

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

3 Laboratorinis darbas

Nr. 17

Atliko:

IFE-8 gr. studentas

Kemežys Martynas

Priėmė:

lekt. Andrius Kriščiūnas

KAUNAS, 2020

TURINYS

[1.UŽDUOTIS 3](#_Toc58535454)

[2.Pagrindinė dalis 4](#_Toc58535455)

[2.1 Interpoliavimas daugianariu. 4](#_Toc58535456)

[2.1.1 Grafikai 4](#_Toc58535457)

[2.1.2 Kodo fragmentas: 6](#_Toc58535458)

[2.2 Interpoliavimas daugianariu ir splainu per duotus taškus 7](#_Toc58535459)

[2.2.1 Daugianariu 7](#_Toc58535460)

[2.2.2 Kodo fragmentas: 7](#_Toc58535461)

[2.4 Aproksimavimas 8](#_Toc58535462)

[2.4.1 Grafikai ir lygtys 8](#_Toc58535463)

[2.4.2 Kodo fragmentas: 10](#_Toc58535464)

[3.IŠVADOS 11](#_Toc58535465)

# 1.UŽDUOTIS

1. Interpoliuojamos fukcijos išraiška ir bazinės funkcijos.



*Lentele. 1*

1. Šalys, metai ir splaino tipas II, III IR IV užduotims



*Lentele. 2*

# 2.Pagrindinė dalis

## 2.1 Interpoliavimas daugianariu.

Duota funkcija:

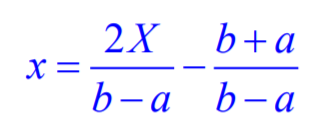


*Lentele. 3*

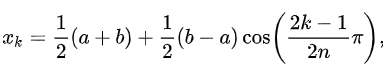
Daugianario formulė:



Interpoliavimo intervalo formulė:

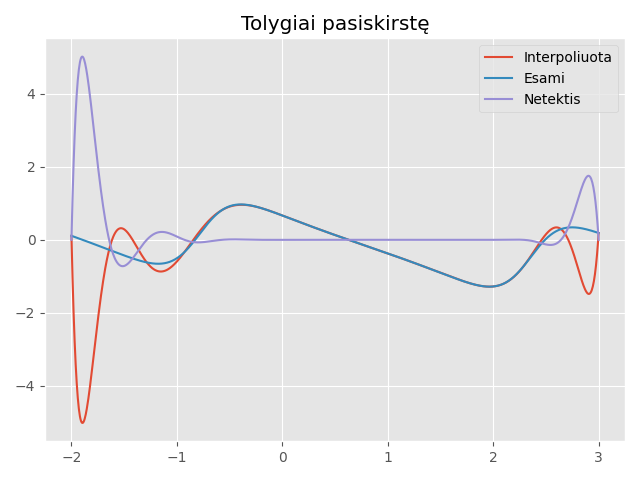


Mazgų transformacijos formulė:

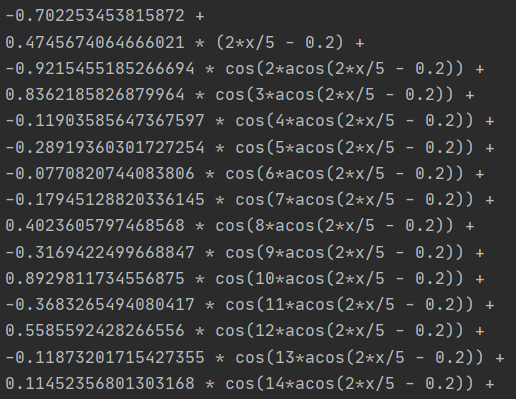


### 2.1.1 Tolygiai pasiskirstę grafikai ir lygtys

Taškai pasiskirstę tolygiai(interpoliavimo taškai: 15)



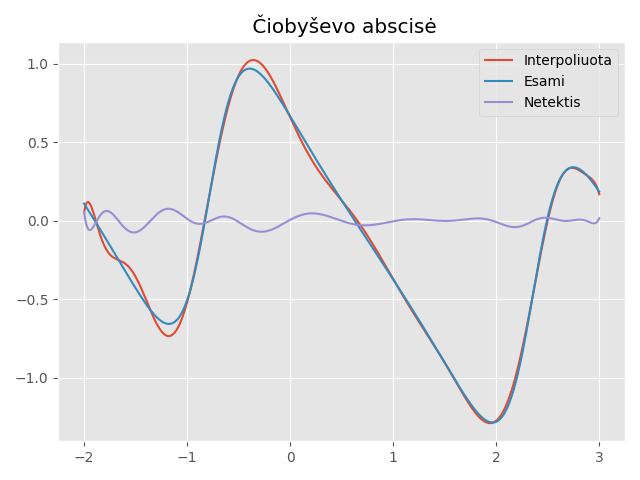
*Pav. 1*



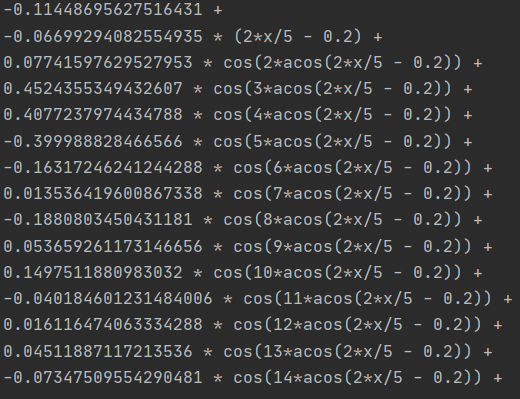
*Pav. 2*

### 2.1.2 Čiobyševo abcisės grafikai ir lygtys

Čiobyševo abcisės(interpoliavimo taškai: 15)



*Pav. 3*



*Pav. 4*

### 2.1.3 Kodo fragmentas:

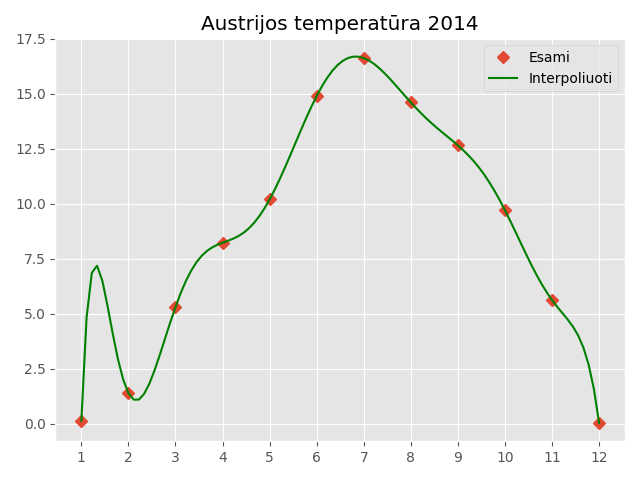
Paleidus programa, paprašoma įvesti tekstą: tl – jei tolygiai išsidėstę taškai, cb čiobyševo abcise.

# duota uzd funkcija  
def funkcija(x): return cos(2 \* x) / (sin(2 \* x) + 1.5) - x / 5  
# daugianario formulė  
def ciobysevo\_daugianaris(x, i): return cos(arccos(x) \* i)  
# duota cb interpoliavimo intervalo formulė  
def ciobysevo\_intervalas(x, a, b): return (2 \* x) / (b - a) - (b + a) / (b - a)  
#cb mazgo transformacija  
def ciobysevo\_mazgas(i, pr, pb, n):  
 # i - iteracijos, pr - pradžia, pb - pabaiga, n - mazgų skaičius.  
 return ((pb - pr) / 2) \* cos(pi \* (2 \* i + 1) / (2 \* n)) + ((pb + pr) / 2)  
  
print('Pasirinkite taškų išdėstymo tipą:')  
print('Tolygiai - tl')  
print('Čiobyševo - cb')  
  
n = 15  
i = np.arange(n)  
  
if str(input("Pasirinkta: ")) == "tl":  
 x = np.linspace(-2, 3, n).reshape(-1, 1); plot\_name = 'Tolygiai pasiskirstę'  
else:  
 x = ciobysevo\_mazgas(i, -2, 3, n).reshape(-1, 1); plot\_name = ' Čiobyševo abscisė'  
  
int\_ciobysevo = ciobysevo\_intervalas(x, -2, 3)  
cb\_daugianaris = ciobysevo\_daugianaris(int\_ciobysevo, i)  
koffs = np.linalg.solve(cb\_daugianaris, funkcija(x))  
  
x = np.linspace(-2, 3, 1000).reshape(-1, 1)  
  
cb\_i\_intervalas = ciobysevo\_intervalas(x, -2, 3)  
  
cb\_i\_daugianaris = ciobysevo\_daugianaris(cb\_i\_intervalas, i)  
int = np.dot(cb\_i\_daugianaris, koffs)

## 2.2 Interpoliavimas daugianariu ir splainu per duotus taškus

### 2.2.1 Daugianariu

a)



*Pav. 5*

### 2.2.2 Kodo fragmentas:

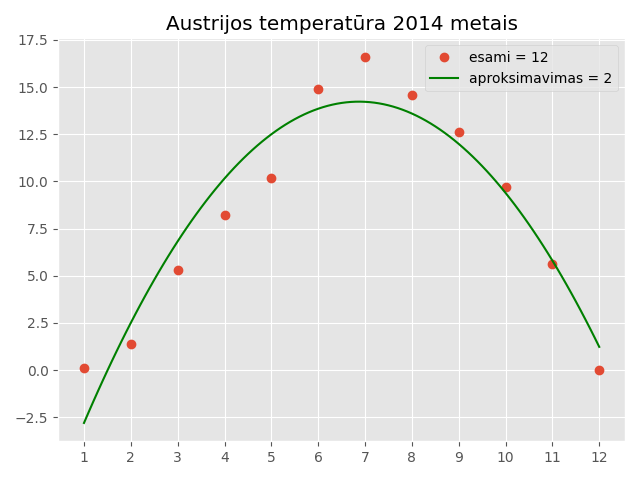
# daugianario formulė  
def ciobysevo\_daugianaris(x, i): return cos(arccos(x) \* i)  
  
# duota cb interpoliavimo intervalo formulė  
# pr - pradžią, b - pabaiga, n - mazgų skaičius FORMULĖ  
def ciobysevo\_intervalas(n, pr, pb): return (2 \* n) / (pb - pr) - (pb + pr) / (pb - pr)  
  
  
def interpoliuoti\_ciobyseva(x, y):  
 int\_ciobysevo = ciobysevo\_intervalas(x, 0, 11)  
 cb\_daugianaris = ciobysevo\_daugianaris(int\_ciobysevo, x.T)  
 koffs = np.linalg.solve(cb\_daugianaris, y)  
  
 x\_n = np.linspace(0, 11, 100).reshape(-1, 1)  
 cb\_i\_int = ciobysevo\_intervalas(x\_n, 0, 11)  
  
 cb\_daugianaris = ciobysevo\_daugianaris(cb\_i\_int, x.T)  
 y\_n = cb\_daugianaris.dot(koffs)

## 2.4 Aproksimavimas

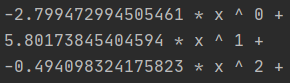
### 2.4.1 Grafikai ir lygtys

**Daugianarų išraiškos su formulėmis**

Antros eilės daugianaris:

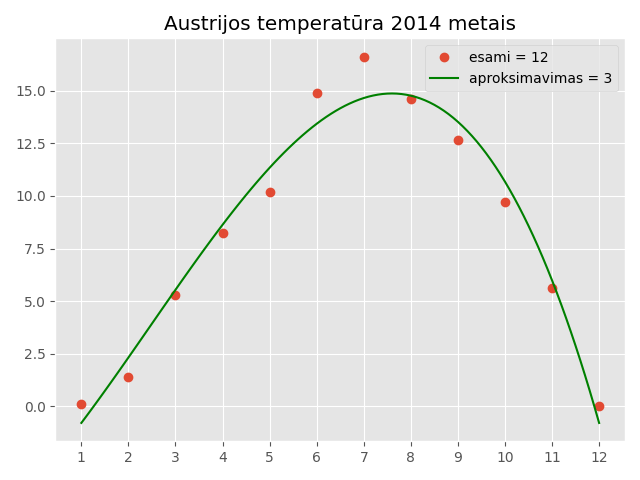
****

*Pav. 6*

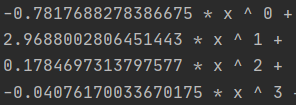


*Pav. 7*

Trečios eilės daugianaris:

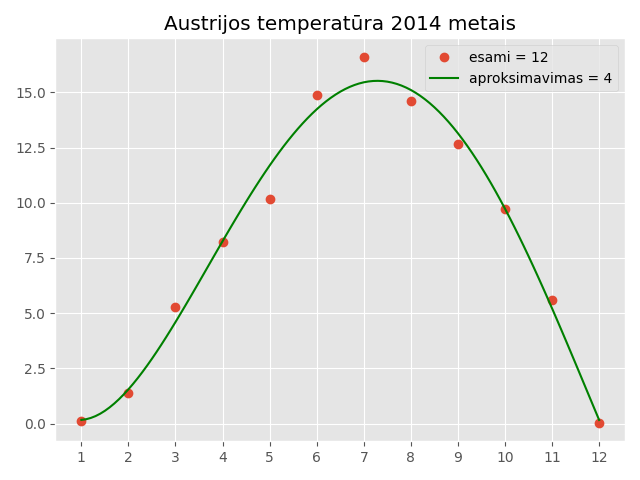


*Pav. 8*

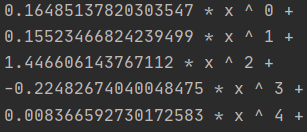
****

*Pav. 9*

Ketvirtos eilės daugianaris:

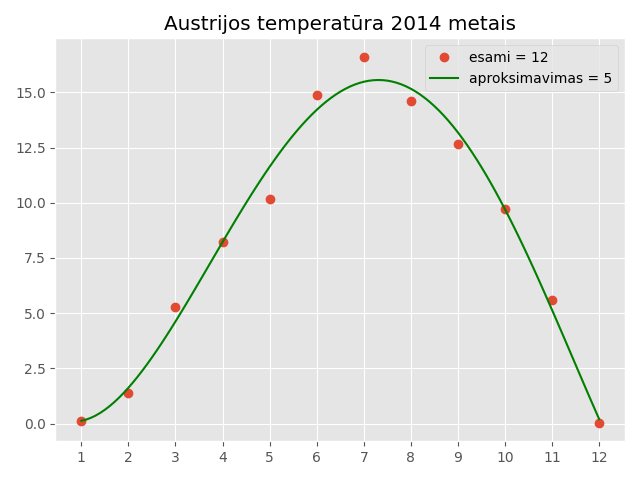


*Pav. 10*

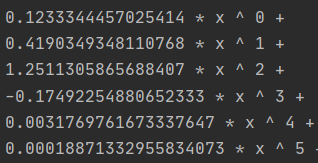
****

*Pav. 11*

Penktos eilės daugianaris:



*Pav. 12*

****

*Pav. 13*

Paleidus programa, prašoma įvesti norimo daugianario laispsnį.

### 2.4.2 Kodo fragmentas:

def kelti(x, laipsnis):  
 x = x.flatten()  
 return np.array([x \*\* i for i in range(laipsnis)]).T.astype(np.float)# skaičius pakeliamas tam tikru laipsniu  
  
def b(x, koffs):  
 koffs = koffs.flatten() # flat list'as  
 ats = 0  
 for i in range(len(koffs)):  
 ats += koffs[i] \* x \*\* i  
 print(f'{koffs[i]} \* x ^ {i} +')  
 return ats  
  
laipsnis = int(input('Daugianario laipsnis: '))  
laipsnis += 1  
duomenys = pd.read\_csv('austria\_temperatures\_2014.csv')  
x = duomenys.index.to\_numpy().reshape(-1, 1) # indeksai/mėnesiai  
y = duomenys.iloc[:, 0].to\_numpy().reshape(-1, 1) # temperatūros  
  
G = kelti(x, laipsnis)  
koffs = np.linalg.solve(G.T.dot(G), G.T.dot(y))  
  
xx = np.arange(0, 11, 0.01).reshape(-1, 1)

# 3.IŠVADOS