

Санкт-Петербургский Политехнический Университет
Высшая школа прикладной математики и вычислительной физики, ФизМех
01.03.02 Прикладная математика и информатика

Отчет по лабораторной работе № 12
"Решение задачи Коши для ОДУ 1 порядка методами
Рунге-Кутты"
Метод Рунге-Кутты-Мерсона
дисциплина "Численные методы"

Выполнил студент гр. 5030102/20003
Преподаватель

Жохов О. Д.
Козлов К.Н.

Сентябрь, 2024

Содержание

1	Формулировка задачи и её формализация	3
1.1	Формализация	3
1.2	Поставленная задача	3
2	Алгоритм метода и условия его применимости	3
2.1	Алгоритм	3
2.2	Условия применимости	3
3	Тестовый пример	3
4	Модульная структура программы и контрольные тесты	5
4.1	Контрольные тесты	5
4.2	Модульная структура	5
5	Численный анализ	7
5.1	Графики	7
5.2	Анализ графиков	11
6	Выводы	11

1 Формулировка задачи и её формализация

1.1 Формализация

Дано ОДУ 1-го порядка:

$$y' = f(x, y); \quad a \leq x \leq b$$

Для данной задачи Коши обеспечивается существование и единственность решения на отрезке $[a, b]$, $y(x_0) = y_0$. Необходимо найти общее решение $y = (x, C)$, чтобы выделить из него интегральную кривую $y = y(x)$, проходящую через заданную точку (x_0, y_0) .

1.2 Поставленная задача

Необходимо решить задачу Коши методом Кутты-Мерсона:

$$\begin{cases} y' = \frac{y^2(\ln(x)+2)\ln(x)}{x}, & 1 \leq x \leq 3 \\ y(1) = 1 \end{cases}$$

2 Алгоритм метода и условия его применимости

2.1 Алгоритм

$$h = \frac{b - a}{n}$$

$$k_1 = f(x_i, y_i)$$

$$k_2 = f\left(x_i + \frac{h}{3}, y_i + \frac{h}{3}k_1\right)$$

$$k_3 = f\left(x_i + \frac{h}{3}, y_i + \frac{h}{6}k_1 + \frac{h}{6}k_2\right)$$

$$k_4 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{8}k_1 + \frac{3h}{8}k_3\right)$$

$$k_5 = f\left(x_i + h, y_i + \frac{h}{2}k_1 - \frac{3h}{2}k_3 + 2hk_4\right)$$

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 4k_4 + k_5)$$

$$R = |2k_1 - 9k_3 + 8k_4 - k_5| \frac{h}{6}$$

Если $R > \epsilon$, то $h/=2$

Если $32R < \epsilon$, то $h*=2$

2.2 Условия применимости

$f(x, y)$ и $f'_y(x, y)$ непрерывны на отрезке $[a, b]$. То есть выполняются теоремы о существовании и единственности задачи Коши.

3 Тестовый пример

Ручной расчёт

$$\begin{cases} xy' - y = -y^2 \ln x (\ln x + 2) \\ y(1) = 1 \end{cases} \quad x \in [1, 2]$$

Точное решение: $\frac{3}{\ln^3 x + 3 \ln^2 x + 3}$

$$f(x, y) = y' = \frac{-y^2 (\ln x + 2) \ln x + y}{x}$$

$$I \quad k_1 = h \cdot f(x_0, y_0) = 0$$

$$k_2 = -0,2896 = f\left(x + \frac{h}{3}, y + \frac{k_1}{3}\right)$$

$$k_3 = f\left(x + \frac{h}{3}, y + h \cdot \frac{(k_1 + k_2)}{6}\right) = -0,213$$

$$k_4 = f\left(x + \frac{h}{2}, y + h \cdot \frac{k_1 + 3k_3}{8}\right) = -0,3575$$

$$k_5 = f\left(x + h, y + h \left(\frac{k_1 + 3k_3}{2} + 2k_4\right)\right) = -0,4659$$

$$y_{next} = y_0 + \frac{h}{6} (k_4 + 4k_5 + k_6) = 0,8421$$

Рис. 1: Ручной расчёт, часть 1

$k_1 = -0,461; k_2 = -0,4506; k_3 = -0,4516;$
 $k_4 = -0,4345; k_5 = -0,4717; y_2 = 0,6278$

N	x	\tilde{y}	y	$ y - \tilde{y} $
1	1,5	0,8421	0,8427	0,0006
2	2	0,6278	0,6284	0,0006

Рис. 2: Ручной расчёт, часть 2

4 Модульная структура программы и контрольные тесты

4.1 Контрольные тесты

Для исследования метода средних прямоугольников для функции на отрезке $[1; 3]$ будем рассматривать точность вычисления от 10^{-13} до 10^{-3} порядков.

4.2 Модульная структура

```
typedef struct {
    double y;
    double h;
    double R;
} current;
```

- структура, представляющая из себя промежуточный этап вычисления значения искомой функции со значением y , шагом h и точностью R в точке.

```
double f(double x, double y)
```

- функция-производная искомой функции

```
double F(double x)
```

- искомая функция

```
double KuttaMersonMethodConstantStep(double x, double y, double h)
```

- метод РКМ с постоянным шагом h

```
current KuttaMersonMethod(double x, double y, double h, double delta)
```

- метод РКМ с заданной изначальной точностью `delta` и возвращающий структуру `current`
`void data()`
- функция со всеми необходимыми расчётами, записывающая результаты в файлы `output1.txt`, `output2.txt`, `output3.txt`, `output4.txt`.

5 Численный анализ

5.1 Графики

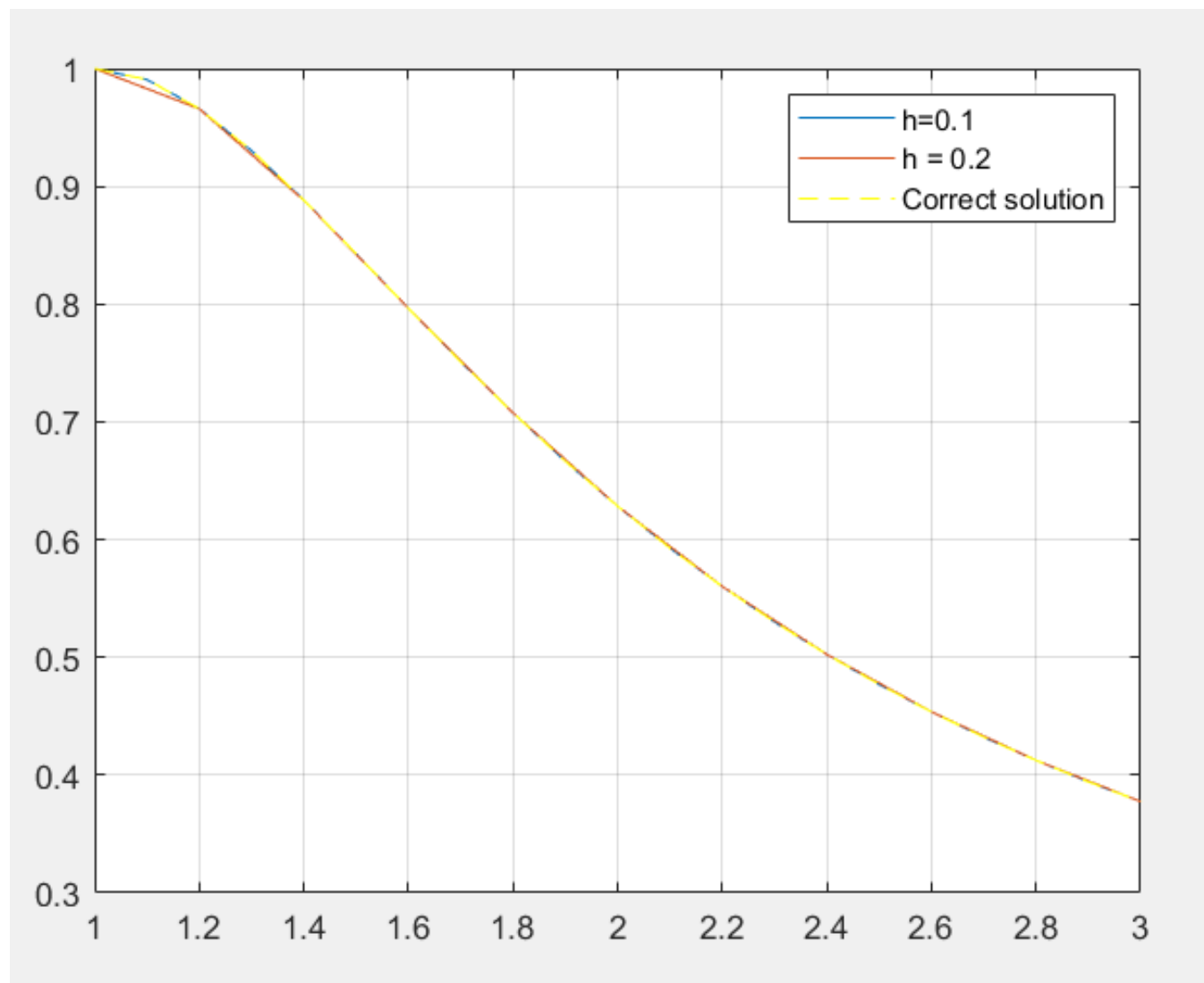


Рис. 3: Графики точного решения, решений с фиксированными шагами 0.1 и 0.2

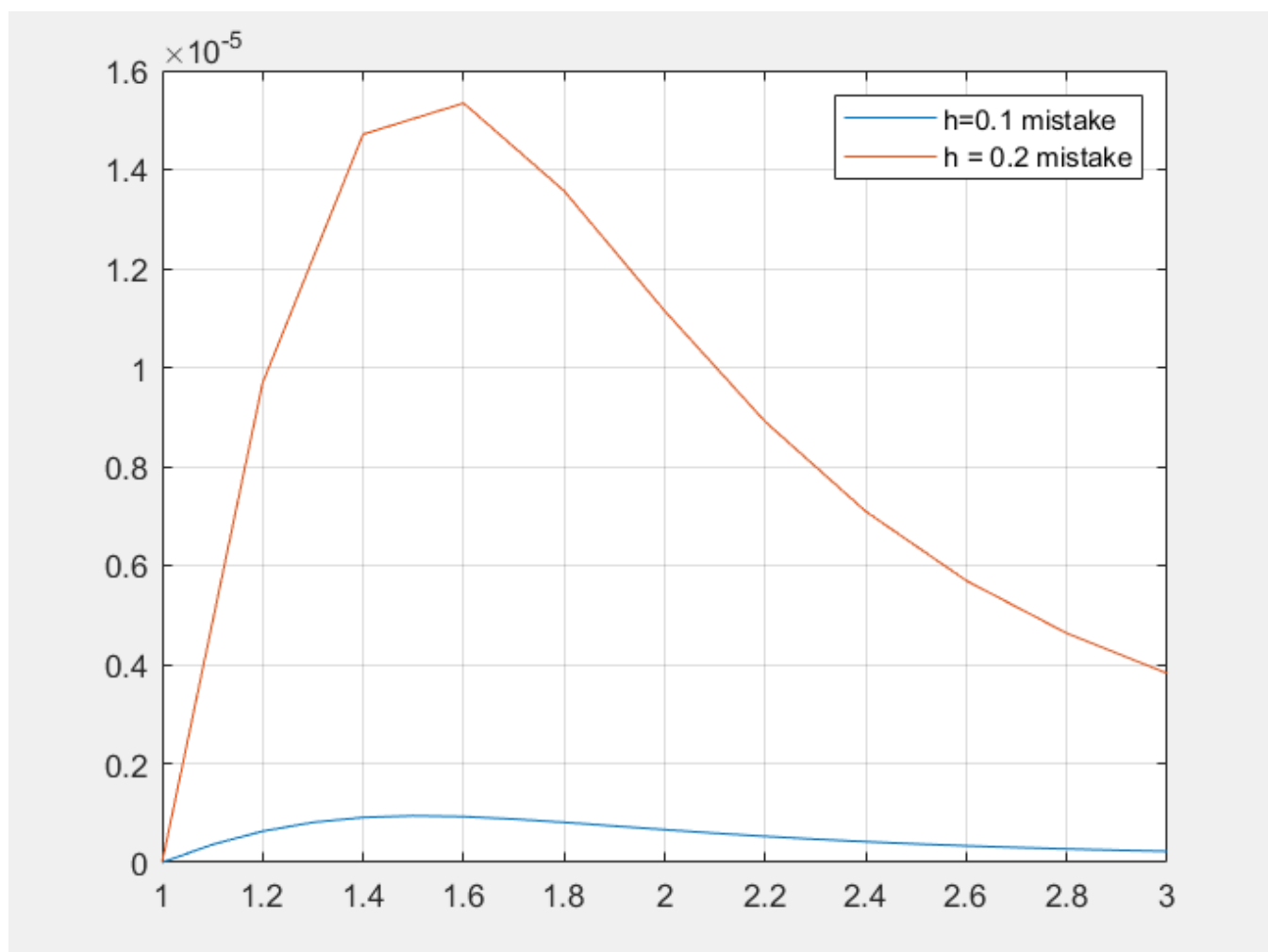


Рис. 4: Графики ошибки решения с фиксированным шагом

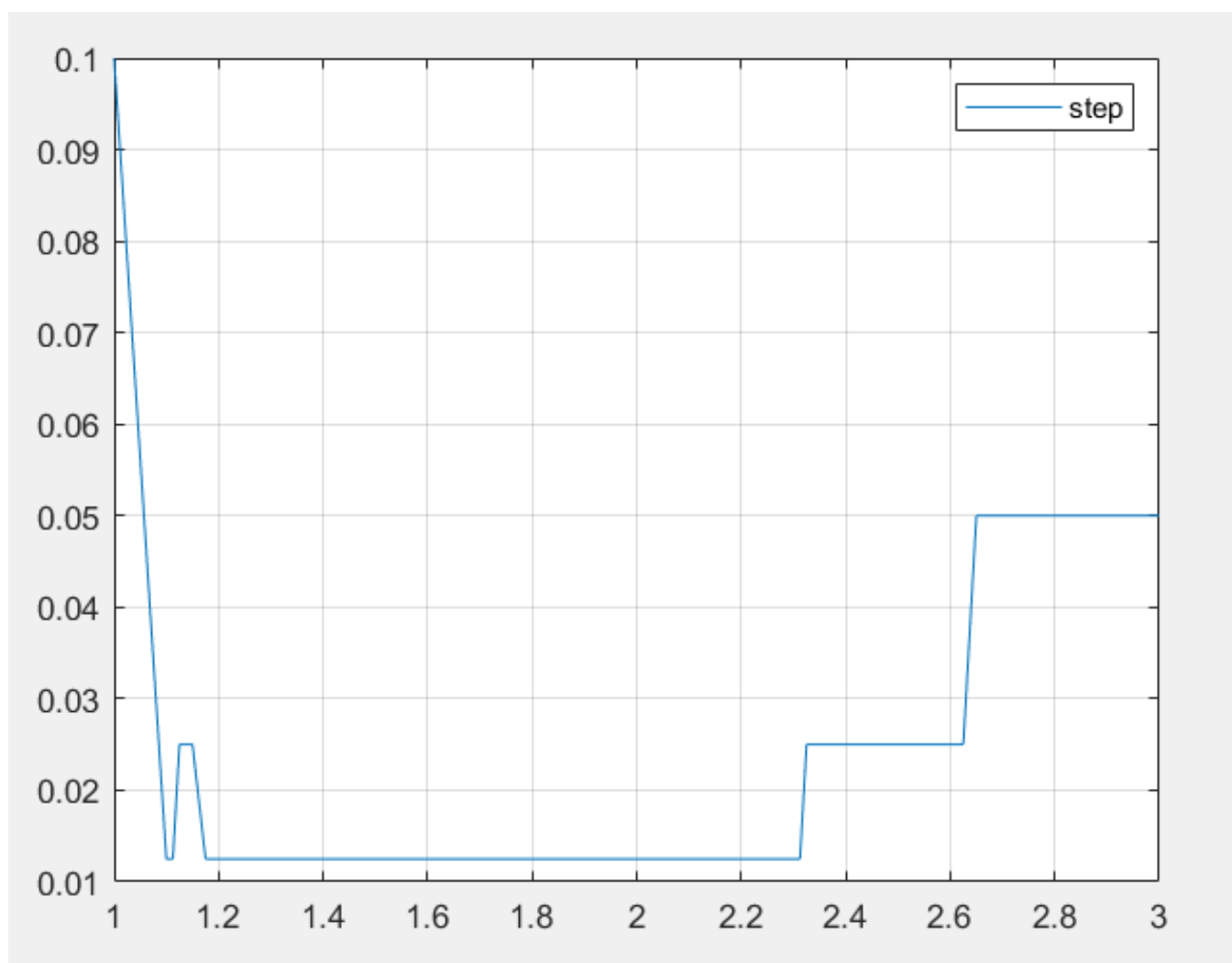


Рис. 5: График величины шага от x при $\epsilon = 10^{-7}$

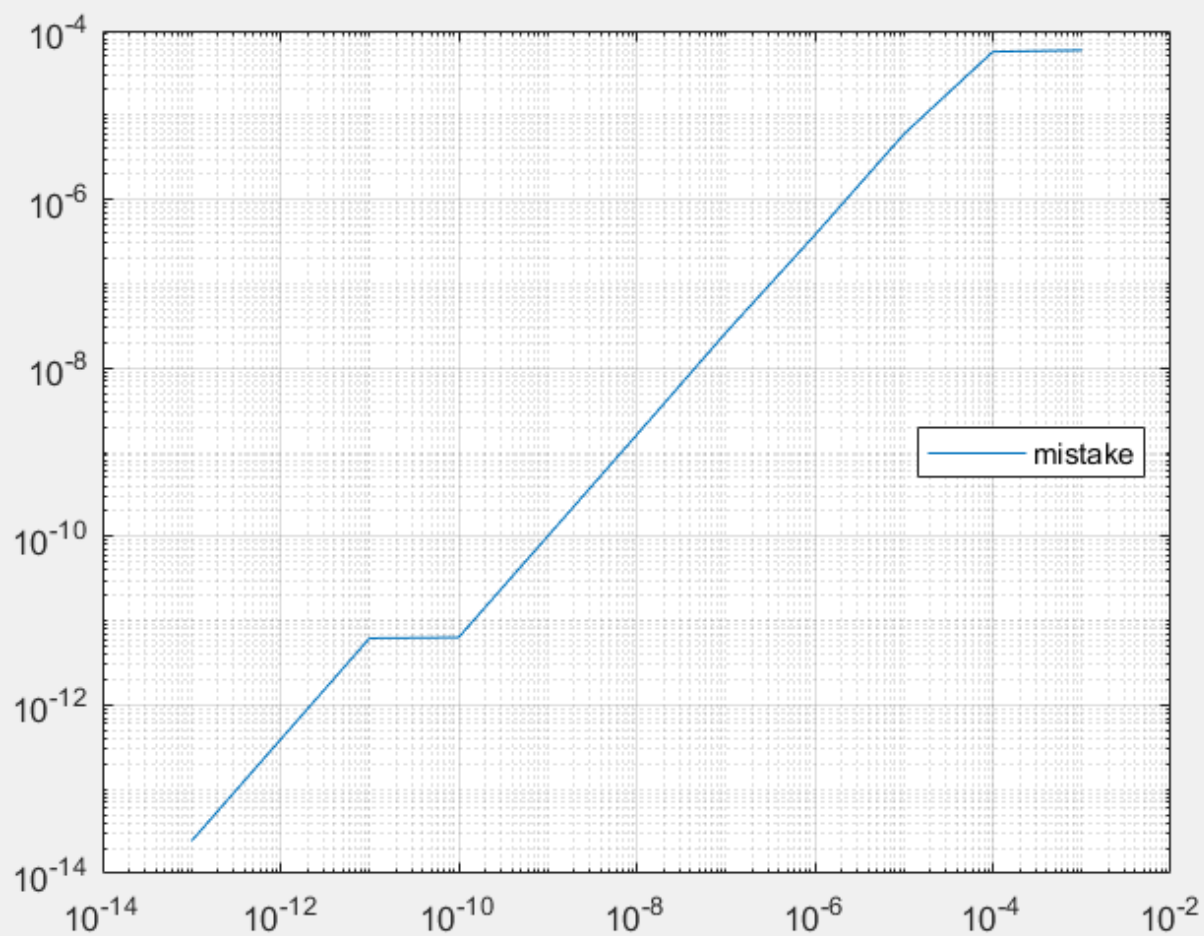


Рис. 6: График ошибки от заданной точности

5.2 Анализ графиков

1. На рисунке 3 видно, что решение с меньшим шагом более близко к точному, чем с большим. Это подтверждается данными с рисунка 4, где видно, что у метода с применением большего шага ошибка больше или равна при любом взятом x .
2. На рисунке 5 видно, что длина шага изменяется под действием заданной точности, причём нет чёткого тренда к понижению/повышению длины шага.
3. На рисунке 6 видно, что метод всегда позволяет добиться заданной точности.

6 Выводы

1. Метод прост в реализации и позволяет добиться решения с заданной точностью.
2. Чем точнее решение мы хотим получить, тем больше итераций надо произвести.