Отчёт

Иван Рубачёв, 161

10 декабря 2018 г.

Содержание

1 Модуль интегрирования

Необходимо реализовать и протестировать модуль интегрирования. Было решено для интегрирования использовать метод трапеций. Реализацию можно найти в файле `integral.py`. Здесь приведен код для тестирования и результаты.

1.1 Сравнение с scipy.integrate.quad

Были выбраны следющие функции:

1. Непрерывная
$$f(x) = \arctan(x+2) + 1 \tag{1}$$
2. Осцилирующая
$$f(x) = \sin(100x) \tag{2}$$
3. Разрывная
$$f(x) = \begin{cases} x^3 - 2x + 2 & \text{if } x \geqslant 0 \\ x - 1 & \text{otherwise} \end{cases} \tag{3}$$

Метод трапеций, заключается в замене на каждом элементарном отрезке подынтегральной функции на многочлен первой степени, то есть линейную функцию. Площадь под графиком функции аппроксимируется прямоугольными трапециями.

$$\int\limits_a^b f(x)dx \approx \sum_{i=0}^{n-1} \frac{f(x_i) + f(x_{i+1})}{2}(x_{i+1} - x_i) \tag{4}$$

Порядок аппроксимации $p = 2$ для метода трапеций (т.к. ошибка оценивается сверху $O(h^2) = O(1/N^2)$). На Рис. 1 изобрашены отклонения от значения интеграла, вычисляемого функцией `scipy.integrate.quad`

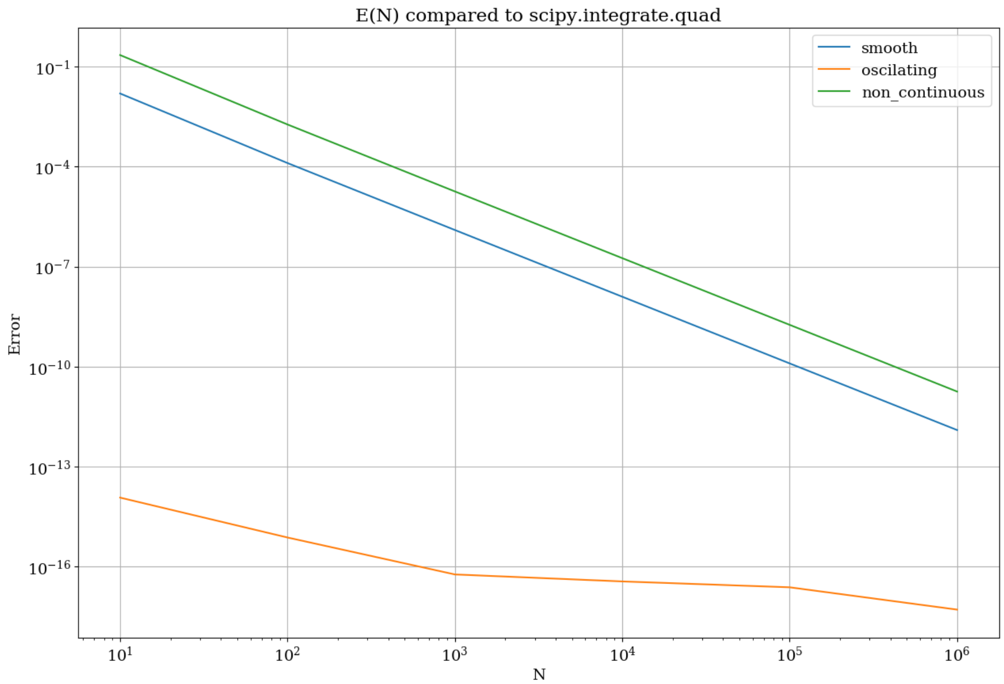


Рис. 1: Графики ошибки в зависимости от числа разбиений сетки

1.2 Оценка с использованием правила Рунге

Оценка на погрешность вычисления интеграла при числе шагов $2n$ можно оценить сверху:

$$\Delta_{2N} \leqslant \frac{|I_{2N} - I_N|}{2^{p-1} - 1} = \frac{|I_{2N} - I_N|}{3} \text{ т.к. для метода трапеций } p = 3 \tag{5}$$

Графики зависимости погрешности (верхней оценки) изображены на Рис. 2

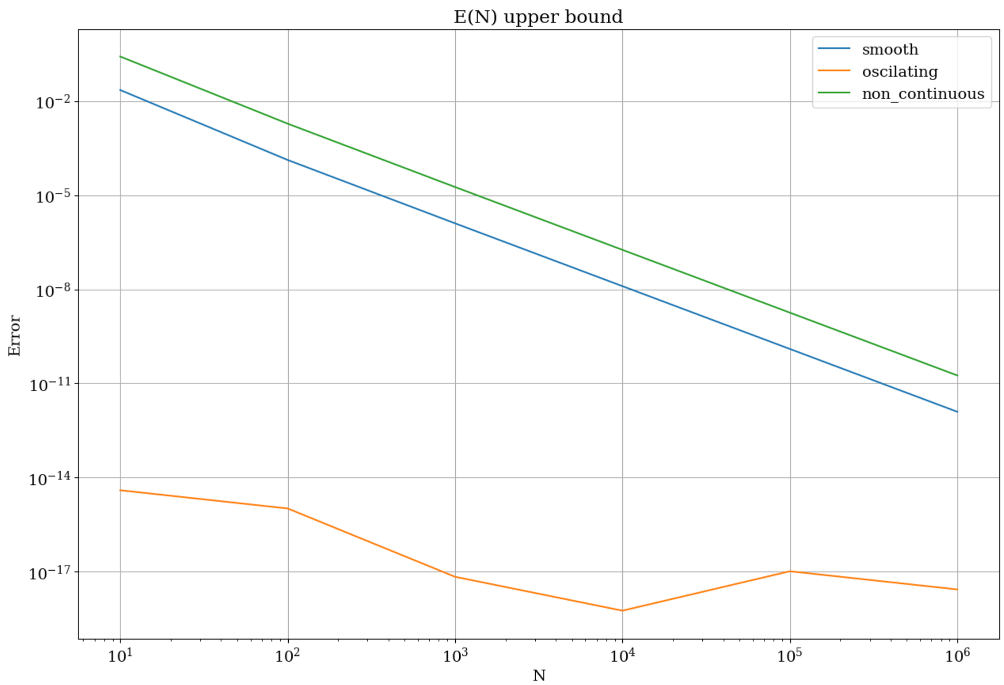


Рис. 2: Оценка на погрешность вычисления интеграла

2 Модуль интерполяции

В данном разд описан метод интерполяции, использующий кубический сплайн дефекта 1 и метод прогонки для вычислени коэффициентов.

2.1 Описание алгоритма

Функция задана на отрезке $[a, b]$, разбитом на части $[x_{i-1}, x_i]$. Кубический сплайн дефекта 1 – функция $S(x)$, которая:

- На каждом подотрезке является многочленом степени $\leqslant 3$
- Имеет непрерывную первую и вторую производные на $[a, b]$
- В точках $S(x_i) = f(x_i)$

На каждом отрезке $[x_{i-1}, x_i]$ функция $S(x)$ – некоторый многочлен 3

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \tag{6}$$

Тогда $S_i(x_i) = a_i, S_i'(x_i) = b_i, S_i''(x_i) = c_i$. Условия непрерывности всех производных до второго порядка включительно записываются следующим образом:

$$\begin{gathered} S_i(x_{i-1}) = S_{i-1}(x_{i-1}) \\ S_i'(x_{i-1}) = S_{i-1}'(x_{i-1}) \\ S_i''(x_{i-1}) = S_{i-1}''(x_{i-1}) \end{gathered}$$

Также запишем условие интерполяции: $S(x_i) = f(x_i)$. Можем выразить неизвестные коэфициенты следующим образом:

$$\begin{aligned} &a_i = f(x_i) \\ &d_i = \frac{c_i - c_{i-1}}{3h_i} \\ &b_i = \frac{a_i - a_{i-1}}{h_i} + \frac{2c_i + c_{i-1}}{3} \cdot h_i \\ &c_{i-1}h_i + 2c_i(h_i + h_{i+1}) + c_{i+1}h_{i+1} = 3\left(\frac{a_{i+1} - a_i}{h_{i+1}} - \frac{a_i - a_{i-1}}{h_i}\right), \quad i = 1, 2, \dots, n-1 \end{aligned}$$

Если предположить, что $c_0 = c_N = 0$, то последнюю систему можно решить методом прогонки, а затем вычислить оставчиеся коэффициенты.

2.1.1 Метод прогонки

Пусть дана СЛАУ с трехдиагональной матрицей A:

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = f_i, \quad i = 0, \dots, n \tag{7}$$

Можно расширеную матрицу системы к следующему виду:

$$\begin{pmatrix} 1 & P_1 & 0 & 0 & \cdots & 0 & Q_1 \\ 0 & 1 & P_2 & 0 & \cdots & 0 & Q_2 \\ 0 & 0 & 1 & P_3 & \cdots & 0 & Q_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & Q_n \end{pmatrix} \tag{8}$$

Откуда получить решение: $x_i = -P_i x_{i+1} + Q_i, \quad i = 1, 2, \dots, n-1$. При этом коэффициенты можно посчитать так:

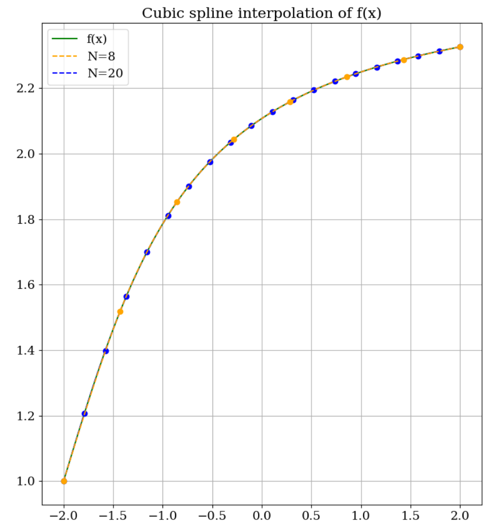
$$P_i = \frac{c_i}{-b_i - a_i P_{i-1}}, \quad Q_i = \frac{a_i Q_{i-1} - f_i}{-b_i - a_i P_{i-1}}, \quad i = 1, 2, \dots, n-1 \tag{9}$$

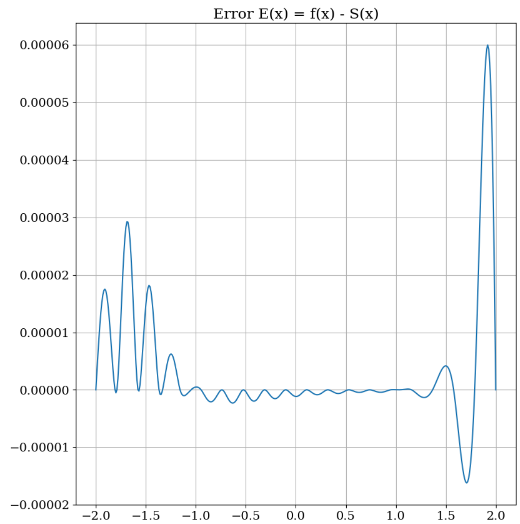
Поиск коэффициентов – прямой ход метода прогонки. Далее находим x_n по формуле, оставшиеся коэфициенты выражаем рекуррентно:

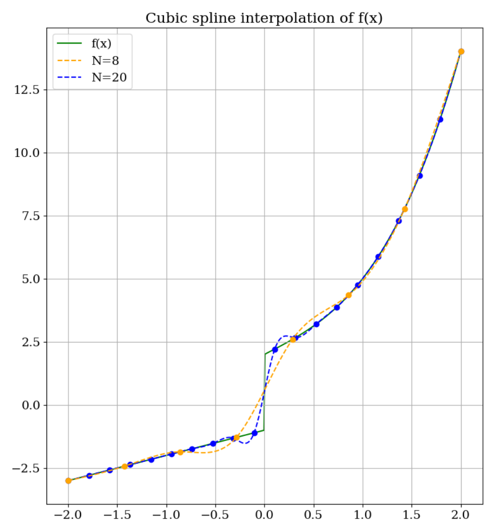
$$x_n = \frac{a_n Q_{n-1} - f_n}{-b_n - a_n P_{n-1}} \tag{10}$$

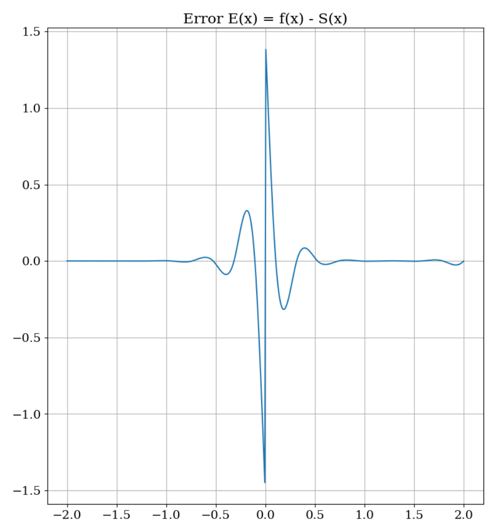
2.2 Графики

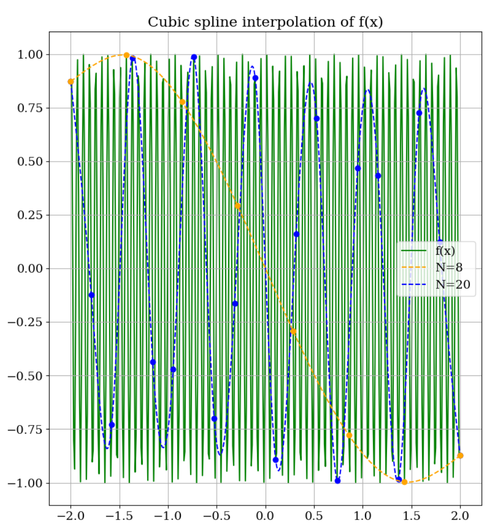
Здесь приведены графики для различных видов функций: гладкой, разрывной и осцилирующей (взяты из предыдущего пункта):

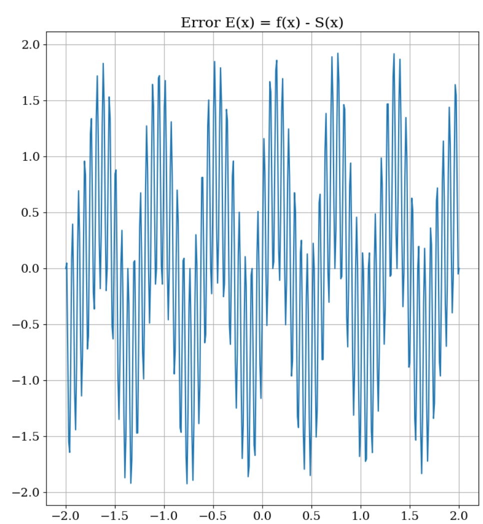












3 Модуль решения задачи Коши

3.1 Описание алгоритма

Для решения ситем дифференциальных уравнений было решено использовать метод Рунге — Кутты четвертого порядка. Рассмотрим задачу Коши для системы обыкновенных дифференциальных уравнений первого порядка. (Далее $x, f, k_i \in \mathbb{R}^n$, а $t, h \in \mathbb{R}$)

$$x' = f(t, x), \quad x(t_0) = x_0 \tag{11}$$

Тогда приближенное значение в последующих точках вычисляется по итерационной формуле:

$$x_{n+1} = x_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{12}$$

Вычисление нового значения проходит в четыре стадии:

$$k_1 = h \cdot f(t_n, x_n) \tag{13}$$

$$k_2 = h \cdot f\left(t_n + \frac{h}{2}, x_n + \frac{k_1}{2}\right) \tag{14}$$

$$k_3 = h \cdot f\left(t_n + \frac{h}{2}, x_n + \frac{k_2}{2}\right) \tag{15}$$

$$k_4 = h \cdot f\left(t_n + h, x_n + k_3\right) \tag{16}$$

Этот метод имеет четвёртый порядок точности. Это значит, что ошибка на одном шаге имеет порядок $O(h^5)$, а суммарная ошибка на конечном интервале имеет порядок $O(h^4)$

3.2 Результаты

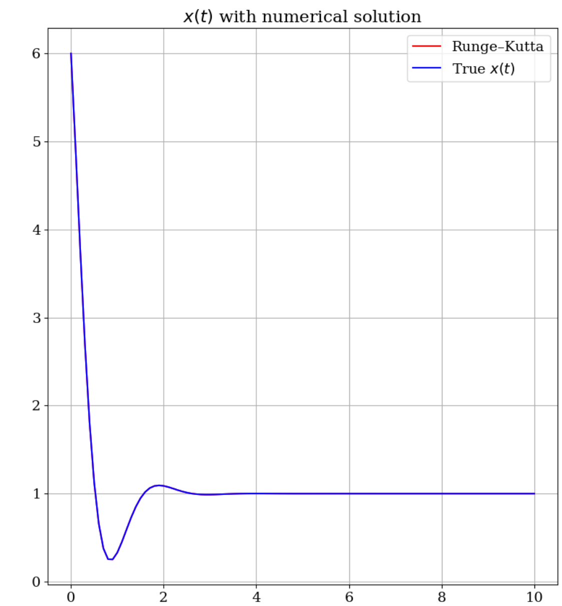
Для проверки реализованного будем использовать следующую задачу Коши

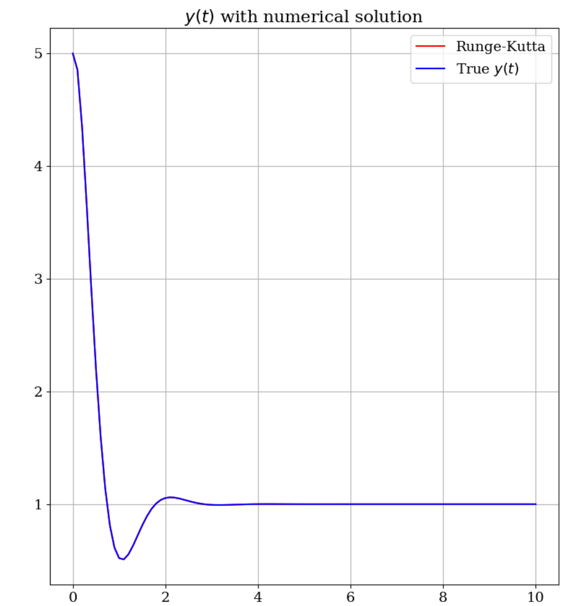
$$\begin{cases} x' = 2x - 5y + 3 \\ y' = 5x - 6y + 1 \\ x(0) = 6 \\ y(0) = 5 \end{cases} \tag{17}$$

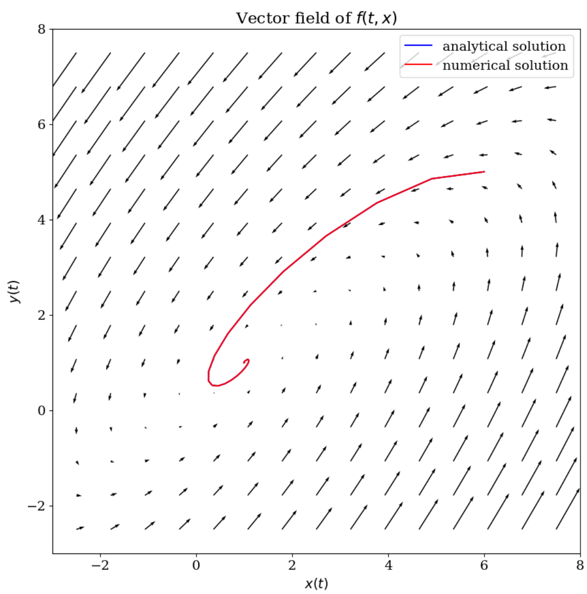
Данная система имеет единственное решение т.к. вектор функция $f(t, x, y)$ липшицева. Она имеет следующее аналитическое решение:

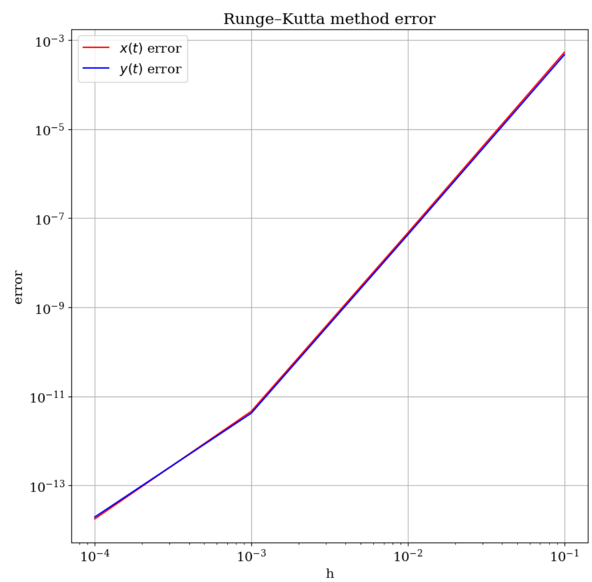
$$\begin{cases} x(t) = 5e^{-2t}\cos(3t) + 1 \\ y(t) = e^{-2t}(4\cos(3t) + 3\sin(3t)) + 1 \end{cases} \tag{18}$$

Далее приведены результаты. На Рис 3.2 изображены аналитические и численные решения. На оставшихся двух изображен фазовый портрет и график зависимости ошибки от величины шага разбиения (из графика видно, что ошибка в действительности равна $O(h^5)$).


$x(t)$ with numerical solution
Runge–Kutta
True $x(t)$
0
1
2
3
4
5
6
8
10


$y(t)$ with numerical solution
Runge–Kutta
True $y(t)$
0
1
2
3
4
5
6
8
10


Vector field of $f(t, x)$
analytical solution
numerical solution
$y(t)$
$x(t)$
−2
0
2
4
6
8


Runge–Kutta method error
$x(t)$ error
$y(t)$ error
error
h
10^{-3}
10^{-5}
10^{-7}
10^{-9}
10^{-11}
10^{-13}
10^{-4}
10^{-2}
10^{-1}