OOPython

Задача 5. Численное решение ОДУ

Введение

Будем рассматривать следующие методы численного решения ОДУ:

(1) явный метод Эйлера (явный, 1-й порядок аппроксимации; реализован в **lecture_8_scalar_ode.ipynb**)

(2) метод Хойна (явный, 2-й порядок аппроксимации; реализован в **lecture_8_scalar_ode.ipynb**)

(3) метод Рунге-Кутты (явный, 4-й порядок аппроксимации; реализован в **lecture_8_scalar_ode.ipynb**)

(4) метод трапеций (неявный, 2-й порядок аппроксимации).

Задание

Определение классов

Для каждого из методов (1)-(4) реализовать соответствующий класс *MethodName*, минимизировав суммарное число строк кода с помощью наследования. Каждый класс должен включать в себя как минимум следующее:

Поля:

- функция правой части ОДУ
- начальное условие
- множество точек сетки и ее параметры: кол-во точек, отрезков, шаг сетки
- массив для хранения значений численного решения
- начальный и конечный моменты времени.

Методы:

- конструктор: задать правую часть, начальное условие, параметры сетки
- решить ОДУ («timestepping» - цикл по точкам сетки)
- построить график численного решения.

Использование классов

Каждым из методов произвести численное решение логистического уравнения и построить графики полученных решений (в одном графическом окне).

Параметры функции правой части уравнения:

- $\alpha = 0.2$
- $R = 100$

Параметры расчетной сетки:

- число отрезков разбиения $N = 30$
- $t_{start} = 0$
- $t_{end} = 80$

Начальное условие:

- $u(0) = 2$

Примечания по реализации

Метод трапеций (неявный, 2-й порядок аппроксимации)

Формула для проведения вычислений:

$$u_{n+1} = u_n + \frac{\Delta t}{2}\big(f(u_n) + f(u_{n+1})\big), \qquad \text{где } n = 0, \dots, N$$

Для решения получившегося нелинейного уравнения относительно u^{n+1} будем использовать итерационный метод Ньютона:

$$F(u_{n+1}) \equiv u_{n+1} - \frac{\Delta t}{2} f(u_{n+1}) - u_n - \frac{\Delta t}{2} f(u_n) = 0$$

$$u_{n+1}^{(k+1)} = u_{n+1}^{(k)} - \frac{F\left(u_{n+1}^{(k)}\right)}{F'\left(u_{n+1}^{(k)}\right)}, = 0, \dots, K \quad \left\| u_{n+1}^{(K)} - u_{n+1}^{(K-1)} \right\| \le \varepsilon$$

Для ускорения сходимости начальное приближение можно рассчитывать по следующим формулам:

$$u_{n+1}^{(0)} = u_n$$

$$u_{n+1}^{(0)} = u_n + \Delta t f(u_n)$$

Для приближенного вычисления производной использовать формулу центральной разности 2-го порядка точности:

$$F'\left(u_{n+1}^{(k)}\right) \approx F'_{num}\left(u_{n+1}^{(k)}\right) = \frac{F\left(u_{n+1}^{(k)} + h\right) - F\left(u_{n+1}^{(k)} - h\right)}{2h}$$

Решение нелинейного уравнения с помощью метода Ньютона можно оформить как виде метода класса, так и в виде отдельной функции. При вычислении производной задействовать функторы, реализованные в **Задаче 2**. Критерий остановки итераций: $\left\| u_{n+1}^{(K)} - u_{n+1}^{(K-1)} \right\| \le \varepsilon = 10^{-3}$.