



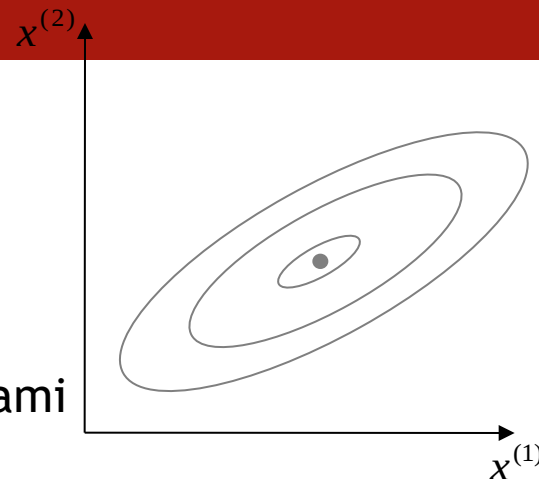
Politechnika Wrocławska

Numeryczne metody optymalizacji Optymalizacja w kierunku

informacje dodatkowe

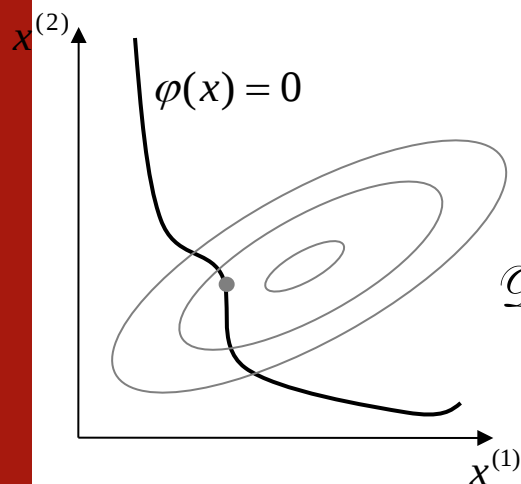
Typowe zadania decyzyjne

Zadanie decyzyjne bez ograniczeń: $\mathcal{D}_x = \mathcal{R}^S$



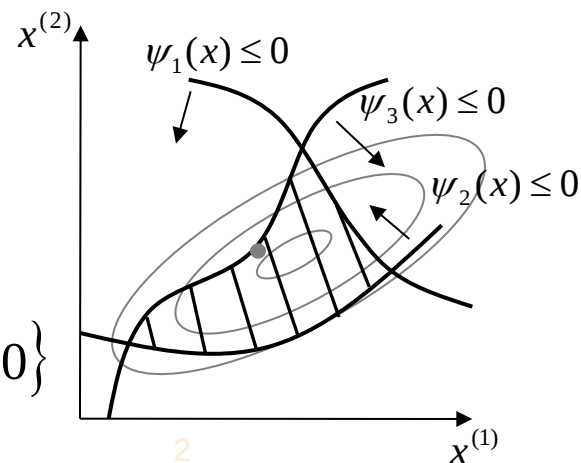
Zadanie decyzyjne z ograniczeniami równościowymi:

$$\mathcal{D}_x = \left\{ x \in \mathcal{R}^S : \varphi_1(x) = 0, \varphi_2(x) = 0, \dots, \varphi_L(x) = 0, L \leq S \right\}$$



Zadanie decyzyjne z ograniczeniami nierównościami:

$$\mathcal{D}_x = \left\{ x \in \mathcal{R}^S : \psi_1(x) \leq 0, \psi_2(x) \leq 0, \dots, \psi_M(x) \leq 0 \right\}$$



Zadanie optymalizacji bez ograniczeń

Zadanie optymalizacji: $x^* \rightarrow F(x^*) = \min_{x \in \mathcal{D}_x = \mathcal{R}^S} F(x)$

Założenie: $F(x)$ jest funkcją ciągłą i różniczkowalną.

Warunkiem koniecznym aby x^* było minimum lokalnym jest:

$$\nabla_x F(x) \Big|_{x^*} = 0_S$$

Jeżeli $F(x)$ jest funkcją pseudo - wypukłą, powyższe równanie jest warunkiem koniecznym i wystarczającym aby x^* było minimum globalnym

Jeżeli $H(x)$ jest dodatnio pół określona $\forall x \in \mathcal{R}^S$ to rozwiązanie powyższego równania jest minimum globalnym

Jeżeli $H(x)$ jest dodatnio określona $\forall x \in \mathcal{R}^S$ powyższe równanie ma jedno rozwiązanie x^* i jest ono minimum globalnym

Zadanie optymalizacji z ograniczeniami równościowymi

Metoda mnożników Lagrange' a

Funkcja Lagrange'a:

$$L(x, \lambda) = F(x) + \sum_{l=1}^L \lambda_l \varphi_l(x) = F(x) + \lambda^T \varphi(x)$$

gdzie: $\varphi(x) = \begin{bmatrix} \varphi_1(x) \\ \varphi_2(x) \\ \vdots \\ \varphi_L(x) \end{bmatrix}, \lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_L \end{bmatrix}$ - wektor współczynników Lagrange' a

Warunki konieczne optymalności:

$$\nabla_x L(x, \lambda) \Big|_{x^*, \lambda^*} = 0_S$$

$$\nabla_\lambda L(x, \lambda) \Big|_{x^*, \lambda^*} = 0_L \Leftrightarrow \text{rank } G(x) = \text{rank} \begin{bmatrix} G(x) & \vdots & -\nabla_x F(x) \end{bmatrix},$$

$$G(x) = \begin{bmatrix} \nabla_x \varphi_1(x) & \vdots & \nabla_x \varphi_2(x) & \vdots & \dots & \vdots & \nabla_x \varphi_L(x) \end{bmatrix}$$



Zadanie optymalizacji z ograniczeniami nierównościami Metoda Kuhna - Tucker'a

Funkcja Lagrange' a:

$$L(x, \mu) = F(x) + \mu^T \psi(x) \Leftrightarrow L(x, \mu) = F(x) + \sum_{m=1}^M \mu_m \psi_m(x)$$

Warunki konieczne optymalności:

$$\nabla_x L(x, \mu) \Big|_{x^*, \mu^*} = 0_S$$

$$\mu^T \nabla_\mu L(x, \mu) \Big|_{x^*, \mu^*} = 0$$

$$\nabla_\mu L(x, \mu) \Big|_{x^*, \mu^*} \leq 0_M$$

$$\mu^* \geq 0_M$$

$\Leftrightarrow$ gdy rozwiązanie jest regularne

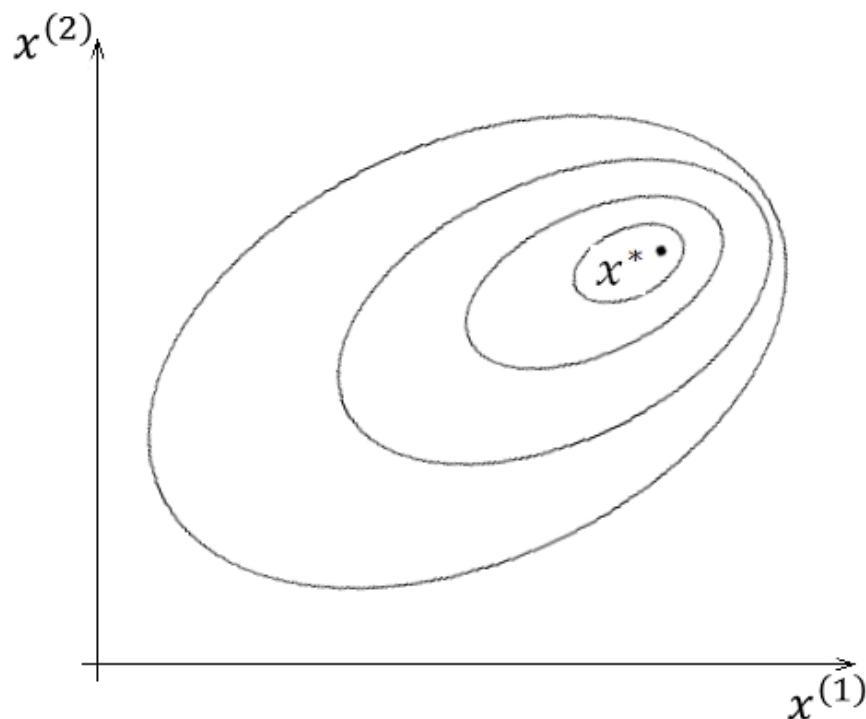
$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_s \end{bmatrix} \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_s \end{bmatrix}$$

$$\alpha \leq \beta \Rightarrow \forall_{s=1, \dots, S} \alpha_s \leq \beta_s$$

gdzie: $\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_M \end{bmatrix}$ - wektor współczynników Lagrange' a

Numeryczne metody optymalizacji

$$x^* \rightarrow F(x^*) = \min_{x \in D_x} F(x)$$



Problemy analityczne:

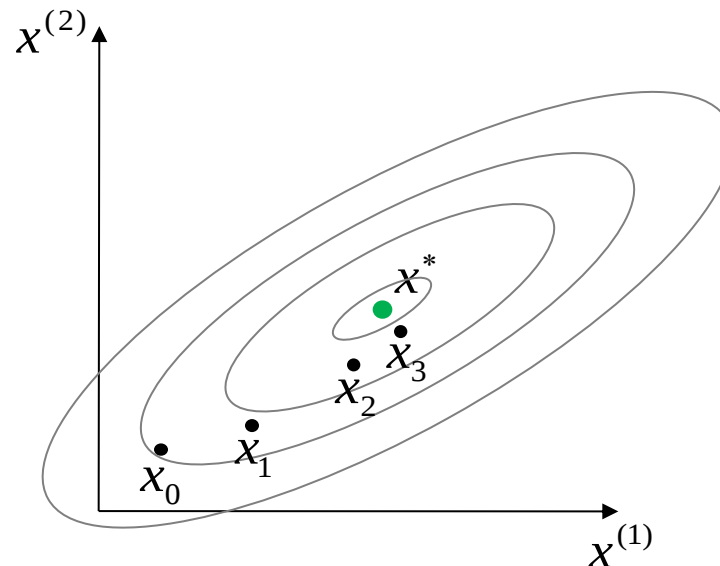
1. Nieliniowa złożona funkcja celu F i ograniczeń φ oraz ψ .
2. Nieróżniczkowalność funkcji F , φ oraz ψ .
3. Nie jest znana postać analityczna funkcji F , φ oraz ψ , można jedynie „zmierzyć” wartość funkcji
4. Duży wymiar wektora zmiennych decyzyjnych.

Metody numeryczne

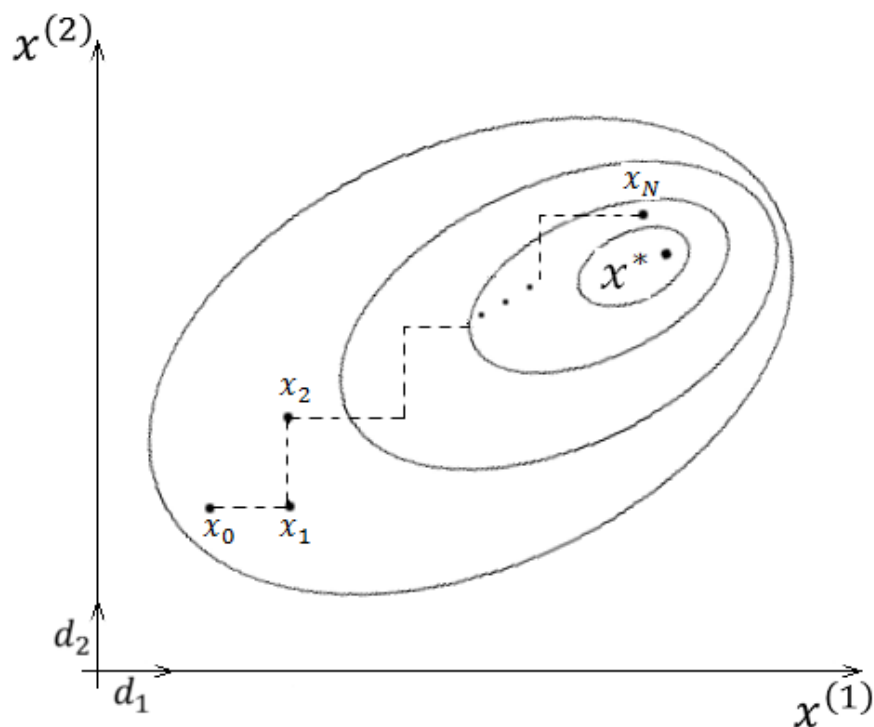
$$x^* \rightarrow F(x^*) = \min_{x^* \in \mathcal{D}_x} F(x),$$

Konstruujemy ciąg przybliżeń na podstawie wartości funkcji $F(x)$ w danym punkcie x

$$\begin{array}{ccccccc} x_0 & \xrightarrow{\quad} & x_1 & \xrightarrow{\quad} & x_2 & \xrightarrow{\quad} & \dots \xrightarrow{\quad} x_N \approx x^* \\ \downarrow & \nearrow & \downarrow & \nearrow & \downarrow & \nearrow & \downarrow \\ F(x_0) & > & F(x_1) & > & F(x_2) & > & \dots > F(x_N) \approx F(x^*) \end{array}$$



Numeryczne metody optymalizacji



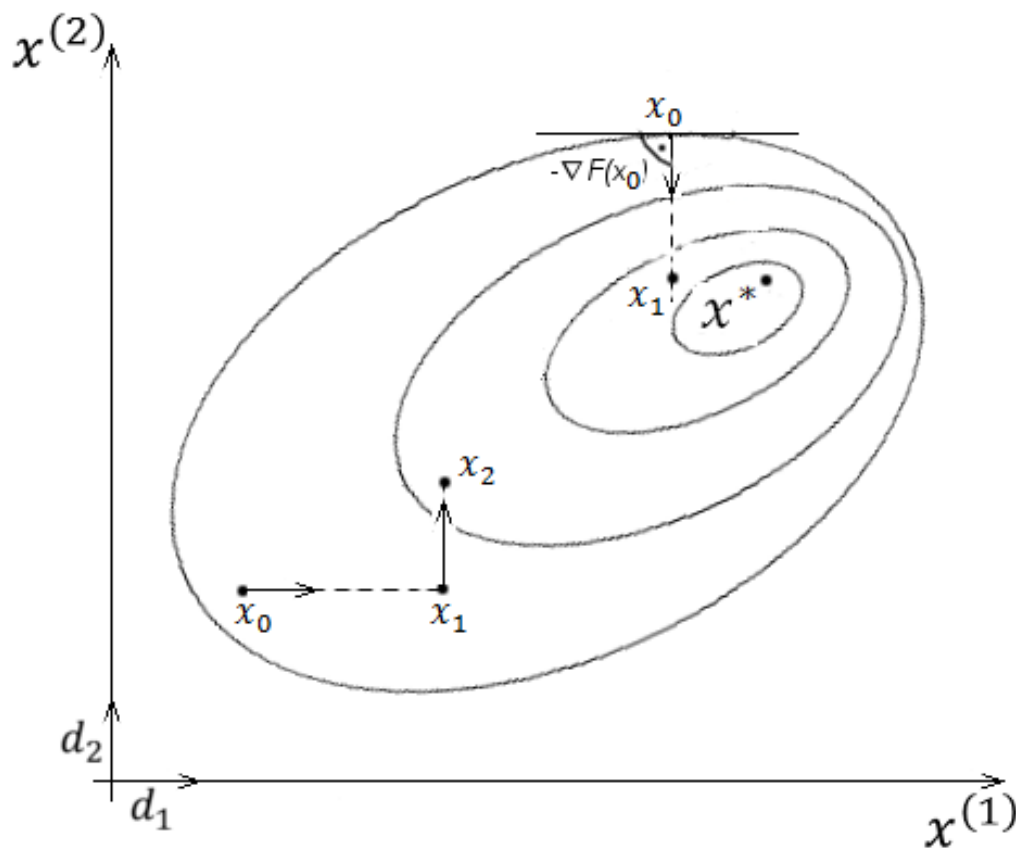
Algorytm

$$x_{n+1} = \Psi(x_n), x_0$$

- Wybór kierunku poszukiwań.
- Optymalizacja w kierunku.
- Warunki zatrzymania procedury.

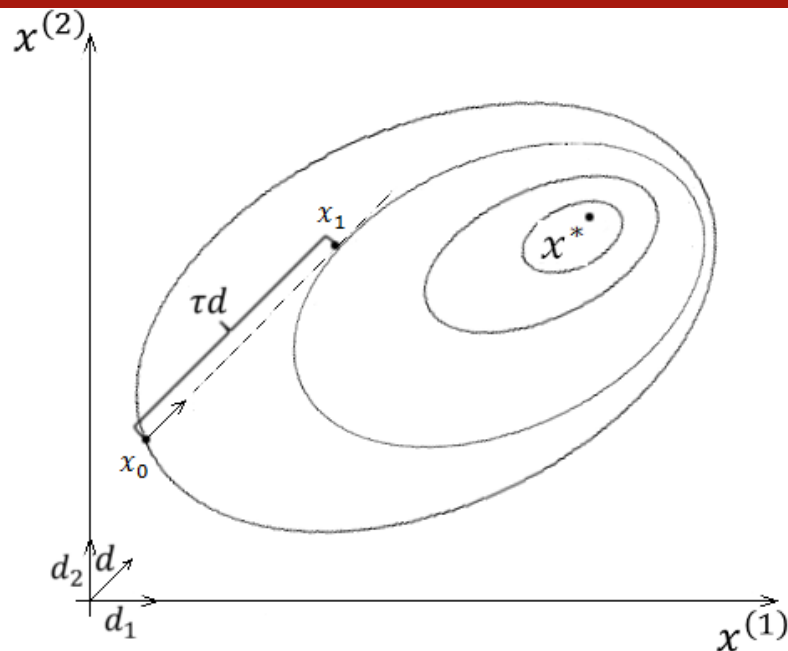
$$x_0, x_1, \dots, x_n, \dots, x_N \approx x^*$$
$$F(x_0) > F(x_1) > \dots > F(x_n) > \dots > F(x_N) \approx F(x^*)$$

Wybór kierunku poszukiwań



- Kierunki bazowe i ich modyfikacje – metody bezgradientowe.
- Kierunki oparte na gradiencie funkcji – metody gradientowe.

Optymalizacja w kierunku



x_0 - punkt początkowy
 x_1 - punkt końcowy
 d - kierunek
 τ - długość kroku w kierunku

$$\tau^* \rightarrow F(x_0 + \tau^* d) = \min_{\tau} F(x_0 + \tau d)$$

$$x_0, d - \text{ustalone } F(x_0 + \tau d) \triangleq f(\tau)$$

$f(\tau)$ – funkcja jednej zmiennej (długości kroku τ)

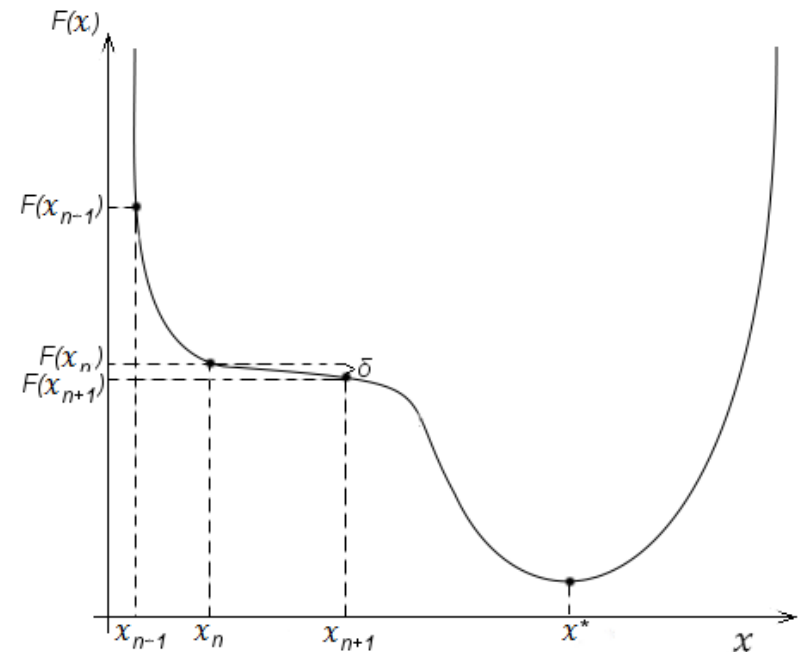
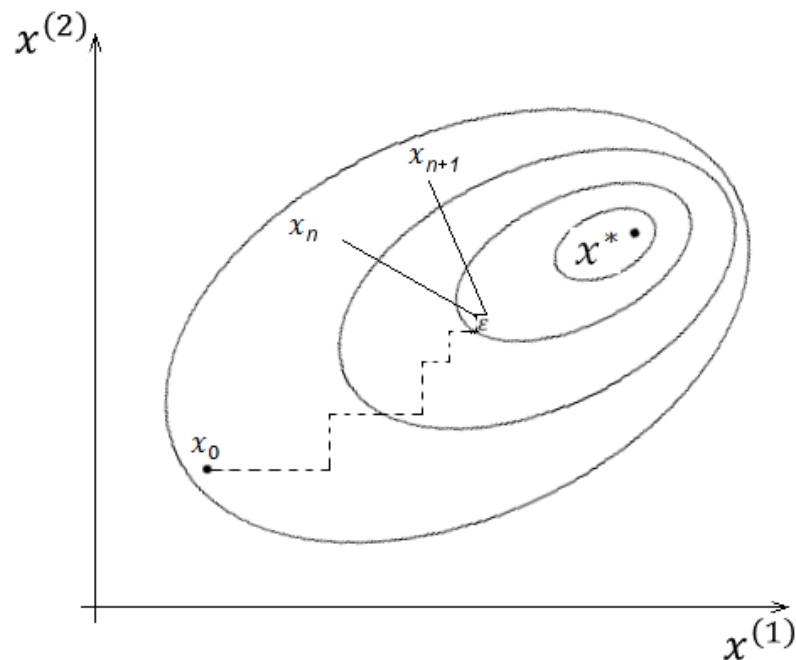
$$\tau^* \rightarrow f(\tau^*) = \min_{\tau} f(\tau)$$

optymalizacja w kierunku $\equiv$ optymalizacja funkcji jednej zmiennej

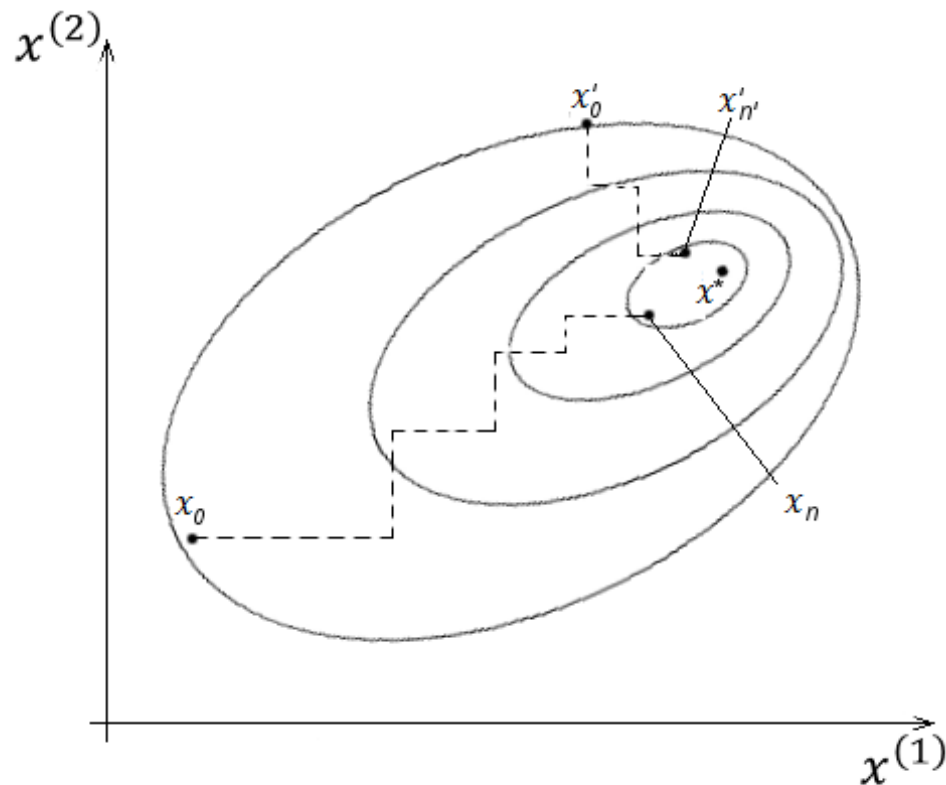
Warunki zatrzymania procedury

$$\|x_{n+1} - x_n\| < \varepsilon; \quad |F(x_{n+1}) - F(x_n)| < \delta; \quad \frac{|F(x_{n+1}) - F(x_n)|}{\|x_{n+1} - x_n\|} < \varrho$$

Uwaga!



Dobra rada



$$\|x_n - x'_{n'}\| < \varepsilon$$

x_0, x'_0 - różne punkty początkowe
 $x_n, x'_{n'}$ - odpowiednie punkty końcowe

- Metody optymalizacji w kierunku
 - Podział równomierny
 - Podział na połowę
 - Złoty podział
 - Aproksymacji kwadratowej
 - Metoda pierwszej pochodnej
 - Metoda znaku pochodnej
 - Metoda Newtona
 - Metoda Bolzano

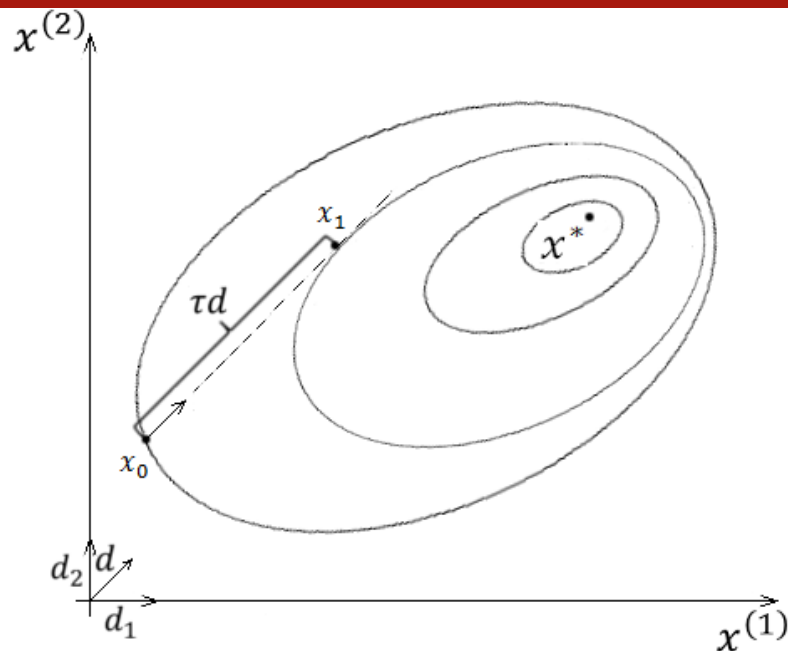
- Bezgradientowe metody optymalizacji
 - Hooka-Jevesa z krokiem dyskretnym
 - Rozendrocka z krokiem dyskretnym
 - Hooka-Jevesa z krokiem optymalnym
 - Rozendrocka z krokiem optymalnym
 - Powella - kierunków sprzężonych
 - Nelder-Mead (simplex)



- Gradientowe metody optymalizacji
 - Gradientu prostego
 - Najszybszego spadku
 - Newtona
 - Fletchera-Reevesa (gradientu sprzężonego)
 - Zmiennej metryki

- Metody optymalizacji z ograniczeniami
 - Transformacji zmiennych
 - Funkcji kary
 - Kary zewnętrznej (kary)
 - Kary wewnętrzne (bariery)
 - Metody modyfikacji kierunku
 - Metoda kompleks

Optymalizacja w kierunku



x_0 - punkt początkowy
 x_1 - punkt końcowy
 d - kierunek
 τ - długość kroku w kierunku

$$\tau^* \rightarrow F(x_0 + \tau^* d) = \min_{\tau} F(x_0 + \tau d)$$

x_0, d – ustalone $F(x_0 + \tau d) \triangleq f(\tau)$

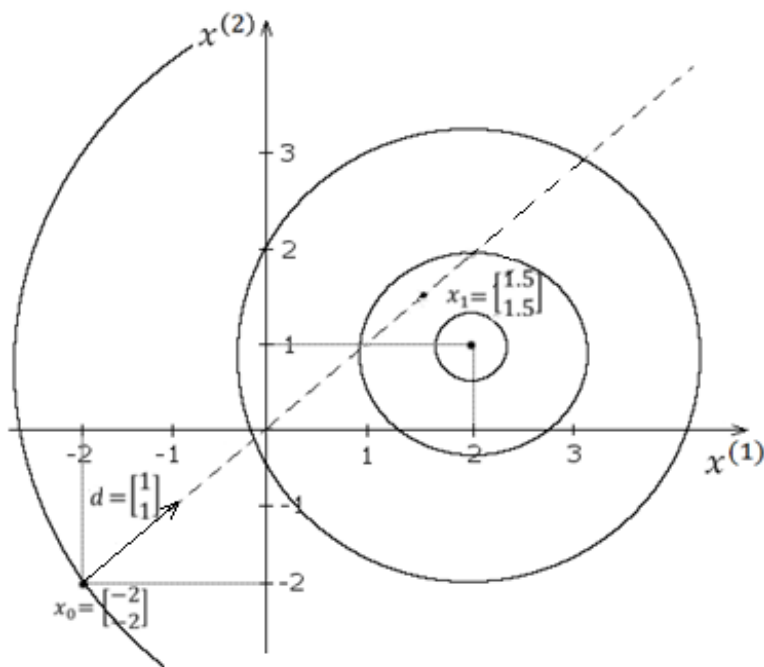
$f(\tau)$ – funkcja jednej zmiennej (długości kroku τ)

$$\tau^* \rightarrow f(\tau^*) = \min_{\tau} f(\tau)$$

optymalizacja w kierunku $\equiv$ optymalizacja funkcji jednej zmiennej

Przykład

$$F(x) = (x^{(1)} - 2)^2 + (x^{(2)} - 1)^2, x_0 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}, d = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$



$$x_1 = x_0 + \tau d = \begin{bmatrix} -2 \\ -2 \end{bmatrix} + \tau \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 + 1 * \tau \\ -2 + 1 * \tau \end{bmatrix}$$

$$F(x_0 + \tau d) = (-2 + \tau - 2)^2 + (-2 + \tau - 1)^2 = (\tau - 4)^2 + (\tau - 3)^2 = 2\tau^2 - 14\tau + 25 \triangleq f(\tau)$$

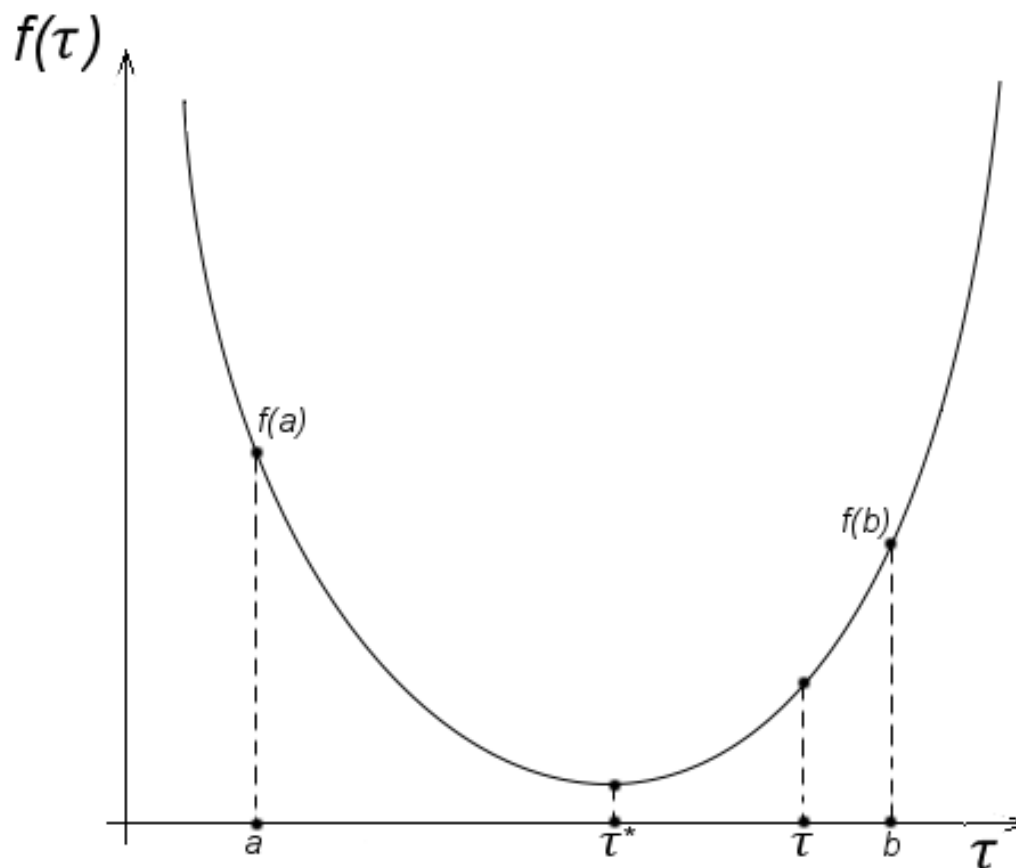
$$f'(\tau) = 4\tau - 14 = 0$$

$$\tau^* = 3.5$$

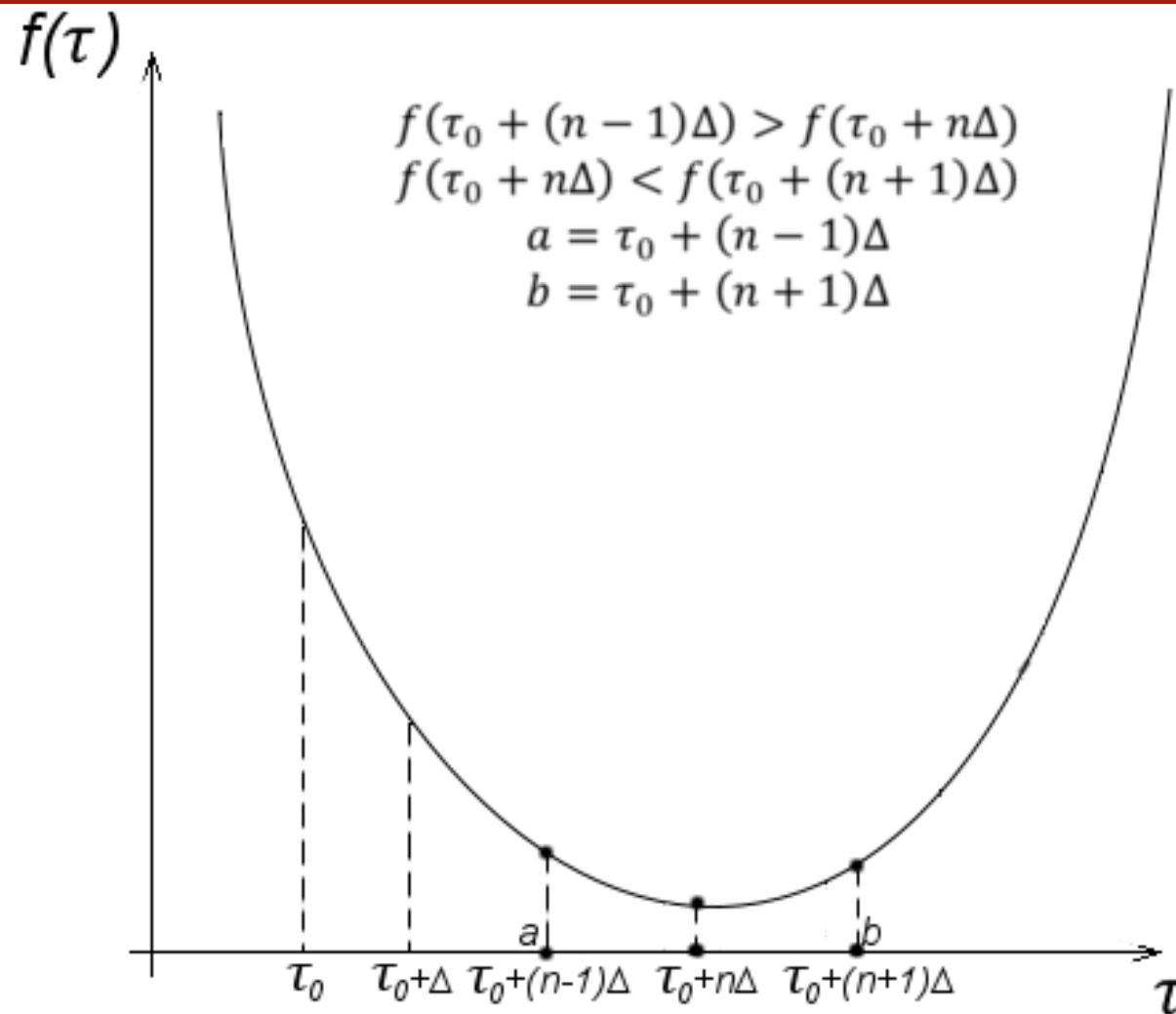
$$x_1 = x_0 + \tau^* d = \begin{bmatrix} -2 \\ -2 \end{bmatrix} + 3.5 * \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}$$

Metody zawężania odcinka

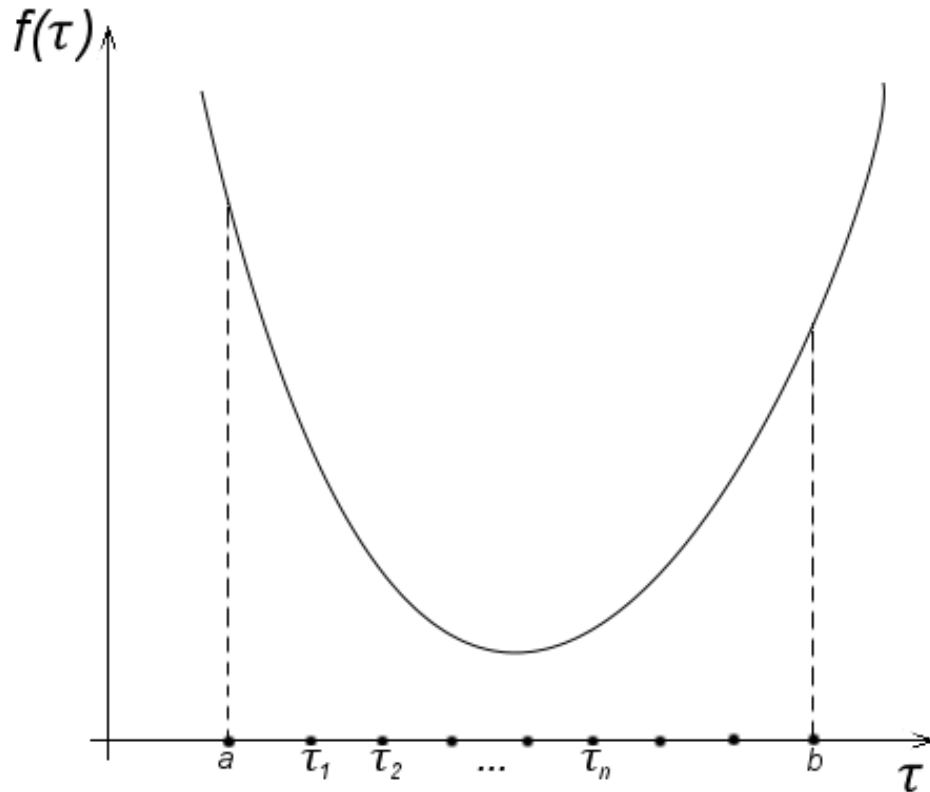
Założenie: $\tau^* \in [a, b]$



Spełnienie założenia $\tau \in [a, b]$



Podział równomierny



$$\tau_0 = a$$

$$\tau_n = \tau_0 + n\varepsilon$$

$$\tau^* \approx \tilde{\tau} \rightarrow f(\tilde{\tau}) = \min_{1 \leq n \leq N} \{f(\tau_n)\}$$

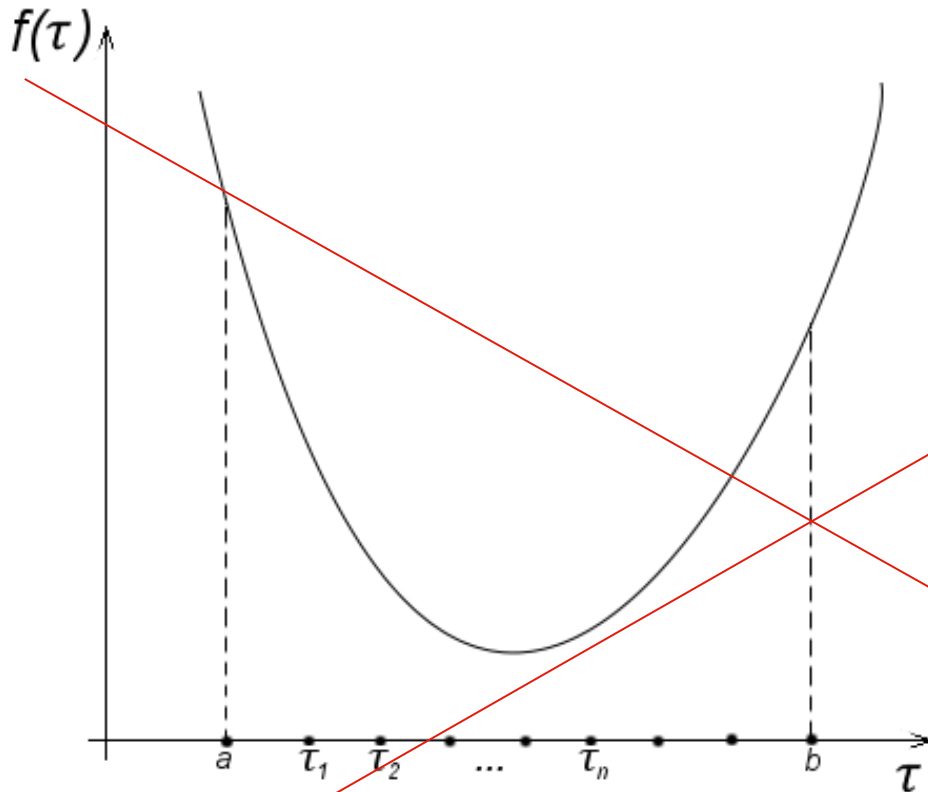
$N = \left\lceil \frac{b-a}{\varepsilon} \right\rceil$ - liczba wyliczeń wartości funkcji

Przykładowo dla:

$$\Delta = b - a = 1, \varepsilon = 0.01$$

$$N = \left\lceil \frac{b-a}{\varepsilon} \right\rceil = \frac{1}{0.01} = 100$$

Podział równomierny



$$\tau_0 = a$$

$$\tau_n = \tau_0 + n\varepsilon$$

$$\tau^* \approx \tilde{\tau} \rightarrow f(\tilde{\tau}) = \min_{1 \leq n \leq N} \{f(\tau_n)\}$$

$N = \left\lceil \frac{b-a}{\varepsilon} \right\rceil$ - liczba wyliczeń wartości funkcji

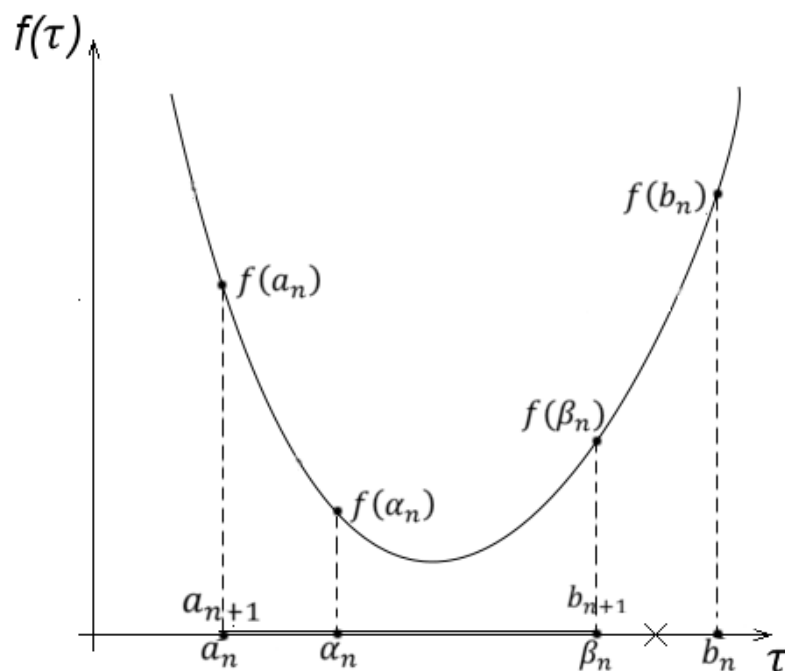
Przykładowo dla:

$$\Delta = b - a = 1, \varepsilon = 0.01$$

$$N = \left\lceil \frac{b-a}{\varepsilon} \right\rceil = \frac{1}{0.01} = 100$$

Zawężanie odcinka

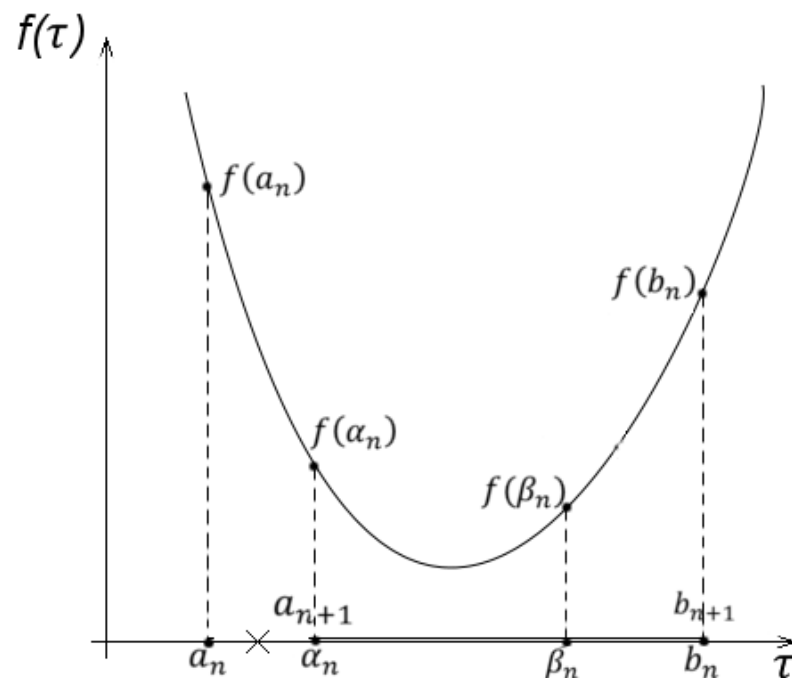
$f(\alpha_n)? f(\beta_n)$



$$f(\alpha_n) \leq f(\beta_n)$$

$$a_{n+1} := a_n$$

$$b_{n+1} := \beta_n$$

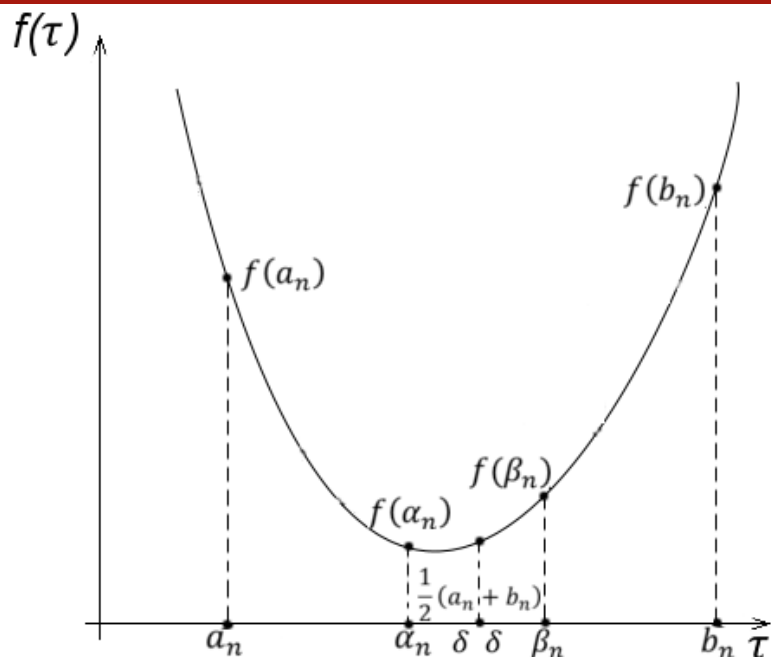


$$f(\alpha_n) > f(\beta_n)$$

$$a_{n+1} := \alpha_n$$

$$b_{n+1} := b_n$$

Metoda podziału na połowę



$$\alpha_n = \frac{1}{2}(a_n + b_n) - \delta$$
$$\beta_n = \frac{1}{2}(a_n + b_n) + \delta$$

$N = ?$ dla $\varepsilon = 0.01, \Delta = b - a = 1$

Dane: $a_0, b_0, \varepsilon, \delta$

Krok 0: $n = 0$

Krok 1: $\alpha_n = \frac{1}{2}(a_n + b_n) - \delta$

$$\beta_n = \frac{1}{2}(a_n + b_n) + \delta$$

Krok 2: Jeśli $f(\alpha_n) \leq f(\beta_n)$ to
 $a_{n+1} := a_n, b_{n+1} := \beta_n$,
w przeciwnym razie

$$a_{n+1} := \alpha_n, b_{n+1} := b_n.$$

Krok 3: Jeśli $|b_{n+1} - a_{n+1}| \geq \varepsilon$ to
 $n := n + 1$, idź do kroku 1,
w przeciwnym razie
 $\tilde{\tau} = \frac{1}{2}(a_{n+1} + b_{n+1})$ (STOP)

Oszacowanie liczby kroków procedury

- $\Delta_0 = b_0 - a_0 = 1$ początkowa długość odcinka
- $\Delta_1 = \frac{1}{2} \Delta_0$ długość odcinka po jednym kroku
- $\Delta_2 = \frac{1}{2} \Delta_1 = \left(\frac{1}{2}\right)^2 \Delta_0$ długość odcinka po dwóch krokach
- $\vdots$
- $\Delta_N = \frac{1}{2} \Delta_{N-1} = \dots = \left(\frac{1}{2}\right)^N \Delta_0$ długość odcinka po N krokach procedury
- Oczekujemy, że po N krokach procedury długość odcinka będzie mniejsza od ε .
- Zatem liczba niezbędnych kroków procedury spełnia warunek
$$\Delta_N = \left(\frac{1}{2}\right)^N \Delta_0 \leq \varepsilon$$
- dzielimy przez Δ_0



Oszacowanie liczby kroków procedury

- $\left(\frac{1}{2}\right)^N \leq \frac{\varepsilon}{\Delta_0}$ logarytmujemy obustronnie
- $\ln \left(\frac{1}{2}\right)^N \leq \ln \frac{\varepsilon}{\Delta_0}$ czyli
- $N \ln \left(\frac{1}{2}\right) \leq \ln \frac{\varepsilon}{\Delta_0}$ dzielimy obustronnie przez $\ln \left(\frac{1}{2}\right)$.

Ponieważ $\ln \left(\frac{1}{2}\right)$ jest liczbą ujemną zmieniamy znak nierówności.

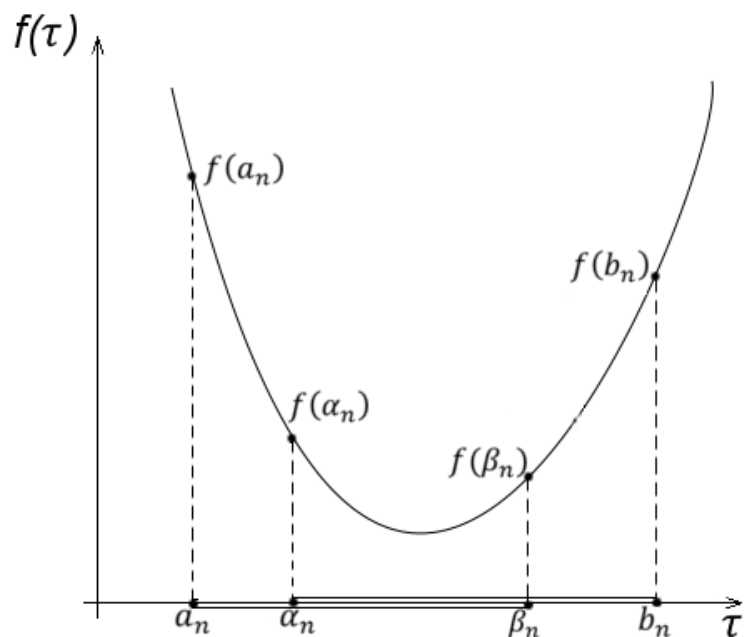
Czyli minimalna liczba kroków procedury spełnia warunek:

- $$N \geq \frac{\ln \frac{\varepsilon}{\Delta_0}}{\ln \frac{1}{2}} = \frac{\ln 0.01}{\ln 0.5} = \frac{4.605}{0.693} \approx 7$$

Ponieważ w każdym kroku procedury wyliczany dwie wartości funkcji to liczba koniecznych wyliczeń wartości funkcji wynosi:

- Liczba wyliczeń wartości funkcji = $2N = 14$

Metoda γ podziału



$$\frac{b_n - \alpha_n}{b_n - a_n} = \frac{\beta_n - a_n}{b_n - a_n} = \gamma$$

$$\alpha_n = b_n + \gamma(a_n - b_n)$$

$$\beta_n = a_n + \gamma(b_n - a_n)$$

Dane: $a_0, b_0, \varepsilon, \gamma$

Krok 0: $n = 0$

Krok 1: $\alpha_n = b_n + \gamma(a_n - b_n)$

$\beta_n = a_n + \gamma(b_n - a_n)$

Krok 2: Jeśli $f(\alpha_n) \leq f(\beta_n)$ to

$a_{n+1} := a_n, b_{n+1} := \beta_n,$

w przeciwnym razie

$a_{n+1} := \alpha_n, b_{n+1} := b_n.$

Krok 3: Jeśli $|b_{n+1} - a_{n+1}| \geq \varepsilon$ to

$n := n + 1$, idź do kroku 1,

w przeciwnym razie

$\tilde{\tau} = \frac{1}{2}(a_{n+1} + b_{n+1})$ (STOP)

$N = ?$ dla $\varepsilon = 0.01, \Delta = b - a = 1$

Oszacowanie liczby kroków procedury

- $\Delta_0 = b_0 - a_0 = 1$ początkowa długość odcinka, $\varepsilon = 0.01$
- $\Delta_1 = \gamma \Delta_0$ długość odcinka po jednym kroku
- $\Delta_2 = \gamma \Delta_1 = (\gamma)^2 \Delta_0$ długość odcinka po dwóch krokach
- $\vdots$
- $\Delta_N = \gamma \Delta_{N-1} = \dots = (\gamma)^N \Delta_0$ długość odcinka po N krokach procedury
- Oczekujemy, że po N krokach procedury długość odcinka będzie mniejsza od ε .
- Zatem liczba niezbędnych kroków procedury spełnia warunek
$$\Delta_N = (\gamma)^N \Delta_0 \leq \varepsilon$$
- dzielimy przez Δ_0

Oszacowanie liczby kroków procedury

- $(\gamma)^N \leq \frac{\varepsilon}{\Delta_0}$ logarytmujemy obustronnie
- $\ln(\gamma)^N \leq \ln \frac{\varepsilon}{\Delta_0}$ czyli
- $N \ln(\gamma) \leq \ln \frac{\varepsilon}{\Delta_0}$ dzielimy obustronnie przez $\ln(\gamma)$.

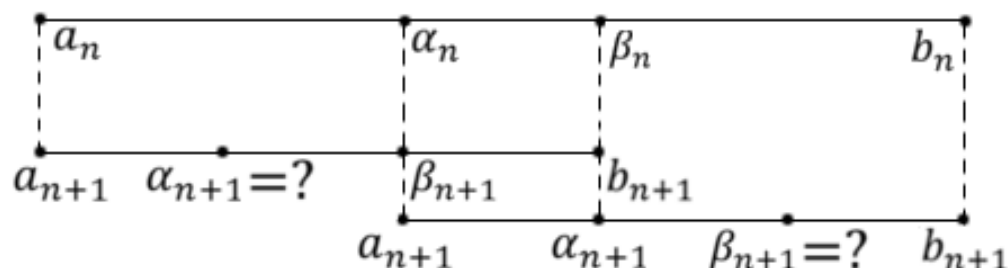
Ponieważ $\ln(\gamma)$ jest liczbą ujemną zmieniamy znak nierówności.

Czyli minimalna liczba kroków procedury spełnia warunek:

- $N \geq \frac{\ln \frac{\varepsilon}{\Delta_0}}{\ln \gamma}$
- Ponieważ w każdym kroku procedury wyliczany dwie wartości funkcji to liczba koniecznych wyliczeń wartości funkcji wynosi:
- Liczba wyliczeń wartości funkcji = $2N$

$$\gamma = ??$$

Metoda złotego podziału



$$\gamma^2 + \gamma - 1 = 0$$

$$\gamma = \frac{\sqrt{5} - 1}{2} \approx 0.618$$

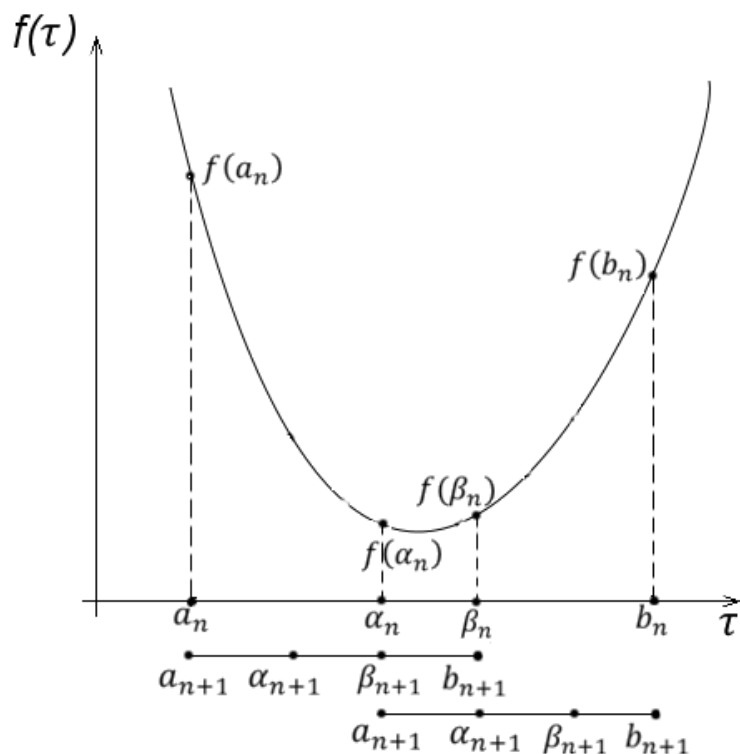
1. $\frac{\beta_{n+1} - a_{n+1}}{b_{n+1} - a_{n+1}} = \gamma$, czyli

$$\frac{\alpha_n - a_n}{\beta_n - a_n} = \frac{b_n + \gamma(a_n - b_n) - a_n}{a_n + \gamma(b_n - a_n) - a_n} = \frac{b_n - a_n + \gamma(a_n - b_n)}{\gamma(b_n - a_n)} = \frac{1}{\gamma} - 1 = \gamma$$

2. $\frac{b_{n+1} - \alpha_{n+1}}{b_{n+1} - a_{n+1}} = \gamma$, czyli

$$\frac{b_n - \beta_n}{b_n - a_n} = \frac{b_n - a_n + \gamma(b_n - a_n)}{b_n - b_n - \gamma(b_n - a_n)} = \frac{b_n - a_n + \gamma(b_n - a_n)}{\gamma(b_n - a_n)} = \frac{1}{\gamma} - 1 = \gamma$$

Metoda złotego podziału c.d.



$$\gamma^2 + \gamma - 1 = 0$$

$$\gamma = \frac{\sqrt{5} - 1}{2} \approx 0.618$$

$$N = ? \text{ dla } \varepsilon = 0.01, \Delta = b - a = 1$$

Dane: $a_0, b_0, \varepsilon, \gamma = \frac{\sqrt{5}-1}{2}$

Krok 0: $n = 0$

$$\alpha_0 = b_0 + \gamma(a_0 - b_0)$$

$$\beta_0 = a_0 + \gamma(b_0 - a_0)$$

Krok 1: Jeśli $|b_n - a_n| < \varepsilon$, to
 $\tilde{\tau} = \frac{1}{2}(a_n + b_n)$ (STOP)

w przeciwnym razie → krok 2

Krok 2: Jeśli $f(\alpha_n) \leq f(\beta_n)$ to

$$a_{n+1} := a_n, b_{n+1} := \beta_n,$$

$$\beta_{n+1} := \alpha_n,$$

$$\alpha_{n+1} := \beta_n + \gamma(a_n - b_n)$$

$$n := n + 1, \text{ idź do kroku 1}$$

w przeciwnym razie

$$a_{n+1} := \alpha_n, b_{n+1} := b_n,$$

$$\alpha_{n+1} := \beta_n,$$

$$\beta_{n+1} := \alpha_n + \gamma(b_n - \alpha_n)$$

$$n := n + 1, \text{ idź do kroku 1}$$



Oszacowanie liczby kroków procedury

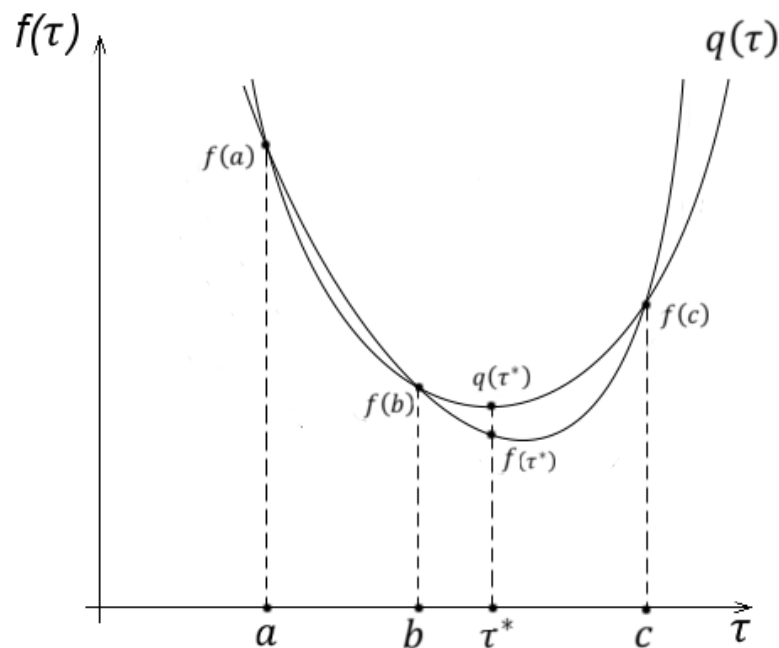
- $(\gamma)^N \leq \frac{\varepsilon}{\Delta_0}$ logarytmujemy obustronnie
- $\ln(\gamma)^N \leq \ln \frac{\varepsilon}{\Delta_0}$ czyli
- $N \ln(\gamma) \leq \ln \frac{\varepsilon}{\Delta_0}$ dzielimy obustronnie przez $\ln(\gamma)$.

Ponieważ $\ln(\gamma)$ jest liczbą ujemną zmieniamy znak nierówności.

Czyli minimalna liczba kroków procedury spełnia warunek:

- $N \geq \frac{\ln \frac{\varepsilon}{\Delta_0}}{\ln \gamma}$ teraz $\gamma = \frac{\sqrt{5}-1}{2} \approx 0.618$
- Ponieważ w każdym kroku procedury wyliczany dodatkowo wartości funkcji to liczba koniecznych wyliczeń wartości funkcji wynosi:
- $\ln \gamma = \ln 0.618 = -0.481$
- $N \geq \frac{\ln \frac{\varepsilon}{\Delta_0}}{\ln \gamma} = \frac{\ln 0.01}{\ln 0.618} = \frac{4.605}{0.481} \approx 9,578 \approx 10$
- Liczba wyliczeń wartości funkcji = $N+1 = 10+1 = 11$

Metoda aproksymacji kwadratowej



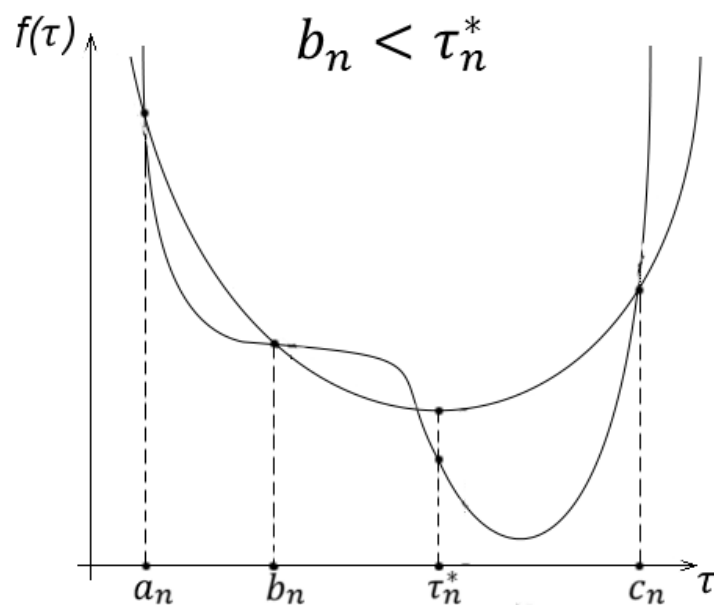
$$\begin{aligned} a &< b < c \\ f(a) &\geq f(b) \\ f(b) &\leq f(c) \end{aligned}$$

$q(\tau)$ – aproksymacja kwadratowa
 τ^* - minimum funkcji $q(\tau)$

$$q(\tau) = \frac{f(a)(\tau - b)(\tau - c)}{(a - b)(a - c)} + \frac{f(b)(\tau - a)(\tau - c)}{(b - a)(b - c)} + \frac{f(c)(\tau - a)(\tau - b)}{(c - a)(b - c)}$$

$$\tau^* = \frac{1}{2} \frac{f(a)(b^2 - c^2) + f(b)(c^2 - a^2) + f(c)(a^2 - b^2)}{f(a)(b - c) + f(b)(c - a) + f(c)(a - b)}$$

Metoda aproksymacji kwadratowej



$$f(b_n) \geq f(\tau_n^*)$$

$$a_{n+1} := b_n$$

$$b_{n+1} := \tau_n^*$$

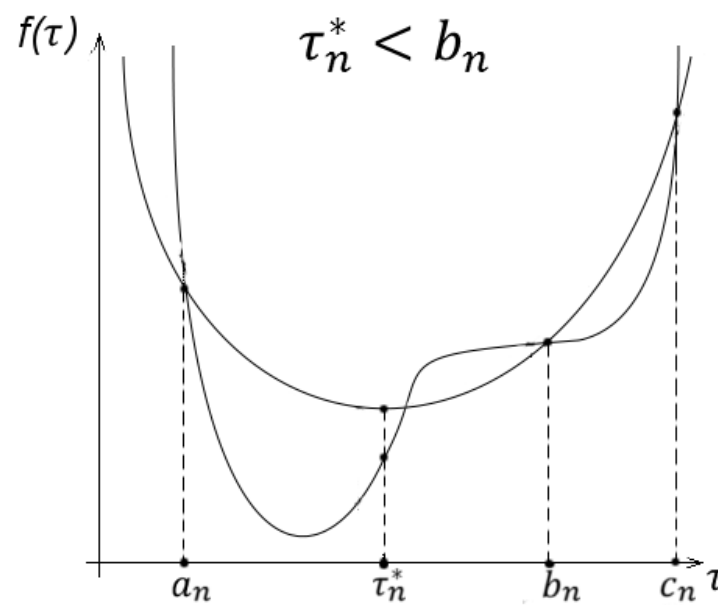
$$c_{n+1} := c_n$$

$$f(b_n) < f(\tau_n^*)$$

$$a_{n+1} := a_n$$

$$b_{n+1} := b_n$$

$$c_{n+1} := \tau_n^*$$



$$f(b_n) \geq f(\tau_n^*)$$

$$a_{n+1} := a_n$$

$$b_{n+1} := \tau_n^*$$

$$c_{n+1} := b_n$$

$$f(b_n) \geq f(\tau_n^*)$$

$$a_{n+1} := \tau_n^*$$

$$b_{n+1} := b_n$$

$$c_{n+1} := c_n$$

$$|c_{n+1} - a_{n+1}| < \varepsilon \quad \tilde{\tau} = \tau_{n+1}^*$$

Metoda aproksymacji kwadratowej

Dane: $a_0, b_0, c_0, \varepsilon$

Krok 0: $n = 0$

Krok 1: $\tau_n = \frac{1}{2} \frac{f(a_n)(b_n^2 - c_n^2) + f(b_n)(c_n^2 - a_n^2) + f(c_n)(a_n^2 - b_n^2)}{f(a_n)(b_n - c_n) + f(b_n)(c_n - a_n) + f(c_n)(a_n - b_n)}$

Krok 2: Jeśli $b_n < \tau_n$ to idź do kroku 3

w przeciwnym razie

Jeśli $f(b_n) \geq f(\tau_n)$ to $a_{n+1} := a_n, b_{n+1} := \tau_n, c_{n+1} := b_n$

w przeciwnym razie $a_{n+1} := \tau_n, b_{n+1} := b_n, c_{n+1} := c_n$

Krok 3: Jeśli $f(b_n) \geq f(\tau_n)$ to $a_{n+1} := b_n, b_{n+1} := \tau_n, c_{n+1} := c_n$

w przeciwnym razie $a_{n+1} := a_n, b_{n+1} := b_n, c_{n+1} := \tau_n$

Krok 4: Jeśli $|c_{n+1} - a_{n+1}| \geq \varepsilon$ to $n := n + 1$

w przeciwnym razie $\tilde{\tau} = \frac{1}{2}(a_{n+1} + c_{n+1})$ (STOP)

idź do kroku 4

idź do kroku 4

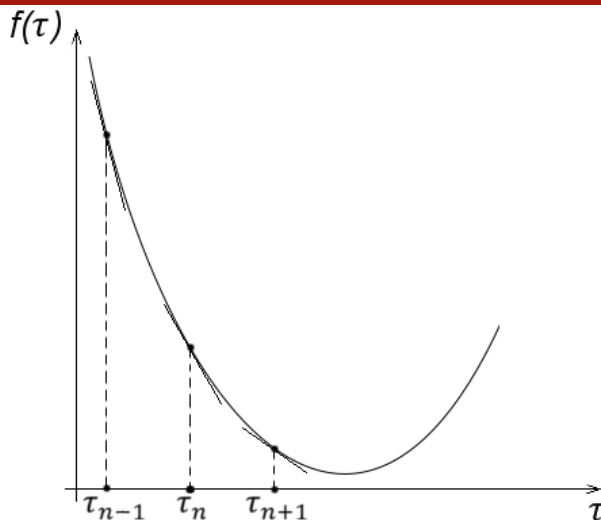
idź do kroku 4

idź do kroku 4

idź do kroku 1



Metoda pierwszej pochodnej



$$\tau_{n+1} = \tau_n - \gamma_n f'(\tau_n) \quad \gamma_n > 0, \tau_0$$

$$\lim_{n \rightarrow \infty} \gamma_n = \gamma$$

$$\sum_{n=0}^{\infty} \gamma_n = \infty$$

$$\text{np. } |\tau_{n+1} - \tau_n| < \varepsilon \quad (\text{STOP})$$

$$\tau_0$$

$$\tau_1 = \tau_0 - \gamma_0 f'(\tau_0)$$

$$\tau_2 = \tau_1 - \gamma_1 f'(\tau_1) = \tau_0 - \gamma_0 f'(\tau_0) - \gamma_1 f'(\tau_1)$$

$$\tau_{n+1} = \tau_n - \gamma_n f'(\tau_n) = \dots = \tau_0 - \gamma_0 f'(\tau_0) - \gamma_1 f'(\tau_1) - \dots - \gamma_n f'(\tau_n)$$

$$|\tau_{n+1} - \tau_0| = \left| \sum_{k=0}^n \gamma_k f'(\tau_k) \right| \leq \sum_{k=0}^n \gamma_k |f'(\tau_k)| \leq \max_{0 \leq k \leq n} |f'(\tau_k)| \sum_{k=0}^n \gamma_k$$

$$|\tau_{\infty} - \tau_0| \leq \sum_{k=0}^{\infty} \gamma_k = \infty$$



Metoda znaku pierwszej pochodnej

$$\tau_{n+1} = \tau_n - \vartheta_n \operatorname{sign}[f'(\tau_n)]$$

$$\gamma_n f'(\tau_n) = \gamma_n |f'(\tau_n)| * \operatorname{sign} f'(\tau_n) = \vartheta_n \operatorname{sign}[f'(\tau_n)], \text{ gdzie } \vartheta_n = \gamma_n |f'(\tau_n)|$$

$$\vartheta_n > 0$$

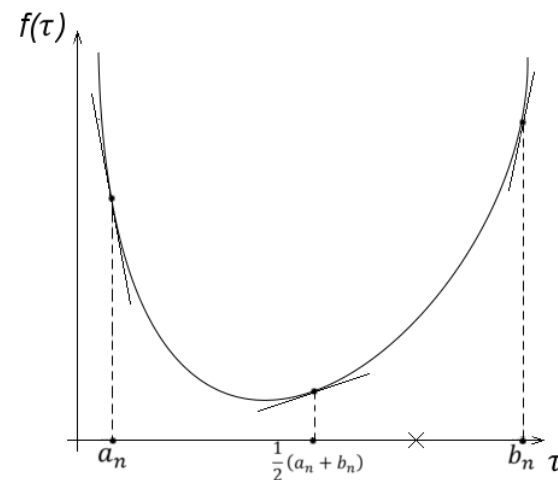
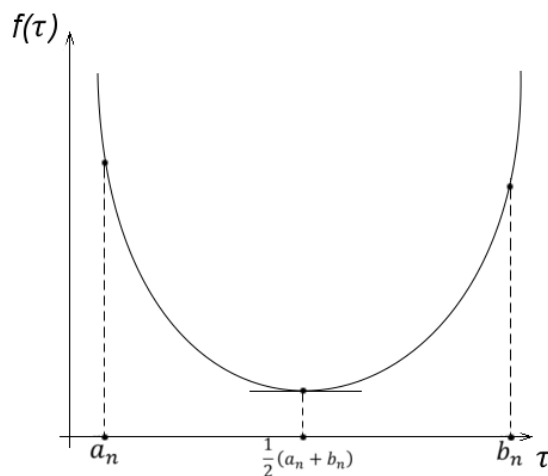
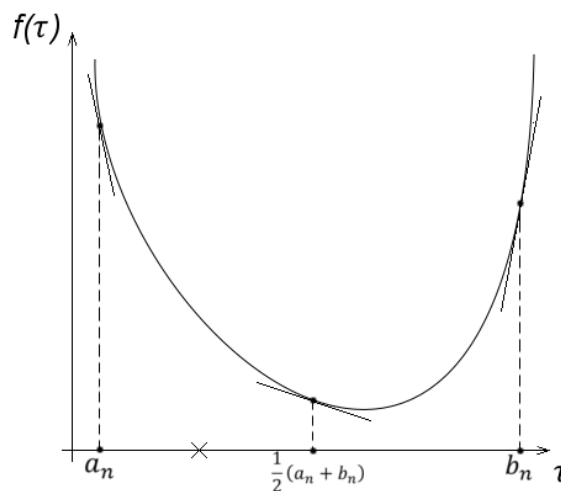
$$\lim_{n \rightarrow \infty} \vartheta_n = 0, \text{ bo } \lim_{n \rightarrow \infty} |f'(\tau_n)| = 0, \lim_{n \rightarrow \infty} \gamma_n = \gamma$$

$$\sum_{n=0}^{\infty} \vartheta_n = \infty \qquad \lim_{n \rightarrow \infty} \vartheta_n = \lim_{n \rightarrow \infty} \gamma_n |f'(\tau_n)| = 0$$



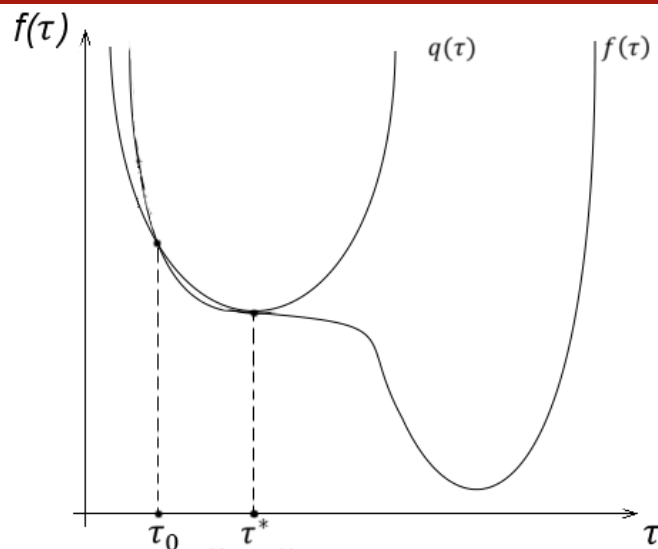
Metoda Bolzano

$$\text{sign } a_n \neq \text{sign } b_n$$



$$\begin{aligned} \text{sign } a_n &= \text{sign}\left(f'\left(\frac{1}{2}(a_n + b_n)\right)\right) & f'\left(\frac{1}{2}(a_n + b_n)\right) &= 0 & \text{sign } b_n &= \text{sign}\left(f'\left(\frac{1}{2}(a_n + b_n)\right)\right) \\ a_{n+1} &:= \frac{1}{2}(a_n + b_n) & \tilde{\tau} &:= \frac{1}{2}(a_n + b_n) & a_{n+1} &:= a_n \\ b_{n+1} &:= b_n & (\text{rzadko}) & & b_{n+1} &:= \frac{1}{2}(a_n + b_n) \end{aligned}$$

Metoda drugiej pochodnej



$$\tau_0$$
$$\tau_{n+1} = \tau_n - \frac{f'(\tau_n)}{f''(\tau_n)}$$
$$|\tau_{n+1} - \tau_n| < \varepsilon \text{ (STOP)}$$

$$f(\tau) = \underbrace{f(\tau_0) + (\tau - \tau_0)f'(\tau_0) + \frac{1}{2}(\tau - \tau_0)^2 f''(\tau_0)}_{q(\tau)} + o_3(|\tau - \tau_0|)$$

$$q'(\tau) = f'(\tau_0) + (\tau^* - \tau_0)f''(\tau_0) = 0$$

$$\tau^* = \tau_0 - \frac{f'(\tau_0)}{f''(\tau_0)}$$

Dziękuję za uwagę

