

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.](#page3) | [UŽDUOTIS ..............................................................................................................................................](#page3) | | [2](#page3) |
| [2.](#page4) | [PAGRINDINĖ DALIS ............................................................................................................................](#page4) | | [3](#page4) |
|  | [2.1](#page4) | [Daugianario “Grubus” šaknų įvertis .......................................................................................................](#page4) | [3](#page4) |
|  | [2.2](#page4) | [Daugianario “Tikslesnis” šaknų įvertis ...................................................................................................](#page4) | [3](#page4) |
|  | [2.3](#page6) | [Daugianario šaknų radimas ....................................................................................................................](#page6) | [5](#page6) |
|  | [2.4](#page9) | [Transcendentinės funkcijos šaknų radimas ............................................................................................](#page9) | [8](#page9) |
|  | [2.5](#page12) | [Tekstinio uždavinio sprendimas ...........................................................................................................](#page12) | [11](#page12) |
| [3.](#page13) | [IŠVADOS ..............................................................................................................................................](#page13) | | [12](#page13) |
| [4.](#page14) | [PRIEDAI ................................................................................................................................................](#page14) | | [13](#page14) |
|  | [4.1](#page14) | P[rograminio kodo fragmentai: ...........................................................................](#page14) | [13](#page14) |

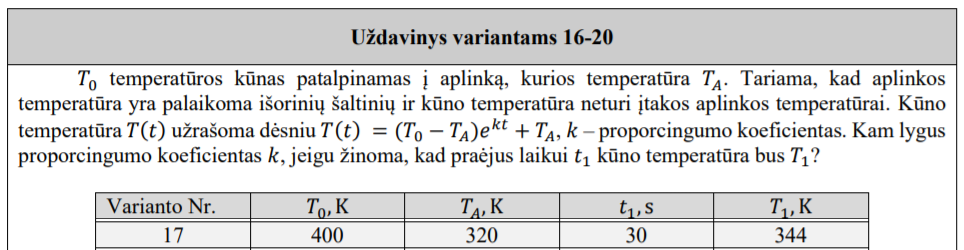
1

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



*Pav. 1*

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



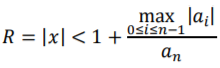
*Pav. 2*

1. PAGRINDINĖ DALIS

2.1 Daugianario "Grubus” šaknų įvertis

Daugianaris:

Grubaus įverčio formulė:

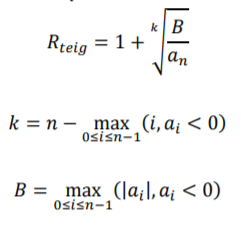


Įsistate kof. gauname:

Gautas grubus šaknies įvertis:

2.2 Daugianario "Tikslesnis” šaknų įvertis

Tikslesnio įverčio formulės:



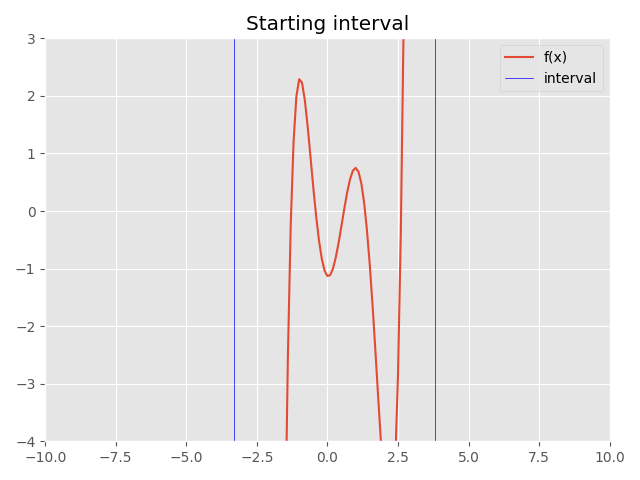
Pilna įverčio formulė:

Skaičiuojame Rteig:

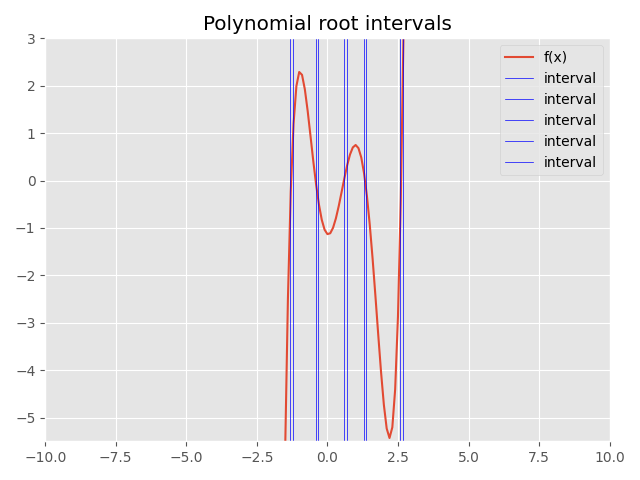
Skaičiuojame Rneig:

Gauname tikslesnį šaknies įvertį:

2.3 Daugianario šaknų radimas

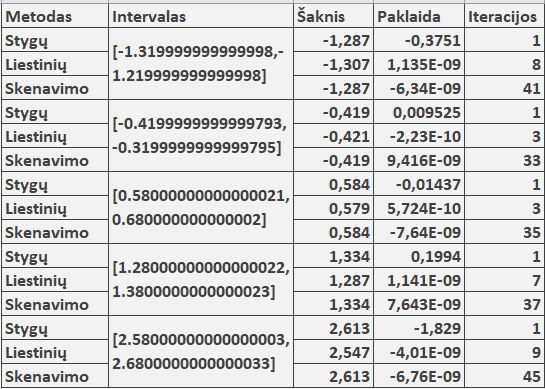


*Diagrama. 1*



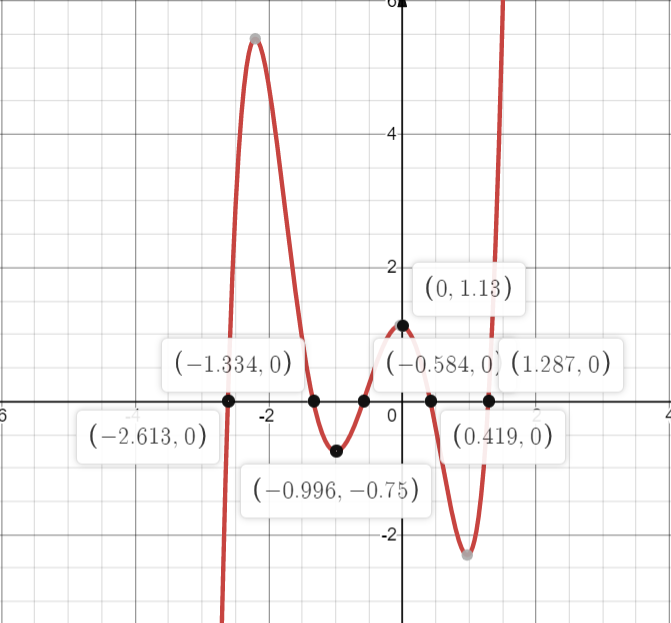
*Diagrama. 2*

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



*Lentele. 1*

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



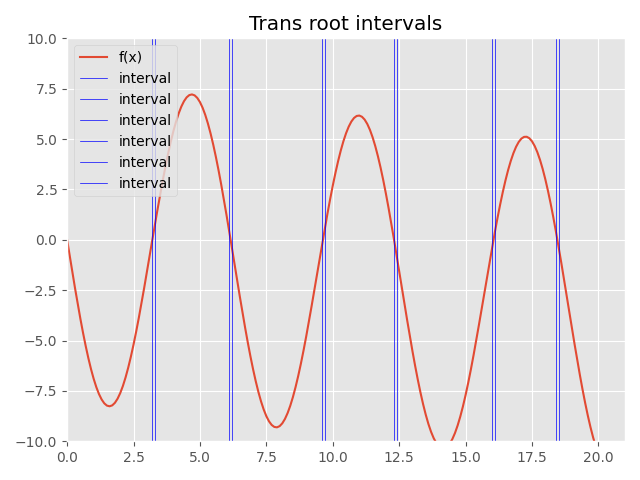
*Diagrama. 3*

2.4 Transcendentinės funkcijos šaknų radimas

Duota transcendentinė funkcija:



Pavaizduojame funkciją nurodytame intervale ir atskiriame šaknų intervalus



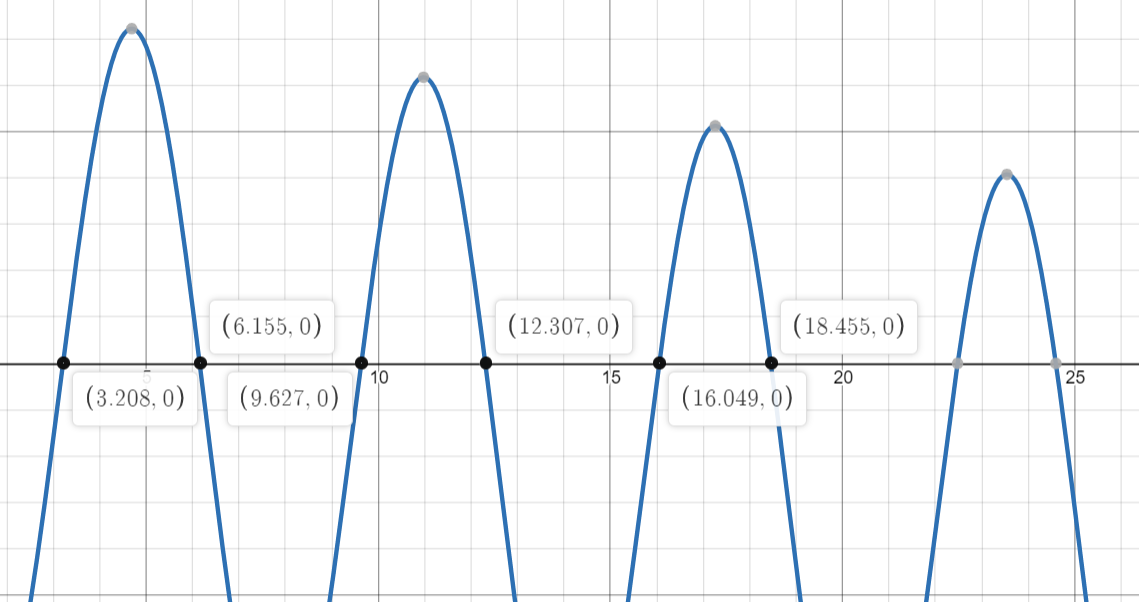
*Diagrama. 4*

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



*Lentele. 2*

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



*Diagrama. 5*

2.5 Tekstinio uždavinio sprendimas

Kūno temperatūros pokytis apskaičiuojamas formule:



Duoti duomenys:



Į lygtį įrašome turimus duomenis:

Susumuojame:

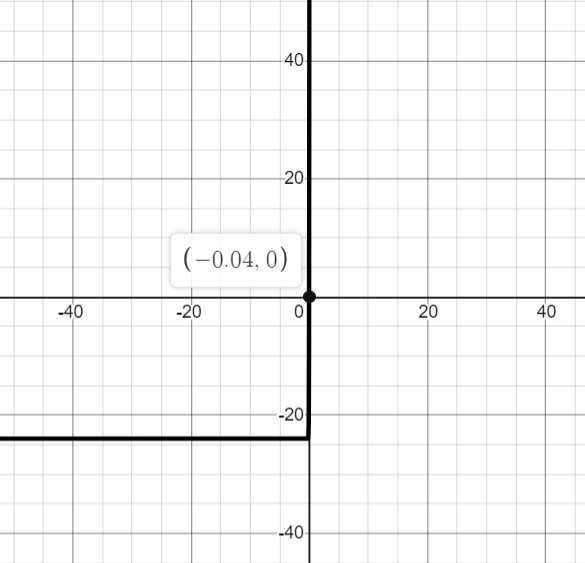
Pasirinktas metodas: skenavimo su mažėjančiu žingsniu.

Skenavimo metodo rezultatai:



*Lentele. 3*

Grafinės lygties sprendimas:



*Diagrama. 6*

1. 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.

1. 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