

Optimizavimo metodai

Laboratorinis darbas nr. 1
Vienmatis optimizavimas

Tomas Giedraitis
MIF INFO 3 kursas 1 grupė
2019-09-18

Algoritmų trumpiniai:

IDP – Intervalo dalijimo pusiau

AP – Auksinio pjūvio

NM – Niutono metodas

Algoritmų palyginimas

Algoritmas	k (iteracijų kiekis)	Funkcijos iškvietimai	Funkcijos iškvietimų kiekis	Tikslus minimumas	Minimumas (gautas sprendinys), y_m	Vidurinio taško (gauto sprendinio – minimumo atstumas nuo tikslo minimumo)	Argumento reikšmė (minimumo taškas, minimumo artinys) x_m
IDP	17	$2k + 1$	35	-1	-0.999999997871295	0.000000002128705	1.9999694824
AP	24	$k+2$	26	-1	-0.999999998700534	0.000000001299466	2.0000238434
NM	7	$2k$	14	-1	-1.0	0.0	2.0

tikslo funkcija

$$f = \frac{(x^2 - a)^2}{b} - 1$$

a = 4

b = 7

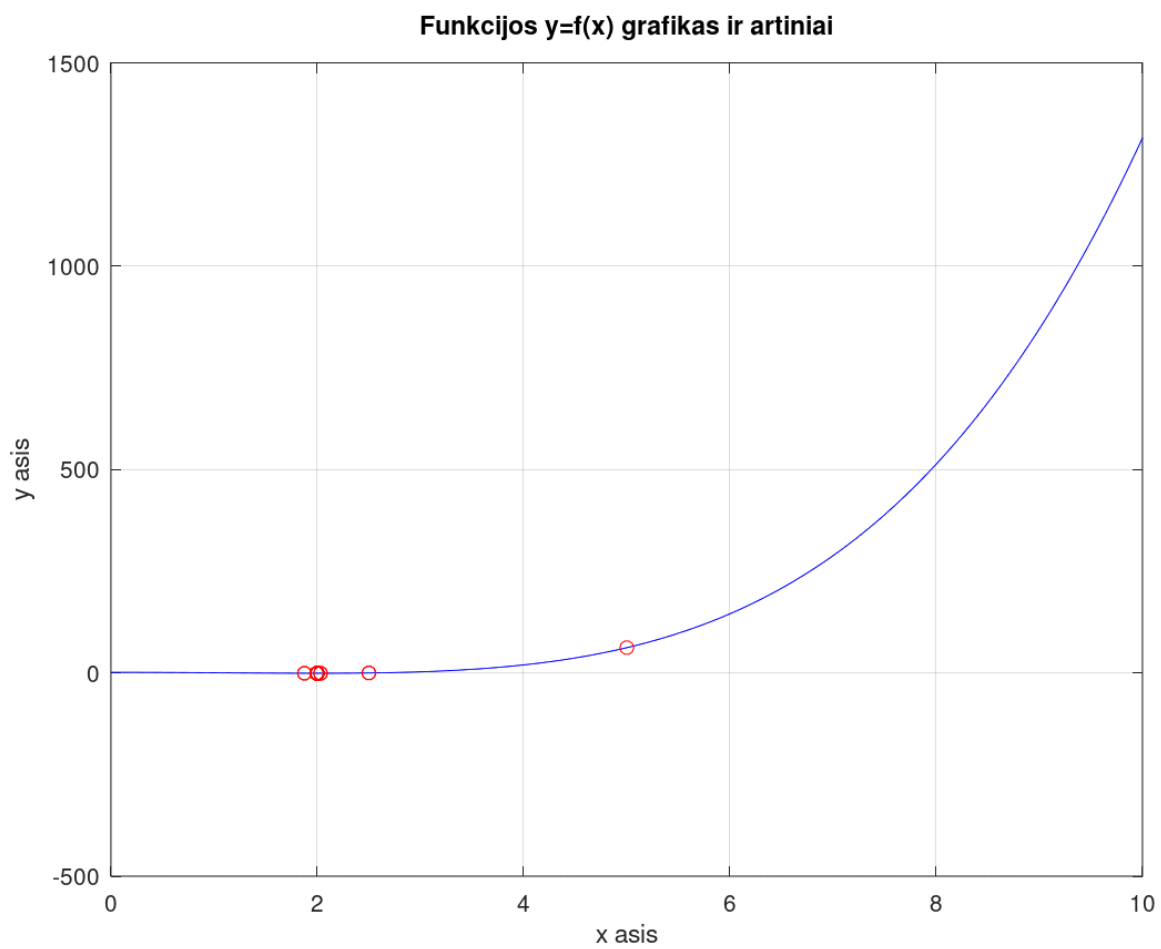
tikslumas = 0.0001

intervalas = [0,10]

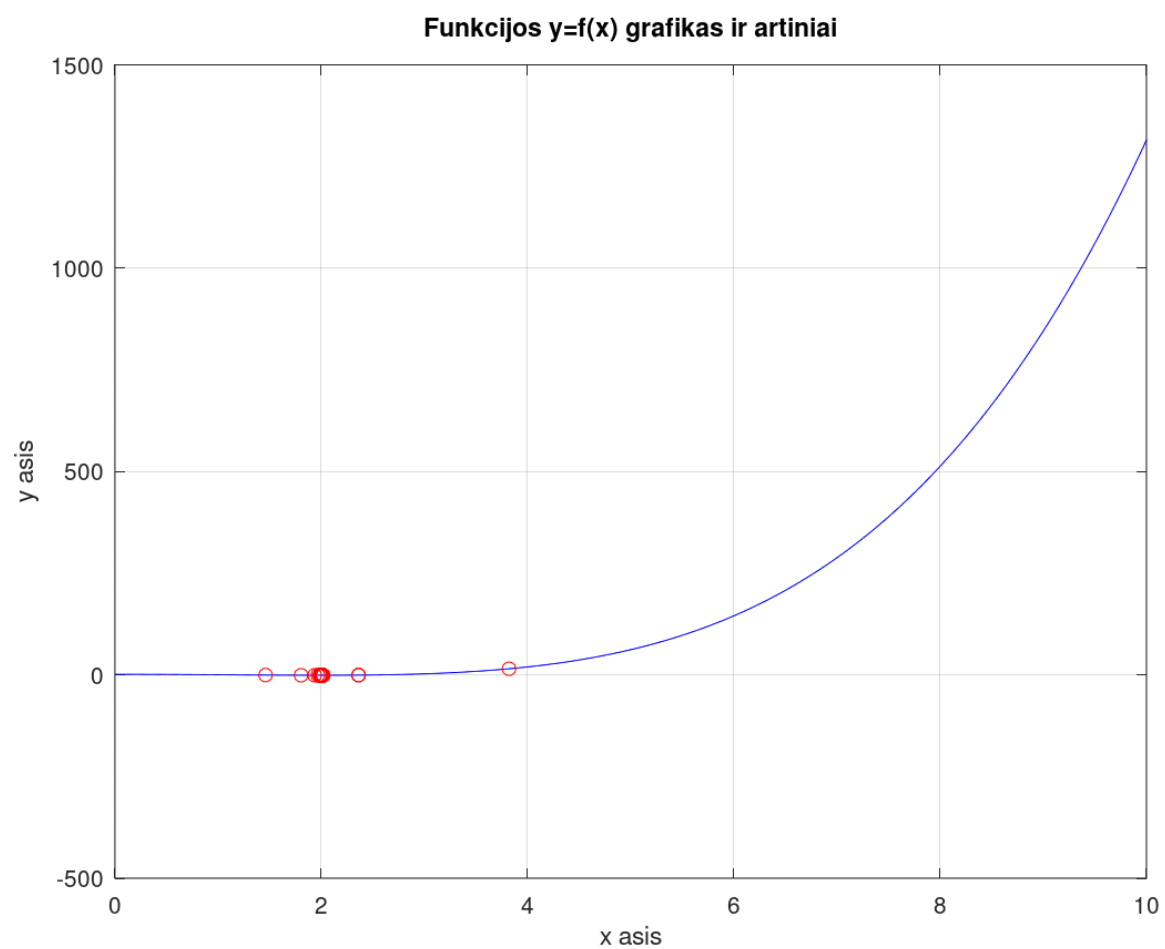
Bandymų išvados:

Kaip matome, IDP metodas artėja greičiau prie minimumo nei AP metodas (mažesnis iteracijų skaičius), tačiau daugiau kartų iškviečia funkciją. Tuo tarpu Niutono metodas pareikalavo daug mažiau iteracijų bei funkcijos kvietimų. Taip pat verta pažymėti, kad AP metode iteracijų skaičius tik nežymiai skiriasi nuo funkcijų kvietimo skaičiaus, priešingai nei kituose metoduose.

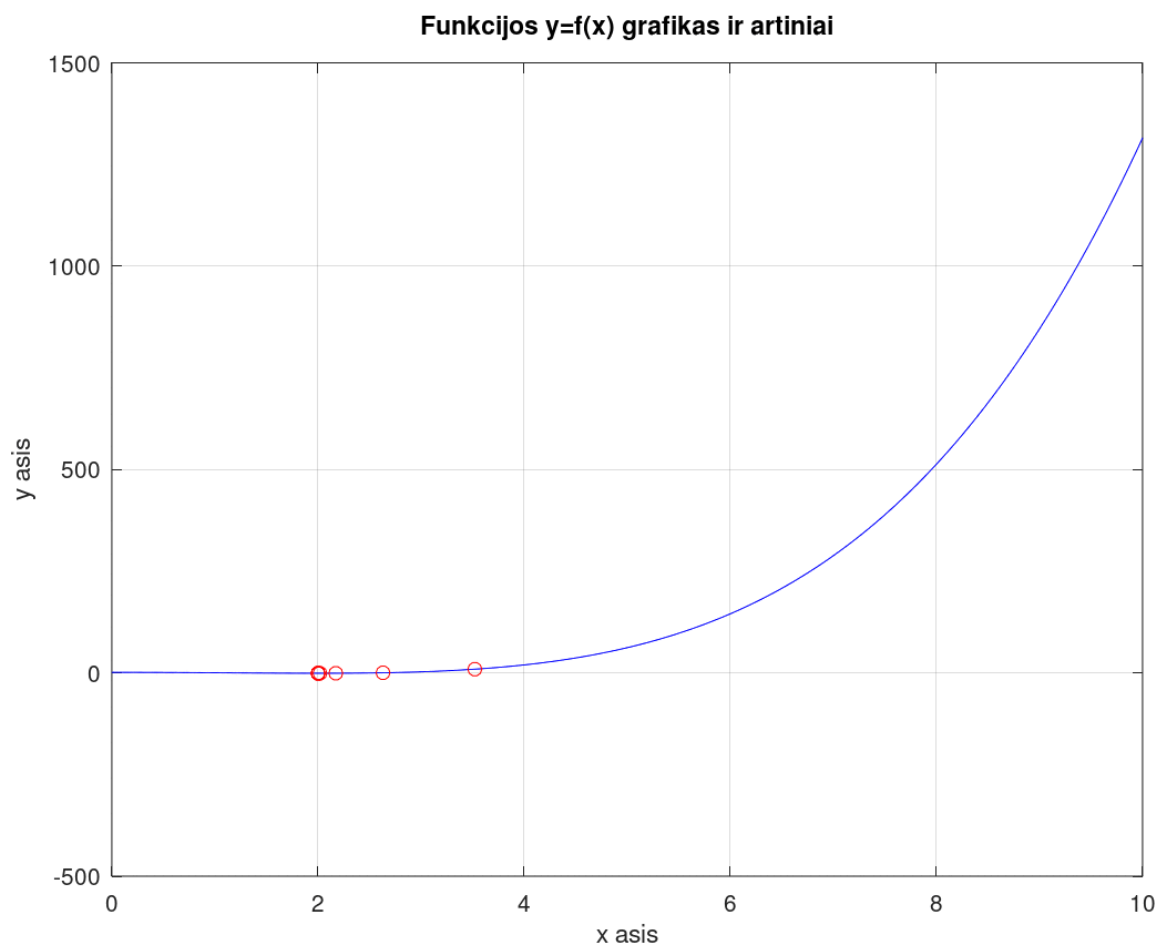
IDP bandymo taškų ir tikslo funkcijos vizualizacija



AP bandymo taškų ir tikslo funkcijos vizualizacija



NM bandymo taškų ir tikslo funkcijos vizualizacija



Programų išvestis (iteracijų rezultatai)

IDP:

xm	ym	k	funkc. kviet. sk		
5			62	1	3
2.500000000000000e+00			-2.767857142857143e-01	2	5
2.500000000000000e+00			-2.767857142857143e-01	3	7
1.875000000000000e+00			-9.664829799107143e-01	4	9
1.875000000000000e+00			-9.664829799107143e-01	5	11
2.031250000000000e+00			-9.977328436715263e-01	6	13
2.031250000000000e+00			-9.977328436715263e-01	7	15
1.992187500000000e+00			-9.998610354959965e-01	8	17
1.992187500000000e+00			-9.998610354959965e-01	9	19
2.001953125000000e+00			-9.999912721749362e-01	10	21
2.001953125000000e+00			-9.999912721749362e-01	11	23
1.999511718750000e+00			-9.999994551762857e-01	12	25
1.999511718750000e+00			-9.999994551762857e-01	13	27
2.000122070312500e+00			-9.999999659381241e-01	14	29
2.000122070312500e+00			-9.999999659381241e-01	15	31
1.999969482421875e+00			-9.999999978712951e-01	16	33
1.999969482421875e+00			-9.999999978712951e-01	17	35

AP:

x1	y1	k	funkc.	kviet.	sk
3.819660112501051e+00	1.502056221740420e+01	1		3	
2.360679774997897e+00	-6.466102641791984e-01	2		4	
1.458980337503154e+00	-4.997072091311895e-01	3		5	
2.360679774997897e+00	-6.466102641791984e-01	4		6	
2.016261237511566e+00	-9.993906693084861e-01	5		7	
1.803398874989484e+00	-9.201237431719084e-01	6		8	
2.016261237511566e+00	-9.993906693084861e-01	7		9	
1.934955049953733e+00	-9.906414475155996e-01	8		10	
2.016261237511566e+00	-9.993906693084861e-01	9		11	
1.985205037360148e+00	-9.995033721769061e-01	10		12	
1.966011250105151e+00	-9.974041462145048e-01	11		13	
1.985205037360148e+00	-9.995033721769061e-01	12		14	
1.997067450256570e+00	-9.999803720162765e-01	13		15	
1.992536411718722e+00	-9.998731486541225e-01	14		16	
1.997067450256570e+00	-9.999803720162765e-01	15		17	
1.999867786077296e+00	-9.999999600471638e-01	16		18	
1.998798152973692e+00	-9.999967004150644e-01	17		19	
1.999867786077296e+00	-9.999999600471638e-01	18		20	
1.999459222587211e+00	-9.999993317459589e-01	19		21	
1.999867786077296e+00	-9.999999600471638e-01	20		22	
2.000120292200730e+00	-9.999999669232368e-01	21		23	
2.000023843443946e+00	-9.999999987005335e-01	22		24	
1.999964234834080e+00	-9.999999970762875e-01	23		25	
2.000023843443946e+00	-9.999999987005335e-01	24		26	

NM:

x1	y1	k	funkc. kviet. sk		
3.521126760563380			9.076001190161325	1	2
2.630281858478884e+00			2.167081887509414e-01	2	4
2.172143919641523e+00			-9.263107905810557e-01	3	6
2.018514084916996e+00			-9.992092531830250e-01	4	8
2.000251644239276e+00			-9.999998552393347e-01	5	10
2.000000047479678e+00			-9.999999999999949e-01	6	12
2.0000000000000002			-1.0000000000000000	7	14

Programų rašymui naudota priemonė: Octave

IDP kodas:

```
function DalijimasPusiau
% Dalijimo pusiau metodu randamas funkcijos f(x) minimumas intervale [l,r].

f=@(x)(((x.^2-4).^2)/7) - 1;

l=0; % apatinis intervalo rezis
r=10; % desinysis intervalo rezis

epsilon=10^(-4); %tikslumas

k=1; %iteracijų skaitliukas
kmax=100; % maksimalus iteracijų 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
L=r-l; %intervalo ilgis
xm=(l+r)/2; %intervalo vidurio taskas
ym=f(xm);

disp(['  xm      ym      k      funkc. kviet. sk']);

format long

while L>= epsilon

    format long
    disp([xm, ym]);
    format short
    disp([k, 2*k+1]);

    hold on;
    plot(xm, ym, 'ro');

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

    if y1 < ym
        r = xm;
        xm = x1;
        ym = y1;
```

```

        % ciklas tesiasi
elseif y2 < ym
    l = xm;
    xm = x2;
    ym = y2;
else
    l = x1;
    r = x2;
end

if k==kmax
    format short
    disp(['Pasiektas maksimalus iteraciju skaicius k=', num2str(kmax)]);
    break
end

k=k+1;
L=r-l;
end
end

```

AP kodas:

```
function AuksinisPjuvis
% Aksinio pjuvio metodu randamas funkcijos f(x) minimumas intervale [l,r].

f=@(x)(((x.^2-4).^2)/7) - 1;

l=0; % apatinis 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
%L=r-l; %intervalo ilgis
%xm=(l+r)/2; %intervalo vidurio taskas
%ym=f(xm);

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

format long

gR = (sqrt(5) - 1) / 2;

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

while L>= epsilon
    format long
    disp([x1, y1]);
    format short
    disp([k, k+2]);

    hold on;
    plot(x1, y1, 'ro');

    if y2 < y1
        l = x1;
```

```

    L = r - l;
    x1 = x2;
    y1=y2;
    x2 = l + gR*L;
    y2 = f(x2);
else
    r = x2;
    L = r - l;
    x2 = x1;
    y2=y1;
    x1 = r - gR*L;
    y1 = f(x1);
end

if k==kmax
    format short
    disp(['Pasiekta maksimalus iteracijų skaičius k=', num2str(kmax)]);
    break
end

k=k+1;
L=r-l;
end
end

```

NM kodas:

```
function NiutonoMetodas
% Niutono metodu randamas funkcijos f(x) minimumas intervale [l,r].

f=@(x)((x.^2-4).^2)/7-1;
f1=@(x)4*x*(x.^2-4)/7;
f2=@(x)4*(3*x.^2-4)/7;

l=0; % apatinis 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

x0=5;
delta=1;
disp([' x1      y1      k      funkc. kviet. sk']);

format long

while delta>=epsilon
    x1=x0-f1(x0)/f2(x0);
    y1=f(x1);
    delta=abs(x1-x0);
    x0=x1;

    format long
    disp([x1, y1]);
    format short
    disp([k, 2*k]);

    hold on
    plot(x1, y1, 'ro')

    if k==kmax
        format short
        disp(['Pasiektas maksimalus iteraciju skaicius k=', num2str(kmax)]);
        break
    end
    k=k+1;
end
end
```