

Санкт – Петербургский политехнический университет Петра Великого
Физико-механический институт

Отчет по лабораторной работе №4 по Численным методам
«Численное решение задачи Коши»
Метод Рунге-Кутты 3 порядка

Выполнил студент

Группы 5030301/10002

Тугай В.В.

Преподаватель:

К.Н. Козлов

Санкт-Петербург

2023

Содержание отчета:

1. Ручной расчет
2. Метод Рунге-Кутты 3 порядка
3. Структура программы
4. Изменение фактической ошибки от шага на отрезке
5. Изменение шага на отрезке
6. Зависимость фактической от заданной точности
7. Изучение солверов MATLAB ode**
8. Реализация и исследование поведения явного метода Эйлера для неустойчивой задачи
9. Реализация и исследование поведения явного и неявного метода Эйлера
10. Исследование влияния шага на решение для системы
11. Вывод

Вариант 21
Метод Рунге-Кутты 3 порядка

Ручной расчет

$$y' = 2x(x^2 + y)$$

$x_0 = 1.0; x_{\text{max}} = 2.0; y(x_0) = e$
 $\varepsilon = 0.1; h = 0.1$
 $K_1 = f(x_n, y_n)h$
 $K_2 = f(x_n + \frac{h}{2}, y_n + \frac{K_1}{2})h$
 $K_3 = f(x_n + h, y_n + K_1)h$
 $\Delta y = K_1 + 4K_2 + K_3$
 $x_{n+1} = x_n + h$
 $y_{n+1} = y_n + \Delta y$
 (если $h > \varepsilon$ или $\Delta y > \varepsilon$, то $h = h \cdot \sqrt{\frac{\varepsilon}{\Delta y}}$)

i=1: $h=0.1; K_1=0.3446; K_2=0.8804; K_3=1.0881; \Delta y=0.8802; x=1.1; y=3.61$
 i=2: $h=0.0344; K_1=0.355; K_2=0.346; K_3=0.4017; \Delta y=0.344; x=1.1347; y=3.98$
 i=3: $K_1=0.447; K_2=0.476; K_3=0.566; \Delta y=0.475; x=1.2; y=4.84$
 i=4: $K_1=0.565; K_2=0.599; K_3=0.641; \Delta y=0.6; x=1.267; y=5.919$
 i=5: $K_1=0.715; K_2=0.757; K_3=0.813; \Delta y=0.761; x=1.334; y=7.2788$
 i=6: $K_1=0.904; K_2=0.965; K_3=1.034; \Delta y=0.967; x=1.401; y=9.004$
 i=7: $K_1=1.156; K_2=1.230; K_3=1.32; \Delta y=1.233; x=1.468; y=11.207$
 i=8: $K_1=1.444; K_2=1.544; K_3=1.693; \Delta y=1.548; x=1.535; y=14.018$
 i=9: $K_1=1.896; K_2=2.023; K_3=2.149; \Delta y=2.028; x=1.602; y=17.624$
 i=10: $K_1=2.445; K_2=2.616; K_3=2.818; \Delta y=2.603; x=1.669; y=22.27$
 i=11: $K_1=3.164; K_2=3.388; K_3=3.663; \Delta y=3.338; x=1.736; y=28.286$
 i=12: $K_1=4.125; K_2=4.418; K_3=4.786; \Delta y=4.43; x=1.8; y=36.113$
 i=13: $K_1=5.4; K_2=5.79; K_3=6.288; \Delta y=5.809; x=1.865; y=46.352$
 i=14: $K_1=7.107; K_2=7.637; K_3=8.3104; \Delta y=7.662; x=1.932; y=59.823$
 i=15: $K_1=9.4156; K_2=10.1308; K_3=11.0487; \Delta y=10.164; x=2; y=77.65$

Метод Рунге–Кутты 3 порядка

Рассмотрим задачу Коши для системы обыкновенных дифференциальных уравнений первого порядка (далее $y, g, k_i \in \mathbb{R}^n$; $x, h \in \mathbb{R}^1$)

$$y' = f(x, y), y(x_0) = y_0$$

Тогда приближенное значение в последующих точках вычисляется по итерационной формуле $y_{n+1} = y_n + \frac{k_1 + 4k_2 + k_3}{6}h$, где:

$$k_1 = f(x_n, y_n)h,$$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right)h,$$

$$k_3 = f(x_n + h, y_n - k_1 + 2k_2)h,$$

где h – величина шага сетки по x .

Структура программы

Программа lab4.f95 состоит из основного тела и модуля:

RK_Runge.mod - модуль, содержащий две функции и подпрограмму. Функция func(x,y) инициализирует исходную функцию задачи Коши. Функция Runge(a,b,eps) является алгоритмом, реализующим правило Рунге. Подпрограмма RK реализует метод Рунге–Кутты 3 порядка.

Программа lab4.m реализует численное решение задачи Коши с помощью средств Matlab. После выполнения в папке будут созданы рисунки, все они представлены далее в отчете.

Изменение фактической ошибки от шага на отрезке

На рисунке 1 изображены графики точного и численного решений для значений шага на отрезке $2.4 \cdot 10^{-5}$ и $1.2 \cdot 10^{-5}$. Как можно увидеть по рисунку, численное решение задачи Коши методом Эйлера с шагом 10^{-2} и 10^{-6} совпадает с точным решением. При излишне малом значении величины шага может происходить численной неустойчивости и колоссальный рост вычислительной сложности. Для предотвращения этих процессов в программе имеется параметр h_{\max} , ограничивающий минимальное значение шага на отрезке ($h_{\max}=\text{eps}$).

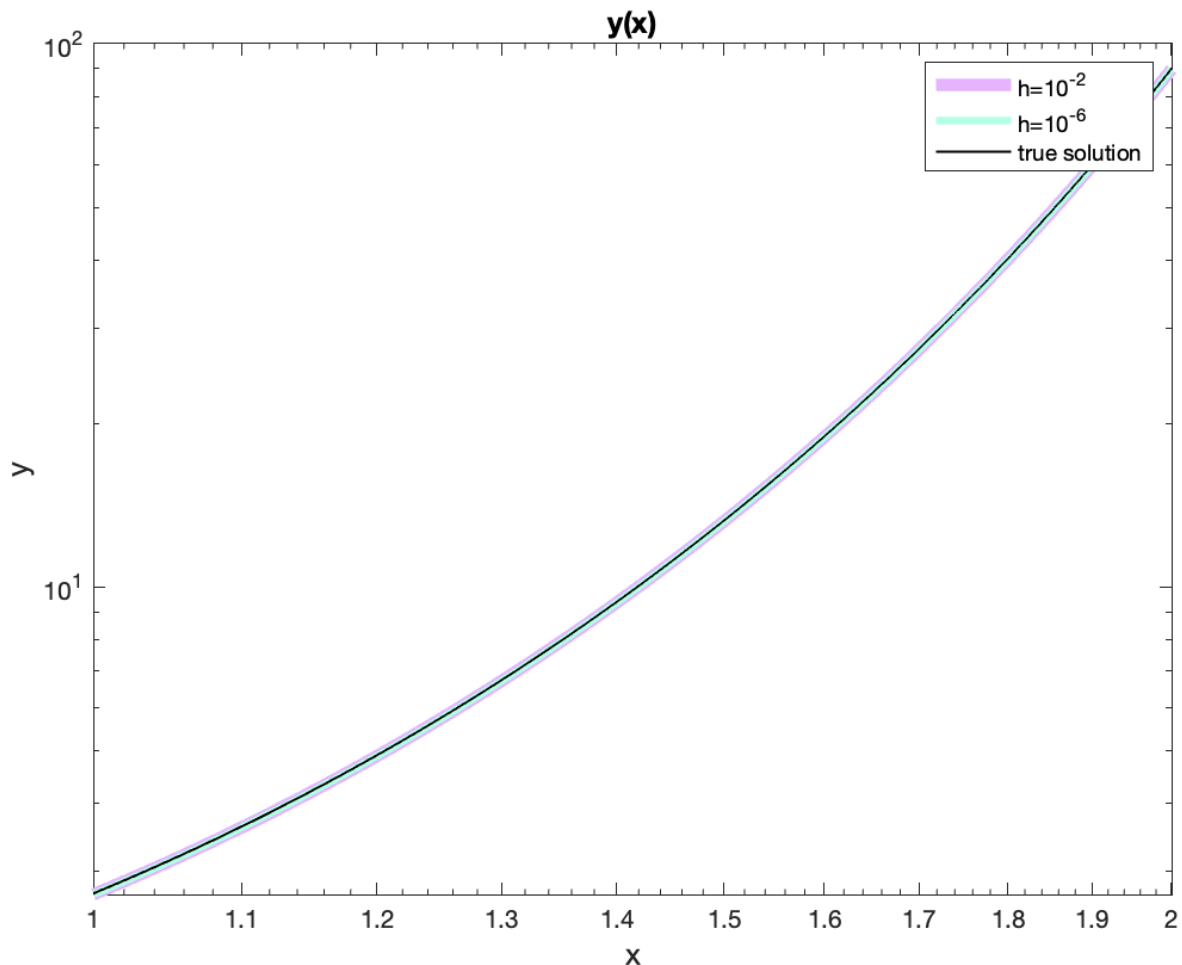


Рисунок 1
График точного и численного решений
для шага на отрезке $h=2.4 \cdot 10^{-5}$

Изменение шага на отрезке

На рисунке 2 изображен график изменения шага на отрезке. Как видно из рисунка, шаг на заданном отрезке резко уменьшается до 10^{-6} , что связано с формулой, по которой считается длина шага $h=h*(\varepsilon/\text{error})^{1/2}$.

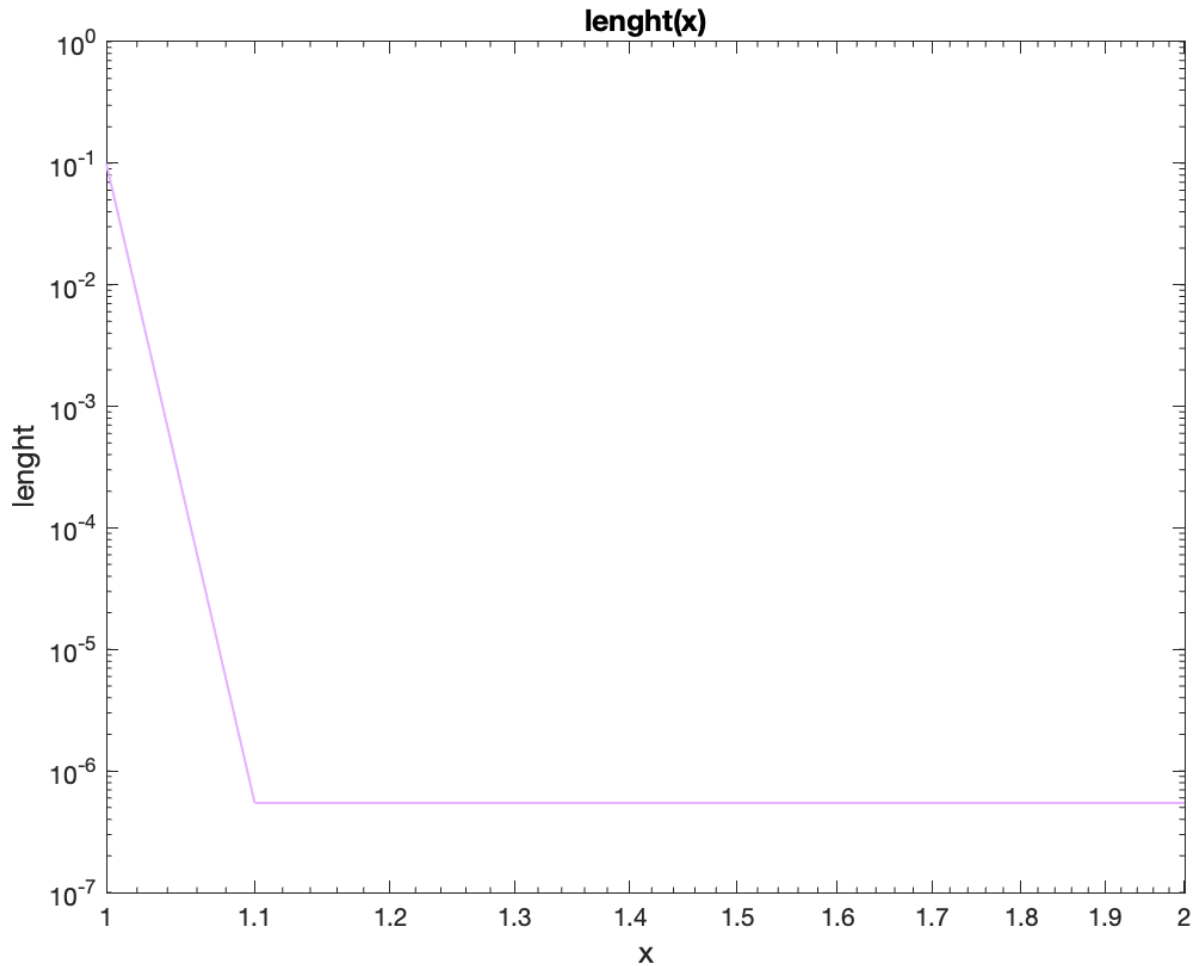


Рисунок 2
График изменения шага на отрезке

Зависимость фактической от заданной точности

На рисунке 3 изображен график зависимости фактической точности от заданной точности. Как можно увидеть по рисунку, метод Рунге-Кутты 3 порядка достигает заданной точности.

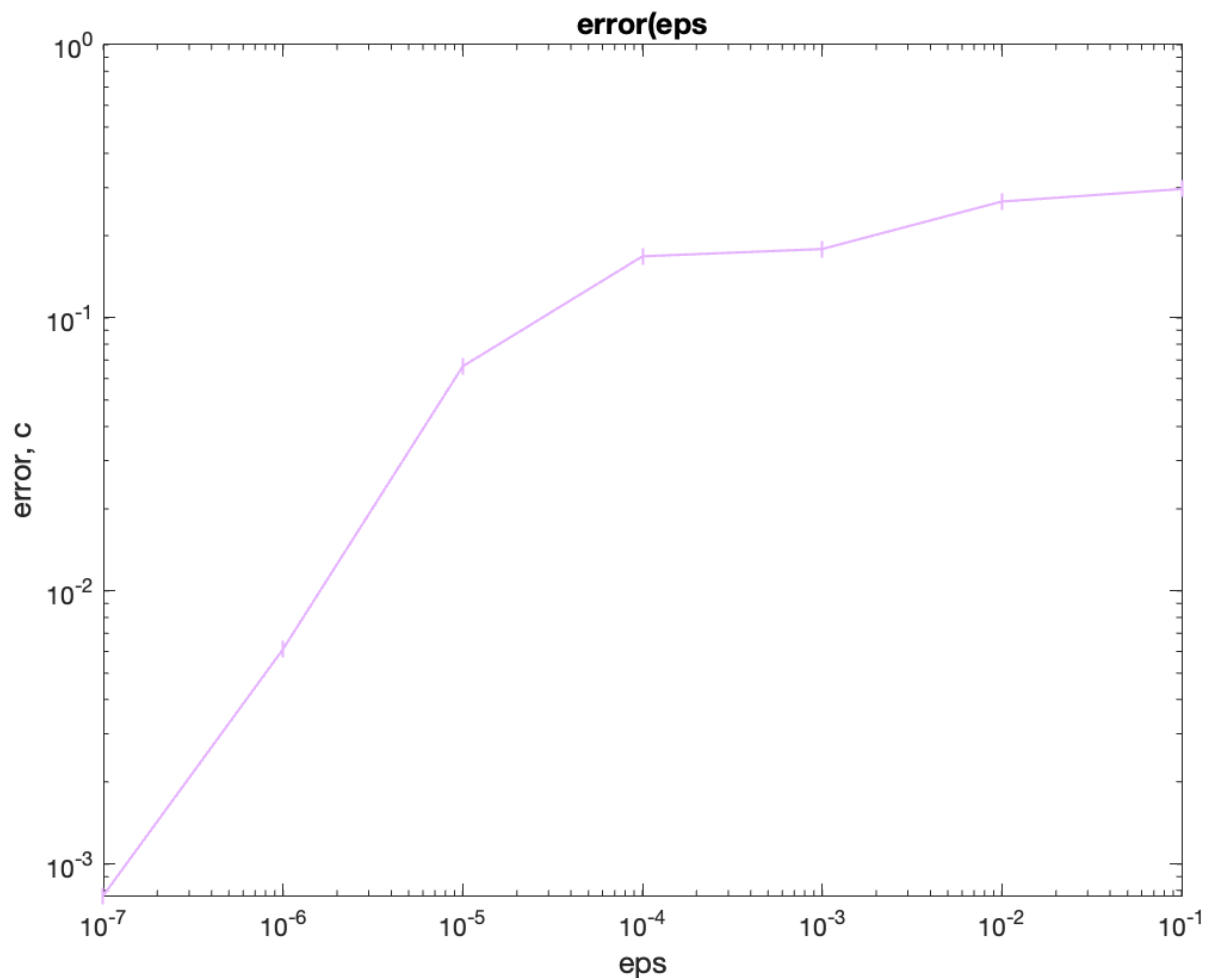


Рисунок 3
Зависимость фактической от заданной точности

Изучение солверов MATLAB ode**

На рисунках 5-11 изображены зависимости фактических ошибок от параметров встроенной функции MATLAB, которые используются для решения уравнительной задачи. Некоторые функции ode требуют явной информации о свойствах якобинана. Обычно это требуется только в том случае, если уравнения являются жесткими (например, функции ode14s и ode22t. Все функции ode по умолчанию адаптируют шаг интегрирования с целью достижения заданной точности решения.

Ode автоматически подстраиваются к шагу интегрирования для достижения требуемой точности. Они используют встроенные алгоритмы для анализа поведения системы и изменения шага интегрирования. Некоторые функций ode не поддерживают обнаружение событий. Обнаружение событий позволяет остановить интегрирование, когда определенное условие или событие происходит в процессе решения дифференциального уравнения (функции ode45, ode23, ode15s, ode23s, ode23t, ode23tb).

В большинстве функций ode в Matlab не требуется явное указание свойств матрицы масс. Однако в некоторых случаях, когда система дифференциальных уравнений имеет особую структуру или требуется дополнительная настройка, можно задать свойства матрицы масс для уточнения процесса интегрирования. Например, функции ode15s и ode23t поддерживают задание свойств матрицы масс.

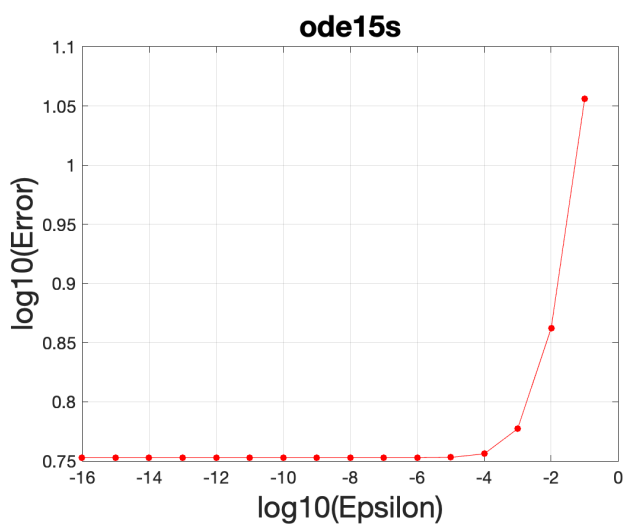


Рисунок 5
Зависимость фактической ошибки от заданной точности для ode15s

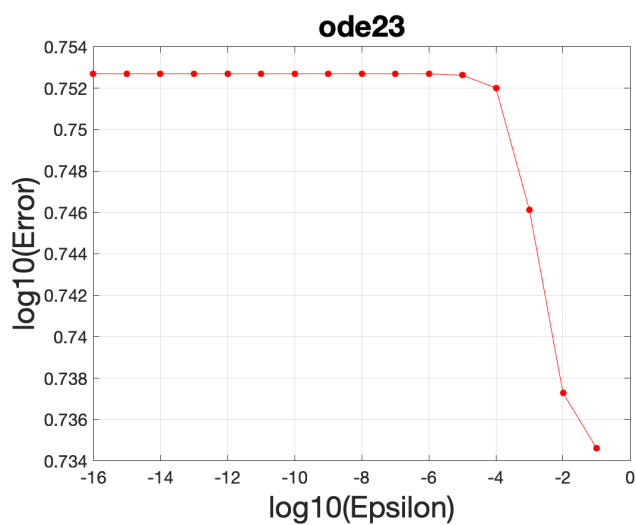


Рисунок 6
Зависимость фактической ошибки от заданной точности для ode23

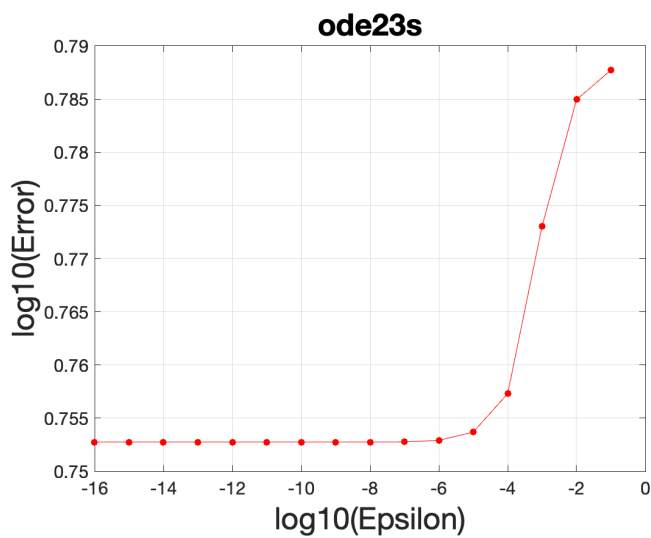


Рисунок 7
Зависимость фактической ошибки от заданной точности для ode23s

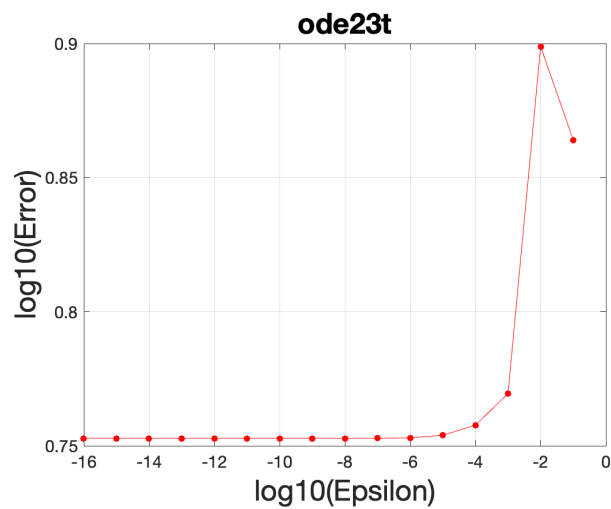


Рисунок 8
Зависимость фактической ошибки от заданной точности для ode23t

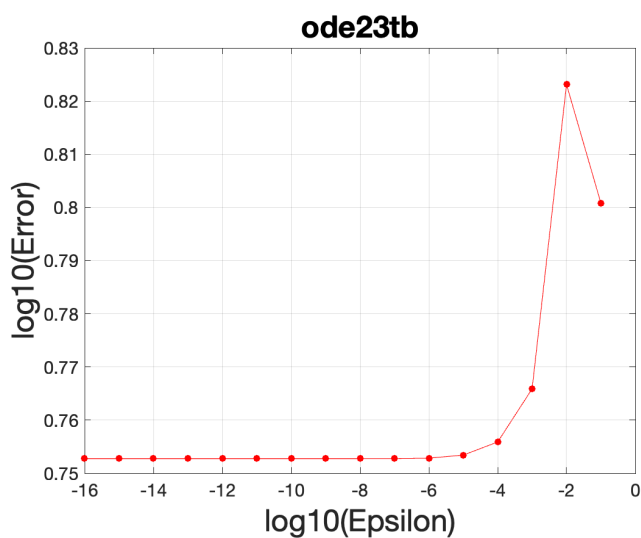


Рисунок 9
Зависимость фактической ошибки от заданной точности для ode23tb

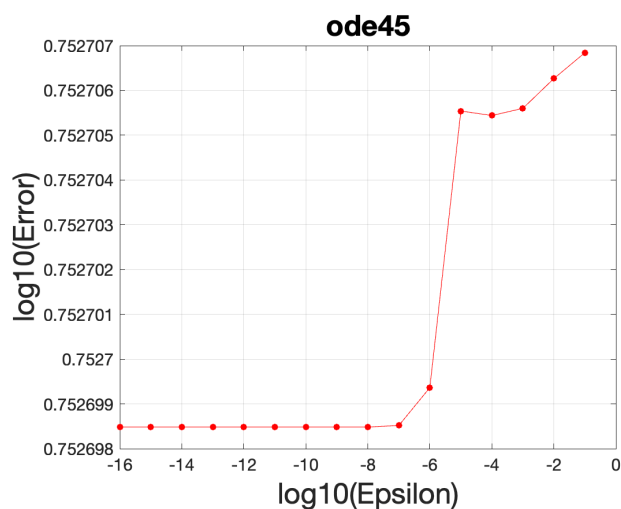


Рисунок 10
Зависимость фактической ошибки от заданной точности для ode45

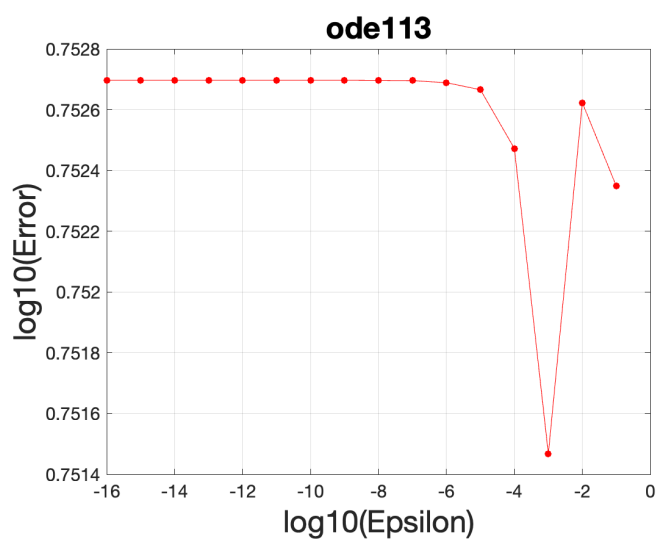


Рисунок 11
Зависимость фактической ошибки от заданной точности для ode113

Реализация и исследование поведения явного метода Эйлера для неустойчивой задачи

На рисунках 12-16 изображены зависимости для явного метода Эйлера. Как видно из рисунков, шаг интегрирования напрямую влияет на точность результата. В случае явного метода Эйлера чем меньше шаг интегрирования, тем точнее может быть получено решение.

Рисунок 12
Поведение неявного метода Эйлера

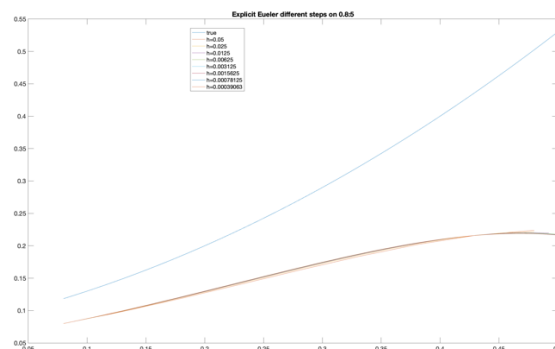


Рисунок 13
Решение явным методом Эйлера для различных
величин шага на отрезке (0.8;5)

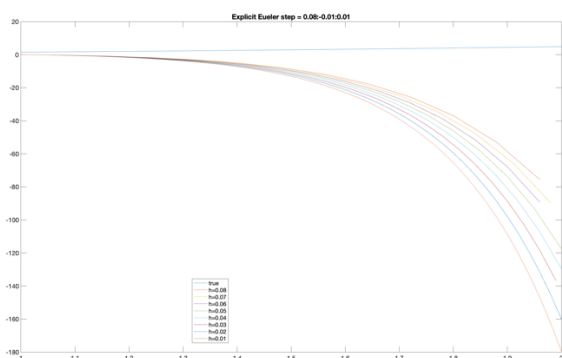


Рисунок 14
Решение неявным методом Эйлера
для разного шага

Рисунок 15
Зависимость погрешности от
шага на отрезке (1;2)

Рисунок 16
Зависимость погрешности от заданного шага на отрезке $(0,08;5)$

Реализация и исследование поведения явного и неявного метода Эйлера

На рисунках 17-21 изображены зависимости для неявного метода Эйлера. Как видно из рисунков, в отличие от явного метода Эйлера, точность неявного метода Эйлера обычно напрямую не зависит от шага интегрирования. Это связано с итерационным процессом, используемым для вычисления значений на следующем шаге. Неявный метод Эйлера может давать более точные результаты даже при больших шагах интегрирования, поскольку учитывает изменение значений на всем интервале шага. Кроме того, неявный метод Эйлера более устойчив, чем явный метод Эйлера.

Рисунок 17
Поведение неявного метода Эйлера

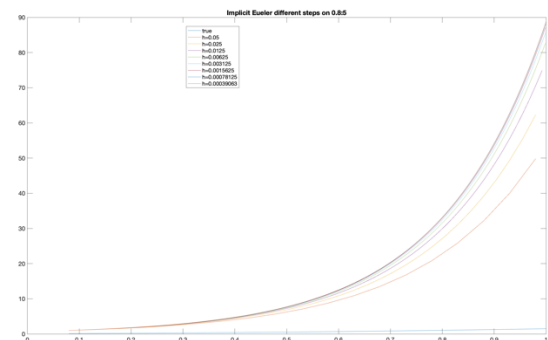


Рисунок 18
Решение явным методом Эйлера для различных
величин шага на отрезке (0.8;5)

Рисунок 19
Решение неявным методом Эйлера
для разного шага

Рисунок 20
Зависимость погрешности от
шага на отрезке (1;2)

Рисунок 21
Зависимость погрешности от заданного шага на отрезке $(0,08;5)$

Исследование влияния шага на решение для системы

На рисунках 22-23 изображены зависимости ошибки от жесткости для явного и неявного методов Эйлера. Как видно из рисунков, при малой жесткости системы явный метод Эйлера может быть достаточно точным, особенно при использовании достаточно малого шага интегрирования. Ошибка метода будет относительно небольшой. Однако при повышенной жесткости системы явный метод Эйлера может давать неудовлетворительные результаты. Неявный метод Эйлера более устойчив по отношению к жестким системам. Он может успешно интегрировать системы с высокой жесткостью, где явный метод Эйлера может дать неудовлетворительные результаты.

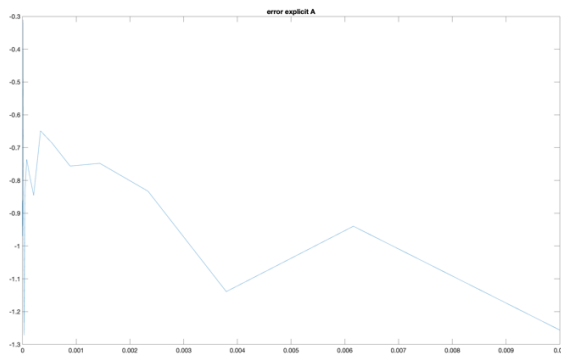


Рисунок 22
Зависимость ошибки от жесткости
системы, явный метод Эйлера

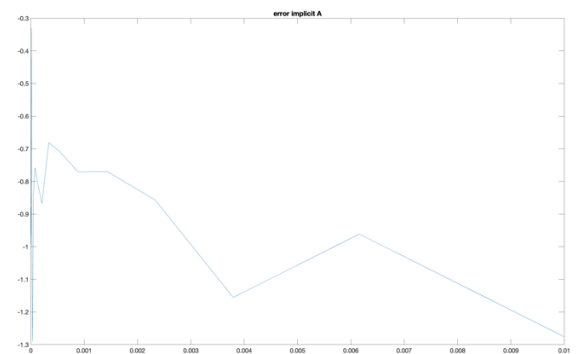


Рисунок 23
Зависимость ошибки от жесткости
системы, неявный метод Эйлера

Вывод

В ходе проделанной лабораторной работы нами был исследован метод Рунге-Кутты 3-го порядка для решения задачи Коши, а также были исследованы встроенные функции ode, явный и неявный метод Эйлера средствами пакета Matlab. Метод и алгоритм были реализованы на языке Fortran, затем по полученным результатам с помощью пакета Matlab были построены графики зависимостей. Исследования же (вместе с построением графиков) были проведены с помощью пакета Matlab.