody jest ciąg punktów $x_1, x_2, \ldots$ zbiegających do minimum lokalnego x^* funkcji $f(x)$
- W każdej iteracji $k = 1, 2, \ldots$ **przybliżamy** funkcję $f(x)$ **wielomianem Taylora drugiego rzędu** wokół aktualnego punktu x_k:

$$f(x) \simeq P_2(x; x_k) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2} f''(x_k)(x - x_k)^2$$

Metoda Newtona

- Metoda Newtona to iteracyjny algorytm minimalizacji **bez ograniczeń** funkcji $f(x)$ o ciągłej pierwszej i drugiej pochodnej
- Wynikiem metody jest ciąg punktów $x_1, x_2, \ldots$ zbiegających do minimum lokalnego x^* funkcji $f(x)$
- W każdej iteracji $k = 1, 2, \ldots$ **przybliżamy** funkcję $f(x)$ **wielomianem Taylora drugiego rzędu** wokół aktualnego punktu x_k:

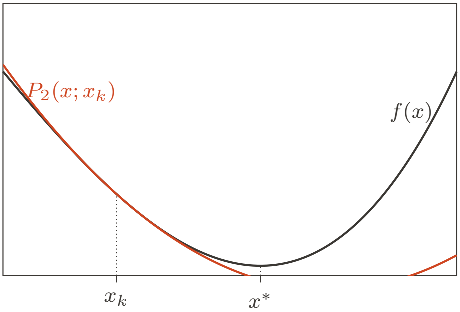
$$f(x) \simeq P_2(x; x_k) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2} f''(x_k)(x - x_k)^2$$



Metoda Newtona

- Metoda Newtona to iteracyjny algorytm minimalizacji **bez ograniczeń** funkcji $f(x)$ o ciągłej pierwszej i drugiej pochodnej
- Wynikiem metody jest ciąg punktów $x_1, x_2, \ldots$ zbiegających do minimum lokalnego x^* funkcji $f(x)$
- W każdej iteracji $k = 1, 2, \ldots$ **przybliżamy** funkcję $f(x)$ **wielomianem Taylora drugiego rzędu** wokół aktualnego punktu x_k:

$$f(x) \simeq P_2(x; x_k) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2} f''(x_k)(x - x_k)^2$$



Przybliżenie funkcji $f(x)$ **funkcją kwadratową** wokół x_k

Powód: Minimum funkcji kwadratowej umiemy znaleźć **analitycznie**!

Metoda Newtona

Z założenia $f''(x) > 0$ w otoczeniu x^*, stąd dla x_k bliskiego x^*

$$P_2(x; x_k) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}\underbrace{f''(x_k)}_{>0}(x - x_k)^2$$

jest **parabolą wypukłą** posiadającą minimum

Metoda Newtona

Z założenia $f''(x) > 0$ w otoczeniu x^*, stąd dla x_k bliskiego x^*

$$P_2(x; x_k) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}\underbrace{f''(x_k)}_{>0}(x - x_k)^2$$

jest **parabolą wypukłą** posiadającą minimum

Kolejny punkt x_{k+1} przyjmujemy więc jako **minimum** $P_2(x; x_k)$ uzyskane poprzez przyrównanie pochodnej $P_2(x; x_k)$ do zera:

$$P_2'(x; x_k) = 0 \qquad \Longleftrightarrow \qquad f'(x_k) + f''(x_k)(x - x_k) = 0$$

Metoda Newtona

Z założenia $f''(x) > 0$ w otoczeniu x^*, stąd dla x_k bliskiego x^*

$$P_2(x;x_k) = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}\underbrace{f''(x_k)}_{>0}(x - x_k)^2$$

jest **parabolą wypukłą** posiadającą minimum

Kolejny punkt x_{k+1} przyjmujemy więc jako **minimum** $P_2(x;x_k)$ uzyskane poprzez przyrównanie pochodnej $P_2(x;x_k)$ do zera:

$$P_2'(x;x_k) = 0 \qquad \Longleftrightarrow \qquad f'(x_k) + f''(x_k)(x - x_k) = 0$$

$$\boxed{x_{k+1} \;=\; x_k - \frac{f'(x_k)}{f''(x_k)}}$$

Metoda Newtona

Wejście: procedura wyznaczająca wartości $f'(x)$ i $f''(x)$ w dowolnym punkcie dziedziny; liczba iteracji n;

Inicjalizuj: x_1 odpowiednio blisko minimum x^*

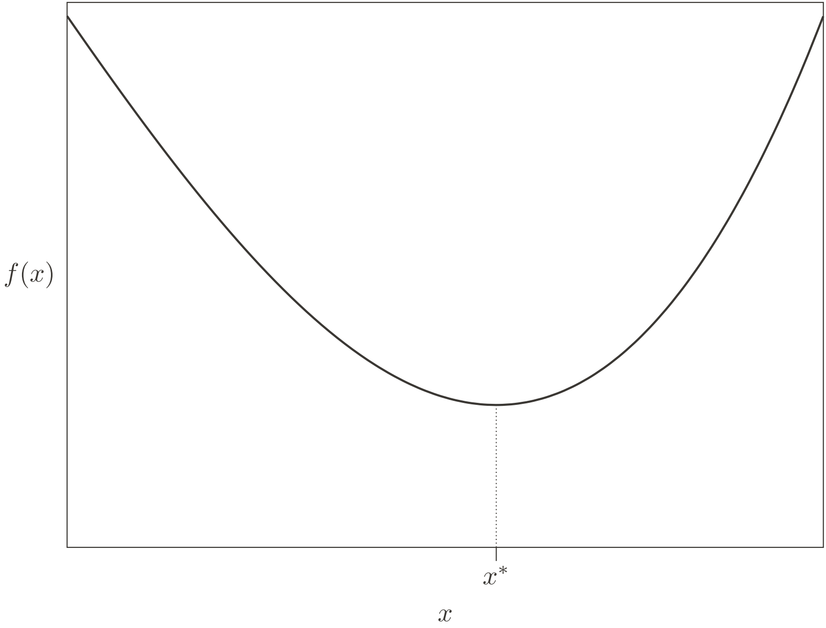
Powtarzaj dla $k = 1, 2, \ldots, n$:

Wyznacz pierwszą i drugą pochodną w x_k: $f'(x_k)$, $f''(x_k)$

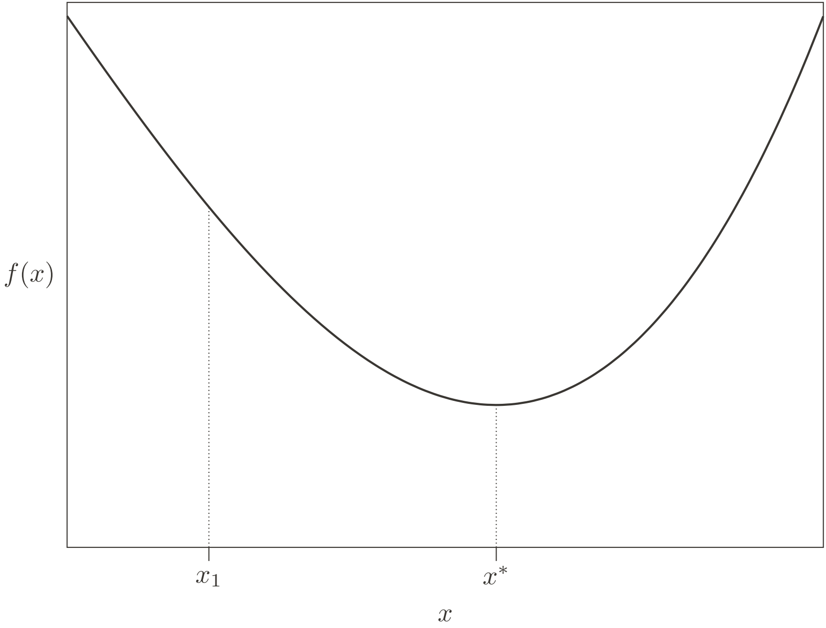
Przypisz $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$

Zwróć wartość x_{n+1}

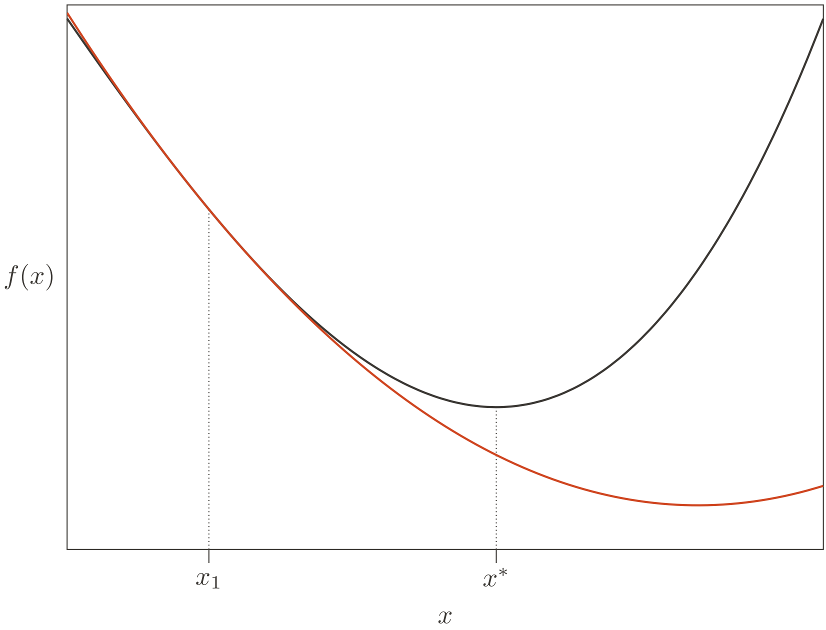
Metoda Newtona – przykład



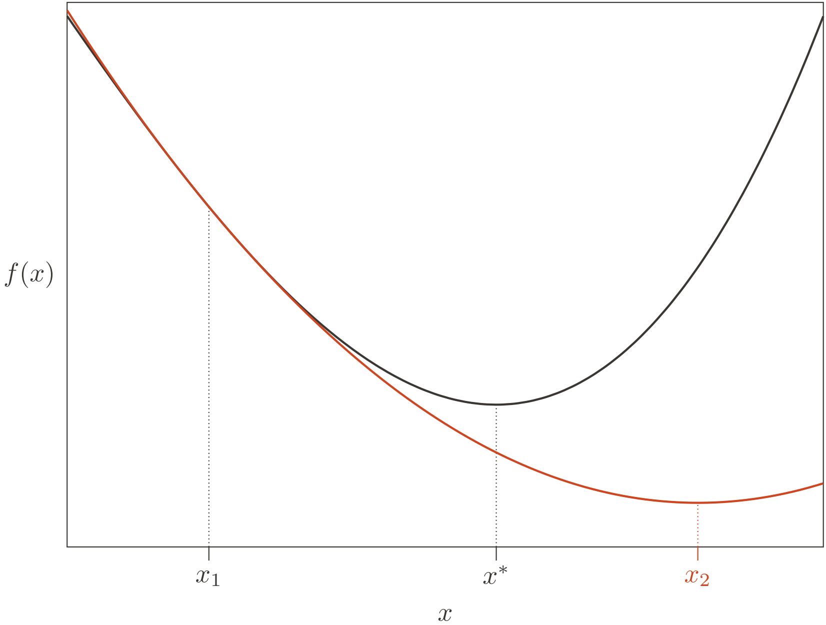
Metoda Newtona – przykład



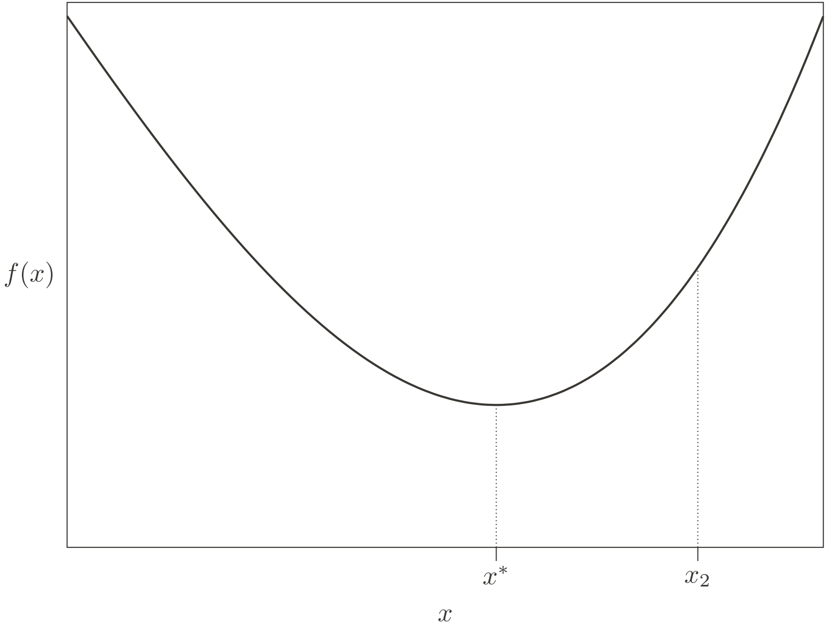
Metoda Newtona – przykład



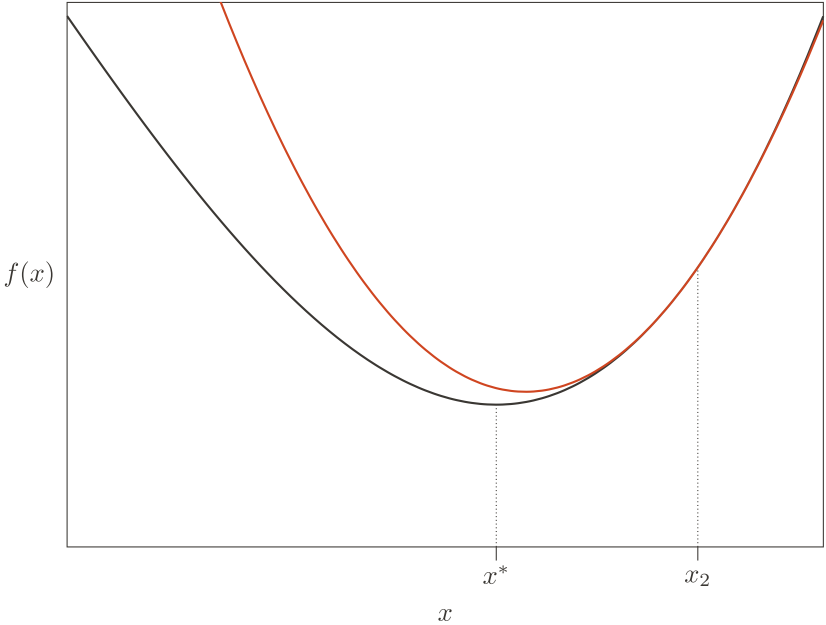
Metoda Newtona – przykład



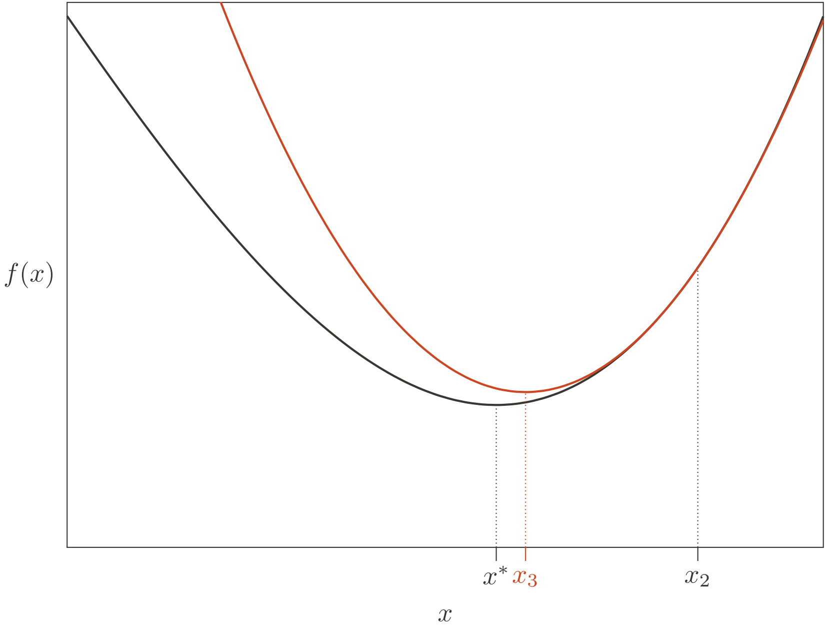
Metoda Newtona – przykład



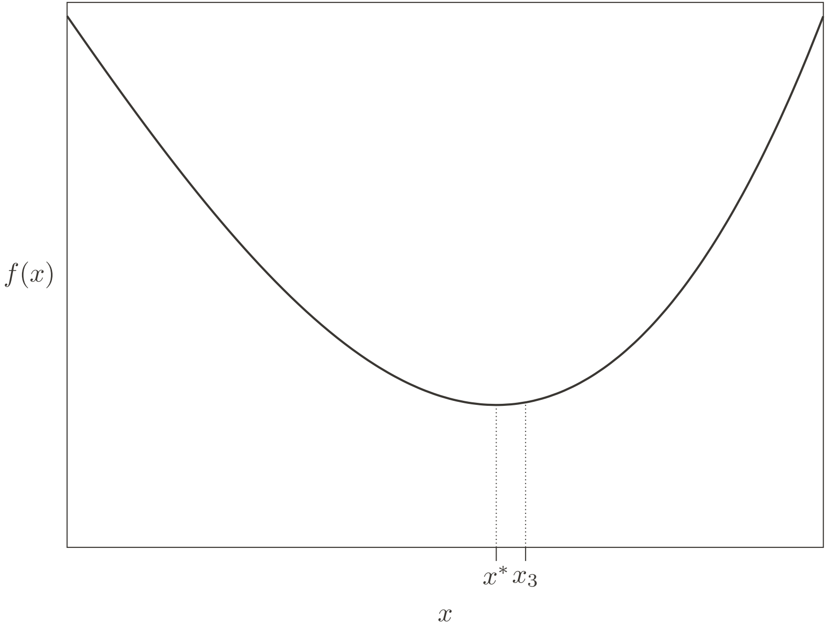
Metoda Newtona – przykład



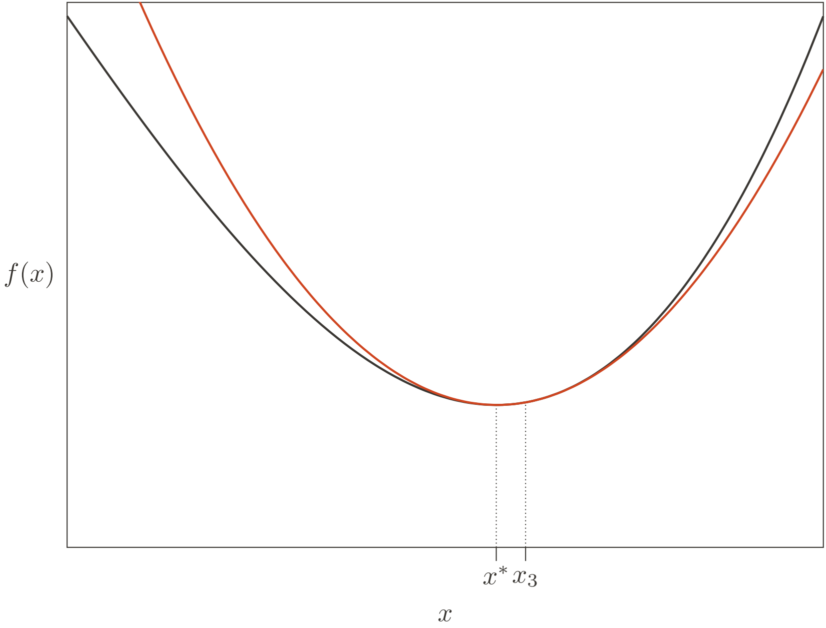
Metoda Newtona – przykład



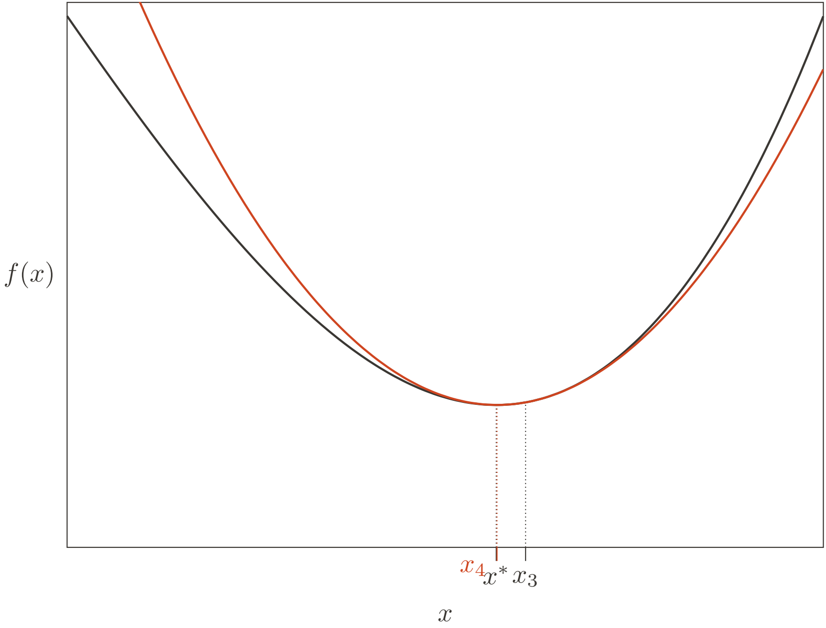
Metoda Newtona – przykład



Metoda Newtona – przykład



Metoda Newtona – przykład



Metoda Newtona – przykład

Szukamy minimum funkcji $f(x) = e^x + e^{-x}$

$$f'(x) = e^x - e^{-x}, \qquad f''(x) = e^x + e^{-x}$$

Łatwo wyznaczyć minimum globalne analitycznie: $x^* = 0$.

Metoda Newtona – przykład

Szukamy minimum funkcji $f(x) = e^x + e^{-x}$

$$f'(x) = e^x - e^{-x}, \qquad f''(x) = e^x + e^{-x}$$

Łatwo wyznaczyć minimum globalne analitycznie: $x^* = 0$.

Rozpoczynamy od rozwiązania $x_1 = 1$

Metoda Newtona – przykład

Szukamy minimum funkcji $f(x) = e^x + e^{-x}$

$$f'(x) = e^x - e^{-x}, \qquad f''(x) = e^x + e^{-x}$$

Łatwo wyznaczyć minimum globalne analitycznie: $x^* = 0$.

Rozpoczynamy od rozwiązania $x_1 = 1$

k	x_k	$\lvert x_k - x^* \rvert$
1	1	1
2	0.238	0.238
3	0.004	0.004
4	$2.9 \cdot 10^{-8}$	$2.9 \cdot 10^{-8}$
5	$-2.51 \cdot 10^{-17}$	$2.51 \cdot 10^{-17}$

Metoda Newtona – przykład

Szukamy minimum funkcji $f(x) = e^x + e^{-x}$

$$f'(x) = e^x - e^{-x}, \qquad f''(x) = e^x + e^{-x}$$

Łatwo wyznaczyć minimum globalne analitycznie: $x^* = 0$.

Rozpoczynamy od rozwiązania $x_1 = 10$

Metoda Newtona – przykład

Szukamy minimum funkcji $f(x) = e^x + e^{-x}$

$$f'(x) = e^x - e^{-x}, \qquad f''(x) = e^x + e^{-x}$$

Łatwo wyznaczyć minimum globalne analitycznie: $x^* = 0$.

Rozpoczynamy od rozwiązania $x_1 = 10$

k	x_k	$\lvert x_k - x^* \rvert$
1	10	10
2	9	9
3	8	8
4	7	7
5	6	6
6	5	5
7	4	4
8	3	3
9	2.005	2.005
10	1.041	1.041
11	0.263	0.263
12	0.006	0.006
13	$6.8 \cdot 10^{-8}$	$6.8 \cdot 10^{-8}$
14	$1.5 \cdot 10^{-17}$	$1.5 \cdot 10^{-17}$

Metoda Newtona – przykład

Szukamy minimum funkcji $f(x) = x^2 + 2x + 1$

$$f'(x) = 2x + 2 \qquad f''(x) = 2$$

Łatwo wyznaczyć minimum globalne analitycznie: $x^* = -1$.

Metoda Newtona – przykład

Szukamy minimum funkcji $f(x) = x^2 + 2x + 1$

$$f'(x) = 2x + 2 \qquad f''(x) = 2$$

Łatwo wyznaczyć minimum globalne analitycznie: $x^* = -1$.

Rozpoczynamy od rozwiązania $x_1 = 5$

k	x_k	$\lvert x_k - x^* \rvert$
1	5	6
2	-1	0

Metoda Newtona – przykład

Szukamy minimum funkcji $f(x) = x^2 + 2x + 1$

$$f'(x) = 2x + 2 \qquad f''(x) = 2$$

Łatwo wyznaczyć minimum globalne analitycznie: $x^* = -1$.

Rozpoczynamy od rozwiązania $x_1 = 5$

k	x_k	$\lvert x_k - x^* \rvert$
1	5	6
2	-1	0

Metoda Newtona znajduje minimum **funkcji kwadratowej** dokładnie w **jednym kroku**!

Dlaczego?

Metoda Newtona – przykład

Szukamy minimum funkcji $f(x) = x^2 + 2x + 1$

$$f'(x) = 2x + 2 \qquad f''(x) = 2$$

Łatwo wyznaczyć minimum globalne analitycznie: $x^* = -1$.

Rozpoczynamy od rozwiązania $x_1 = 5$

k	x_k	$\|x_k - x^*\|$
1	5	6
2	-1	0

Metoda Newtona znajduje minimum **funkcji kwadratowej** dokładnie w **jednym kroku**!

Dlaczego? Ponieważ przybliżenie kwadratowe jest w tym wypadku tożsame z funkcją $f(x)$!

Metoda Newtona – przykład

Szukamy minimum funkcji $f(x) = x - \log(x)$

$$f'(x) = 1 - \frac{1}{x}, \qquad f''(x) = \frac{1}{x^2}$$

Łatwo wyznaczyć minimum globalne analitycznie: $x^* = 1$.

Metoda Newtona – przykład

Szukamy minimum funkcji $f(x) = x - \log(x)$

$$f'(x) = 1 - \frac{1}{x}, \qquad f''(x) = \frac{1}{x^2}$$

Łatwo wyznaczyć minimum globalne analitycznie: $x^* = 1$.

Rozpoczynamy od rozwiązania $x_1 = 1.5$

Metoda Newtona – przykład

Szukamy minimum funkcji $f(x) = x - \log(x)$

$$f'(x) = 1 - \frac{1}{x}, \qquad f''(x) = \frac{1}{x^2}$$

Łatwo wyznaczyć minimum globalne analitycznie: $x^* = 1$.

Rozpoczynamy od rozwiązania $x_1 = 1.5$

k	x_k	$\lvert x_k - x^* \rvert$
1	1.5	0.5
2	0.75	0.25
3	0.9375	0.0625
4	0.996	0.004
5	0.999	$1.53 \cdot 10^{-5}$
6	0.999	$2.33 \cdot 10^{-10}$

Metoda Newtona – przykład

Szukamy minimum funkcji $f(x) = x - \log(x)$

$$f'(x) = 1 - \frac{1}{x}, \qquad f''(x) = \frac{1}{x^2}$$

Łatwo wyznaczyć minimum globalne analitycznie: $x^* = 1$.

Rozpoczynamy od rozwiązania $x_1 = 3$

Metoda Newtona – przykład

Szukamy minimum funkcji $f(x) = x - \log(x)$

$$f'(x) = 1 - \frac{1}{x}, \qquad f''(x) = \frac{1}{x^2}$$

Łatwo wyznaczyć minimum globalne analitycznie: $x^* = 1$.

Rozpoczynamy od rozwiązania $x_1 = 3$

k	x_k	$\lvert x_k - x^* \rvert$
1	3	2
2	-3	4
3	-4	5
4	-15	16
5	-16	17
6	-255	256
...	...	...
10	-4294967295	4294967296

Metoda **rozbiega się** do (minus) nieskończoności!

Metoda Newtona

- Wymaga dostępu do pierwszej i drugiej pochodnej
- Nie radzi sobie z ograniczeniami dziedziny
- W ogólności algorytm trzeba rozpocząć **dostatecznie blisko** optimum, w przeciwnym wypadku może się nawet rozbiec!
- Rozpoczęty blisko optimum, algorytm zbiega **błyskawicznie**, zwykle wystarczy kilka iteracji! (z tego względu zwykle preferowany w praktyce)

Zbieżność metody Newtona

Przypomnienie (wzór Taylora):

$$f(x) \ = \ P_m(x; x_0) + \frac{f^{(m+1)}(\xi)}{(m+1)!}(x - x_0)^{m+1}$$

gdzie ξ jest punktem leżącym między x a x_0. W szczególności dla $m = 1$:

$$f(x) \ = \ f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(\xi)(x - x_0)^2$$

Zbieżność metody Newtona

Przypomnienie (wzór Taylora):

$$f(x) \;=\; P_m(x;x_0) + \frac{f^{(m+1)}(\xi)}{(m+1)!}(x-x_0)^{m+1}$$

gdzie ξ jest punktem leżącym między x a x_0. W szczególności dla $m=1$:

$$f(x) \;=\; f(x_0) + f'(x_0)(x-x_0) + \frac{1}{2}f''(\xi)(x-x_0)^2$$

Wniosek: Stosując to twierdzenie do **pochodnej** $f'(x)$ (dla $m=1$) z $x = x^*$ i $x_0 = x_k$ mamy:

$$\underbrace{f'(x^*)}_{=0} \;=\; f'(x_k) + f''(x_k)(x^*-x_k) + \frac{1}{2}f'''(\xi)(x^*-x_k)^2,$$

(zakładamy, że funkcja jest trzykrotnie różniczkowalna)

Zbieżność metody Newtona

Przypomnienie (wzór Taylora):

$$f(x) \;=\; P_m(x;x_0) + \frac{f^{(m+1)}(\xi)}{(m+1)!}(x-x_0)^{m+1}$$

gdzie ξ jest punktem leżącym między x a x_0. W szczególności dla $m=1$:

$$f(x) \;=\; f(x_0) + f'(x_0)(x-x_0) + \frac{1}{2}f''(\xi)(x-x_0)^2$$

Wniosek: Stosując to twierdzenie do **pochodnej** $f'(x)$ (dla $m=1$) z $x=x^*$ i $x_0=x_k$ mamy:

$$\underbrace{f'(x^*)}_{=0} \;=\; f'(x_k) + f''(x_k)(x^*-x_k) + \frac{1}{2}f'''(\xi)(x^*-x_k)^2,$$

(zakładamy, że funkcja jest trzykrotnie różniczkowalna)

$$\text{co daje:}\quad -f'(x_k) \;=\; f''(x_k)(x^*-x_k) + \frac{1}{2}f'''(\xi)(x^*-x_k)^2$$

Zbieżność metody Newtona

$$\boxed{-f'(x_k) \; = \; f''(x_k)(x^* - x_k) + \frac{1}{2}f'''(\xi)(x^* - x_k)^2}$$

Z definicji metody Newtona mamy:

$$x_{k+1} - x^* \; = \; x_k - x^* - \frac{f'(x_k)}{f''(x_k)}$$

Zbieżność metody Newtona

$$\boxed{-f'(x_k) \;=\; f''(x_k)(x^* - x_k) + \frac{1}{2}f'''(\xi)(x^* - x_k)^2}$$

Z definicji metody Newtona mamy:

$$x_{k+1} - x^* \;=\; x_k - x^* - \frac{f'(x_k)}{f''(x_k)}$$

Używając wzoru w ramce:

$$x_{k+1} - x^* \quad = \quad x_k - x^* + \frac{f''(x_k)(x^* - x_k) + \frac{1}{2}f'''(\xi)(x^* - x_k)^2}{f''(x_k)}$$

Zbieżność metody Newtona

$$\boxed{-f'(x_k) \;=\; f''(x_k)(x^* - x_k) + \frac{1}{2}f'''(\xi)(x^* - x_k)^2}$$

Z definicji metody Newtona mamy:

$$x_{k+1} - x^* \;=\; x_k - x^* - \frac{f'(x_k)}{f''(x_k)}$$

Używając wzoru w ramce:

$$\begin{aligned} x_{k+1} - x^* &= x_k - x^* + \frac{f''(x_k)(x^* - x_k) + \frac{1}{2}f'''(\xi)(x^* - x_k)^2}{f''(x_k)} \\ &= \frac{1}{2}\frac{f'''(\xi)}{f''(x_k)}(x^* - x_k)^2 \end{aligned}$$

Zbieżność metody Newtona

$$\boxed{-f'(x_k) \;=\; f''(x_k)(x^* - x_k) + \frac{1}{2} f'''(\xi)(x^* - x_k)^2}$$

Z definicji metody Newtona mamy:

$$x_{k+1} - x^* \;=\; x_k - x^* - \frac{f'(x_k)}{f''(x_k)}$$

Używając wzoru w ramce:

$$\begin{aligned} x_{k+1} - x^* &= x_k - x^* + \frac{f''(x_k)(x^* - x_k) + \frac{1}{2} f'''(\xi)(x^* - x_k)^2}{f''(x_k)} \\ &= \frac{1}{2} \frac{f'''(\xi)}{f''(x_k)} (x^* - x_k)^2 \end{aligned}$$

Zbieżność **kwadratowa**:

$$|x_{k+1} - x^*| = \beta_k (x_k - x^*)^2, \qquad \beta_k = \left| \frac{1}{2} \frac{f'''(\xi)}{f''(x_k)} \right|$$

Rząd zbieżności

Niech $p \geqslant 1$ będzie maksymalną wartością, dla której istnieje granica:

$$\beta \;=\; \lim_{k\to\infty} \beta_k, \qquad \text{gdzie } \beta_k = \frac{|x_{k+1} - x^*|}{|x_k - x^*|^p}.$$

Wtedy p nazywamy **rzędem zbieżności** ciągu x_k do x^*.

Rząd zbieżności

Niech $p \geqslant 1$ będzie maksymalną wartością, dla której istnieje granica:

$$\beta \;=\; \lim_{k\to\infty} \beta_k, \qquad \text{gdzie } \beta_k = \frac{|x_{k+1} - x^*|}{|x_k - x^*|^p}.$$

Wtedy p nazywamy **rzędem zbieżności** ciągu x_k do x^*.

- Jeśli $p = 1$ i $\beta \in (0,1)$ to ciąg ma **zbieżność liniową** (przeszukiwanie dychotomiczne, złoty podział, bisekcja)
- Jeśli $p = 2$ to ciąg ma **zbieżność kwadratową** (metoda Newtona)

Porównanie rzędów zbieżności

Załóżmy dla uproszczenia, że zachodzi $|x_{k+1} - x^*| = \beta |x_k - x^*|^p$ dla wszystkich n. Porównamy rzędy zbieżności dla $\beta = 0.5$ i $p = 1, 2$, tzn:

$$|x_{k+1} - x^*| = 0.5 |x_k - x^*|, \qquad |x_{k+1} - x^*| = 0.5 |x_k - x^*|^2$$

Porównanie rzędów zbieżności

Załóżmy dla uproszczenia, że zachodzi $|x_{k+1} - x^*| = \beta |x_k - x^*|^p$ dla wszystkich n. Porównamy rzędy zbieżności dla $\beta = 0.5$ i $p = 1, 2$, tzn:

$$|x_{k+1} - x^*| = 0.5 |x_k - x^*|, \qquad |x_{k+1} - x^*| = 0.5 |x_k - x^*|^2$$

Rozpoczynamy od $|x_1 - x^*| = 1$.

k	wartość $\lvert x_k - x^*\rvert$	
	$p = 1$	$p = 2$
1	1	1
2	0.5	0.5
3	0.25	0.125
4	0.125	0.00781
5	0.0625	$3.05 \cdot 10^{-5}$
6	0.03125	$4.66 \cdot 10^{-10}$
7	0.01563	$1.08 \cdot 10^{-19}$

Porównanie rzędów zbieżności

Załóżmy dla uproszczenia, że zachodzi $|x_{k+1} - x^*| = \beta |x_k - x^*|^p$ dla wszystkich n. Porównamy rzędy zbieżności dla $\beta = 0.5$ i $p = 1, 2$, tzn:

$$|x_{k+1} - x^*| = 0.5|x_k - x^*|, \qquad |x_{k+1} - x^*| = 0.5|x_k - x^*|^2$$

Rozpoczynamy od $|x_1 - x^*| = 4$.

k	wartość $\lvert x_k - x^*\rvert$	
	$p = 1$	$p = 2$
1	2	2
2	1	8
3	0.5	512
4	0.25	131072
5	0.125	8589934592

Metoda siecznych

- Druga pochodna potrzebna w metodzie Newtona może być czasem trudna (lub wręcz niemożliwa) do pozyskania

Metoda siecznych

- Druga pochodna potrzebna w metodzie Newtona może być czasem trudna (lub wręcz niemożliwa) do pozyskania
- **Rozwiązanie**: Korzystając z definicji drugiej pochodnej:

$$f''(x) = \lim_{\Delta \to 0} \frac{f'(x + \Delta) - f'(x)}{\Delta}$$

przybliżamy:

$$f''(x_k) \simeq \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$$

Metoda siecznych

- Druga pochodna potrzebna w metodzie Newtona może być czasem trudna (lub wręcz niemożliwa) do pozyskania
- **Rozwiązanie**: Korzystając z definicji drugiej pochodnej:

$$f''(x) = \lim_{\Delta \to 0} \frac{f'(x+\Delta) - f'(x)}{\Delta}$$

  przybliżamy:

$$f''(x_k) \simeq \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$$

- W rezultacie dostajemy tzw. **metodę siecznych**:

$$x_{k+1} \;=\; x_k - \frac{f'(x_k)(x_k - x_{k-1})}{f'(x_k) - f'(x_{k-1})}$$

Metoda siecznych

- Druga pochodna potrzebna w metodzie Newtona może być czasem trudna (lub wręcz niemożliwa) do pozyskania
- **Rozwiązanie**: Korzystając z definicji drugiej pochodnej:

$$f''(x) = \lim_{\Delta \to 0} \frac{f'(x+\Delta) - f'(x)}{\Delta}$$

  przybliżamy:

$$f''(x_k) \simeq \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$$

- W rezultacie dostajemy tzw. **metodę siecznych**:

$$x_{k+1} \; = \; x_k - \frac{f'(x_k)(x_k - x_{k-1})}{f'(x_k) - f'(x_{k-1})}$$

- Można pokazać, że metoda siecznych ma rząd zbieżności $\beta = \frac{1+\sqrt{5}}{2} \simeq 1.618$

Kryteria stopu algorytmów optymalizacji

Poznane algorytmy mogą w zasadzie działać nieskończenie długo, coraz lepiej przybliżając optimum, ale nigdy go dokładnie nie osiągając.

Potrzebne jest więc dodatkowe **kryterium stopu**

Kryteria stopu algorytmów optymalizacji

Poznane algorytmy mogą w zasadzie działać nieskończenie długo, coraz lepiej przybliżając optimum, ale nigdy go dokładnie nie osiągając.

Potrzebne jest więc dodatkowe **kryterium stopu**

Dla uproszczenia dotychczas stosowaliśmy kryterium **czasu obliczeń**, w którym z góry zadana jest liczba iteracji n.

Kryteria stopu algorytmów optymalizacji

Poznane algorytmy mogą w zasadzie działać nieskończenie długo, coraz lepiej przybliżając optimum, ale nigdy go dokładnie nie osiągając.

Potrzebne jest więc dodatkowe **kryterium stopu**

Dla uproszczenia dotychczas stosowaliśmy kryterium **czasu obliczeń**, w którym z góry zadana jest liczba iteracji n.

Alternatywnie, można rozważać kryteria **jakości przybliżenia** x_k wyrażone za pomocą numerycznej stałej ϵ (np. $\epsilon = 10^{-8}$), np. zatrzymaj algorytm:

- Jeśli $|x_k - x^*| \leqslant \epsilon$ (co otrzymujemy z analizy teoretycznej)

Kryteria stopu algorytmów optymalizacji

Poznane algorytmy mogą w zasadzie działać nieskończenie długo, coraz lepiej przybliżając optimum, ale nigdy go dokładnie nie osiągając.

Potrzebne jest więc dodatkowe **kryterium stopu**

Dla uproszczenia dotychczas stosowaliśmy kryterium **czasu obliczeń**, w którym z góry zadana jest liczba iteracji n.

Alternatywnie, można rozważać kryteria **jakości przybliżenia** x_k wyrażone za pomocą numerycznej stałej ϵ (np. $\epsilon = 10^{-8}$), np. zatrzymaj algorytm:

- Jeśli $|x_k - x^*| \leqslant \epsilon$ (co otrzymujemy z analizy teoretycznej)
- Jeśli $|f'(x_k)| \leqslant \epsilon$ (korzystając z faktu, że $f'(x^*) = 0$)

Kryteria stopu algorytmów optymalizacji

Poznane algorytmy mogą w zasadzie działać nieskończenie długo, coraz lepiej przybliżając optimum, ale nigdy go dokładnie nie osiągając.

Potrzebne jest więc dodatkowe **kryterium stopu**

Dla uproszczenia dotychczas stosowaliśmy kryterium **czasu obliczeń**, w którym z góry zadana jest liczba iteracji n.

Alternatywnie, można rozważać kryteria **jakości przybliżenia** x_k wyrażone za pomocą numerycznej stałej ϵ (np. $\epsilon = 10^{-8}$), np. zatrzymaj algorytm:

- Jeśli $|x_k - x^*| \leqslant \epsilon$ (co otrzymujemy z analizy teoretycznej)
- Jeśli $|f'(x_k)| \leqslant \epsilon$ (korzystając z faktu, że $f'(x^*) = 0$)
- Jeśli $\frac{|x_k - x_{k-1}|}{|x_{k-1}|} < \epsilon$ (relatywna zmiana rozwiązania niewielka)

Kryteria stopu algorytmów optymalizacji

Poznane algorytmy mogą w zasadzie działać nieskończenie długo, coraz lepiej przybliżając optimum, ale nigdy go dokładnie nie osiągając.

Potrzebne jest więc dodatkowe **kryterium stopu**

Dla uproszczenia dotychczas stosowaliśmy kryterium **czasu obliczeń**, w którym z góry zadana jest liczba iteracji n.

Alternatywnie, można rozważać kryteria **jakości przybliżenia** x_k wyrażone za pomocą numerycznej stałej ϵ (np. $\epsilon = 10^{-8}$), np. zatrzymaj algorytm:

- Jeśli $|x_k - x^*| \leqslant \epsilon$ (co otrzymujemy z analizy teoretycznej)
- Jeśli $|f'(x_k)| \leqslant \epsilon$ (korzystając z faktu, że $f'(x^*) = 0$)
- Jeśli $\frac{|x_k - x_{k-1}|}{|x_{k-1}|} < \epsilon$ (relatywna zmiana rozwiązania niewielka)
- Jeśli $\frac{|f(x_k) - f(x_{k-1})|}{|f(x_{k-1})|} < \epsilon$ (relatywna zmiana jakości rozwiązania niewielka)