Intervalo dalijimo pusiau metodas

- input: $f()$, l, r, ε.

1. $x_m = (l + r)/2$, $L = r - l$, skaičiuojame $f(x_m)$;
2. $x_1 = l + L/4$, $x_2 = r - L/4$, skaičiuojame $f(x_1)$ ir $f(x_2)$;
3. jei $f(x_1) < f(x_m)$, tai:
   3.1 atmetamas $(x_m, r]$ atliekant keitimą $r = x_m$;
   3.2 intervalo centru tampa x_1, tad keičiamas $x_m = x_1$;
   3.3 einama į 6 punktą
4. jei $f(x_2) < f(x_m)$, tai:
   4.1 atmetamas $[l, x_m)$ atliekant keitimą $l = x_m$;
   4.2 intervalo centru tampa taškas x_2, tad keičiama $x_m = x_2$;
   4.3 einama į 6 punktą
5. priešingu atveju $(f(x_1) \geq f(x_m)$ ir $f(x_2) \geq f(x_m))$:
   5.1 atmetami intervalai $[l, x_1)$ ir $(x_2, r]$ atliekant keitimus $l = x_1$ ir $r = x_2$;
6. skaičiuojamas $L = r - l$; jei L pakankamai mažas ($L < \varepsilon$), baigiame skaičiavimus, jei ne – einame į 2 punktą.

- output: mažiausias iš $f(x_m)$, $f(x_1)$ ir $f(x_2)$ ir atitinkamas x_i.

Auksinio pjūvio algoritmas

- input: $f()$, l, r, ε, $\tau = (-1 \pm \sqrt{5})/2 = 0{,}61803...$
  $(\tau^2 = 1 - \tau)$.

1. $L = r - l$, $x_1 = r - \tau L$ ir $x_2 = l + \tau L$, skaičiuojame $f(x_1)$ ir $f(x_2)$;
2. jei $f(x_2) < f(x_1)$, tai:
   - 2.1 atmetamas $[l, x_1)$ atliekant keitimą $l = x_1$, $L = r - l$;
   - 2.2 kairiuoju tašku tampa ankstesnis dešinysis taškas $x_1 = x_2$;
   - 2.3 naujasis dešinysis taškas $x_2 = l + \tau L$, skaičiuojame $f(x_2)$;
3. priešingu atveju:
   - 3.1 atmetamas $(x_2, r]$ atliekant keitimą $r = x_2$, $L = r - l$;
   - 3.2 dešiniuoju tašku tampa ankstesnis kairysis taškas $x_2 = x_1$;
   - 3.3 naujasis kairysis taškas $x_1 = r - \tau L$, skaičiuojame $f(x_1)$;
4. jei L pakankamai mažas ($L < \varepsilon$), skaičiavimus baigiame, jei ne – einame į 2 punktą.

- output: mažiausias iš $f(x_1)$ ir $f(x_2)$ ir atitinkamas x_i.

Niutono algoritmas

- input: $f()$, x_0, ε.

1. $i = 0$;
2. $$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)};$$
3. jei žingsnio ilgis pakankamai mažas ($|x_i - x_{i+1}| < \varepsilon$), skaičiavimus baigiame, jei ne, tai $i = i + 1$ ir einame į 2 punktą.

- output: x_{i+1} ir $f(x_{i+1})$, $i + 1$.

Gradientinis nusileidimas

- input: $f()$, X_0, γ, ε.

1. $i = 0$;
2. $$X_{i+1} = X_i - \gamma \nabla f(X_i);$$
3. jei gradientas pakankamai mažas ($\|\nabla f(X_i)\| < \varepsilon$), skaičiavimus baigiame, jei ne, tai $i = i + 1$ ir einame į 2 punktą.

- output: X_{i+1} ir $f(X_{i+1})$, $i + 1$.

Greičiausias nusileidimas

- input: $f()$, X_0, ε.

1. $i = 0$;
2. $$X_{i+1} = X_i - \arg\min_{\gamma \geq 0} f(X_i - \gamma \cdot \nabla f(X_i)) \nabla f(X_i);$$
3. jei gradientas pakankamai mažas ($\|\nabla f(X_i)\| < \varepsilon$), skaičiavimus baigiame, jei ne, tai $i = i + 1$ ir einame į 2 punktą.

- output: X_{i+1} ir $f(X_{i+1})$, $i + 1$.

Deformuojamo simplekso algoritmas

- input: $f()$, X_0, n, ε, α, β, γ, ν.

1. Sudaromas simpleksas iš X_0 ir X_i, gautų per X_0 ir α.
2. Nustatoma geriausia X_l, antra pagal gerumą X_g, blogiausia X_h viršūnės.
3. Apskaičiuojamas X_c ir X_{naujas}. Pagal $f(X_{naujas})$ nustatomas Θ ir skaičiuojamas naujas taškas Z:
   a) jei $f(X_l) < f(X_{naujas}) < f(X_g)$, tai $\Theta = 1$: dydis nekinta;
   b) jei $f(X_{naujas}) < f(X_l)$, tai $\Theta = \gamma > 1$: išplečiamas;
   c) jei $f(X_{naujas}) > f(X_h)$, tai $\Theta = \nu$: suspaudžiamas;
   d) jei $f(X_g) < f(X_{naujas}) < f(X_h)$, tai $\Theta = \beta$: suspaudžiamas.
4. Jei žingsnis sėkmingas, $X_h = Z$, kartojame nuo 2 punkto.
5. Jei žingsnis nesėkmingas, $\alpha = K\alpha$, $X_0 = X_l$.
6. Jei $\alpha < \varepsilon$, stabdoma, priešingu atveju einame į 1 punktą.

- output: X_l ir $f(X_l)$.