



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
2.Pagrindinė dalis	4
2.1 Interpoliavimas daugianariu.	4
2.1.1 Grafikai	4
2.1.2 Kodo fragmentas:.....	6
2.2 Interpoliavimas daugianariu ir splineu per duotus taškus	7
2.2.1 Daugianariu	7
2.2.2 Kodo fragmentas:.....	7
2.4 Aproximavimas	8
2.4.1 Grafikai ir lygtys.....	8
2.4.2 Kodo fragmentas:.....	10
3.IŠVADOS	11

1.UŽDUOTIS

- Interpoliuojamos funkcijos išraiška ir bazinės funkcijos.

17	$\cos(2 \cdot x) / (\sin(2 \cdot x) + 1,5) - \frac{x}{5}; -2 \leq x \leq 3;$	Čiobyševo
----	--	-----------

Lentele. 1

- Šalys, metai ir splaino tipas II, III IR IV užduotims

17	Austrija	2014	Globalus
----	----------	------	----------

Lentele. 2

2. Pagrindinė dalis

2.1 Interpoliavimas daugianariu.

Duota funkcija:

17	$\cos(2 \cdot x) / (\sin(2 \cdot x) + 1,5) - \frac{x}{5}; -2 \leq x \leq 3;$	Čiobyševo
----	--	-----------

Lentele. 3

Daugianario formulė:

$$\cos(\arccos x)$$

Interpoliavimo intervalo formulė:

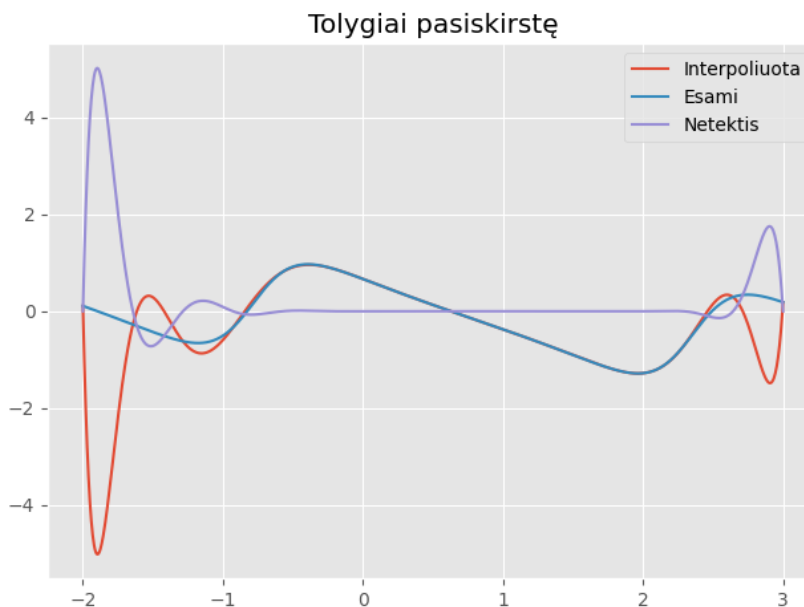
$$x = \frac{2X}{b-a} - \frac{b+a}{b-a}$$

Mazgų transformacijos formulė:

$$x_k = \frac{1}{2}(a+b) + \frac{1}{2}(b-a) \cos\left(\frac{2k-1}{2n}\pi\right),$$

2.1.1 Tolygiai pasiskirstę grafikai ir lygtys

Taškai pasiskirstę tolygiai (interpoliavimo taškai: 15)



Pav. 1

```

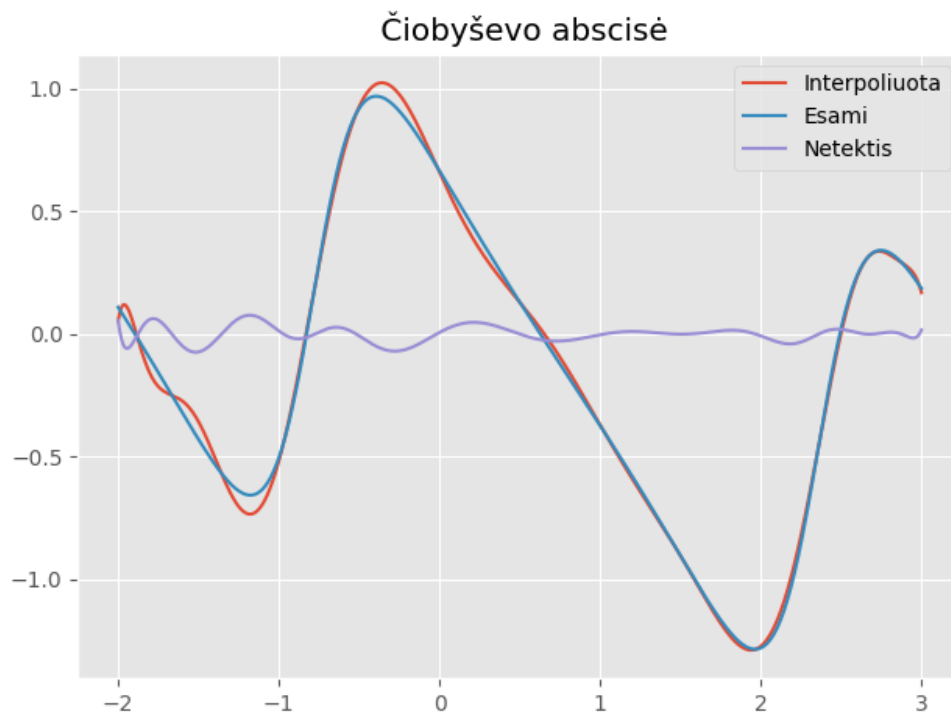
-0.702253453815872 +
0.4745674064666021 * (2*x/5 - 0.2) +
-0.9215455185266694 * cos(2*acos(2*x/5 - 0.2)) +
0.8362185826879964 * cos(3*acos(2*x/5 - 0.2)) +
-0.11903585647367597 * cos(4*acos(2*x/5 - 0.2)) +
-0.28919360301727254 * cos(5*acos(2*x/5 - 0.2)) +
-0.0770820744083806 * cos(6*acos(2*x/5 - 0.2)) +
-0.17945128820336145 * cos(7*acos(2*x/5 - 0.2)) +
0.4023605797468568 * cos(8*acos(2*x/5 - 0.2)) +
-0.3169422499668847 * cos(9*acos(2*x/5 - 0.2)) +
0.8929811734556875 * cos(10*acos(2*x/5 - 0.2)) +
-0.3683265494080417 * cos(11*acos(2*x/5 - 0.2)) +
0.5585592428266556 * cos(12*acos(2*x/5 - 0.2)) +
-0.11873201715427355 * cos(13*acos(2*x/5 - 0.2)) +
0.11452356801303168 * cos(14*acos(2*x/5 - 0.2)) +

```

Pav. 2

2.1.2 Čiobyševio abscisės grafikai ir lygtys

Čiobyševio abscisės(interpoliavimo taškai: 15)



Pav. 3

```

-0.11448695627516431 +
-0.06699294082554935 * (2*x/5 - 0.2) +
0.07741597629527953 * cos(2*acos(2*x/5 - 0.2)) +
0.4524355349432607 * cos(3*acos(2*x/5 - 0.2)) +
0.4077237974434788 * cos(4*acos(2*x/5 - 0.2)) +
-0.399988828466566 * cos(5*acos(2*x/5 - 0.2)) +
-0.16317246241244288 * cos(6*acos(2*x/5 - 0.2)) +
0.013536419600867338 * cos(7*acos(2*x/5 - 0.2)) +
-0.1880803450431181 * cos(8*acos(2*x/5 - 0.2)) +
0.053659261173146656 * cos(9*acos(2*x/5 - 0.2)) +
0.1497511880983032 * cos(10*acos(2*x/5 - 0.2)) +
-0.040184601231484006 * cos(11*acos(2*x/5 - 0.2)) +
0.016116474063334288 * cos(12*acos(2*x/5 - 0.2)) +
0.04511887117213536 * cos(13*acos(2*x/5 - 0.2)) +
-0.07347509554290481 * cos(14*acos(2*x/5 - 0.2)) +

```

Pav. 4

2.1.3 Kodo fragmentas:

Paleidus programa, paprašoma įvesti tekstą: tl – jei tolygiai išsidėstę taškai, cb Čiobyšėvo abscise.

```

# 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('Čiobyšėvo - 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 = 'Čiobyšėvo 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)

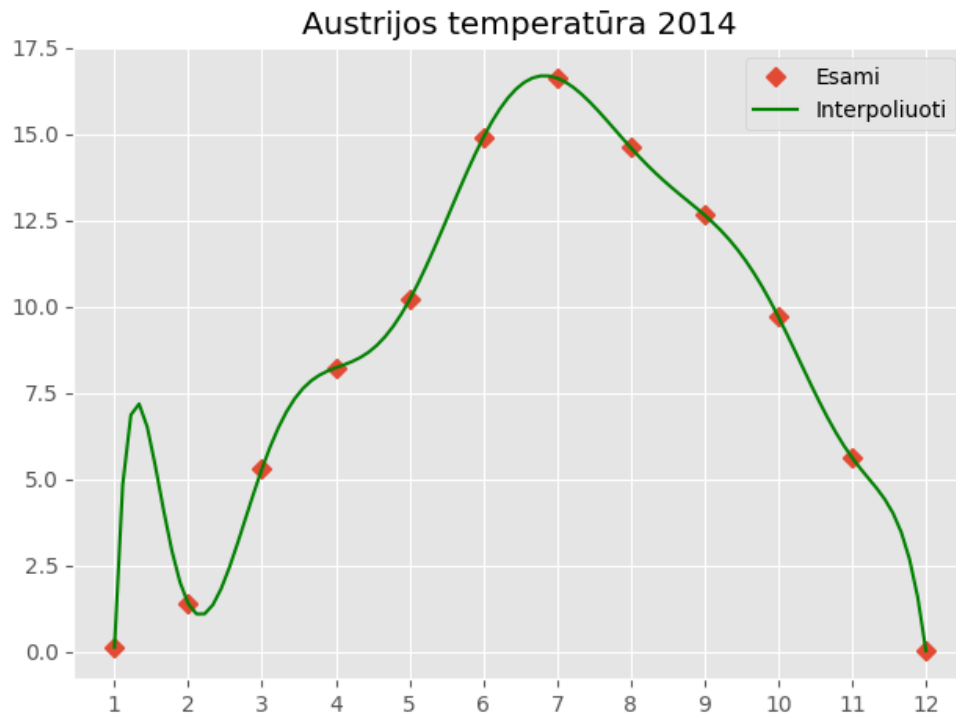
cb_i_daugianaris = ciobysevo_daugianaris(cb_i_intervalas, i)
int = np.dot(cb_i_daugianaris, koffs)

```

2.2 Interpoliavimas daugianariu ir splineu 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 - pradžia, 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)
```

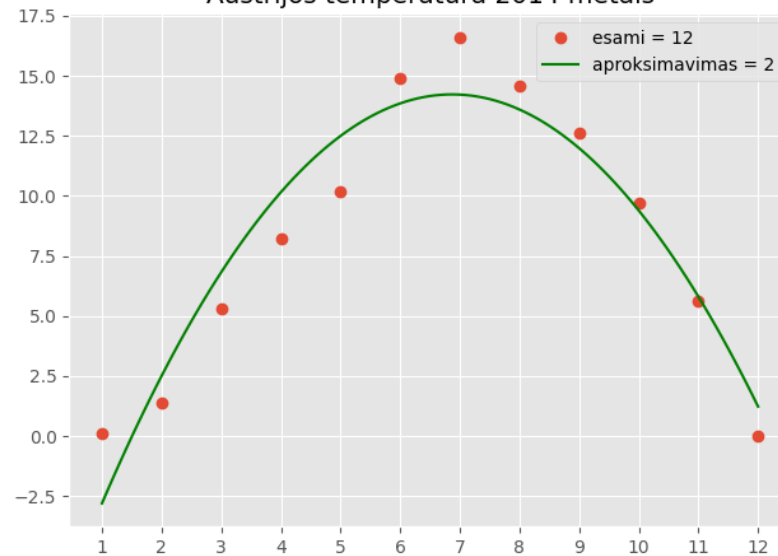
2.4 Aproximavimas

2.4.1 Grafikai ir lygtys

Daugianarų išraiškos su formulėmis

Antros eilės daugianaris:

Austrijos temperatūra 2014 metais



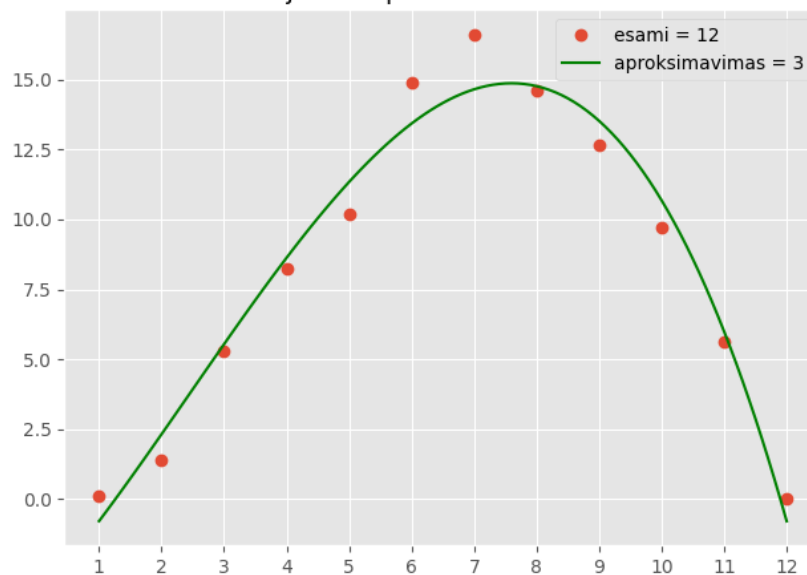
Pav. 6

```
-2.799472994505461 * x ^ 0 +  
5.80173845404594 * x ^ 1 +  
-0.494098324175823 * x ^ 2 +
```

Pav. 7

Trečios eilės daugianaris:

Austrijos temperatūra 2014 metais

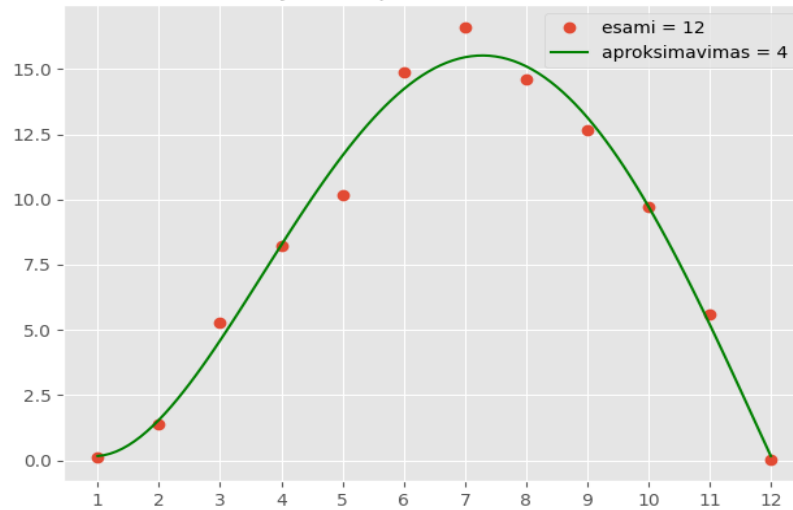


Pav. 8

```
-0.7817688278386675 * x ^ 0 +  
2.9688002806451443 * x ^ 1 +  
0.1784697313797577 * x ^ 2 +  
-0.04076170033670175 * x ^ 3
```


Ketvirtos eilės daugianaris:

Austrijos temperatūra 2014 metais



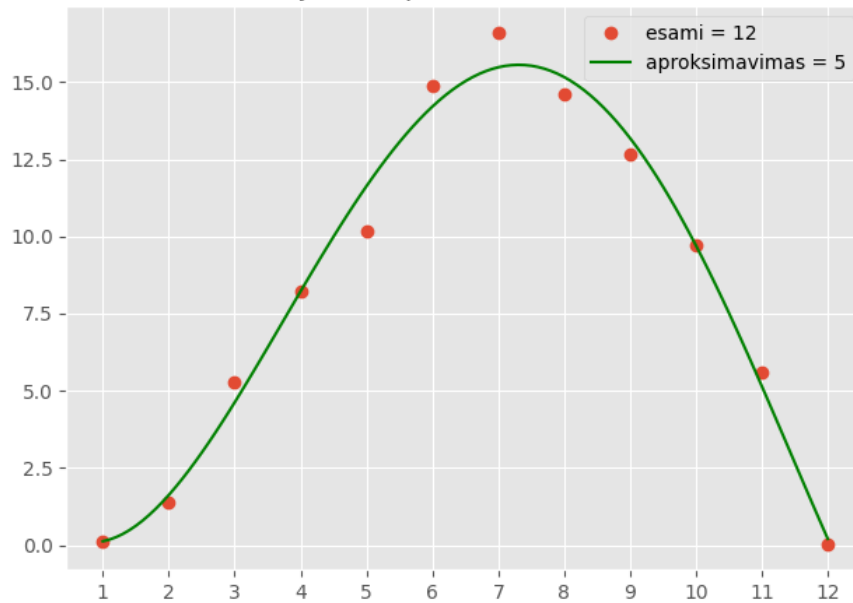
Pav. 10

```
0.16485137820303547 * x ^ 0 +
0.15523466824239499 * x ^ 1 +
1.446606143767112 * x ^ 2 +
-0.22482674040048475 * x ^ 3 +
0.008366592730172583 * x ^ 4 +
```

Pav. 11

Penktos eilės daugianaris:

Austrijos temperatūra 2014 metais



Pav. 12

```
0.1233344457025414 * x ^ 0 +
0.4190349348110768 * x ^ 1 +
1.2511305865688407 * x ^ 2 +
-0.17492254880652333 * x ^ 3 +
0.0031769761673337647 * x ^ 4 +
0.00018871332955834073 * x ^ 5 +
```

Paleidus programa, prašoma įvesti norimo daugianario laipsnį.

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