

ООPython

Задача 2. Численное дифференцирование

Введение

В лекционном материале `lecture_5_classes_b.ipynb` рассматривалось представление производной f' в виде класса, экземпляры которого являются функторами:

```
class Derivative:

    def SetFunction(self, f):
        self._f = f

    def SetStep(self, h):
        self._h = h

    def __call__(self, x):
        f, h = self._f, self._h
        return (f(x + h) - f(x - h)) / (2.*h)
```

В методе `__call__` для приближенного вычисления производной f' используется формула центральной разности:

$$f'(x_0) \approx f'_{num}(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} \quad (1)$$

Существует множество формул различных порядков точности для расчета численной производной $f'_{num}(x_0)$. Ограничимся следующими:

- $f'_{num}(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} \quad (2)$

- $f'_{num}(x_0) = \frac{f(x_0) - f(x_0 - h)}{h} \quad (3)$

- $f'_{num}(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} \quad (4)$

- $f'_{num}(x_0) = \frac{4}{3} \frac{f(x_0 + h) - f(x_0 - h)}{2h} - \frac{1}{3} \frac{f(x_0 + 2h) - f(x_0 - 2h)}{4h} \quad (5)$

- $f'_{num}(x_0) = \frac{3}{2} \frac{f(x_0 + h) - f(x_0 - h)}{2h} - \frac{3}{5} \frac{f(x_0 + 2h) - f(x_0 - 2h)}{4h} + \frac{1}{10} \frac{f(x_0 + 3h) - f(x_0 - 3h)}{6h} \quad (6)$

Любую из вышеприведенных формул можно записать в общем виде:

$$f'_{num}(x_0) = \frac{1}{h} \sum_{i=-N}^N \alpha_i f(x_i) \quad (7)$$

где $\{\alpha_i\}_{i=-N}^N$ - набор коэффициентов, $\{x_i\}_{i=-N}^N$ - набор точек шаблона, $N = 3$.

Задание

Определение классов

Для каждой из формул (2)-(7) программно реализовать соответствующий класс

DerivativeName, минимизировав суммарное число строк кода с помощью наследования.

Каждый класс должен включать в себя как минимум следующее:

Поля:

- дифференцируемая функция f
- шаг численного дифференцирования h

Методы:

- конструктор
- «сеттеры» для f и h
- задать коэффициенты $\{\alpha_i\}_{i=-N}^N$
- вычислить значение $f'_{num}(x_0)$ по формуле (7) (реализовать через магический метод `__call__`)

Тестирование классов

Использовать созданные классы для расчета численных производных: для каждой

формулы (2)-(6) в точке $x_0 = 5$ построить графики абсолютного значения погрешности в

зависимости от шага численного дифференцирования $h_n = \frac{1}{2^{n-1}}$, $n = 1, \dots, 21$ для следующих

функций:

$$\sin(x^2), \cos(\sin(x)), \exp(\sin(\cos(x))), \ln(x+3), \sqrt{x+3}$$

Использовать логарифмический масштаб по обеим осям. Графики погрешностей для

каждой функции строить в отдельном окне. Задействовать библиотеку **SymPy** для

аналитического вычисления производных. Примеры реализации – см. в

lecture_5_classes_b.ipynb.