Vilniaus universitetas

Informatika

3 kursas 2 grupė 1 pogrupis

Optimizavimo metodai

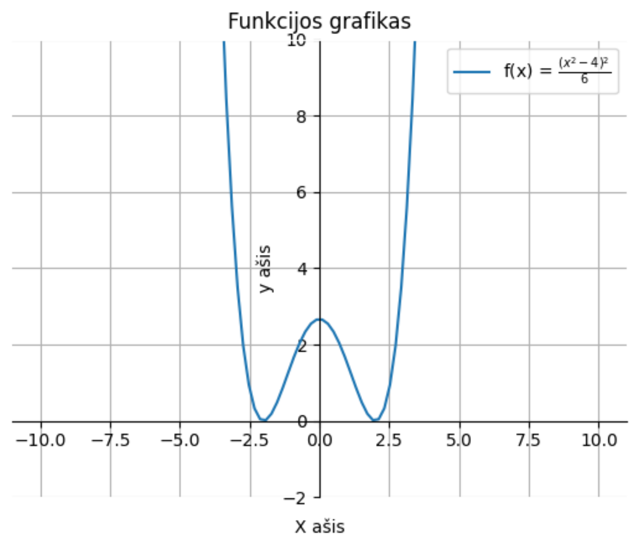
1 Laboratorinis darbas vienmatis optimizavimas

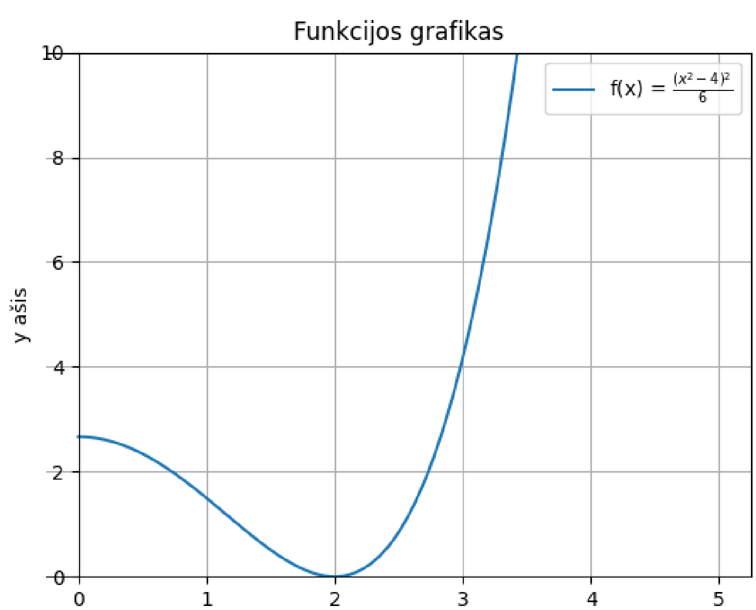
Vilnius 2022

Laboratorinio darbo tikslas

Suprogramuoti vienmačio optimizavimo intervalo dalijimo pusiau, auksinio pjūvio ir Niutono metodo algoritmus.

Tikslo funkcija: $f(x) = \frac{(x^2-4)^2}{6}$ Minimizuoti šią funkciją intervalo dalijimo pusiau ir auksinio pjūvio metodais intervale [0, 10] iki tikslumo 10^{-4}, bei Niutono metodu nuo $X_0 = 5$, kol žingsnio ilgis bus didesnis už 10^{-4}





Vilnius 2022

Auksinio pjūvio algoritmas

GR (Fibonačio skaičius) = $\frac{\sqrt{5}-1}{2}$ = 0,61803

1. L = b − a, X_1= b – GR * L ir X_2= a + GR *L, skaičiuojame $f(X_1)\ ir\ f(X_2)$

2. jei $f(X_2) < f(X_1)$ tai:

    2.1 atmetamas [a,X_1) atliekant keitimą a = x1, L = b − a;

    2.2 kairiuoju tašku tampa ankstesnis dešinysis taškas x1 = x2;

    2.3 naujasis dešinysis taškas X_2= a + GR * L, skaiciuojame ˇ f(x2);

3. priešingu atveju:

    3.1 atmetamas (x2, b] atliekant keitimą b =X_2, L = b − a;

    3.2 dešiniuoju tašku tampa ankstesnis kairysis taškas$X_2 = X_1$;

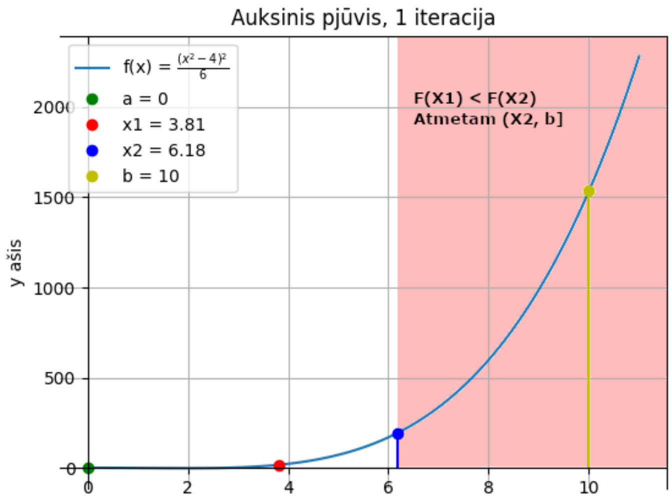
    3.3 naujasis kairysis taškas$X_1 = b - GR * L$, skaiciuojame $f(X_1)$;

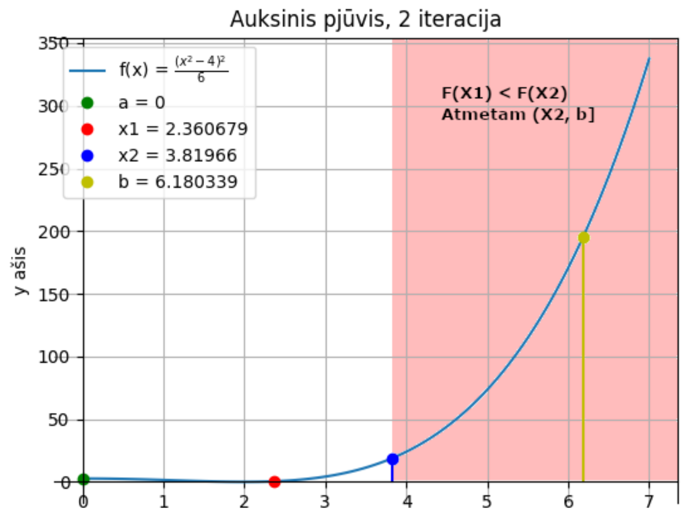
4. jei L pakankamai mažas (L < ε) ε = 0.0004, skaičiavimus baigiame, jei ne – einame į 2 punktą.

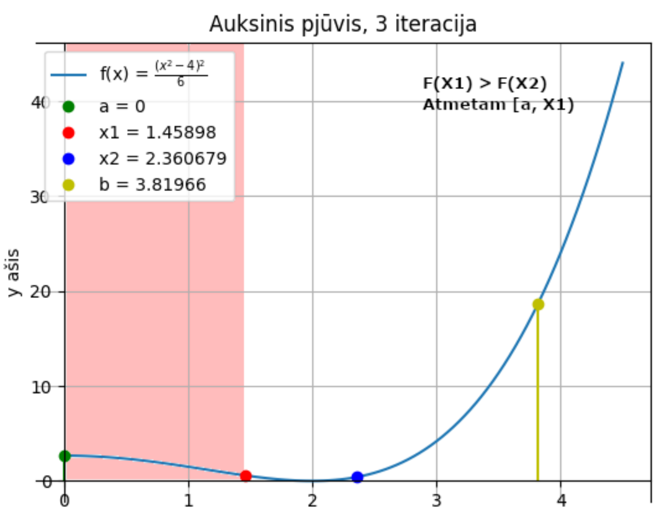
Paskaičiuokime kelias iteracijas ranka:

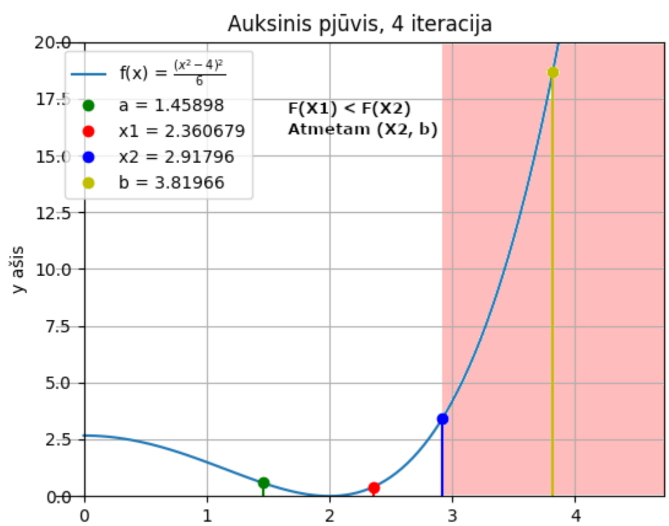
1. $L = 10 - 0 = 10,\ X_1 = 10 - 0.61803 * 10 = 3.81966,\ X_2 = 0 + 0.61803 * 10 = \ 6.180339,$ $f(X_1) = \ 18.69065,\ f(X_2) = \ 194.90125$
    a. $Kadangi, f(X_1) < f(X_2), atmetame\ (X_2, b], b\ patampa\ X_2, skaičiuojame,\ L = 6.18033 - 0 = 6.18033$
    b. Skaičiuojame $X_1 = 6.18033 - 0.61803 * 6.18033 = \ 2.36067, f(X_1) = \ 0.412288$
2. L = 6.18033, $X_1(jau\ suskaičiavome) = 2.36067, X_2 = \ 0 + 0.61803 * 6.18033 = \ 3.81966$, $f(X_1) = \ 0.41228,\ f(X_2) = \ 18.690655$
    a. $Kadangi, f(X_1) < f(X_2), atmetame\ (X_2, b], b\ patampa\ X_2, skaičiuojame,\ L = 3.819660 - 0 = 3.819660$
    b. Skaičiuojame $X_1 = 3.819660 - 0.61803 * 3.819660 = \ 1.458980, f(X_1) = 0.583674$
3. L = 3.819660, $X_1 = \ 1.458980, X_2$ =0 + 0.61803 $*$ 3.819660 = 2.36067, $f(X_1) = \ 0.583674,$ $f(X_2) = \ 0.412288$
    a. $Kadangi, f(X_1) > \ f(X_2), atmetame$ [a, $X_1), a\ patampa\ X_1, skaičiuojame\ L = 3.819660 - \ 1.458980 = 2.36068$
    b. Skaičiuojame $X_2 = 1.458980 + 0.61803 * 2.36068 = \ 2.9179606, f(X_2) = 3.3967767$

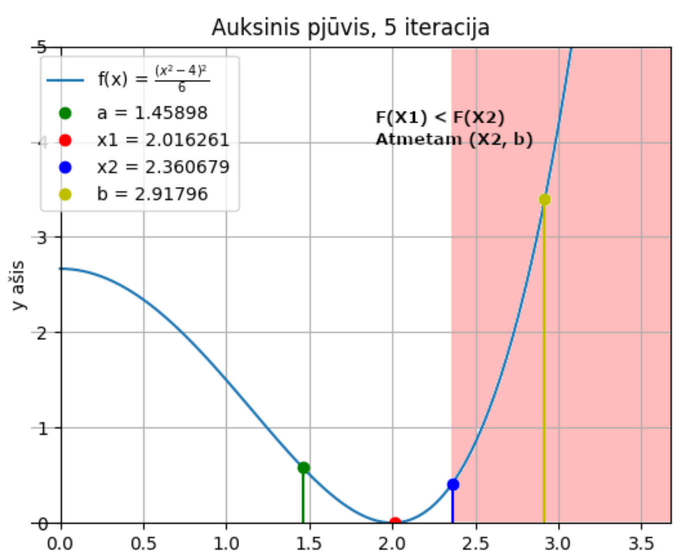
```python
def goldenSearch(intervalA, intervalB):
    #for c in range(10):
    GR = (math.sqrt(5) - 1)/2
    for iteratorius in range(100):
        print("iteracija nr:", iteratorius+1)
        L = intervalB - intervalA  # GR * (Intervalo pabaiga - Intervalo pradzia) # CIA YRA L
        intervalx1 = intervalB - GR * L
        intervalx2 = intervalA + GR * L
        fx1Reiksme = f(intervalx1)
        fx2Reiksme = f(intervalx2)
        round(fx1Reiksme, 4)
        round(fx2Reiksme, 4)
        printData("gold",intervalA, intervalx1, 0, intervalx2 ,intervalB, fx1Reiksme,0, fx2Reiksme)
        if(L < 0.0001):
            break
        if fx1Reiksme < fx2Reiksme:
            intervalB = intervalx2
        elif fx1Reiksme > fx2Reiksme:
            intervalA = intervalx1
```

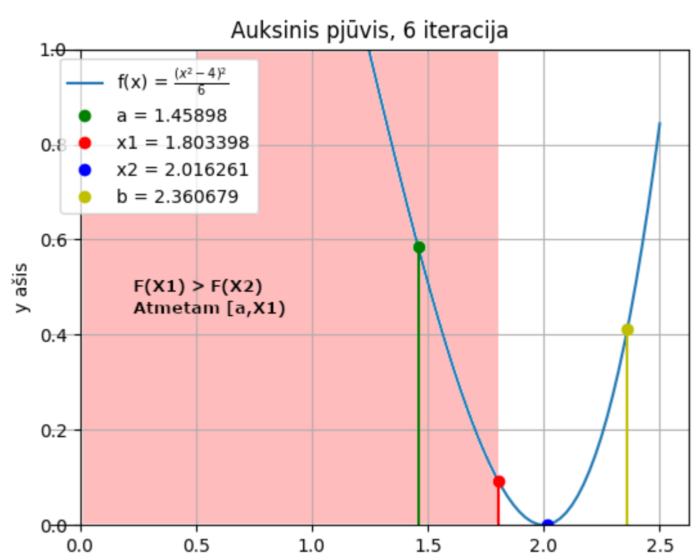

Auksinis pjūvis, 1 iteracija
f(x) = $\frac{(x^2-4)^2}{6}$
a = 0
x1 = 3.81
x2 = 6.18
b = 10
F(X1) < F(X2)
Atmetam (X2, b]
y ašis


Auksinis pjūvis, 2 iteracija
f(x) = $\frac{(x^2-4)^2}{6}$
a = 0
x1 = 2.360679
x2 = 3.81966
b = 6.180339
F(X1) < F(X2)
Atmetam (X2, b]
y ašis


Auksinis pjūvis, 3 iteracija
f(x) = $\frac{(x^2-4)^2}{6}$
a = 0
x1 = 1.45898
x2 = 2.360679
b = 3.81966
F(X1) > F(X2)
Atmetam [a, X1)
y ašis


Auksinis pjūvis, 4 iteracija
f(x) = $\frac{(x^2-4)^2}{6}$
a = 1.45898
x1 = 2.360679
x2 = 2.91796
b = 3.81966
F(X1) < F(X2)
Atmetam (X2, b)
y ašis


Auksinis pjūvis, 5 iteracija
f(x) = $\frac{(x^2-4)^2}{6}$
a = 1.45898
x1 = 2.016261
x2 = 2.360679
b = 2.91796
F(X1) < F(X2)
Atmetam (X2, b)
y ašis


Auksinis pjūvis, 6 iteracija
f(x) = $\frac{(x^2-4)^2}{6}$
a = 1.45898
x1 = 1.803398
x2 = 2.016261
b = 2.360679
F(X1) > F(X2)
Atmetam [a,X1)
y ašis

Intervalo dalijimas pusiau algoritmas

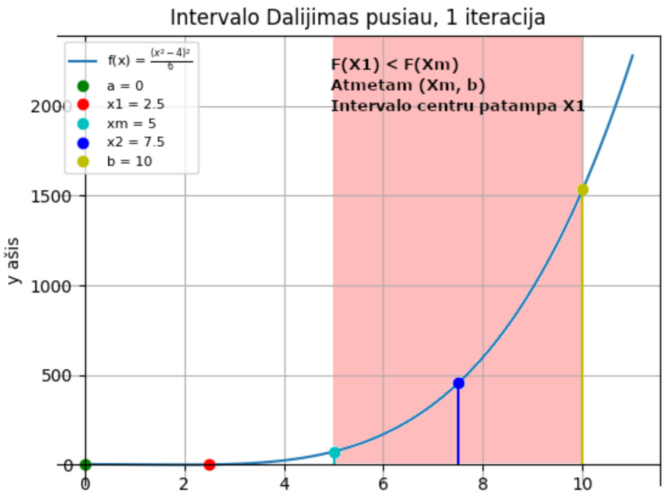
Intervalo dalijimo pusiau algoritmas: Turime du pradinius taškus a ir b (intervalas kuriame ieškome minimumo taško)

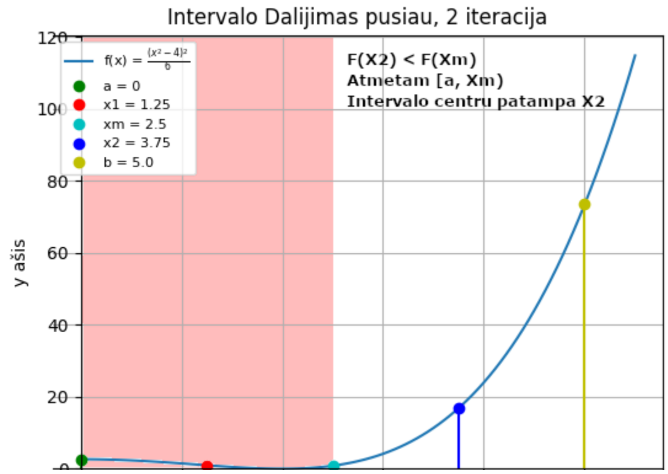
1. Apsiskaičiuojame vidurinį taškąX_m, $X_m = \frac{a+b}{2}$, apsiskaičiuojame L (skirtumas tarp a ir b) = b – a, $f(X_m)$
2. Apsiskaičiuojame $X_1 = a + \frac{L}{4}$, $X_2 = b - \frac{L}{4}$, $f(X_1)$, $f(X_2)$
3. Jeigu $f(X_1) < f(X_m)$:
    a. Atmetame intervalą (X_m, b] atliekant keitimą $b = X_m$
    b. Intervalo centru patampa X_1, tad keičiamas $X_m = X_1$
    c. Skaičiuojame L = b – a, jeigu L < 0.0004 baigiame, jeigu ne – einame į 2 punktą
4. Jeigu $f(X_2) < f(X_m)$:
    a. Atmetame intervalą [a, X_m) atliekant keitimą a = X_m
    b. Interval centru patampa X_2, tad keičiamas $X_m = X_2$
    c. Skaičiuojame L = b – a, jeigu L < 0.0004 baigiame, jeigu ne – einame į 2 punktą
5. Jeigu $f(X_1) \geq f(X_m)$ ir $f(X_x) \geq f(X_m)$:
    a. Atmetame intervalus [a, X_1) ir (X_2, b] atliekant keitimus a = X_1 ir $b = X_2$
    b. Skaičiuojame L = b – a, jeigu L < 0.0004 baigiame, jeigu ne – einame į 2 punktą

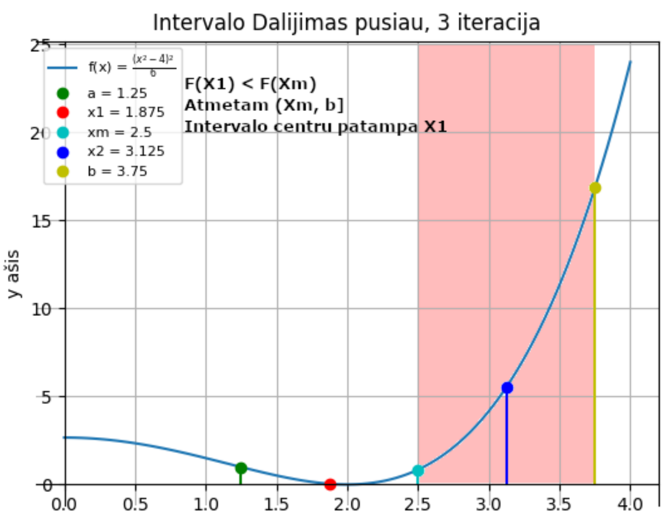
Paskaičiuokime kelias iteracijas "ranka":

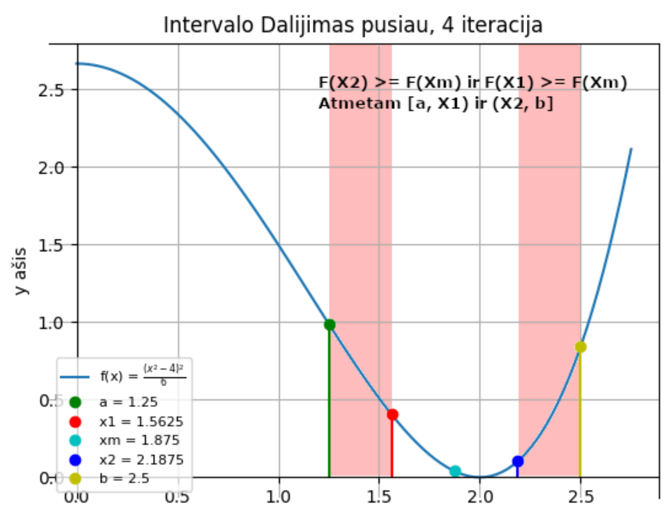
1. $a = 0,\ b = 10,\ X_m = \frac{0+10}{2} = 5, f(X_m) = 73.5,\ L = 10 - 0 = 10$
    a. $X_1 = 0 + \frac{10}{4} = 2.5, X_2 = 10 - \frac{10}{4} = 7.5, f(X_1) = 0.84375,\ f(X_2) = 455.01041$
    b. Kadangi, $f(X_1) < f(X_m) atmetame$ $(X_m$, b], intervalo centru patampa $X_1, skaičiuojame\ L = 5 - 0 = 5$
2. $A = 0, b = 5,\ X_m = \frac{0+5}{2} = 2.5, f(X_m) = 0.84375,\ L = 5$
    a. $X_1 = 0 + \frac{5}{4} = 1.25, X_2 = 5 - \frac{5}{4} = 3.75, f(X_1) = 0.990234, f(X_2) = 16.875651$
    b. Kadangi, $f(X_1) \geq f(X_m)\ ir\ f(X_x) f(X_m)\ atmetame\ intervalus$ [a, X_1)ir (X_2, b], L = 3.75 – 1.25 = 2.5
3. $a = 1.25,\ b = 3.75,\ X_m = \frac{1.25+3.75}{2} = 2.5, f(X_m) = 0.84375,\ L = 2.5$
    a. $X_1 = 1.25 + \frac{2.5}{4} = 1.875, X_2 = 3.75 - \frac{2.5}{4} = 3.125, f(X_1) = 0.039103,\ f(X_2) = 5.54040$
    b. Kadangi, $f(X_1) < f(X_m) atmetame$ $(X_m$, b], intervalo centru patampa $X_1, skaičiuojame\ L = 2.5 - 1.25 = 1.25$

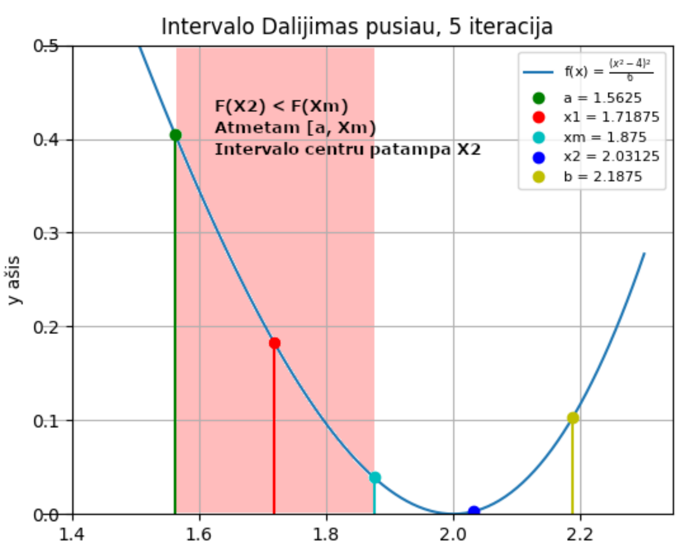
```
def sectionSearch(intervalA, intervalB):
    # intervalA = l  # intervalB = r
    for iteratorius in range(50):
        print("iteracija nr:", iteratorius+1)
        intervalxm = (intervalA+intervalB)/2
        L = intervalB - intervalA
        intervalx1 = intervalA + (L/4)
        intervalx2 = intervalB - (L/4)
        fx1Reiksme = f(intervalx1)
        fx2Reiksme = f(intervalx2)
        fxmReiksme = f(intervalxm)
        printData("split",intervalA, intervalx1, intervalxm, intervalx2, intervalB, fx1Reiksme,fxmReiksme, fx2
        if fx1Reiksme < fxmReiksme:
            intervalB = intervalxm
            intervalxm = intervalx1
        elif fx2Reiksme < fxmReiksme:
            intervalA = intervalxm
            intevalxm = intervalx2
        elif fx1Reiksme >= fxmReiksme and fx2Reiksme >= fxmReiksme:
            intervalA = intervalx1
            intervalB = intervalx2
        else:
            print("Klaida")
        if(L < 0.000001):
            break
```

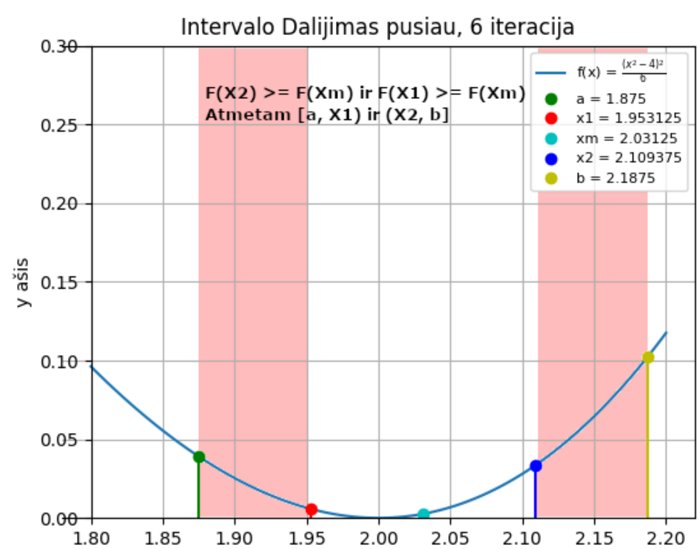












Vilnius 2022

Niutono metodas

Niutono metodo algoritmo iteracinė formulė:

$$X_{i+1} = X_i - \frac{f'(X_i)}{f''(X_i)}$$

Mano parašytas Niutono metodo algoritmo kodas:

```
def newtonsMethod1(x0):

    for iteratorius in range(1000):
        print("iteracija: ", iteratorius+1)
        print(f"reiksme x{iteratorius} : ", x0)
        theOneBefore = x0
        if(secondDegreeDerivative(theOneBefore) <= 0)
            print("Klaida, vardiklis <= 0")
            break
        x0 = theOneBefore - ((firstDegreeDerivative(theOneBefore))/(secondDegreeDerivative(theOneBefore)))
        zingsnioIlgis = theOneBefore - x0
        print("Zingsnio ilgis: ", zingsnioIlgis)
        if(zingsnioIlgis < 0.0002):
            break
```

Paskaičiuokime "ranka":

$$X_{0+1} = X_0 - \frac{f'(X_{0)}}{f''(X_{0)}} = 5 - \frac{f'(5)}{f''(5)} = 5 - \frac{70}{47.3333} = 5 - 1.478873 = 3.521126$$

$$X_2 = X_1 - \frac{f'(X_1)}{f''(X_1)} = 3.521126 - \frac{f'(3.521126)}{f''(3.521126)} = 3.521126 - 0.890844 = 2.630281$$

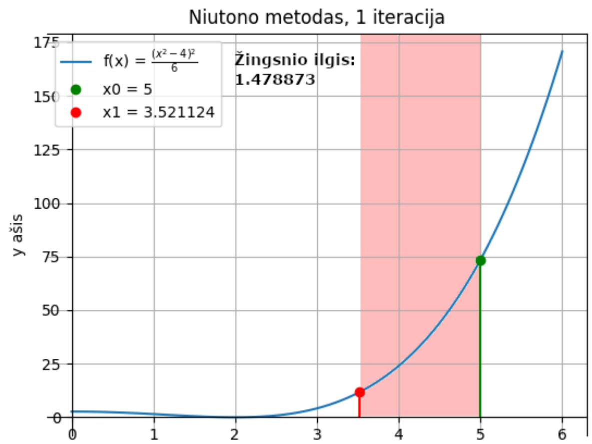
$$X_3 = X_2 - \frac{f'(X_2)}{f''(X_2)} = 2.630281 - \frac{f'(2.630281)}{f''(2.630281)} = 2.630281 - 0.458137 = 2.172143$$

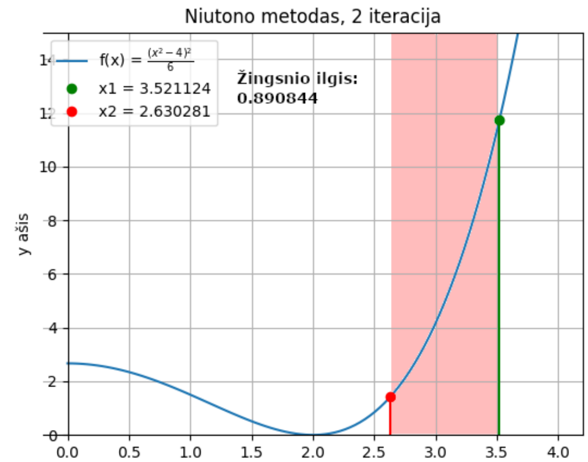
$$X_4 = X_3 - \frac{f'(X_3)}{f''(X_3)} = 2.172143 - \frac{f'(2.172143)}{f''(2.172143)} = 2.172143 - 0.153629 = 2.018514$$

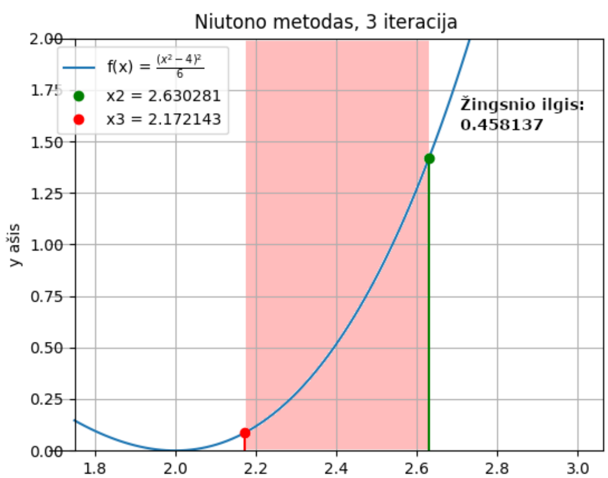
$$X_5 = X_4 - \frac{f'(X_4)}{f''(X_4)} = 2.018514 - \frac{f'(2.018514)}{f''(2.018514)} = 2.018514 - 0.018262 = 2.000251$$

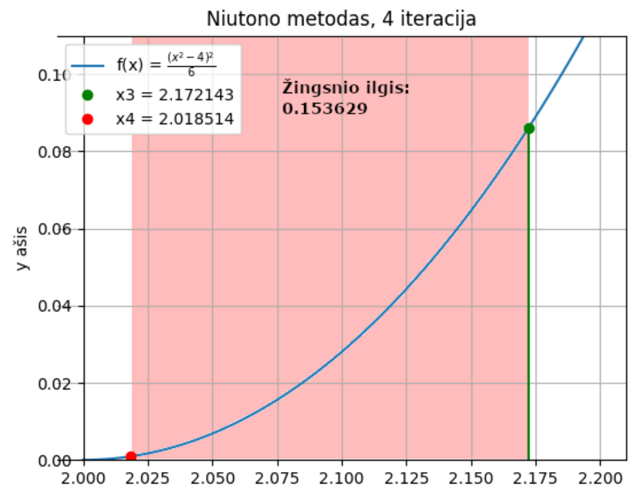
$$X_6 = X_5 - \frac{f'(X_5)}{f''(X_5)} = 2.000251 - \frac{f'(2.000251)}{f''(2.000251)} = 2.000251 - 0.018262 = 2.000000047$$

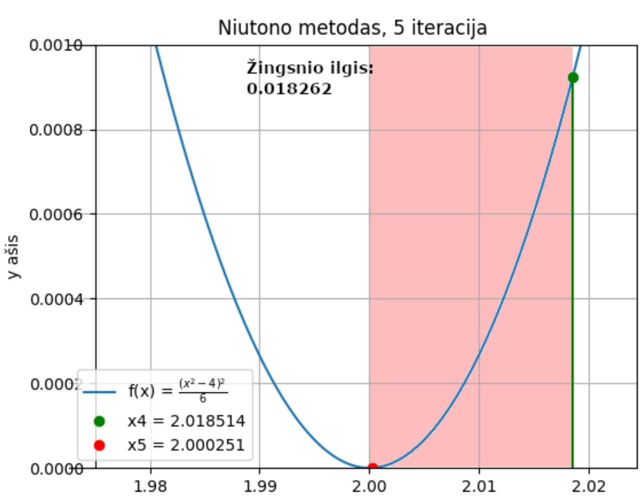
Nutraukiame iteracijas, iteracijos žingnis $2.000251 - 2.000000047 = 0.00025$ < 0.0004

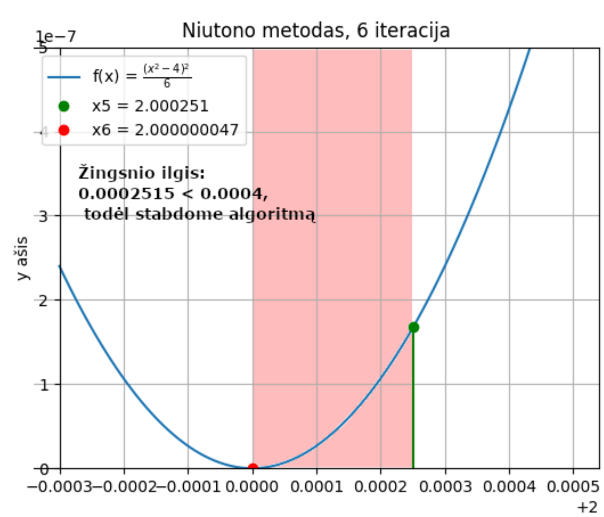












Vilnius 2022

Algoritmas	Iteracijų sk.	Minimumo įvertis
Niutono	5	2.00000004
Intervalo	25	1.9999997
Auksinio pjūvio	25	2.000012