

Санкт-Петербургский политехнический университет
Высшая школа прикладной математики
и вычислительной физики, ФизМех

Направление подготовки
«Прикладная математика и информатика»

Отчет по лабораторной работе №6
«Численное решение ОДУ многошаговыми методами»

Дисциплина: «Численные методы»

Выполнил студент гр. 5030102/00003

Анищенко М.Д.

Преподаватель:

Курц В.В.

Санкт-Петербург 2022

Оглавление

1	Формулировка задачи и её формализация	3
1.1	Формулировка задачи	3
1.2	Формализация задачи	3
2	Алгоритм метода и условия его применимости	4
2.1	Алгоритм метода	4
2.2	Условия применимости	4
3	Анализ задачи	5
4	Тестовый пример	5
5	Контрольные тесты	8
6	Модульная структура программы	8
7	Численный анализ	9
7.1	Графики функций	9
7.2	График ошибки	9
7.3	Зависимость локальной и глобальной погрешностей от h	10
7.4	Влияние ошибок в начальных данных	11
7.5	Сравнение погрешностей двух методов	12
8	Общие выводы	13

Глава 1

Формулировка задачи и её формализация

1.1 Формулировка задачи

Пусть дано дифференциальное уравнение 1-го порядка

$$F(x, y, y') = 0, \quad (1.1)$$

где $y(x)$ — неизвестная функция. Поставлена задача Коши:

$$\begin{cases} y' = f(x, y), x \in [a, b] \\ y(a) = y_0 \end{cases} \quad (1.2)$$

Необходимо найти приближенное решение этой задачи.

1.2 Формализация задачи

В данном варианте необходимо решить следующую задачу Коши методом Эйлера с итерационной обработкой:

$$\begin{cases} (2x + 1)y' = 4x + 2y, x \in [0, 4] \\ y(0) = 1 \end{cases} \quad (1.3)$$

Из условия известно точное решение, с которым мы будем сравнивать решение, полученное методом Эйлера:

$$y = (2x + 1) \ln |2x + 1| + 1 \quad (1.4)$$

Глава 2

Алгоритм метода и условия его применимости

2.1 Алгоритм метода

Чтобы решить ОДУ $y' = f(x, y)$ необходимо проинтегрировать это выражение на отрезке $[x_{k-1}, x_k]$:

$$y_k - y_{k-1} = \int_{x_{k-1}}^{x_k} f(x, y(x)) dx \quad (2.1)$$

Воспользуемся для этого методом трапеций и получим

$$y_k = y_{k-1} + \frac{h}{2}(f(x_{k-1}, y_{k-1}) + f(x_k, y_k)) \quad (2.2)$$

Получаем уравнение вида $x = \phi(x)$, где $x = y_k$. Теперь зададим $y_k^{(0)}$ и

$$y_k^{(j+1)} = y_{k-1} + \frac{h}{2}(f(x_{k-1}, y_{k-1}) + f(x_k, y_k^{(j)})) \quad (2.3)$$

Точку $y_k^{(0)}$ называют разгонной (стартовой) и находят с помощью явной формулы метода Эйлера

$$y_k^{(0)} = y_{k-1} + hf(x_{k-1}, y_{k-1}) \quad (2.4)$$

Применяя формулы (2.3) и (2.4), получаем метод Эйлера с итерационной обработкой.

2.2 Условия применимости

Для данного метода существует следующее условие сходимости:

$$|\phi'(x)| < 1 \Rightarrow \left| \frac{\partial}{\partial y_k} \left(\frac{h}{2} f(x_k, y_k) \right) \right| = \frac{h}{2} \left| \frac{\partial f}{\partial y} \right| \leq \frac{h}{2} L < 1 \Rightarrow h < \frac{2}{L} \quad (2.5)$$

Глава 3

Анализ задачи

Проверим выполнение условия применимости для нашей задачи (1.3). Для этого нужно потребовать $h \leq \frac{2}{L}$. Найдем константу Липшица. Так как функция f непрерывна на выбранном отрезке, то по теореме Лагранжа

$$L = \sup \left| \frac{\partial f}{\partial y} \right| = \sup \left| \frac{\partial}{\partial y} \left(\frac{4x + 2y}{2x + 1} \right) \right| = \sup \left| \frac{2}{2x + 1} \right| = 2 \quad (3.1)$$

Тогда из пункта 2.2 следует, что

$$h < \frac{2}{L} = \frac{2}{2} = 1 \quad (3.2)$$

С учетом выбранного отрезка $[0, 4]$ имеем

$$h = \frac{b - a}{n} = \frac{4}{n} < 1 \Rightarrow n > 4 \quad (3.3)$$

Отсюда следует, что метод Эйлера с итерационной обработкой для данной задачи будет сходиться при $n > 4$.

Глава 4

Тестовый пример

Решим нашу задачу при $n = 5$, итерационную формулу будем применять для каждого y_k 3 раза.

$$X = (0, 0.8, 1.6, 2.4, 3.2, 4)$$

$$y_0 = y(0) = 1$$

$$y_1^0 = y_0 + hf(0, 1) = 1 + 0.8 \cdot 2 = 2.6$$

$$y_1^1 = y_0 + \frac{h}{2}(f(0, 1) + f(0.8, 2.6)) = 1 + \frac{0.8}{2}(2 + 3.2308) = 3.09232$$

$$y_1^2 = y_0 + \frac{h}{2}(f(0, 1) + f(0.8, 3.09232)) = 1 + \frac{0.8}{2}(2 + 3.6095) = 3.24379$$

$$y_1^3 = y_0 + \frac{h}{2}(f(0, 1) + f(0.8, 3.24379)) = 1 + \frac{0.8}{2}(2 + 3.72599) = 3.2904$$

$$y_2^0 = y_1 + hf(0.8, 3.2904) = 3.2904 + 0.8 \cdot 3.7618 = 6.29987$$

$$y_2^1 = y_1 + \frac{h}{2}(f(0.8, 3.2904) + f(1.6, 6.29987)) = 3.2904 + \frac{0.8}{2}(3.7618 + 4.52375) = 6.60463$$

$$y_2^2 = y_1 + \frac{h}{2}(f(0.8, 3.2904) + f(1.6, 6.60463)) = 3.2904 + \frac{0.8}{2}(3.7618 + 4.66887) = 6.66268$$

$$y_2^3 = y_1 + \frac{h}{2}(f(0.8, 3.2904) + f(1.6, 6.66268)) = 3.2904 + \frac{0.8}{2}(3.7618 + 4.69651) = 6.67374$$

$$y_3^0 = y_2 + hf(1.6, 6.67374) = 6.67374 + 0.8 \cdot 4.70178 = 10.4352$$

$$y_3^1 = y_2 + \frac{h}{2}(f(1.6, 6.67374) + f(2.4, 10.4352)) = 6.67374 + \frac{0.8}{2}(4.70178 + 5.25352) = 10.6559$$

$$y_3^2 = y_2 + \frac{h}{2}(f(1.6, 6.67374) + f(2.4, 10.6559)) = 6.67374 + \frac{0.8}{2}(4.70178 + 5.32962) = 10.6863$$

$$y_3^3 = y_2 + \frac{h}{2}(f(1.6, 6.67374) + f(2.4, 10.6863)) = 6.67374 + \frac{0.8}{2}(4.70178 + 5.3401) = 10.6905$$

$$y_4^0 = y_3 + hf(2.4, 10.6905) = 10.6905 + 0.8 \cdot 5.34155 = 14.9637$$

$$y_4^1 = y_3 + \frac{h}{2}(f(2.4, 10.6905) + f(3.2, 14.9637)) = 10.6905 + \frac{0.8}{2}(5.34155 + 5.77397) = 15.1367$$

$$y_4^2 = y_3 + \frac{h}{2}(f(2.4, 10.6905) + f(3.2, 15.1367)) = 10.6905 + \frac{0.8}{2}(5.34155 + 5.82073) = 15.1554$$

$$y_4^3 = y_3 + \frac{h}{2}(f(2.4, 10.6905) + f(3.2, 15.1554)) = 10.6905 + \frac{0.8}{2}(5.34155 + 5.82578) = 15.1574$$

$$y_5^0 = y_4 + hf(3.2, 15.1574) = 15.1574 + 0.8 \cdot 5.82632 = 19.8185$$

$$y_5^1 = y_4 + \frac{h}{2}(f(3.2, 15.1574) + f(4, 19.8185)) = 15.1574 + \frac{0.8}{2}(5.82632 + 6.1818) =$$

$$= 19.9607$$

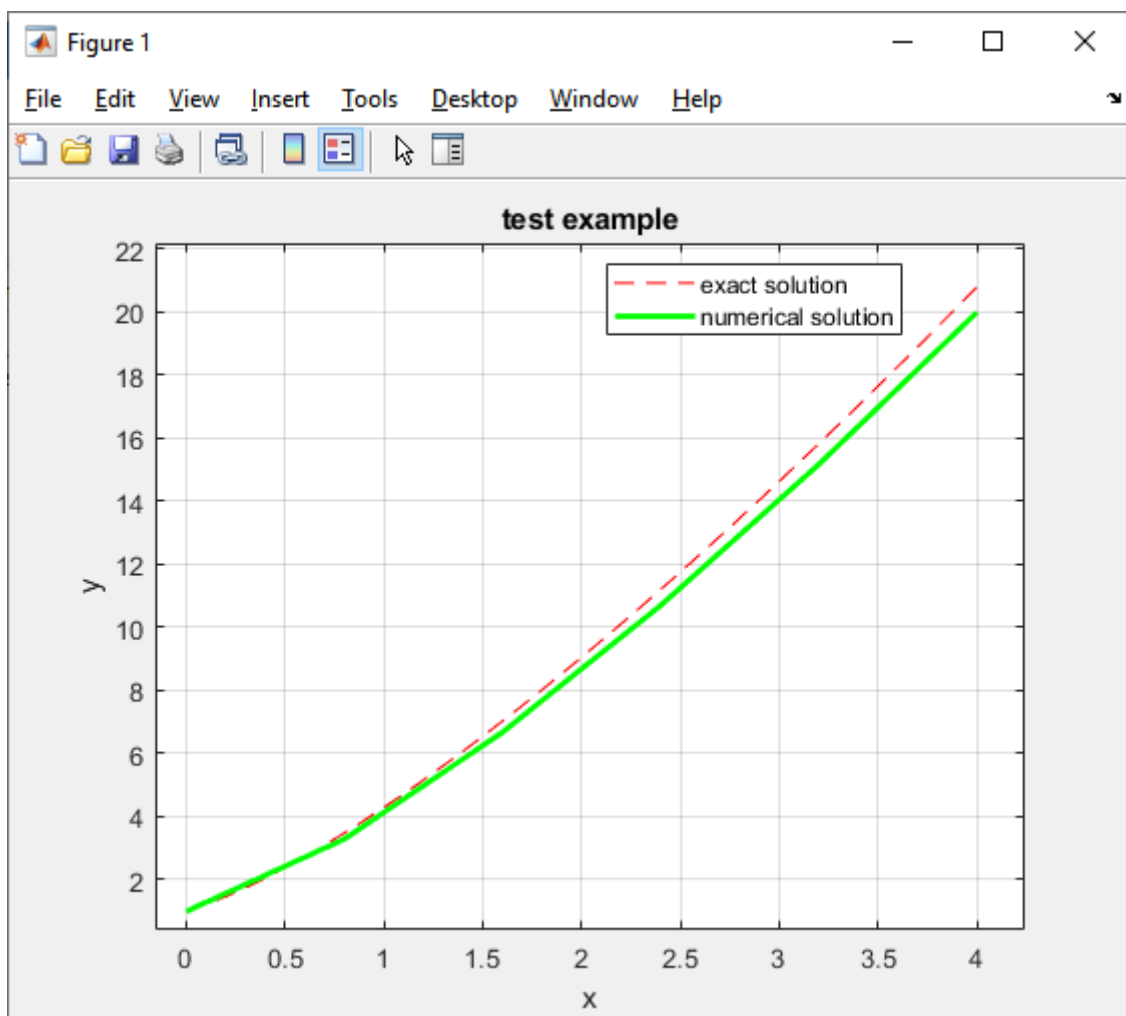
$$y_5^2 = y_4 + \frac{h}{2}(f(3.2, 15.1574) + f(4, 19.9607)) = 15.1574 + \frac{0.8}{2}(5.82632 + 6.21349) = 19.9734$$

$$y_5^3 = y_4 + \frac{h}{2}(f(3.2, 15.1574) + f(4, 19.9734)) = 15.1574 + \frac{0.8}{2}(5.82632 + 6.21631) = 19.9745$$

Итого получаем

$$Y = (1, 3.2904, 6.67374, 10.6905, 15.1574, 19.9745)$$

Построим график по полученным значениям



Ошибка вычисления:

x	0	0.8	1.6	2.4	3.2	4
ε	0	0.1939	0.3536	0.5051	0.6536	0.8005

Глава 5

Контрольные тесты

В рамках данного исследования проведем следующие тесты:

1. Построим графики точного и численного решения.
2. Построим график ошибки.
3. Исследуем зависимость погрешности от h . Сравним с теоретической ошибкой.
4. Исследуем влияние ошибок в начальных данных.
5. Сравним полученные результаты с предыдущей работой (явным методом Эйлера).

Глава 6

Модульная структура программы

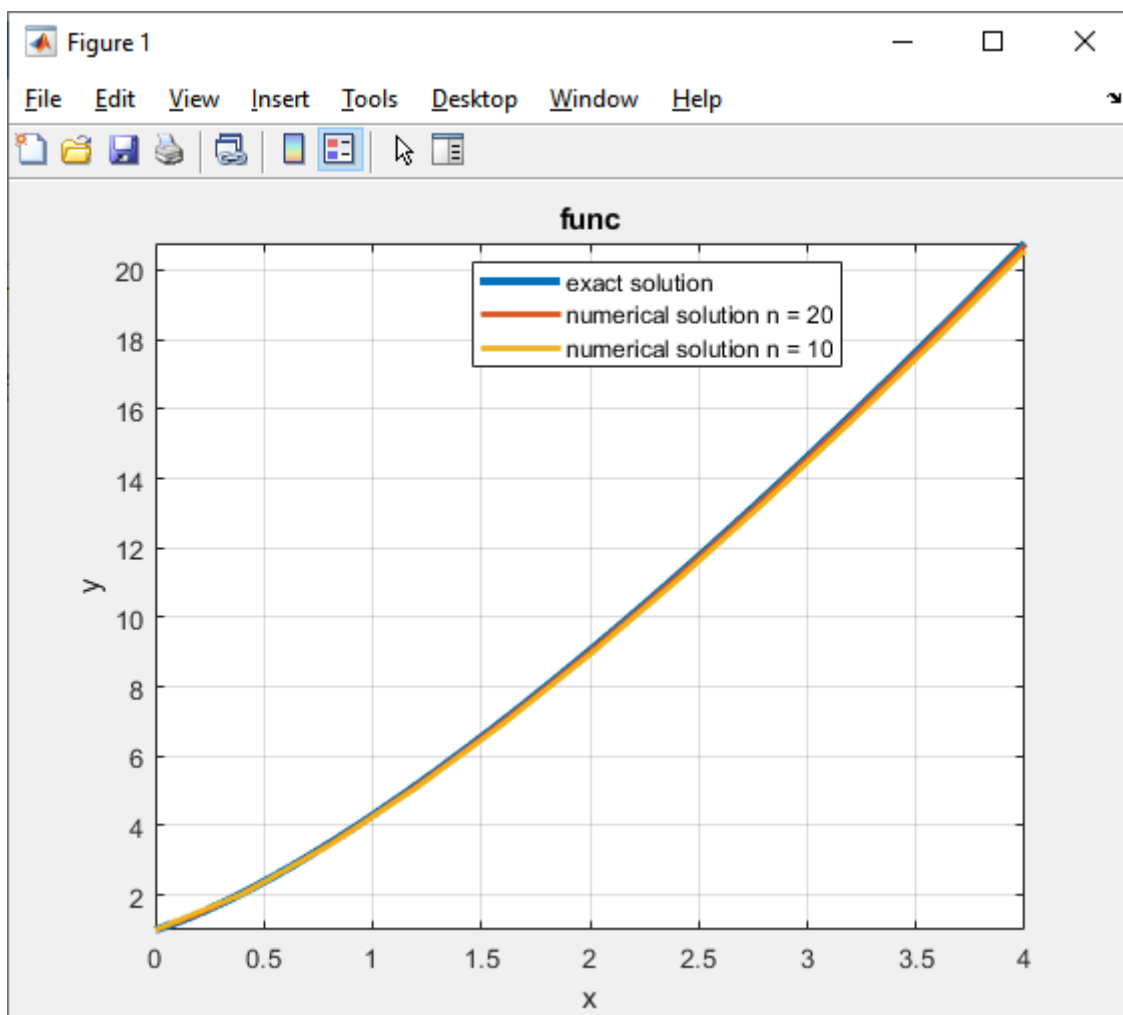
- `double f(double x, double y)` — функция, вычисляющая значение $f(x, y)$.
- `void CreateGrid(double a, double b, int n, vector<double>& x)` — функция для составления равномерной сетки.
- `vector<double> EulerIter(int n, double f_a)` — функция, реализующая итерационную составляющую исследуемого метода.
- `double EulerStep(double x, double y, double h)` — функция, реализующая шаг по явному методу Эйлера для разгонных точек.
- `void WriteResult(vector<double>& y, FILE* file)` — функция для записи всего полученного вектора y в файл.
- `void WriteParcialResult(vector<double>& y, FILE* file)` — функция для записи в файл данных, необходимых для исследований 2-5.

Глава 7

Численный анализ

7.1 Графики функций

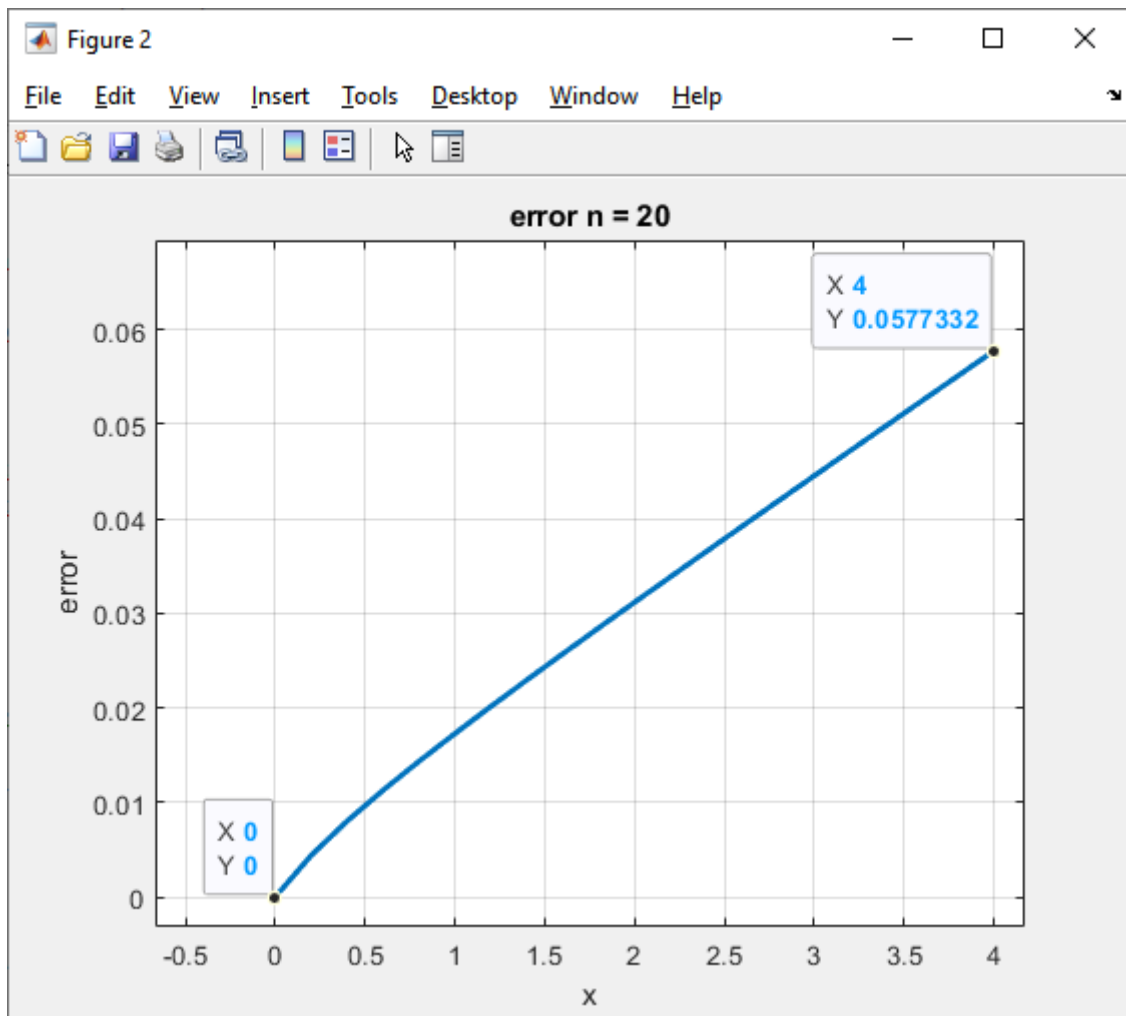
Построим графики точного и численного решения для $n = 20$ и $n = 10$.



Из графика видно, что график численного решения очень близок к точному решению. Это было заметно уже при $n = 5$ в тестовом примере.

7.2 График ошибки

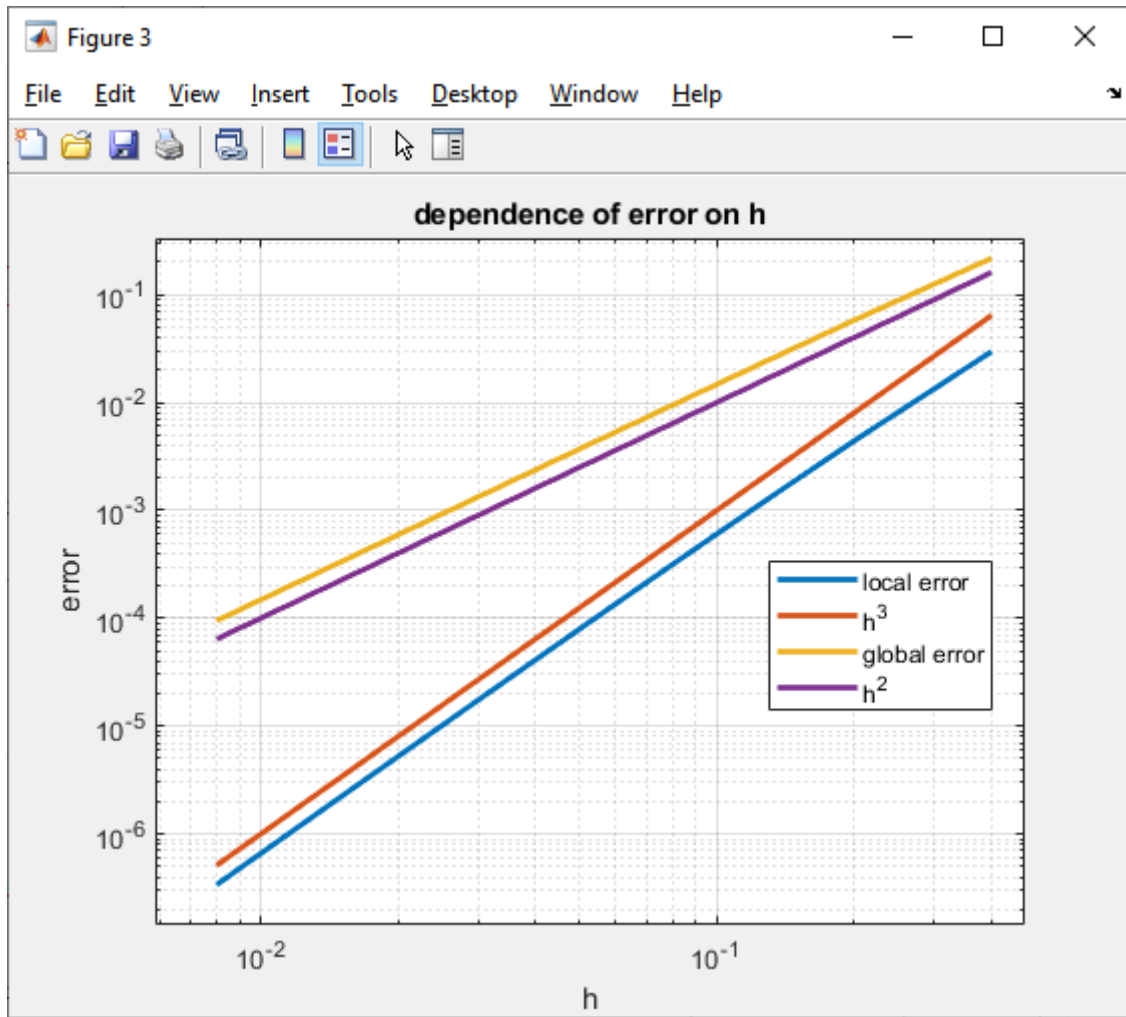
Для решений из предыдущего пункта построим графики ошибок.



Заметим, что ошибка на левом конце равна нулю, так как эта точка дана в условии. И ошибка постепенно увеличивается с приближением к правому концу, поскольку ошибка накапливается из локальных погрешностей на каждом шаге.

7.3 Зависимость локальной и глобальной погрешностей от h

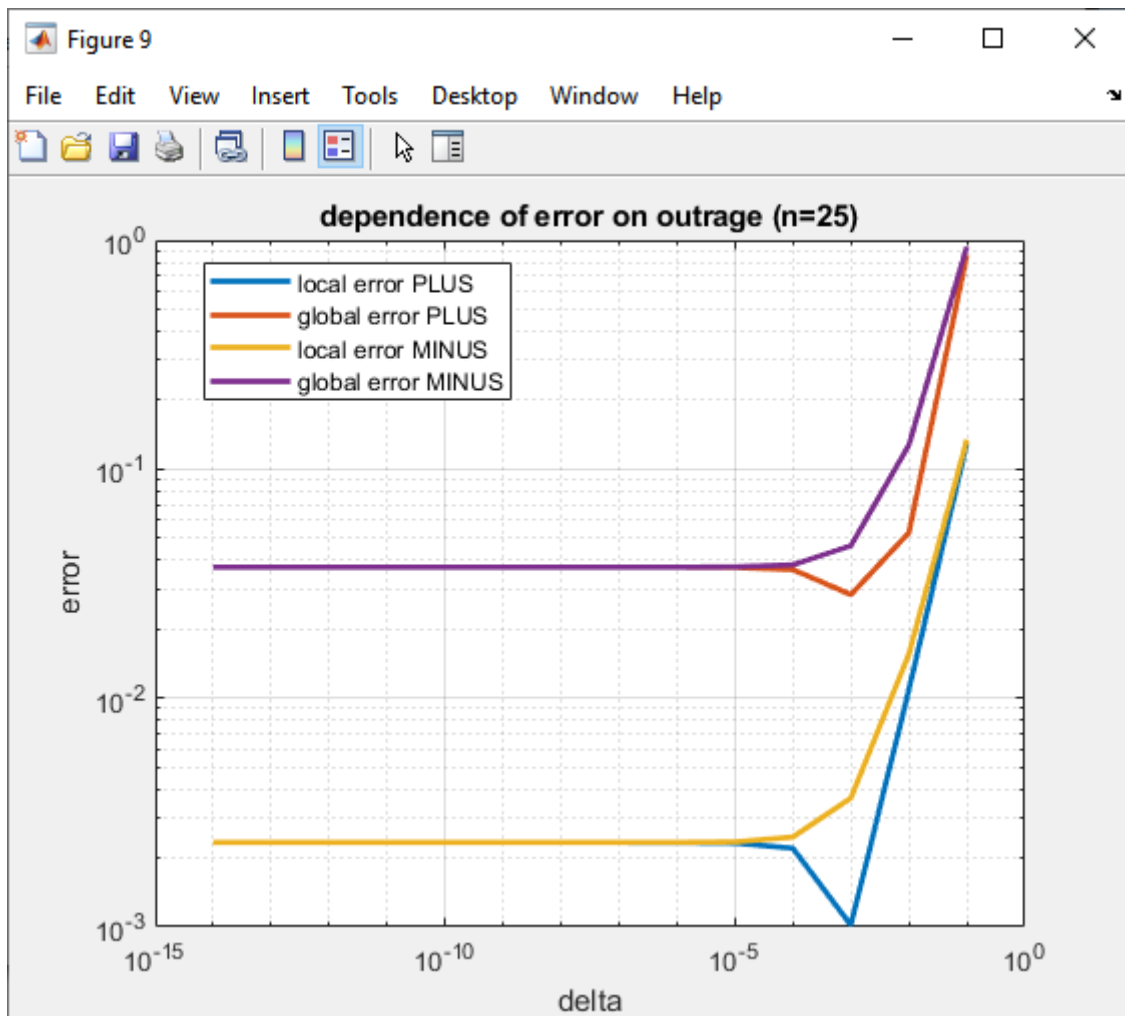
Будем изменять количество разбиений n от 500 до 10 и построим графики зависимости локальной и глобальной ошибок от h .



На графике дополнительно изображены линии $h^2(h)$ и $h^3(h)$. Заметим, что локальная ошибка соответствует $\mathcal{O}(h^3)$, а глобальная — $\mathcal{O}(h^2)$. Порядок точности метода Эйлера с итерационной обработкой на единицу выше, чем у явного метода Эйлера, как и ожидалось.

7.4 Влияние ошибок в начальных данных

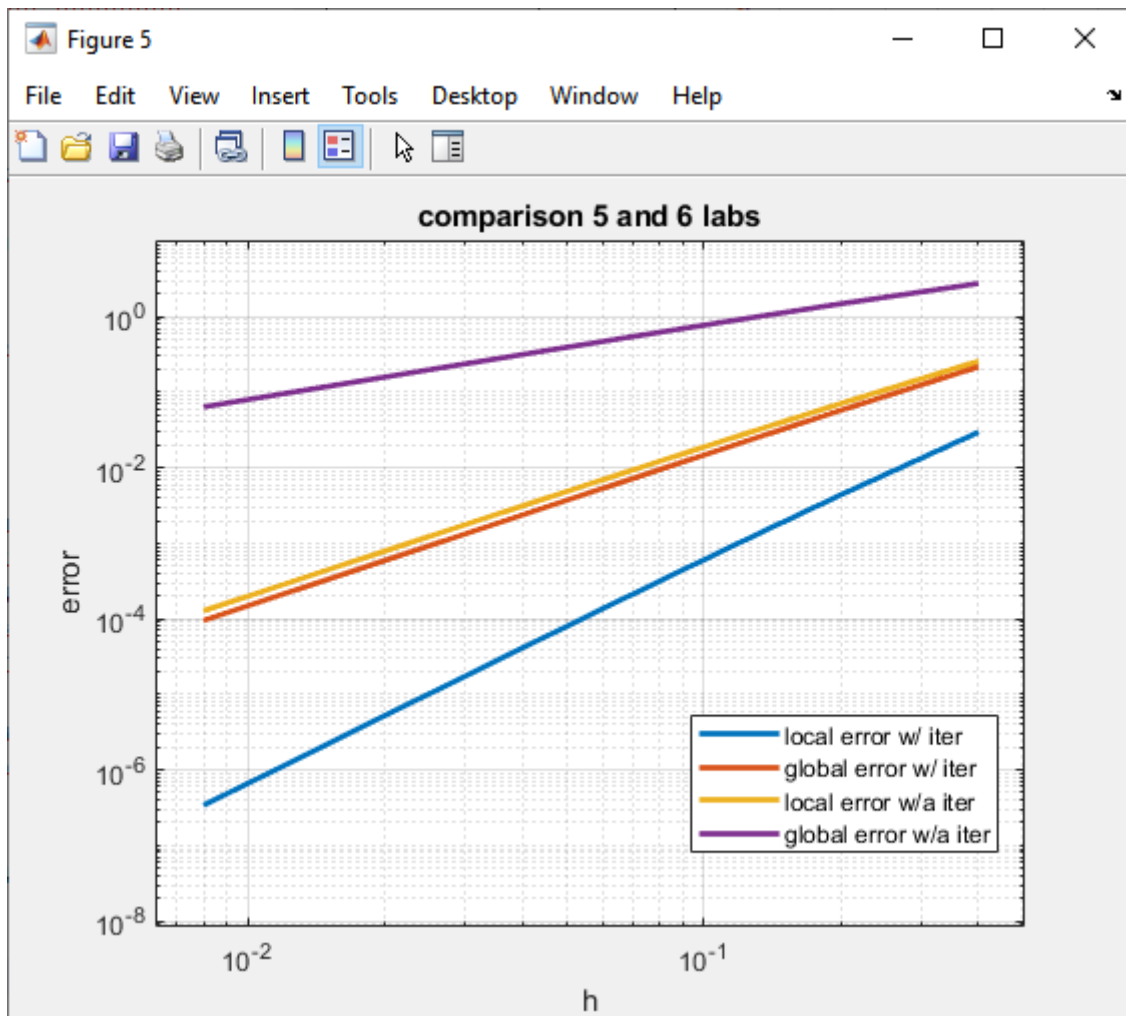
В этом пункте исследования будем вносить возмущение в начальные данные задачи Коши, т.е. y_0 . Порядок возмущения будем менять от 10^{-14} до 10^{-1} , при этом возмущение будет как положительным, так и отрицательным. Количество отрезков разбиения $n = 25$.



Поскольку каждое следующее значение y_k по сути является точкой касания с точным графиком в точке x_k , а график возрастает, то график численного решения приближается к точному снизу. Это можно заметить на графиках из пунктов 4 и 7.1. Поэтому при внесении положительного возмущения мы "поднимаем" график на некоторую величину. При этом возрастает ошибка на левом конце, т.к. без возмущения она равна нулю. Но при этом как локальная ошибка в точке x_1 , так и глобальная ошибка в точке x_n уменьшается до тех пор, пока график численного решения не пересечет график точного решения. После этого обе ошибки увеличиваются. При внесении отрицательного возмущения мы "опускаем" график, и, как следствие, ошибка увеличится. При этом пока величина возмущения не превосходит порядок теоретической ошибки, это возмущение практически не оказывает влияния на конечный ответ.

7.5 Сравнение погрешностей двух методов

Сравним локальную и глобальную ошибки явного метода Эйлера и метода Эйлера с итерационной обработкой в зависимости от h .



Из этого графика видно, что метод Эйлера с итерационной обработкой на порядок точнее явного метода Эйлера, что достигается благодаря итерационному уточнению.

Глава 8

Общие выводы

Данное исследование показало, что метод Эйлера с итерационной обработкой достаточно точен, не требует объемных вычислений (ошибка невелика даже при $n = 10$, которому соответствует около 100 обращений к функции $f(x, y)$). При этом метод устойчив к возмущению невысокого порядка. В сравнении с явным методом Эйлера он очевидно выигрывает, поскольку дает более точное решение при не более трудной реализации и несущественно возрастающем объеме вычислений.