

3. Опис програми і отриманих результатів

3.1. Призначення програми

Програма призначена для побудови апроксимацій функцій чебишовськими сплайнами. Функція може бути задана двома способами: дискретним (у вигляді таблиці) або аналітично. Програма дає змогу знайти коефіцієнти многочленів чебишовського наближення, границі кожної ланки сплайна, максимальні похибки на кожній ланці сплайну. А також будувати графіки сплайна, яким наближуємо функцію, функції, яка наближується, графік функції похибки. Є можливість порівняти два наближення для певної функції.

3.2. Умови застосування

Вимоги до ПК:

- 32-розрядний (x86) або 64-розрядний (x64) процесор із тактовою частотою 1 ГГц або швидший;
 - 1 гігабайт (ГБ) RAM (для 32-розрядної версії) або 2 ГБ (для 64-розрядної версії);
- Another entry in the list

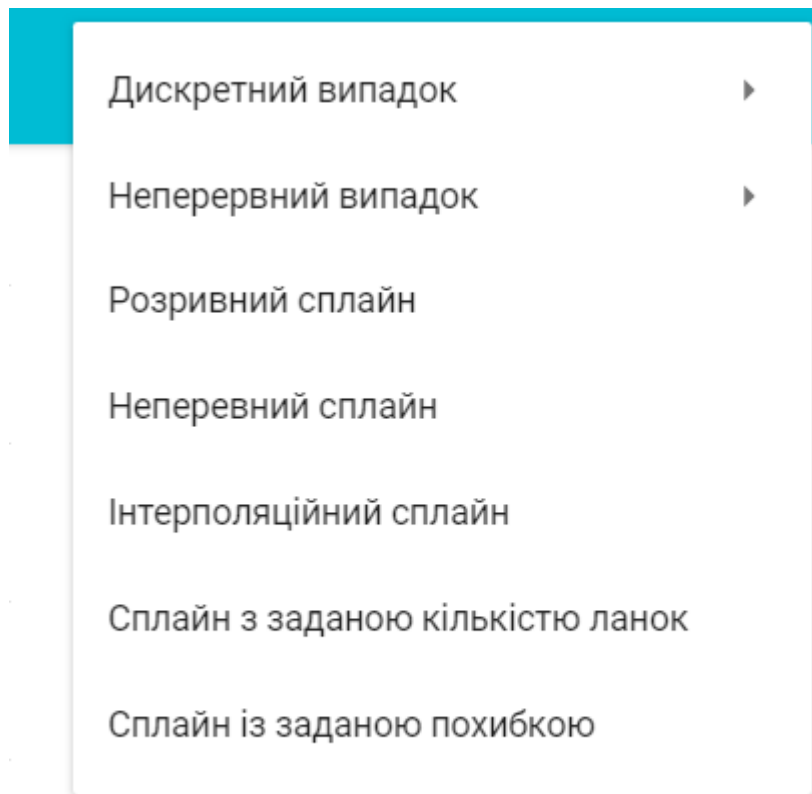
ОС: Windows 7/8/10

3.3. Запуск програми та задання вхідних даних

Для знаходження наближення функції, спочатку потрібно вибрати, яким чином задана функція (таблично чи аналітично). Це можна зробити натиснувши кнопку з правої сторони головного меню сайту.



Далі необхідно вибрати метод яким потрібно апроксимувати функцію.



Для неперервного випадку, потрібно заповнити наступну форму:

Функція, яку апроксимуємо

ln(x)

Степінь многочлена

1

Початок інтервалу

1

Кінець інтервалу

3

Точність

0,01

ОБЧИСЛИТИ


Як видно з рисунку, користувачу потрібно вибрати функцію для апроксимації. Приклади вводу функцій:


e^x	e^x
\sqrt{x}	sqrt(x)
$\cos^2(x)$	cos(x)^2 або (cos(x))^2
$\frac{1}{x}$	1/x

Точність – допустима відносна похибка у визначенні похибки наближення у мінімаксному наближенні. Для дискретного випадку:

Степінь
1

X	0	ПОСОРТУВАТИ	
Y	0	ОЧИСТИТИ	ДОДАТИ


ЗАВАНТАЖИТИ CSV

ЗАВАНТАЖИТИ EXCEL


ОБЧИСЛИТИ

Тут можна задати степінь апроксимуючого многочлена та задати табличну функцію. Це можна зробити двома способами:

1. Вручну. За допомогою кнопки “ДОДАТИ ТОЧКУ” можна додати до таблиці, яка знаходиться лівіше, ще одну точку. Редагувати точки можна відразу в таблиці. При необхідності можна також вилучити точку навівши курсор на неї.

0.25	0.5	0.625	0.8
0.2	0.3	0.4	0.5

✗

Видалити стовпець

Також є можливість посортувати ці точки (по змінній x), натиснувши на відповідну кнопку.

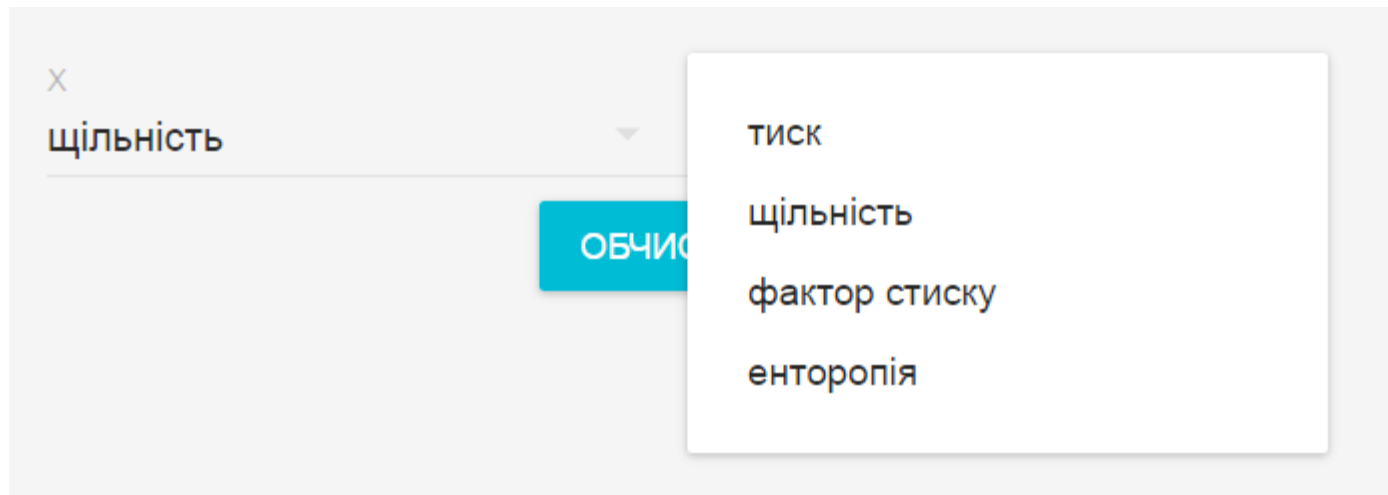
2. Завантажити з файлу. Файл повинен бути у форматі CSV(Comma Separated Values), тобто значення які розділені комою. Перший стовпець – це значення x , другий – y . Приклад файлу CSV:

1	0.1,	0.77
2	1,	7.68
3	2,	15.34
4	3,	22.96
5	4,	30.55
6	5,	38.11
7	6,	45.63
8	8,	60.55
9	10,	75.31
10	12,	89.89
11	16,	118.49
12	20,	146.26
13	25,	179.75
14	30,	211.8
15	35,	242.39
16	40,	271.53
17	45,	299.25
18	50,	325.6

Також файл може бути у форматі $xlsx$ (Excel). Приклад Excel файлу.

	A	B	C	D
1	тиск	щільність	фактор стиску	ентропія
2	0.1	0.048	1.0004	72.293
3	1	0.483	1.004	62.794
4	2	0.962	1.008	59.933
5	3	1.438	1.012	58.259
6	4	1.909	1.016	57.07
7	5	2.377	1.02	56.147
8	6	2.841	1.024	55.393
9	8	3.759	1.032	54.203
10	10	4.663	1.04	53.278
11	12	5.553	1.048	52.522
12	16	7.293	1.0639	51.327
13	20	8.982	1.0797	50.399
14	25	11.026	1.0995	49.47
15	30	12.998	1.1192	48.709
16	35	14.903	1.1389	48.065
17	40	16.745	1.1584	47.506
18	45	18.526	1.1779	47.013
19	50	20.251	1.1973	46.571
20	60	23.542	1.2359	45.805
21	70	26.64	1.2742	45.157
22	80	29.564	1.3122	44.594

Далі необхідно вибрати яка величина – змінна x , а яка – y .



Після цього, незважаючи на спосіб, яким задали функцію (вручну чи завантажили з файлу), потрібно натиснути кнопку “ОБЧИСЛИТИ”. Коли запит обробиться на сервері, результати можна побачити на екрані.

3.4. Опис отриманих результатів

Приклад отриманих результатів наближення функції $\sin(x)$, розривним лінійним сплайном на проміжку $[1, 5]$ із заданою похибкою на кожній ланці – 0.1. Вхідні дані:

Функція, яку апроксимуємо

$\sin(x)$

Степінь многочлена

1

Початок інтервалу

1

Кінець інтервалу

5

Точність

0,1



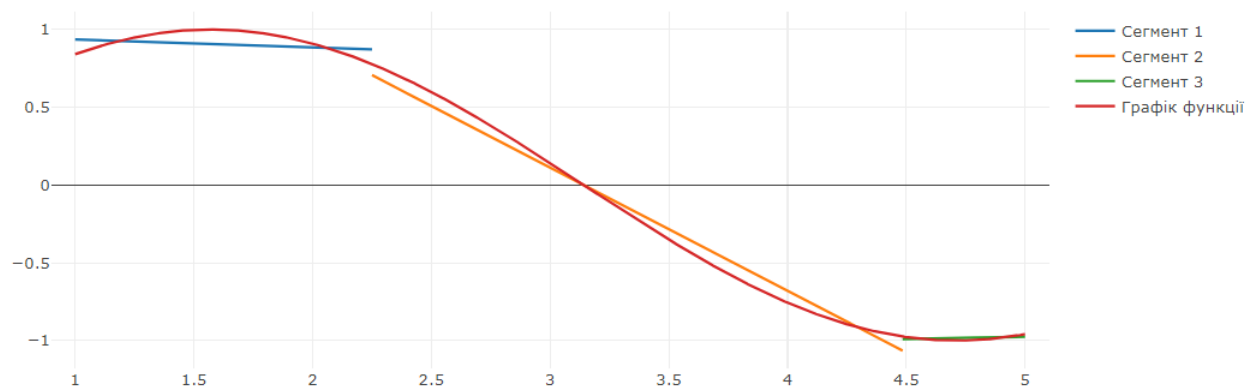
Допустима похибка на одному відрізку сплайна

0,1

ОБЧИСЛИТИ

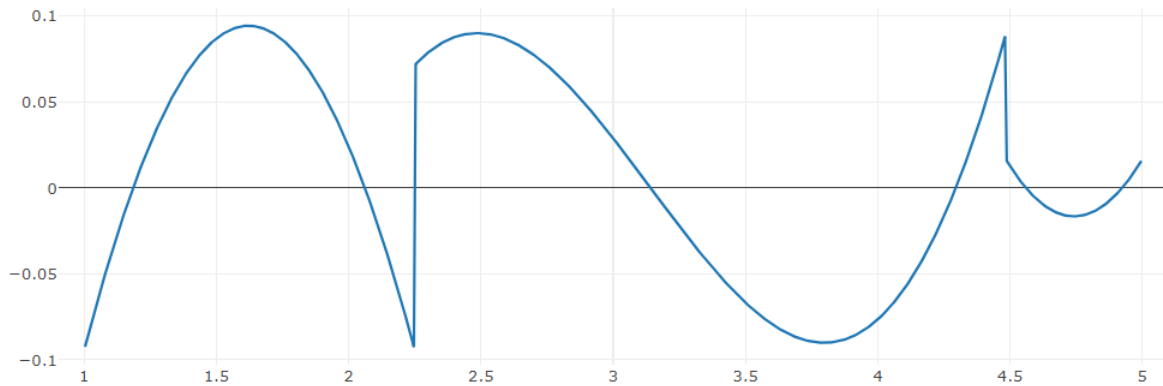
Вихідні дані:

Апроксимація мінімаксними сплайнами



Як можна побачити з рисунку, результатом роботи програми є вивід на екран графік функції, яку наближаємо, а також сплайна.

Графік функції похибки



Також виводиться графік функції похибки на кожній ланці.

Сегмент	Максимальна похибка	Інтервал	Аналітичний вигляд
1	0.094	$[1.000; 2.250]$	$-0.0507x + 0.9865$
2	0.090	$[2.250; 4.484]$	$-0.7924x + 2.4896$
3	0.017	$[4.484; 5.000]$	$0.0294x - 1.1227$

В таблиці представлено максимальну похибку, межі інтервалу і аналітичний вигляд многочлена на всіх ланках чебишовського сплайна.

Додатки

Додаток 1. Код програми

Підпрограма для створення початкового альтернансу (дискретний випадок). Вхідні параметри:

1. масив значень x
2. $degree$ – степінь апроксимуючого многочлена

Результат виконання функції: початковий альтернанс В цій функції використана бібліотека `numpy`(<http://www.numpy.org/>).

```
1 def make_initial_alternance(x, degree):
2     alternance = [x[0]] # add first
3     busy = []
4     for i in range(degree):
5         rand_index = np.random.randint(1, len(x) - 1)
6         while rand_index in busy:
7             rand_index = np.random.randint(1, len(x) - 1)
8         busy.append(rand_index)
9         alternance.append(x[rand_index])
10    alternance.append(x[len(x) - 1]) # add last
11    alternance.sort()
12    return alternance
```

Підпрограма для визначення максимального значення функції в дискретних точках. Вхідні параметри:

1. $func$ – функція
2. x_vals – масив дискретних точок

Результат виконання функції: максимальне значення заданої функції

```
1 def max_error(func, x_vals):
2     y_vals = func(x_vals)
3     neg_err = min(y_vals)
4     pos_err = max(y_vals)
5
6     if abs(neg_err) > pos_err:
7         e_max = neg_err
8     else:
9         e_max = pos_err
10    return e_max
```

Підпрограма для заміни точок альтернансу. Вхідні параметри:

1. `err_func` – функція похибки
2. `alternance` – попередній альтернанс
3. `x_vals` – значення x для таблично заданої функції

Результат виконання функції: новий альтернанс

```
1 def change_alternance(err_func, alternance, x_vals):
2     x_err = x_of_max_error(err_func, x_vals)
3     temp = alternance[:]
4     temp.append(x_err)
5     temp.sort()
6     index_of_x_err = temp.index(x_err)
7     if index_of_x_err != 0 and index_of_x_err != (len(temp) - 1):
8         if sign(err_func(temp[index_of_x_err])) == sign(err_func(temp[
index_of_x_err - 1])):
9             del temp[index_of_x_err - 1]
10
11         else:
12             del temp[index_of_x_err + 1]
13
14     elif index_of_x_err == 0:
15         if sign(err_func(temp[index_of_x_err])) == sign(err_func(temp[1])):
16             del temp[1]
17         else:
18             del temp[len(temp) - 1]
19     elif index_of_x_err == (len(temp) - 1):
20         if sign(err_func(temp[index_of_x_err])) == sign(err_func(temp[
index_of_x_err - 1])):
21             del temp[index_of_x_err - 1]
22         else:
23             del temp[0]
24     return temp
```

Підпрограма для побудови аналітичного вигляду многочлена. Вхідні параметри:

1. `t` – альтернанс
2. `degree` – степінь апроксимуючого многочлена
3. `f_discrete` – дискретна функція

Результат виконання функції: аналітичний вигляд многочлена.

```

1 def pol(t, degree, f_discrete):
2     x = Symbol('x')
3     e = Symbol('e')
4     vars_str = ' '.join(['a' + str(i) for i in range(degree + 1)])
5     variables = symbols(vars_str)
6     eqs = []
7
8     for i in range(degree + 2):
9         eqs.append(make_eq(variables, t[i], f_discrete(t[i])) + e)
10        e *= 1
11    if degree % 2 == 1:
12        e *= 1
13
14    solution = solve(eqs, variables + (e,))
15
16    error_on_iteration = solution[e]
17    polynom = x
18    for i, v in enumerate(variables):
19        polynom += solution[v] * x ** i
20
21    return [polynom, error_on_iteration]

```

Підпрограма *main* для апроксимації чебишовським сплайном з заданою кількістю ланок. Вхідні параметри:

1. func – функція, яку апроксимуємо
2. deg – степінь апроксимуючого многочлена
3. start – початок інтервалу
4. end – кінець інтервалу
5. r – кількістю ланок

```

1 import spline_minmax
2
3
4 def checkIfErrorsOk(specified_precision, approximation, epsilon=0.01):
5     for approx in approximation:
6         if abs(specified_precision - approx["max_error"]) > epsilon:
7             return False
8     return True
9
10
11 def main(func, deg, start, end, r):

```

```

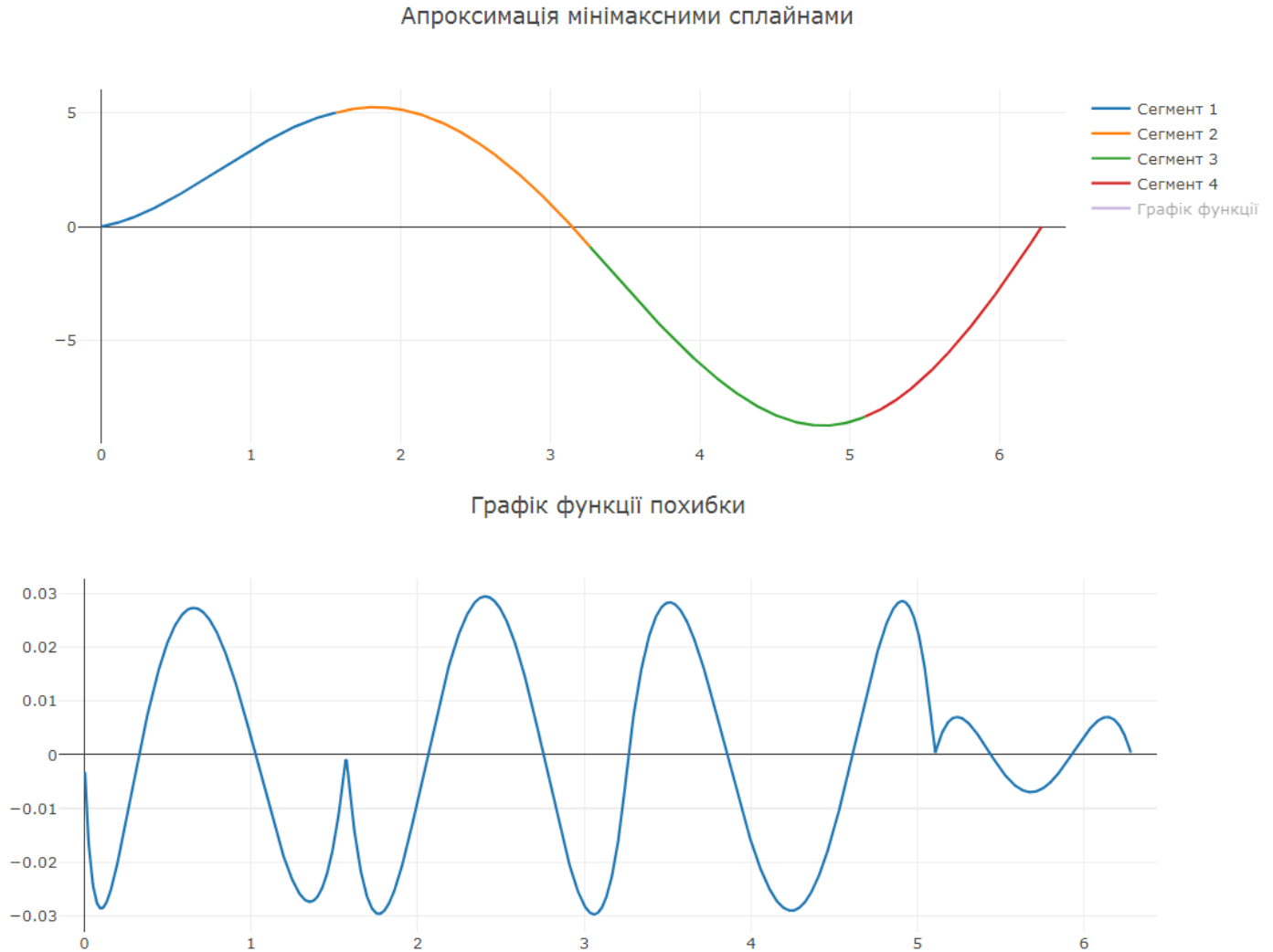
12 mu_left = 0
13 mu_right = 0
14 prev_specified_precision = 0
15 precision = 0.0009
16 mu = 0.1
17
18 approximation = spline_minmax.main(func, deg, start, end, precision, mu)
19 k = len(approximation)
20 while k != r or not checkIfErrorsOk(mu, approximation):
21     if k > r:
22         mu_left = mu
23         if mu_right != 0:
24             mu = (mu + mu_right) / 2
25         else:
26             mu *= 1.1
27
28     if k == r:
29         mu_right = mu
30         if mu_left != 0:
31             mu = (mu + mu_left) / 2
32         else:
33             mu *= 0.9
34     if k < r:
35         mu_right = mu
36         if mu_left != 0:
37             mu = (mu + mu_left) / 2
38         else:
39             mu *= 0.9
40
41     approximation = spline_minmax.main(func, deg, start, end, precision,
mu)
42
43     if len(approximation) == r and abs(prev_specified_precision - mu) <
0.000001:
44         return approximation
45
46     prev_specified_precision = mu
47
48     k = len(approximation)
49
50 return approximation

```

Додаток 2. Результати виконання програми

Приклад 1. Знайти рівномірне наближення чебишовським інтерполяційним сплайном третього степеня для функції $f(x) = 4\sin(x)\sqrt{x}$ на проміжку $[0, 2\pi]$ із заданою похибкою -0.03 .

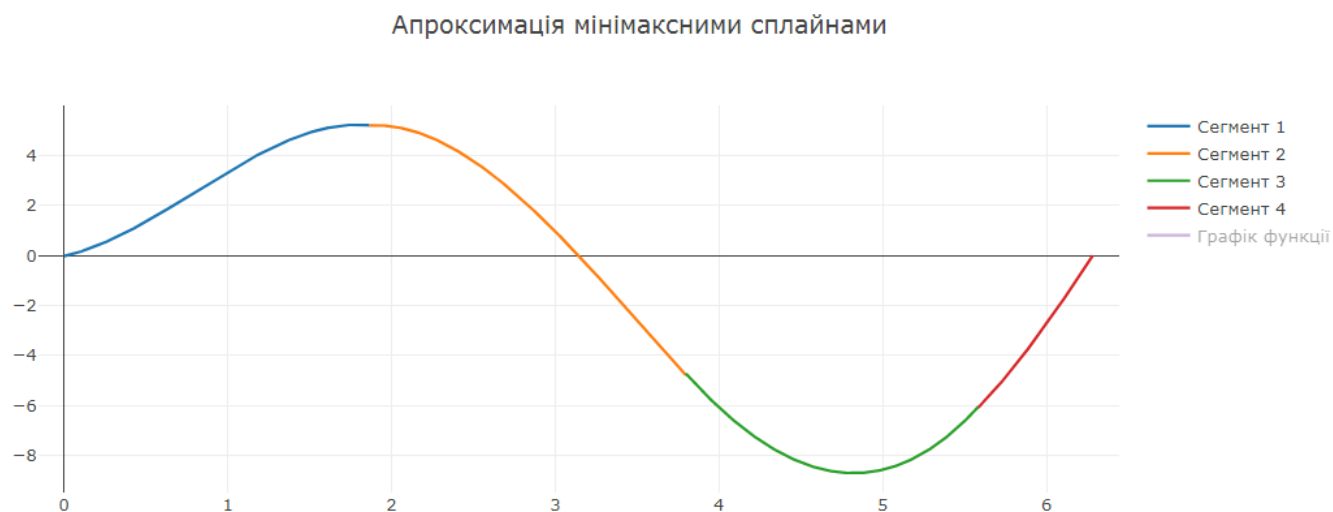
Результат роботи програми:



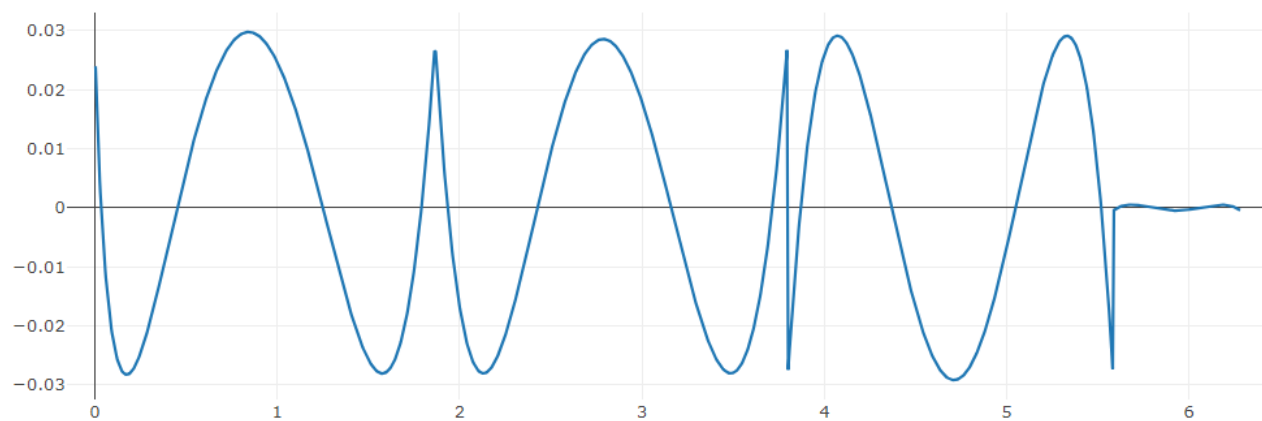
Сегмент	Максимальна похибка	Інтервал	Аналітичний вигляд
1	0.02865	[0.000; 1.571]	$-1.5738x^3 + 3.7462x^2 + 1.1904x$
2	0.02976	[1.571; 3.264]	$0.4431x^3 - 6.0396x^2 + 17.6322x - 9.4986$
3	0.02907	[3.264; 5.104]	$1.0411x^3 - 10.2092x^2 + 25.8096x - 12.5643$
4	0.00702	[5.104; 6.283]	$-1.0929x^3 + 21.9000x^2 - 135.6305x + 258.7080$

Приклад 2. Знайти рівномірне наближення розривним чебишовським сплайном третього степеня для функції $f(x) = 4 * \sin(x)\sqrt{x}$ на проміжку $[0, 2\pi]$ із заданою похибкою $- 0.03$.

Результат роботи програми:



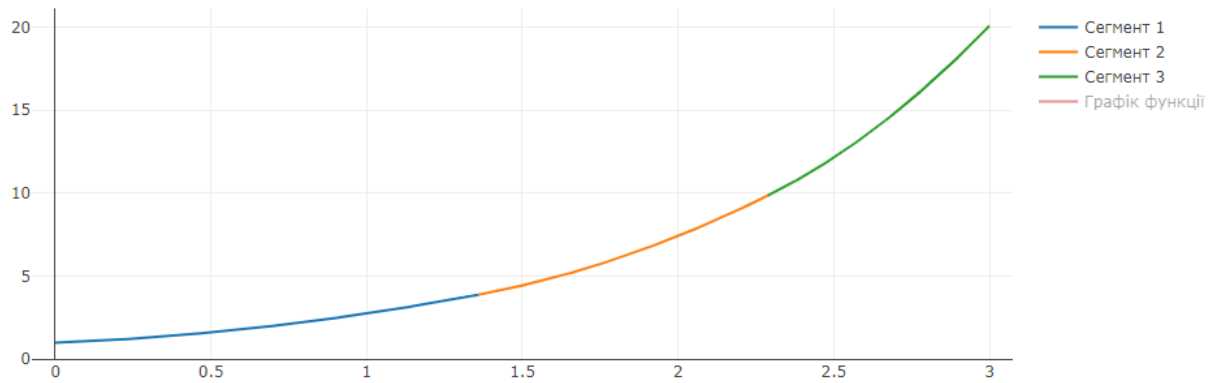
Графік функції похибки



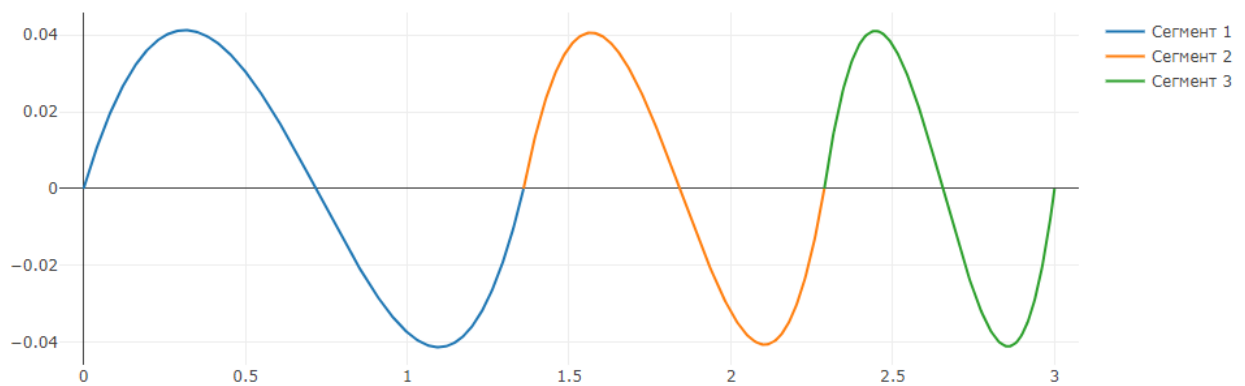
Сегмент	Максимальна похибка	Інтервал	Аналітичний вигляд
1	0.02976	[0.000; 1.865]	$-1.3787x^3 + 3.2956x^2 + 1.4523x - 0.0280$
2	0.02855	[1.865; 3.798]	$0.9199x^3 - 9.7615x^2 + 27.1276x - 17.4082$
3	0.02921	[3.798; 5.584]	$0.4442x^3 - 2.1881x^2 - 9.8981x + 40.0879$
4	0.00050	[5.584; 6.283]	$-1.4059x^3 + 27.4453x^2 - 168.3306x + 322.8903$

Приклад 3. Знайти рівномірне наближення чебишовським інтерполяційним сплайном 2-го степеня функції $f(x) = e^x$ на проміжку $[0, 3]$ з заданою кількістю ланок - 3.

Результат роботи програми:

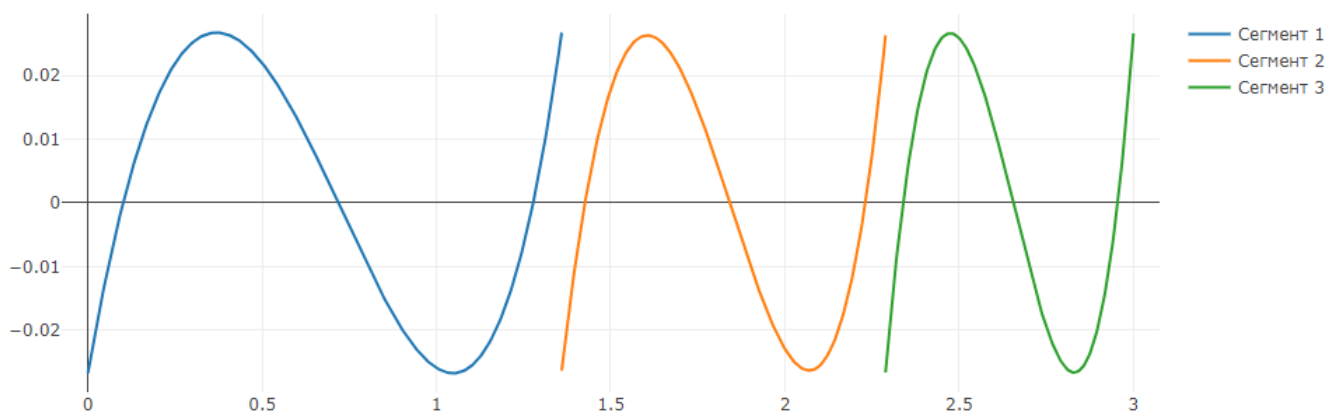
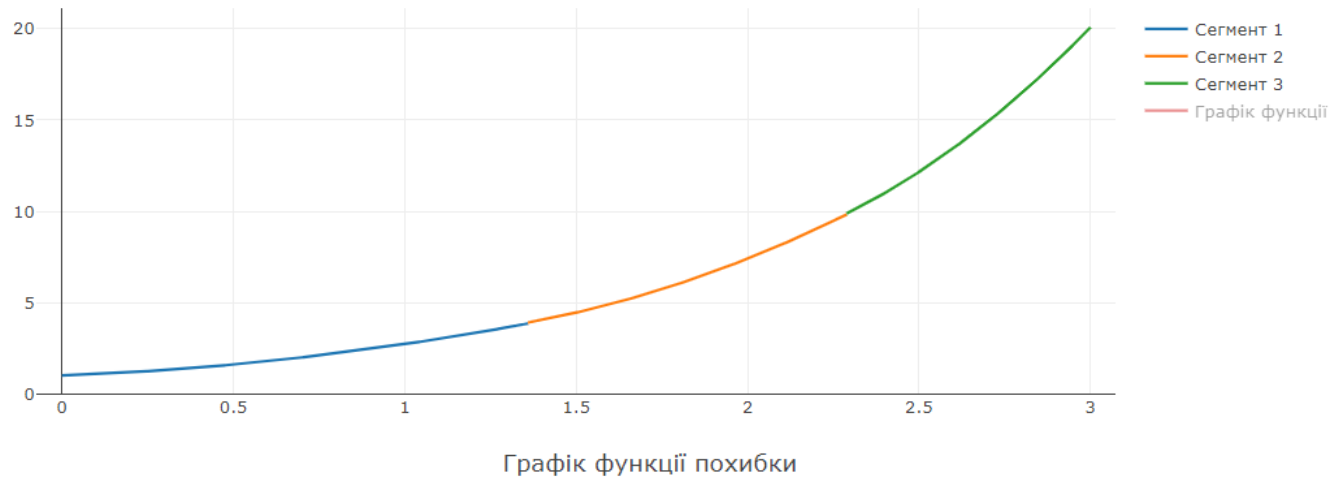


Графік функції похибки



Сегмент	Максимальна похибка	Інтервал	Аналітичний вигляд
1	0.04125	$[0.000; 1.359]$	$1.0381x^2 + 0.7174x + 1.0$
2	0.04058	$[1.359; 2.289]$	$3.1733x^2 - 5.1543x + 5.0364$
3	0.04104	$[2.289; 3.000]$	$7.1362x^2 - 23.3687x + 25.9651$

Приклад 4. Знайти рівномірне наближення чебишовським розривним сплайном 2-го степеня функції $f(x) = e^x$ на проміжку $[0, 3]$ з заданою кількістю ланок - 3. Результат роботи програми:



Сегмент	Максимальна похибка	Інтервал	Аналітичний вигляд
1	0.02676	[0.000; 1.359]	$1.0349x^2 + 0.6825x + 1.0267$
2	0.02634	[1.359; 2.289]	$3.1685x^2 - 5.1935x + 5.1249$
3	0.02665	[2.289; 3.000]	$7.1300x^2 - 23.4108x + 26.1207$