KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

1 Laboratorinis darbas
Nr. 17

Atliko:
IFE-8 gr. studentas
Kemežys Martynas

Priėmė:
lekt. Andrius Kriščiūnas

KAUNAS, 2020

TURINYS

1. UŽDUOTIS

1. Išspręsti netiesines lygtis: a) Daugianaris f(x) = 0, b) Transcendentinė funkcija g(x) = 0

17	$1.03x^5 - 2.91x^4 - 1.44x^3 + 5.56x^2 - 0.36x - 1.13$	$\sin(x)\ln(x) - \frac{x}{6}; 1 \le x \le 20$	1, 3, 5

Pav. 1

2. Pagal pateiktą uždavinio sąlygą sudaryti netiesinę lygtį ir pasirinktu skaitiniu metodu ją išspręsti.

Uždavinys variantams 16-20

T_0 temperatūros kūnas patalpinamas į aplinką, kurios temperatūra T_A. Tariama, kad aplinkos temperatūra yra palaikoma išorinių šaltinių ir kūno temperatūra neturi įtakos aplinkos temperatūrai. Kūno temperatūra $T(t)$ užrašoma dėsniu $T(t) = (T_0 - T_A)e^{kt} + T_A$, k – proporcingumo koeficientas. Kam lygus proporcingumo koeficientas k, jeigu žinoma, kad praėjus laikui t_1 kūno temperatūra bus T_1?

Varianto Nr.	T_0, K	T_A, K	t_1, s	T_1, K
17	400	320	30	344

Pav. 2

2. PAGRINDINĖ DALIS

2.1 Daugianario "Grubus" šaknų įvertis

Daugianaris:

$$1.03x^5 - 2.91x^4 - 1.44x^3 + 5.56x^2 - 0.36x - 1.13$$

Grubaus įverčio formulė:

$$R = |x| < 1 + \frac{\max\limits_{0 \le i \le n-1} |a_i|}{a_n}$$

Įsistate kof. gauname:

$$R = 1 + \frac{5.56}{1.03} \approx 6{,}398$$

Gautas grubus šaknies įvertis:

$$-6.398 < x < 6.398$$

2.2 Daugianario "Tikslesnis" šaknų įvertis

Tikslesnio įverčio formulės:

$$R_{teig} = 1 + \sqrt[k]{\frac{B}{a_n}}$$

$$k = n - \max_{0 \le i \le n-1} (i, a_i < 0)$$

$$B = \max_{0 \le i \le n-1} (|a_i|, a_i < 0)$$

Pilna įverčio formulė:

$$-\min(R, Rneig) \le x \le \min(R, Rteig)$$

Skaičiuojame Rteig:

$$k = 5 - 4 = 1$$
$$B = 5{,}56$$
$$Rteig = 1 + \frac{2{,}91}{1{,}03} \approx 3{,}825$$

Skaičiuojame Rneig:

$$1.03x^5 + 2.91x^4 - 1.44x^3 - 5.56x^2 - 0.36x + 1.13$$

$$k = 5 - 3 = 2$$
$$B = 5{,}56$$
$$Rneig = 1 + \sqrt{\frac{5{,}56}{1{,}03}} \approx 3{,}32$$

Gauname tikslesnį šaknies įvertį:

$$-3{,}32 \le x \le 3{,}825$$

2.3 Daugianario šaknų radimas

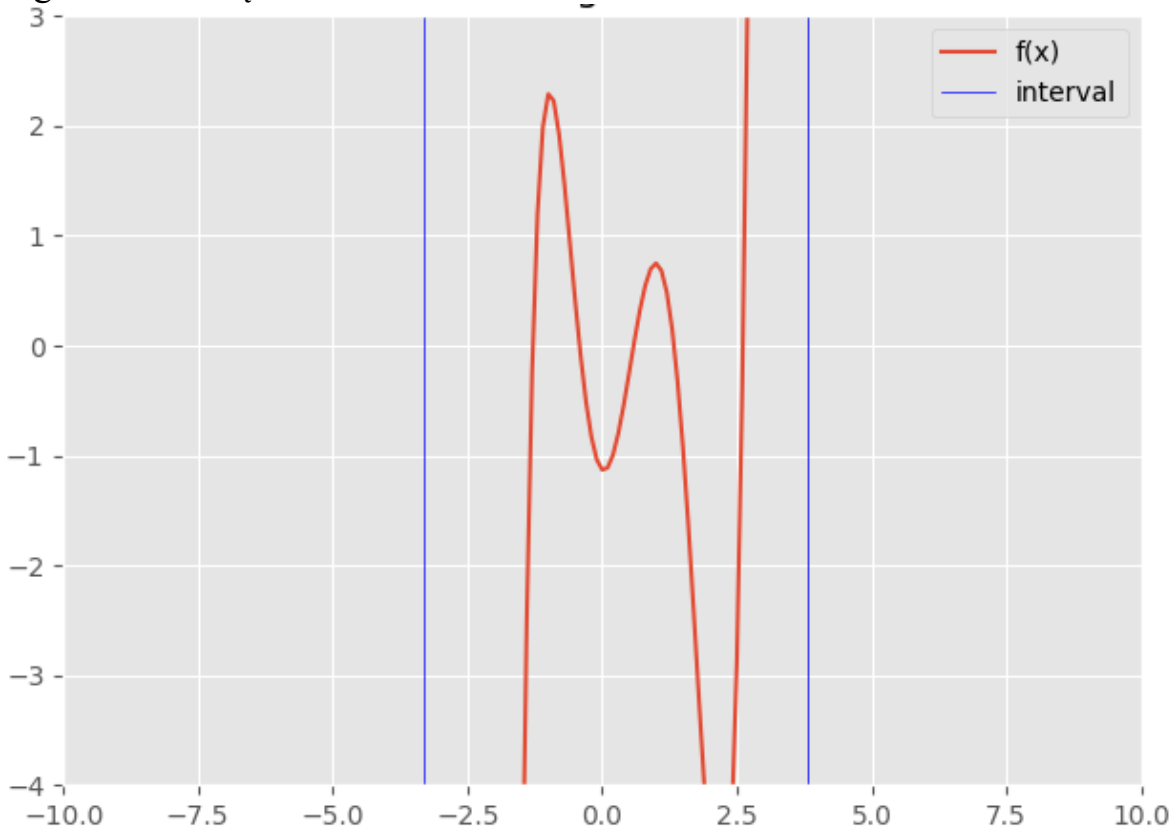

Diagrama. 1

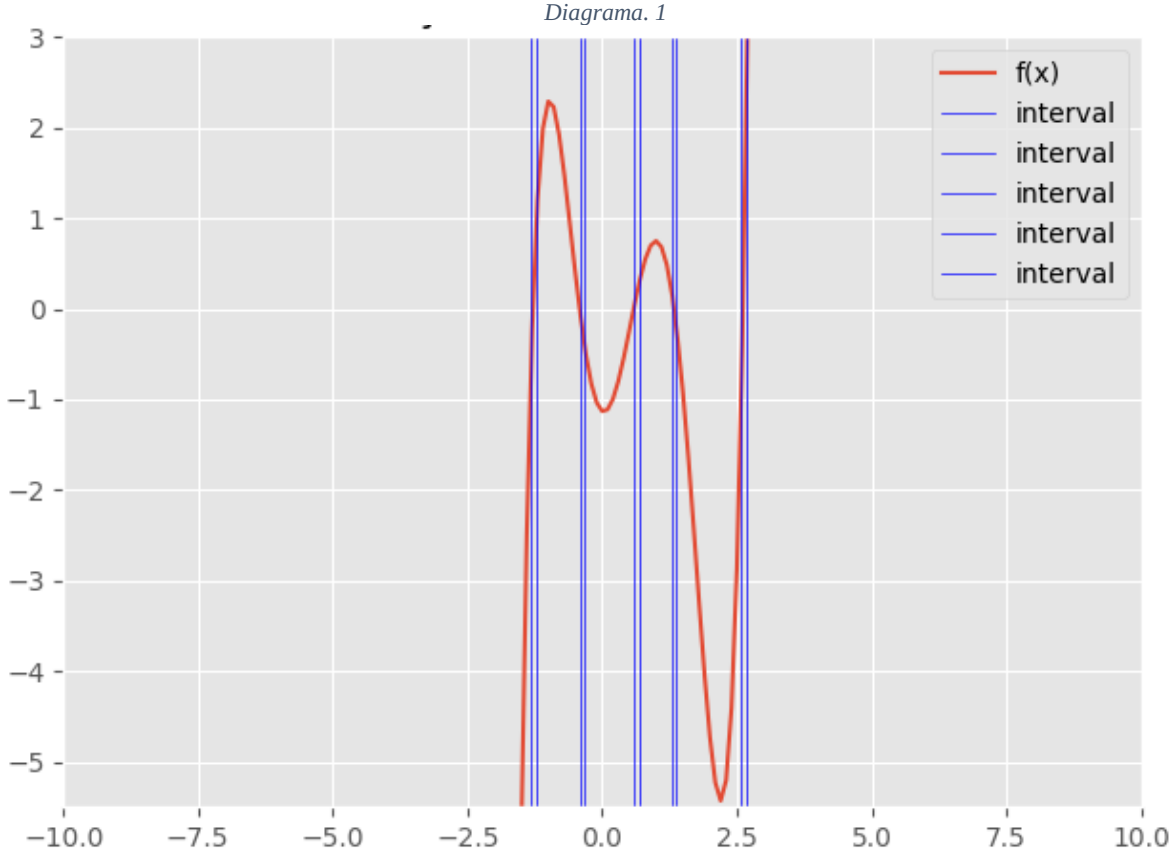

Diagrama. 2

Gautas šaknis tiksliname su „Stygų“, „Liestinių“ ir „Skenavimo su mažėjančiu žingsniu“ metodais.

Metodas	Intervalas	Šaknis	Paklaida	Iteracijos
Stygų	[-1.319999999999998,-1.219999999999998]	-1,287	-0,3751	1
Liestinių		-1,307	1,135E-09	8
Skenavimo		-1,287	-6,34E-09	41
Stygų	[-0.4199999999999793, -0.3199999999999795]	-0,419	0,009525	1
Liestinių		-0,421	-2,23E-10	3
Skenavimo		-0,419	9,416E-09	33
Stygų	[0.58000000000000021, 0.680000000000002]	0,584	-0,01437	1
Liestinių		0,579	5,724E-10	3
Skenavimo		0,584	-7,64E-09	35
Stygų	[1.28000000000000022, 1.3800000000000023]	1,334	0,1994	1
Liestinių		1,287	1,141E-09	7
Skenavimo		1,334	7,643E-09	37
Stygų	[2.58000000000000003, 2.6800000000000033]	2,613	-1,829	1
Liestinių		2,547	-4,01E-09	9
Skenavimo		2,613	-6,76E-09	45

Lentele. 1

To paties daugianario šaknis randame „Desmos Graphing Calculator“ įrankio pagalba:

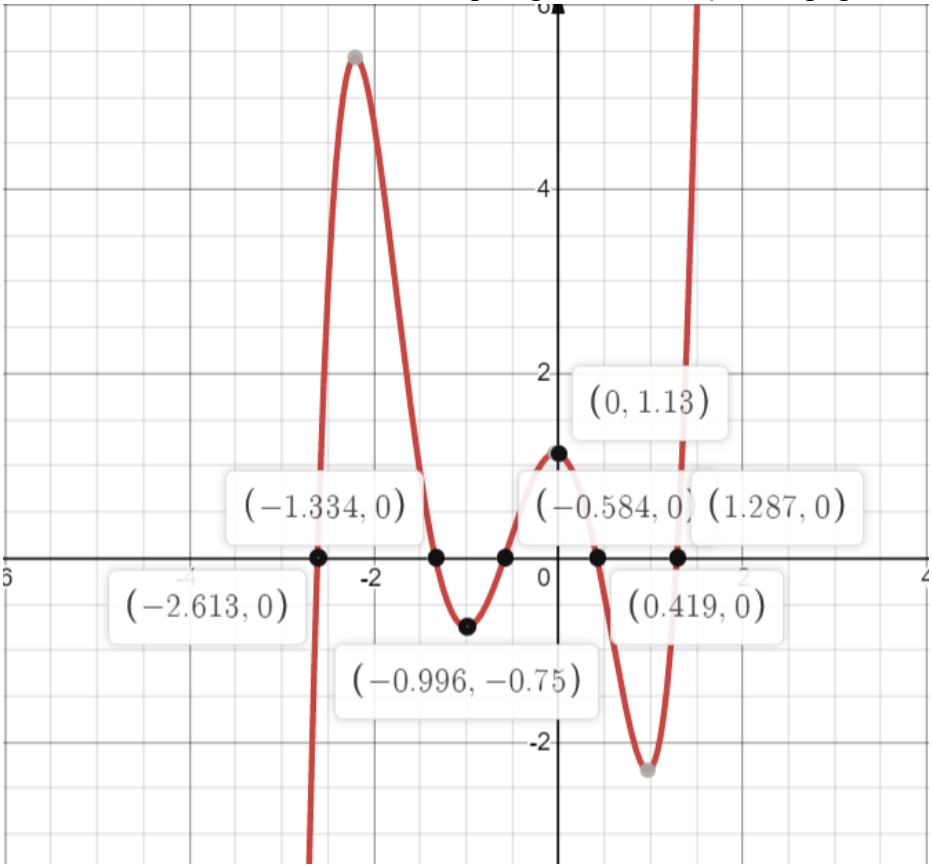

Diagrama. 3

2.4 Transcendentinės funkcijos šaknų radimas

Duota transcendentinė funkcija:

$$\sin(x)\ln(x) - \frac{x}{6}; 1 \leq x \leq 20$$

Pavaizduojame funkciją nurodytame intervale ir atskiriame šaknų intervalus

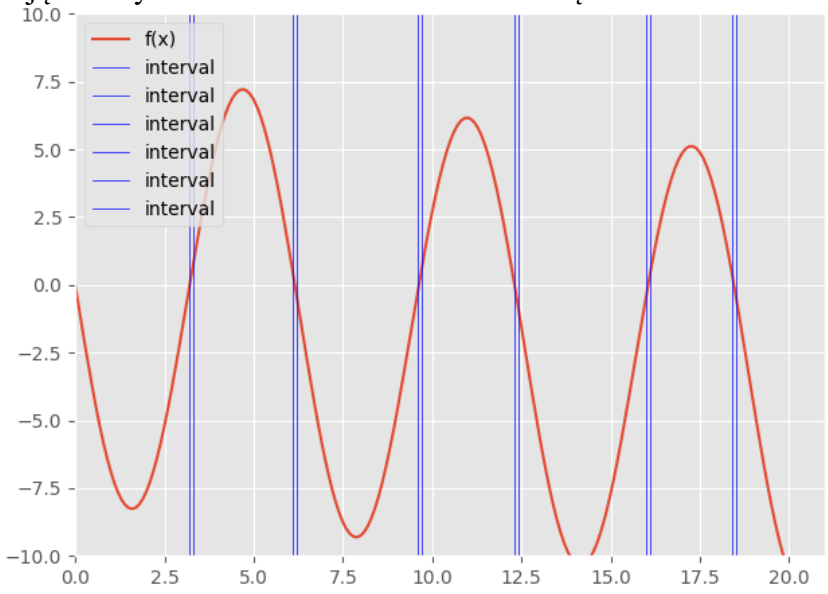

Diagrama. 4

Gautas šaknis tiksliname su „Stygų“, „Liestinių“ ir „Skenavimo su mažėjančiu žingsniu“ metodais.

Metodas	Intervalas	Šaknis	Paklaida	Iteracijos
Stygų	[3.20000000000000002 ,3.300000000000002]	3,209	1,37E-04	1
Liestinių		3,209	2,95E-04	1
Skenavimo		3,208	-6,89E-09	37
Stygų	[6.09999999999999995 ,6.199999999999995]	6,154	-3,30E-04	1
Liestinių		6,155	1,30E-03	1
Skenavimo		6,155	7,64E-09	39
Stygų	[9.599999999999984, 9.699999999999983]	9,627	-3,72E-05	1
Liestinių		9,627	1,70E-03	1
Skenavimo		9,627	-5,01E-09	40
Stygų	[12.299999999999974, 12.399999999999974]	12,307	9,91E-05	1
Liestinių		12,307	5,95E-04	1
Skenavimo		12,307	5,30E-09	34
Stygų	[15.999999999999961, 16.099999999999962]	16,049	-1,99E-03	1
Liestinių		16,049	3,35E-03	1
Skenavimo		16,049	-7,36E-09	37
Stygų	[18.399999999999995, 18.499999999999996]	18,454	-3,82E-03	1
Liestinių		18,455	3,85E-03	1
Skenavimo		18,455	5,04E-09	38

Lentele. 2

„Desmos Graphing Calculator“ įrankio pagalba gauta diagrama ir šaknys:

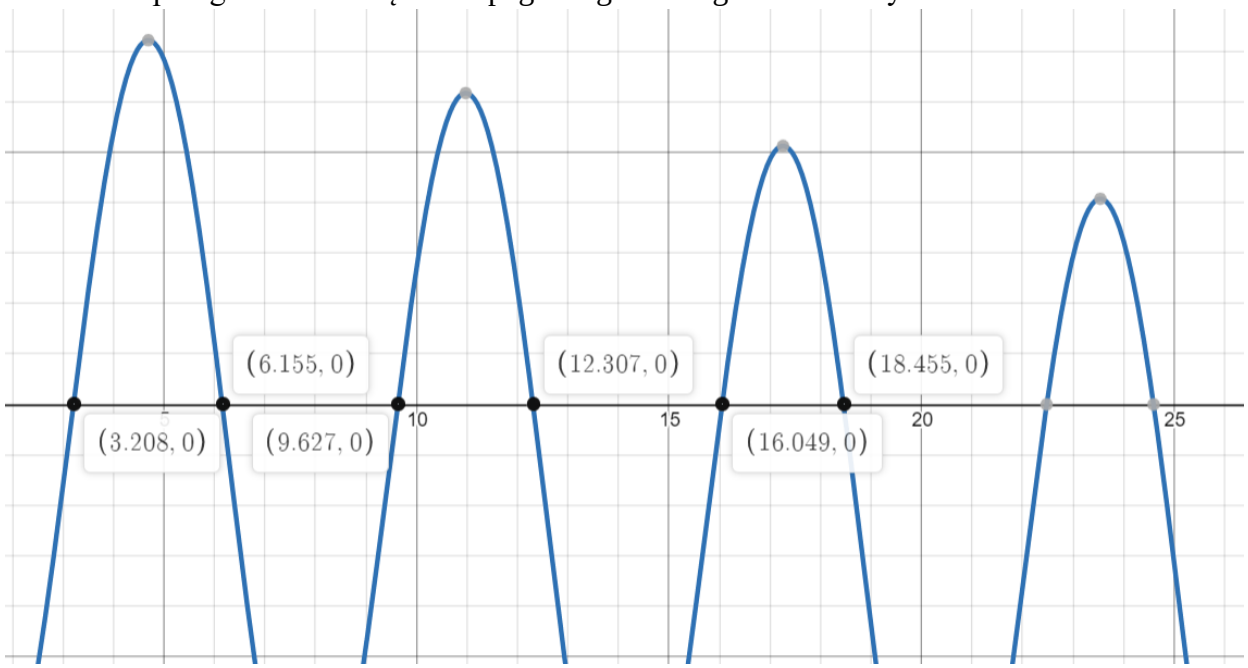

Diagrama. 5

2.5 Tekstinio uždavinio sprendimas

Kūno temperatūros pokytis apskaičiuojamas formule:

$$T(t) = (T_0 - T_A)e^{kt} + T_A$$

Duoti duomenys:

Varianto Nr.	T_0, K	T_A, K	t_1, s	T_1, K
17	400	320	30	344

Į lygtį įrašome turimus duomenis:

$$T(t) = (400 - 320) * e^{30*k} + 320 = 344$$

Susumuojame:

$$T(t) = 80e^{30k} - 24$$

Pasirinktas metodas: skenavimo su mažėjančiu žingsniu.

Skenavimo metodo rezultatai:

Metodas	Intervalas	Šaknis	Paklaida	Iteracijos
Skenavimo	[-1,1]	-0,0401324	-6,41E-09	53

Lentele. 3

Grafinės lygties sprendimas:

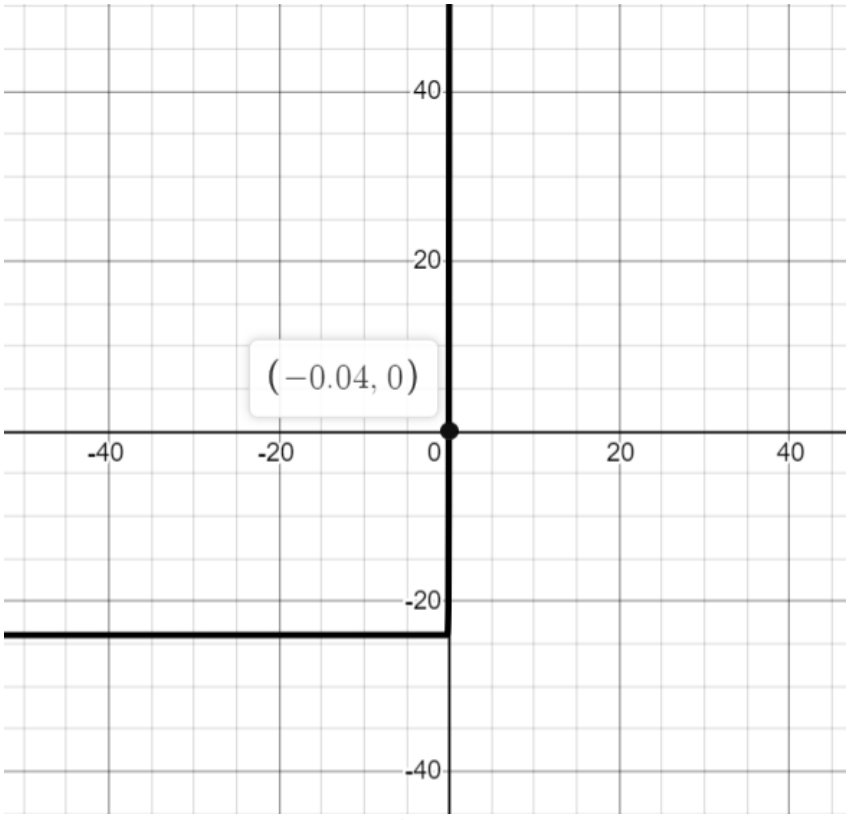

Diagrama. 6

3. IŠVADOS

- Stygų metodo esmė yra ta, kad kreivės lankas pakeičiamas styga ir taip artėjama prie lygties šaknies.

- Niutono metodas turi kvadratinį konvergavimo greitį ir konverguoja sparčiau, ypač, kai paklaida pasidaro maža.
- Skenavimo intervalas bendruoju atveju iš anksto nežinomas, todėl kyla pavojus parinkti per didelį skenavimo žingsnį ir “peršokti” kelias šaknis.

4. PRIEDAI

4.1 Realizuotų algoritmų programinio kodo fragmentai:

```
#------------------CHORD-1--------------------
def chord(points, func, tolerance=1e-8):
    iter = 0
    x_n = points[0]
    x_new = points[1]
    x_mid = 0

    while abs(func(x_mid)) < tolerance:
        iter += 1
        k = abs(func(x_n) / func(x_new))
        x_mid = (x_n + k * x_new) / (1 + k)
        if func(x_mid) * func(x_n) > 0:
            x_n = x_mid
        else:
            x_new = x_mid

    return iter, x_mid
#CHORD
#---------------------SCAN-5----------------------------
def scan(interv, func, step=0.1, tolerance=1e-8):
    iter = 1
    x_new = interv[0]
    x_s = interv[0]

    while (abs(func(x_s))) > tolerance:
        iter += 1
        x_s += step
        if func(x_s) * func(x_new) < 0:
            x_s -= step
            step /= 2
            x_new = x_s

    return iter, x_s
#SCAN
#---------------------NEWTON-3---------------------------------
def newton(points, func, dfunc, tolerance=1e-8):
    iter = 0

    x_n = points[0]
    x_new = points[1]

    while abs(x_new - x_n) > tolerance:
        x_new = x_n - func(x_n) / dfunc(x_n)
        iter += 1
        x_n = x_new

    return iter, x_new
#NEWTON
```