Санкт-Петербургский Государственный университет ИТМО
Факультет програмной инженерии и компьютерной техники

Лабораторная работа №6 по дисциплине "Вычислительная математика"

Численное решение обыкновенных дифференциальных уравнений

Вариант №7

Работу выполнила:
Д. А. Карасева
Группа: P3217
Преподаватель:
Т. А. Малышева

Санкт-Петербург
2024

Содержание

1. Цель работы

Решить задачу Коши для обыкновенных дифференциальных уравнений численными методами. Выполнить программную реализацию *метода Эйлера, метода усовершенствованного Эйлера, метода Адамса.*

В программе численные методы решения обыкновенных дифференциальных уравнений (ОДУ) должен быть реализован в виде отдельного класса /метода/функции; Пользователь выбирает ОДУ вида (не менее трех уравнений), из тех, которые предлагает программа; Предусмотреть ввод исходных данных с клавиатуры: начальные условия , интервал дифференцирования , шаг h, точность; Для исследования использовать одношаговые методы и многошаговые методы; Составить таблицу приближенных значений интеграла дифференциального уравнения, удовлетворяющего начальным условиям, для всех методов, реализуемых в программе; Для оценки точности одношаговых методов использовать правило Рунге; Для оценки точности многошаговых методов использовать точное решение задачи; Построить графики точного решения и полученного приближенного решения (разными цветами); Программа должна быть протестирована при различных наборах данных, в том числе и некорректных. Проанализировать результаты работы программы.

2. Описание методов. Расчетные формулы

Метод Эйлера

Метод Эйлера является простейшим методом решения задачи Коши и имеет невысокую точность, поэтому на практике его используют достаточно редко. Однако в дальнейшем он послужит основой для более эффективных методов.

В задаче Коши

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

$$y(x_0) = y_0$$

запишем уравнение $y'(x) = f(x, y)$ в узлах $x_i, i = \overline{0, n-1}$, для простоты считаем узлы равноотстоящими, т.е. $\Delta x_i = x_{i+1} - x_i = h = const$. Заменим производную следующим конечно-разностным отношением

$$y'(x_i) \sim \frac{y(x_{i+1} - y(x_i)}{h} \sim f(x_i, y_i)$$

Откуда следует рекуррентная формула метода Эйлера для приближенных значений $y_{i+1} \sim y(x_{i+1})$:

$$y_{i+1} = y_i + hf(x_i, y_i), i = \overline{0, n-1}$$

Усовершенстованный метод Эйлера

Для повышения точности метода Эйлера применяют следующие приемы. Первый улучшенный метод (метод серединных точек) состоит в том, что сначала вычисляют промежуточную (серединную) точку с координатами

$$x_{i+0.5} = x_i + \frac{h}{2}, y_{i+0.5} = y_i + \frac{h}{2} f_i,$$

где $f_i = f(x_i, y_i)$.Затем находят число

$$f_{i+0.5} = f(x_{i+0.5}, y_{i+0.5}),$$

определяющее уточненное направление, и берут $y_{i+1} = y_i + hf_{i+0.5}$

Второй улучшенный метод (метод Эйлера-Коши). Сначала находят приближенное значение решения по методу Эйлера:

$$\overline{y_{i+1}} = y_i + hf_i,$$

исходя из которого определяют направление поля интегральных кривых

$$\overline{f_{i+1}} = f(x_{i+1}, y_{i+1}),$$

а затем уточняют его по формуле

$$y_{i+1} = y_i + h * \frac{f_i + \overline{f_{i+1}}}{2}, i = \overline{0, n-1}$$

Метод Адамса

При использовании интерполяционного многочлена 3-ей степени построенного по значениям подынтегральной функции в последних четырех узлах получим метод Адамса четвертого порядка точности: $y_{k+1} = y_k + \frac{h}{24}(55f_k - 59f_{k-1} + 37f_{k-2} - 9f_{k-3}$. Метод Адамса как и все многошаговые методы не является самостартующим, то есть для того, что бы использовать метод Адамса необходимо иметь решения в первых четырех узлах. В узле x0 решение y0 известно из начальных условий, а в других трех узлах x_1, x_2, x_3решения y_1, y_2, y_3 можно получить с помощью подходящего одношагового метода.

3. Листинг программы

Репозиторий на GitHub

Функции, отвечающие за реализацию методов Эйлера

```
   def euler(self):
      x = np.arange(self.x0, self.xn + self.h, self.h)
      y = np.zeros_like(x)
      y[0] = self.y0
      for i in range(len(x) - 1):
          y[i + 1] = y[i] + self.h * self.f(x[i], y[i])
      return x, y

  def improved_euler(self):
      x = np.arange(self.x0, self.xn + self.h, self.h)
      y = np.zeros_like(x)
      y[0] = self.y0
      for i in range(len(x) - 1):
          y_star = y[i] + self.h * self.f(x[i], y[i])
          y[i + 1] = y[i] + self.h / 2 * (self.f(x[i], y[i]) + self.f(x[i
      return x, y
```

Функция, отвечающая за реализацию метода Адамса

```
      def adams(self, n=10000):
      x = np.arange(self.x0, self.xn + self.h, self.h)
      y = np.zeros_like(x)
      y[0] = self.y0
      x_euler, y_euler = self.improved_euler()
      y[:n] = y_euler[:n]
      for i in range(n - 1, len(x) - 1):
          y[i + 1] = y[i] + self.h / 24 * (
          55 * self.f(x[i], y[i]) - 59 * self.f(x[i - 1], y[i - 1])
          + 37 * self.f(x[i - 2],y[i - 2]) - 9 * self.f(
              x[i - 3], y[i - 3]))
      return x, y
```

4. Примеры и результаты работы программы

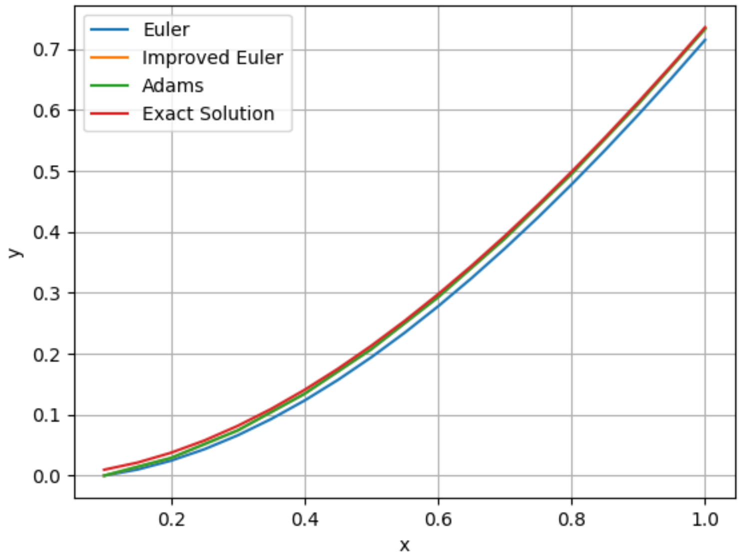

Рисунок 4.1

Выберите ОДУ:
1. y' = -y + 2*x
2. y' = $(y^2 - x^2)/(y^2 + x^2)$
3. y' = y - $x^2 + 1$

Ваш выбор: 1
Введите начальное значение y0: 0
Введите значение x0: 0.1
Введите значение xn: 1
Введите размер шага: 0.2
Введите требуемую точность: 0.01
Метод | x | y

Euler | 0.10 | 0.0000
Euler | 0.30 | 0.0661
Euler | 0.50 | 0.1942
Euler | 0.70 | 0.3726
Euler | 0.90 | 0.5922
R = 0.0085, h = 0.05, h_2 = 0.025

Improved Euler | 0.10 | 0.0000
Improved Euler | 0.30 | 0.0742
Improved Euler | 0.50 | 0.2074
Improved Euler | 0.70 | 0.3889
Improved Euler | 0.90 | 0.6100
R = 0.0060 h = 0.1, h_2 = 0.05

Adams | 0.10 | 0.0000 | 0.00967483607191899

Adams | 0.30 | 0.0742 | 0.08163644136343584
Adams | 0.50 | 0.2074 | 0.21306131942526685
Adams | 0.70 | 0.3889 | 0.3931706075828191
Adams | 0.90 | 0.6100 | 0.6131393194811983
R = 0.0097, $h_c = 0.1, h_2 = 0.05$

5. Вывод

В результате выполнения лабораторной работы я изучила численные методы решения обыкновенных дифференциальных уравнений, смогла программно реализовать методы Эйлера и Адамса на языке Python.