Vilniaus universitetas

Matematikos ir Informatikos fakultetas

Optimizavimo metodai

Vienmatis optimizavimas

Aurimas Petrėtis, Informatika, 3 kursas, 3 grupė

2019

Užduoties formuluotė

Duota funkcija $f(x) = (x^2 - 7)^2/9 - 1$,

Intervalas [0,10],

Tikslumas $10^{(-4)}$,

Pradinis artinys $x0 = 5$.

Spręsta metodais:

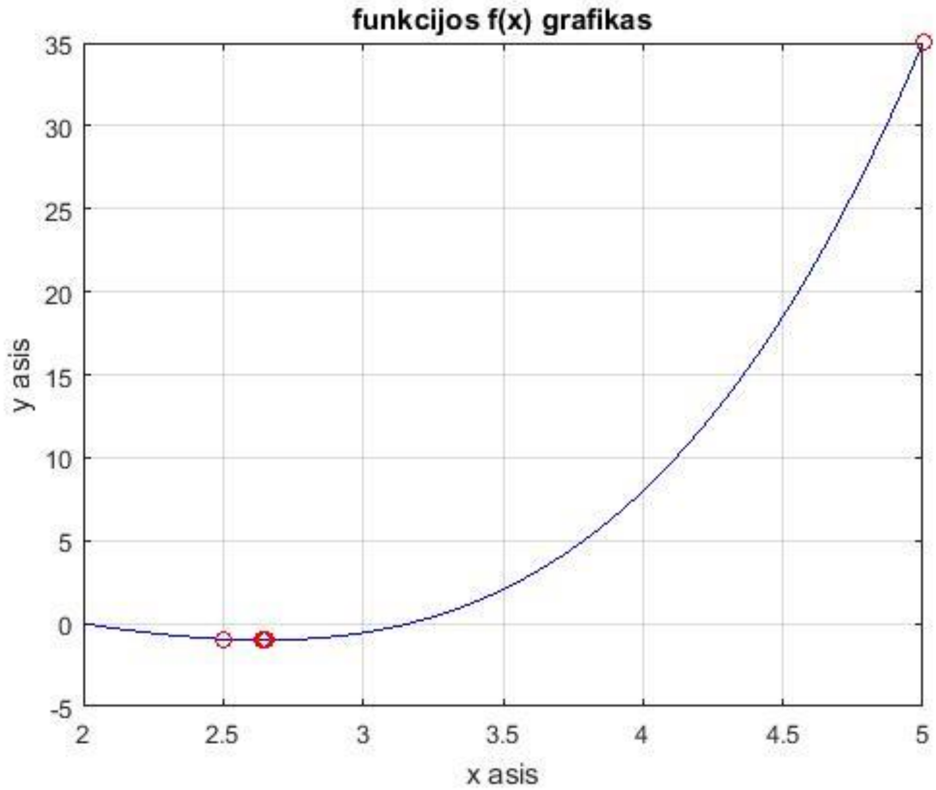
1. Intervalų dalijimo pusiau metodu
2. Auksinio pjūvio metodu
3. Niutono metodu

Sprendimui naudotas MATLAB matematinis paketas.

Intervalų dalijimo pusiau metodas

$$\min_{x \in [l, r]} f(x).$$

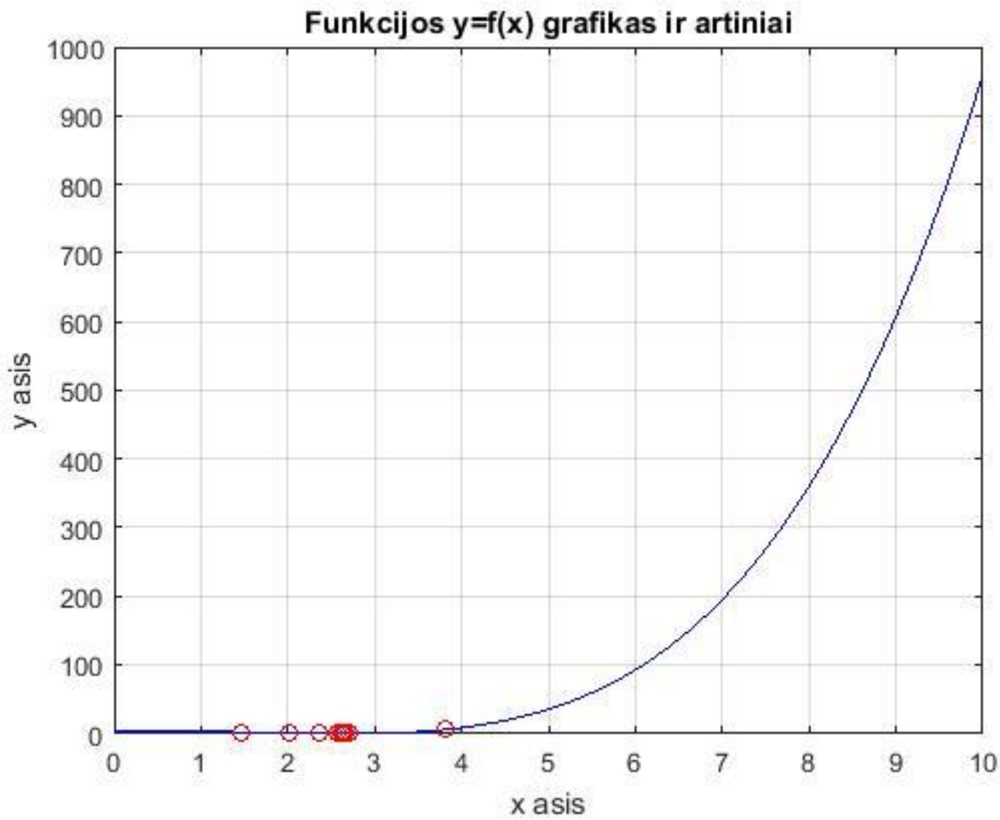
- Pradiniame intervale parenkami trys tolygiai pasiskirstę bandymo taškai x_m, x_1 ir x_2.
  1. $x_m = (l + r)/2$, $L = r - l$, skaičiuojame $f(x_m)$;
  2. $x_1 = l + L/4$, $x_2 = r - L/4$, skaičiuojame $f(x_1)$ ir $f(x_2)$;
  3. jei $f(x_1) < f(x_m)$, tai:
     - 3.1 atmetamas $(x_m, r]$ atliekant keitimą $r = x_m$;
     - 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.



```
xm        ym       k       funkcijos kvietimu sk
     5     35      1       3

    2.5000   -0.9375    2.0000    5.0000

    2.5000   -0.9375    3.0000    7.0000

    2.5000   -0.9375    4.0000    9.0000

    2.5000   -0.9375    5.0000   11.0000

    2.6563   -0.9997    6.0000   13.0000

    2.6563   -0.9997    7.0000   15.0000

    2.6563   -0.9997    8.0000   17.0000

    2.6367   -0.9997    9.0000   19.0000

    2.6465   -1.0000   10.0000   21.0000

    2.6465   -1.0000   11.0000   23.0000

    2.6465   -1.0000   12.0000   25.0000

    2.6453   -1.0000   13.0000   27.0000

    2.6459   -1.0000   14.0000   29.0000

    2.6459   -1.0000   15.0000   31.0000

    2.6457   -1.0000   16.0000   33.0000

    2.6457   -1.0000   17.0000   35.0000
```

Auksinio pjūvio metodas

- Intervale du bandymo taškai, vienodai nutolę nuo vidurio.
- Atmetama santykinai ta pati intervalo dalis.
- Skaičiuojama viena tikslo funkcijos reikšmė iteracijoje.
  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.



--

x1	y1	k	funkc. kviet. sk
3.8197	5.4006	1.0000	3.0000
2.3607	-0.7737	2.0000	4.0000
1.4590	1.6367	3.0000	5.0000
2.3607	-0.7737	4.0000	6.0000
2.0163	-0.0431	5.0000	7.0000
2.3607	-0.7737	6.0000	8.0000
2.5735	-0.9842	7.0000	9.0000
2.7051	-0.9888	8.0000	10.0000
2.6548	-0.9997	9.0000	11.0000
2.6238	-0.9985	10.0000	12.0000
2.6548	-0.9997	11.0000	13.0000
2.6430	-1.0000	12.0000	14.0000
2.6357	-0.9997	13.0000	15.0000
2.6430	-1.0000	14.0000	16.0000
2.6475	-1.0000	15.0000	17.0000
2.6458	-1.0000	16.0000	18.0000
2.6447	-1.0000	17.0000	19.0000
2.6458	-1.0000	18.0000	20.0000
2.6454	-1.0000	19.0000	21.0000
2.6458	-1.0000	20.0000	22.0000
2.6456	-1.0000	21.0000	23.0000
2.6458	-1.0000	22.0000	24.0000
2.6457	-1.0000	23.0000	25.0000
2.6457	-1.0000	24.0000	26.0000

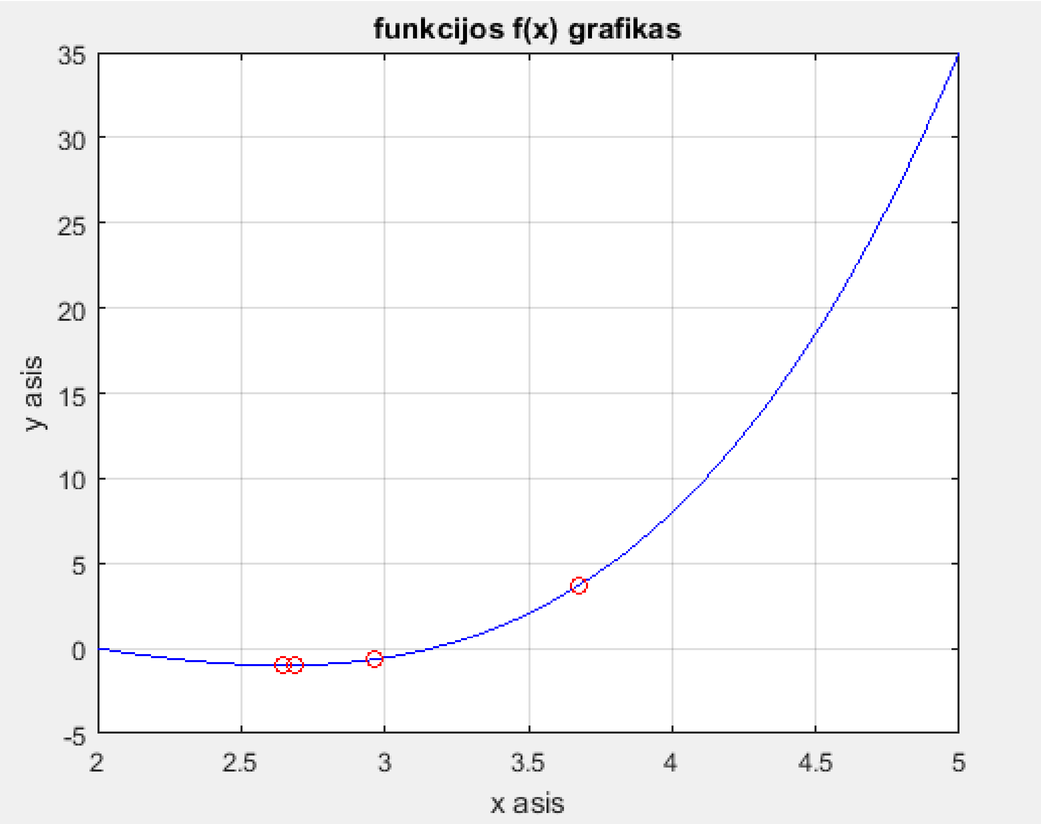
Niutono metodas

- Remdamiesi būtinosiomis minimumo sąlygomis, galime uždavinį spręsti netiesiogiai, pakeisdami jį lygties $f'(x) = 0$ sprendimu.
- Niutono metodas yra vienas iš klasikinių metodų lygčių sistemoms spręsti. Kairiosiose lygčių sistemos pusėse esančios funkcijos ištiesinamos taške X_i ir, išsprendus tiesinę lygčių sistemą, turėtasis taškas pakeičiamas tiesinės lygčių sistemos sprendiniu X_{i+1}.
- Norėdami realizuoti šią idėją ir ištiesinti išvestinės funkciją, pasinaudosime tikslo funkcijos $f(x)$ Teiloro eilutės nariais iki antrosios eilės:

$$f'(x) \approx f'(x_i) + f''(x_i) \cdot (x - x_i).$$

Prilyginę nuliui, gausime iteracinę metodo formulę

$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)}.$$



xm	ym	k	funkcijos kvietimu sk
3.6765	3.7182	1.0000	2.0000
2.9624	-0.6497	2.0000	4.0000
2.6902	-0.9937	3.0000	6.0000
2.6468	-1.0000	4.0000	8.0000
2.6458	-1.0000	5.0000	10.0000
2.6458	-1.0000	6.0000	12.0000

Išvados

Visi algoritmai surado funkcijos f(x) = (x^2 - 7)^2/9 – 1 minimumo tašką xmin ≈ 2,6457 ir minimumą ymin = -1.

Intervalų dalijimo pusiau algoritmas truko 17 iteracijų ir funkcijos kviestos 35 kartus. Auksinio pjūvio algoritmas truko 24 iteracijas (ilgiau, nei pirmasis algoritmas), bet funkcijos kviestos 26 kartus (geriau, nei pirmasis algoritmas). Niutono metodo algoritmas truko 6 iteracijas ir funkcijos kviestos 12 kartų, taigi jis veikė efektyviau už pirmuosius du algoritmus.

PRIEDAI. Programų kodai

```
function DalijimoPusiauAlgoritmas

f=@(x)(x.^2-7).^2/9 - 1;
l=0;
r=10;

epsilon=10^(-4);

k=1;
kmax=100;

x=2:0.01:5;
y=f(x);
plot(x, y, 'b');
grid on;
xlabel('x asis');
ylabel('y asis');
title(['funkcijos f(x) grafikas'])


%Metodo realizavimas
L=r-l;
xm=(l+r)/2;
ym=f(xm);

disp([' xm         ym        k       funkcijos kvietimu sk']);

format short;

while L>=epsilon
    disp([xm, ym, k, 2*k+1]);
    hold on
    plot(xm, ym, 'ro');

    x1=l+L/4;
    x2=r-L/4;
    y1=f(x1);
    y2=f(x2);

    if y1 < ym
        r=xm;
        xm=x1;
        ym=y1;
    else if y2 < ym
        l=xm;
        xm=x2;
        ym=y2;

        else
            l=x1;
```

```
            r=x2;
        end
    end

    if k==kmax
        disp('Pasiektas maksimalus iteraciju skaicius');
        break;
    end
    k=k+1;
    L=r-l;

end
end
```

```
function AuksinioPjuvioAlgoritmas

f=@(x)(x.^2-7).^2/9 - 1;
l=0;   % kairysis intervalo rezis
r=10;  % desinysis intervalo rezis

epsilon=10^(-4); %tikslumas

k=1; %iteraciju skaitliukas
kmax=100; % maksimalus iteraciju skaitliukas

%Funkcijos grafiko y=f(x) braizymas
x=l:0.01:r;
y=f(x);
plot(x,y,'b');
grid on;
xlabel('x asis');
ylabel('y asis');
title(['Funkcijos y=f(x) grafikas ir artiniai']);


%Metodo realizavimas
t=(sqrt(5)-1)/2;
L=r-l;       %intervalo ilgis
x1=r-t*L;
x2=l+t*L;
y1=f(x1);
y2=f(x2);


disp(['     x1        y1         k          funkc. kviet. sk']);

format short;
while L>= epsilon

    disp([x1, y1, k, k+2]);
    hold on;
    plot(x1, y1, 'ro');
```

```
    if y2 < y1

        l=x1;
        L=r-l;
        x1=x2;
        y1=y2;
        x2=l+t*L;
        y2=f(x2);

    else

        r=x2;
        L=r-l;
        x2=x1;
        y2=y1;
        x1=r-t*L;
        y1=f(x1);

    end

      if k==kmax
          disp(['Pasiektas maksimalus iteraciju skaicius k=', num2str(kmax)]);
          break
      end
      k=k+1;
      L=r-l;
end
end
```

```
function NiutonoAlgoritmas

f=@(x)(x.^2-7).^2/9 - 1;
f1=@(x)(4/9)*(x.^3) - (28/9)*x;
f2=@(x)(4/3)*(x.^2) - 28/9;
l=2;
r=5;
x0=5;

epsilon=10^(-4);

k=1;
kmax=100;

x=l:0.01:r;
y=f(x);
plot(x, y, 'b');
grid on;
xlabel('x asis');
ylabel('y asis');
title(['funkcijos f(x) grafikas'])
L=Inf;

disp(['     xm          ym          k    funkcijos kvietimu sk'])

while L>=epsilon
```

```
    x1=x0 - f1(x0)/f2(x0);

    y1=f(x1);
    disp([x1, y1, k, 2*k]);
    hold on
    plot(x1, y1, 'ro');
    if k==kmax
        disp('Pasiektas maksimalus iteraciju skaicius');
        break;
    end
    k=k+1;

    L=abs(x1-x0);
    x0=x1;

end
end
```