

Практикум 9. Многочлен Тейлора.

Цель работы – научиться использовать инструменты языка программирования Python для произведения операций над полиномами, представленными векторами коэффициентов, вычисления значений полиномов, произведения действий с полиномами (сложение, вычитание, умножение, деление, дифференцирование), произведения вычислений с использованием цикла for; разложения многочленов по степеням $x - x_0$.

Продолжительность работы - 4 часа.

Оборудование, приборы, инструментарий – работа выполняется в компьютерном классе с использованием языка программирования Python и интерактивного блокнота jupyter-notebook.

Порядок выполнения

1. Упражнения выполняются параллельно с изучением теоретического материала.
2. После выполнения каждого упражнения результаты заносятся в отчёт.
3. При выполнении упражнений в случае появления сообщения об ошибке рекомендуется сначала самостоятельно выяснить, чем оно вызвано, и исправить команду; если многократные попытки устранить ошибку не привели к успеху, то проконсультироваться с преподавателем.
4. Дома доделать упражнения из раздела «Краткие теоретические сведения и практические упражнения», которые Вы не успели выполнить во время аудиторного занятия.
5. После выполнения упражнений выполнить дополнительные упражнения для самостоятельной работы и ответить на контрольные вопросы и (см. ниже).
6. Подготовить отчёт, в который включить упражнения из раздела «Краткие теоретические сведения и практические упражнения» и упражнения для самостоятельной работы. Отчёт представить в виде документа Microsoft Word, имя файла (пример): mp_10_Ivanov_P_01_s_1 (факультет_группа_Фамилия студента_Инициал_номер лабораторной, семестр). Отчет должен содержать

по каждому выполненному упражнению: № упражнения, текст упражнения; команды, скопированные из командного окна, с комментариями к ним и результаты их выполнения, включая построенные графики; сценарии на языке программирования Python; выводы.

Краткие теоретические сведения

и практические упражнения

1. Полиномы. Действия с полиномами. Полином в языке программирования Python задаётся вектором его коэффициентов, начиная со старшего. Число элементов вектора на один больше степени полинома. Функция *numpy.polyval* предназначена для вычисления значений полинома от некоторого аргумента. Аргумент может быть матрицей или вектором.

Пример 1. Вычислим значение полинома

$$p(x) = x^5 + 5x^3 + 7x^2 + 8x + 3 \text{ в точке } x = -2.$$

```
import numpy as np
pol = [1, 0, 5, 7, 8, 3]
print(f"Значение полинома в точке x = -2 равно {np.polyval(pol, -2)}.")
```

Значение полинома в точке x = -2 равно -57.

Упражнение 1. Вычислить значения полинома

$$p(x) = x^7 - 3.2x^4 + 3x^2 + 3 \text{ в точках } -1; 4; 2, 2; \pi. \text{ Значения аргументов задать в виде вектора. Сохранить значения полинома.}$$

Нахождение всех корней полинома производится с помощью функции *numpy.roots*.

Пример 2.

```
import numpy as np
pol = [1, 0, 5, 7, 8, 3]
print(f"Корни полинома равны:")
for root in np.roots(pol):
    print(f"{root : .3f}")
```

Корни полинома равны:
0.759+2.283j
0.759-2.283j
-0.500+0.866j
-0.500-0.866j
-0.518+0.000j

Обратная функция к *numpy.roots* – *numpy.poly* – вычисляет коэффициенты многочлена с данными корнями.

Пример 3.

```
import numpy as np

pol = [1, 0, 5, 7, 8, 3]

print(f"Коэффициенты полинома равны:")
for root in np.poly(np.roots(pol)):
    print(f"{root : .3f}")
```

```
Коэффициенты полинома равны:
1.000
-0.000
5.000
7.000
8.000
3.000
```

Вопрос. Как вы думаете, получится ли тот же результат при выполнении следующей команды?

```
np.poly([0.7592 + 2.2827i, 0.7592 - 2.2827i, -0.5000 + 0.8660i, -0.5000 - 0.8660i, -0.5184])
```

Почему?

Упражнение 2. Вычислить корни полинома $p(x) = 2x^8 - 3x^5 + 7x^2 - 2$, сохранить их, сделать проверку.

Для умножения полиномов используется функция *numpy.convolve*. Для деления полиномов с остатком используется функция *numpy.polydiv*, имеющая два выходных аргумента – частное и остаток.

Упражнение 3. Вычислить произведение полиномов

$$p(x) = x^6 + 2x^5 - 3x^2 + 4 \quad \text{и} \quad q(x) = x^3 - 3x + 1,$$

а также частное и остаток от деления P на Q .

Для сложения и вычитания полиномов нет специальной функции. Использование знака + приведёт к ошибке при сложении полиномов разной степени.

Упражнение 4. Написать файл-функцию с двумя аргументами, осуществляющую сложение полиномов разной степени. Алгоритм:

1) Найти большую из длин входных аргументов (обозначим её m).

2) Создать вспомогательные векторы длины m , представляющие те же самые полиномы, что и аргументы. Для заполнения части элементов нулями можно использовать функцию `numpy.zeros`.

3) Вычислить сумму.

Протестировать файл-функцию, используя полиномы

$$p(x) = x^6 + 2x^5 - 3x^2 + 4 \quad \text{и} \quad q(x) = x^3 - 3x + 1,$$

2. Формула Тейлора для многочленов.

Производная многочлена. Встроенная функция `numpy.polyder` предназначена для вычисления не только производной полинома, но и для вычисления производной произведения и частного полиномов.

`q = numpy.polyder(p)` – выходной аргумент `q` равен производной полинома `p`;

`q = numpy.polyder(p, n)` – выходной аргумент `q` равен n -ной производной полинома `p`.

Упражнение 5. Для многочленов $p(x) = x^5 - x^4 - 3x^2 - 2$ и $q(x) = x^6 - 3x + 1$ найти их производные, производную произведения и частного.

Нахождение производных высших порядков. Цикл `for`. Цикл `for` предназначен для выполнения заданного числа повторяющихся действий. Самое простое использование цикла `for` выполняется следующим образом:

```
for count in range(start, step, final + 1):  
    команды Python
```

Здесь `count` – переменная цикла, `start` – её начальное значение, `final` – конечное значение, `step` – шаг, на который увеличивается `count` при каждом следующем заходе в цикл. Цикл заканчивается, как только значение `count` становится больше `final`. Значения `start` и `step` (а значит, и `count`) не обязательно целые.

Другим способом является итерация непосредственно по вектору значений:

```
vector = np.linspace(0, 1, 10)  
for x in vector:  
    команды Python
```

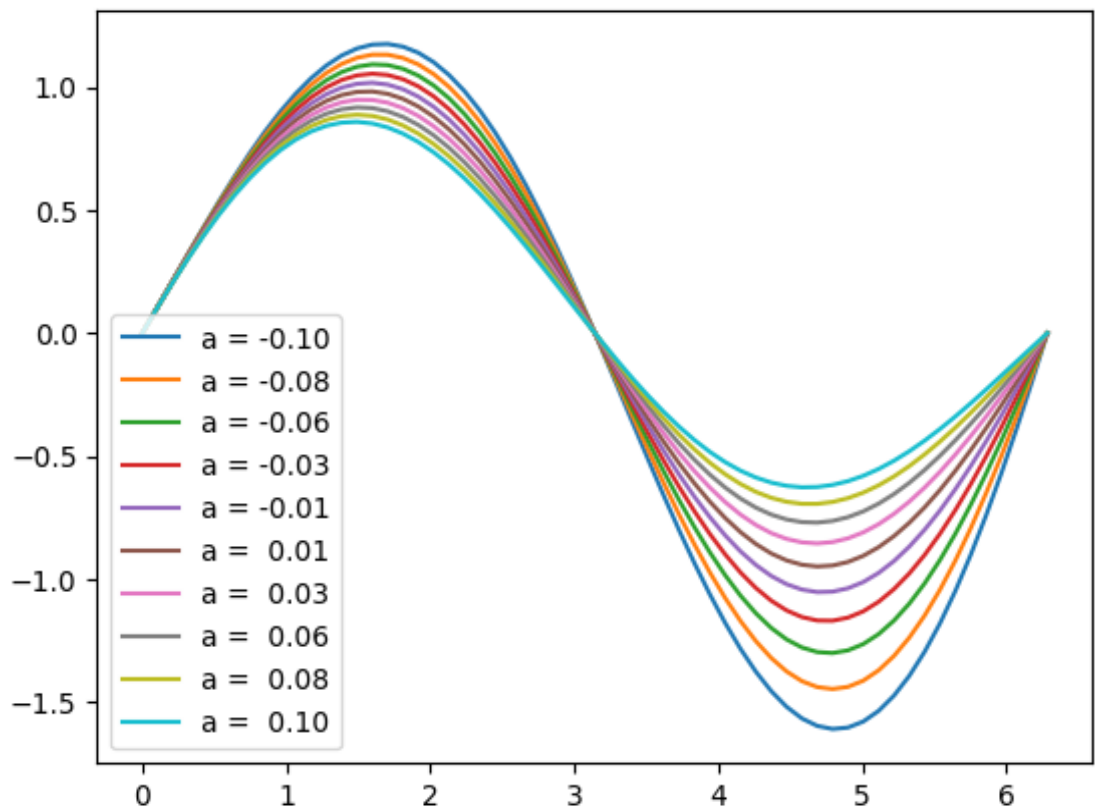
где vector – какой-либо вектор, переменная count принимает поочерёдно все значения элементов vector. Это действительно так.

Пример 4. Пусть требуется вывести семейство кривых, которое задано функцией, зависящей от параметра: $y(x, a) = e^{-ax} \sin x$ для $x \in [0; 2\pi]$ и значений параметра от $-0,1$ до $0,1$ с шагом $0,2$. Создадим сценарий на языке программирования Python

```
import numpy as np
import matplotlib.pyplot as plt

x = 2 * np.pi * np.linspace(0, 1, 60)
for a in 0.2 * np.linspace(0, 1, 10) - 0.1:
    y = np.exp(-a * x) * np.sin(x)
    plt.plot(x, y, label=f"{a = : .2f}")
plt.legend()
plt.show()
```

и запустим его на выполнение.



Упражнение 6. Создать файл-функцию, вычисляющую производную порядка n полинома $P(x)$, заданного вектором коэффициентов. Производную задать вектором коэффициентов. С помощью созданной файл-функции найти 5-ую производную

водную полинома $p(x) = x^{10} - x^7 + 3x^2 - 2$ и вычислить её значение в точке 0,2.

Разложение полинома по степеням $x - x_0$. Полином $P(x)$ степени n раскладывается по степеням $x - x_0$

$$P(x) = b_n(x - x_0)^n + b_{n-1}(x - x_0)^{n-1} + \dots + b_0,$$

где $b_k = \frac{P^{(k)}(x_0)}{k!}$. Разложение называется n -м *многочленом Тейлора* по степеням $x - x_0$.

Упражнение 7. Создать сценарий на языке программирования Python, вычисляющий значения коэффициентов b_k (в виде вектора) в многочлене Тейлора для полинома произвольной степени в произвольной точке. С помощью созданной функции вычислить коэффициенты b_k для разложений полинома $p(x) = x^6 + 2x^4 - 3x - 2$ в многочлен Тейлора по степеням $x - 1$ и $x + 2$.

Задания для самостоятельной работы

1. Выполнить упражнения из раздела «Краткие теоретические сведения и практические упражнения», которые не успели сделать в аудитории.
2. Выполнить упражнения:

Упражнение С1. Вычислить корни полинома $p(x) = x^6 - 1$ сохранить их, сделать проверку.

Упражнение С2. Вычислить произведение полиномов $p(x)$ и $q(x)$, частное и остаток от деления $p(x)$ на $q(x)$, сумму полиномов, если

$$p(x) = x^2 + x + 1 \text{ и } q(x) = x - 1.$$

Упражнение С3. Для многочлена $p(x) = x^3 - 2x^2 + 1$ записать многочлен Тейлора по степеням $x - 2$.

3. Ответить на контрольные вопросы:

- 1) Рассмотрим многочлен $p(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$. Как расположены относительно друг друга графики его многочленов Тейлора порядка n , выписанные по степеням $x - x_1$ и $x - x_2$, если $x_1 \neq x_2$?
- 2) Рассмотрим многочлен $p(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n$. Как расположен его график относительно графика его многочлена Тейлора порядка n , выписанного по степеням $x - x_0$, если $x_0 \neq 0$?

Список рекомендуемой литературы

1. Официальная документация по языку программирования Python
<https://docs.python.org/3/>.
2. Официальная документация к библиотеке numpy
<https://numpy.org/doc/stable/index.html>.
3. Официальная документация к библиотеке matplotlib
<https://matplotlib.org/stable/api/index>.