Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 1

Тема Методы Пикара, Эйлера, Рунге-Кутта

Студент Брянская Е.В.

Группа ИУ7-62Б

Оценка (баллы) ________________

Преподаватель Градов В.М.

Москва.
2021 г.

Задание

Тема. Программная реализация приближенного аналитического метода и численных алгоритмов первого и второго порядков точности при решении задачи Коши для ОДУ.

Цель работы. Получение навыков решения задачи Коши для ОДУ методами Пикара и явными методами первого порядка точности (Эйлера) и второго порядка точности (Рунге-Кутта).

Исходные данные.
ОДУ, не имеющее аналитического решения:

$$\begin{cases} u'(x) = x^2 + u^2, \\ u(0) = 0 \end{cases} \tag{1}$$

Результат работы программы. Таблица, содержащая значения аргумента с заданным шагом в интервале $[0,\ x_{max}]$ и результаты расчета функции $u(x)$ в приближениях Пикара (от 1-го до 4-го), а также численными методами. Границу интервала x_{max} выбирать максимально возможной из условия, чтобы численные методы обеспечивали точность вычисления решения уравнения $u(x)$ до второго знака после запятой.

Описание алгоритмов

Задача Коши

Общее решение дифференциального уравнения n-ого порядка зависит от n констант. Требуется задать n дополнительных условий:

$$u(x) = \phi(x, c_1, c_2, ...c_n) \tag{2}$$

В задаче Коши все дополнительные условия задаются в одной точке ξ:

$$u_k(\xi) = \eta_k, k = 1, ...n \tag{3}$$

Задачу Коши можно решить с помощью следующих алгоритмов.

Приближённый аналитический метод Пикара

$$\begin{cases} u'(x) = f(x, u), \\ u(\xi) = \eta \end{cases} \tag{4}$$

$$u(x) = \eta + \int\limits_{\xi}^{x} f(t, u(t))dt \tag{5}$$

Получается, что

$$y^{(s)}(x) = \eta + \int\limits_{\xi}^{x} f(t, y^{(s-1)}(t))dt \tag{6}$$

$$y^{(0)} = \eta \tag{7}$$

Найдём 1, 2, 3 и 4 приближение для (1).

$$y^{(1)} = 0 + \int\limits_{0}^{x} t^2 dt = \frac{t^3}{3}\Big|_0^x = \frac{x^3}{3} \tag{8}$$

$$y^{(2)} = 0 + \int\limits_{0}^{x} [(\frac{t^3}{3})^2 + t^2]dt = \frac{t^7}{63}\Big|_0^x + \frac{t^3}{3}\Big|_0^x = \frac{x^7}{63} + \frac{x^3}{3} \tag{9}$$

$$\begin{gathered} y^{(3)} = 0 + \int\limits_{0}^{x} [(\frac{t^3}{3} + \frac{t^7}{63})^2 + t^2]dt = \frac{t^{15}}{15 \cdot 63^2}\Big|_0^x + \frac{2 \cdot t^{11}}{3 \cdot 63 \cdot 11}\Big|_0^x + \frac{t^7}{63}\Big|_0^x + \frac{t^3}{3}\Big|_0^x = \\ = \frac{x^{15}}{59535} + \frac{2 \cdot x^{11}}{2079} + \frac{x^7}{63} + \frac{x^3}{3} \end{gathered} \tag{10}$$

$$\begin{gathered} y^{(4)} = 0 + \int\limits_{0}^{x} [(\frac{t^{15}}{59535} + \frac{2 \cdot t^{11}}{2079} + \frac{t^7}{63} + \frac{t^3}{3})^2 + t^2]dt = \frac{x^{31}}{109\,876\,902\,975} + \frac{4 \cdot x^{27}}{3\,341\,878\,155} + \\ + \frac{4 \cdot x^{23}}{99\,411\,543} + \frac{2 \cdot x^{23}}{86\,266\,215} + \frac{2 \cdot x^{19}}{3\,393\,495} + \frac{4 \cdot x^{19}}{2\,488\,563} + \frac{4 \cdot x^{15}}{93\,555} + \frac{x^{15}}{59\,535} + \frac{2 \cdot x^{11}}{2079} + \frac{x^7}{63} + \frac{x^3}{3} \end{gathered} \tag{11}$$

Реализация представлена на листинге 1.

Кроме того, поставленную задачу можно решить с помощью численных методов.

Метод Эйлера

Явная схема выглядит следующим образом (12).

$$y_{n+1} = y_n + hf(x_n, y_n) \tag{12}$$

В этом случае нужно выбирать шаг, так как он определяет точность и устойчивость. Реализация представлена на листинге 1.

Метод Рунге-Кутта

Будем рассматривать метод второго порядка точности.

$$y_{n+1} = y_n + h[(1-\alpha)k_1 + \alpha k_2], \tag{13}$$

где

$$\begin{gathered} k_1 = f(x_n, y_n), \\ k_2 = f(x_n + \frac{h}{2\alpha}, y_n + \frac{h}{2\alpha}k_1), \alpha = \frac{1}{2} \text{ или } \alpha = 1 \end{gathered} \tag{14}$$

Реализация представлена на листинге 1.

Результаты работы программы

Результаты представлены в таблице ниже (для наглядности выведен каждый 1800ый результат).

Условные обозначения:

1 - метод Пикара 1е приближение,

2 - метод Пикара 2е приближение,

3 - метод Пикара 3е приближение,

4 - метод Пикара 4е приближение,

5 - метод Эйлера,

6 - метод Рунге-Кутта

x	1	2	3	4	5	6
0.000e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
9.000e-04	2.43e-10	2.43e-10	2.43e-10	2.43e-10	2.43e-10	2.43e-10
1.800e-03	1.94e-09	1.94e-09	1.94e-09	1.94e-09	1.94e-09	1.94e-09
2.700e-03	6.56e-09	6.56e-09	6.56e-09	6.56e-09	6.56e-09	6.56e-09
3.600e-03	1.56e-08	1.56e-08	1.56e-08	1.56e-08	1.55e-08	1.56e-08
4.500e-03	3.04e-08	3.04e-08	3.04e-08	3.04e-08	3.04e-08	3.04e-08
5.400e-03	5.25e-08	5.25e-08	5.25e-08	5.25e-08	5.25e-08	5.25e-08
6.300e-03	8.33e-08	8.33e-08	8.33e-08	8.33e-08	8.33e-08	8.33e-08
7.200e-03	1.24e-07	1.24e-07	1.24e-07	1.24e-07	1.24e-07	1.24e-07
8.100e-03	1.77e-07	1.77e-07	1.77e-07	1.77e-07	1.77e-07	1.77e-07
9.000e-03	2.43e-07	2.43e-07	2.43e-07	2.43e-07	2.43e-07	2.43e-07
9.900e-03	3.23e-07	3.23e-07	3.23e-07	3.23e-07	3.23e-07	3.23e-07
1.080e-02	4.20e-07	4.20e-07	4.20e-07	4.20e-07	4.20e-07	4.20e-07
1.170e-02	5.34e-07	5.34e-07	5.34e-07	5.34e-07	5.34e-07	5.34e-07
1.260e-02	6.67e-07	6.67e-07	6.67e-07	6.67e-07	6.67e-07	6.67e-07
1.350e-02	8.20e-07	8.20e-07	8.20e-07	8.20e-07	8.20e-07	8.20e-07
1.440e-02	9.95e-07	9.95e-07	9.95e-07	9.95e-07	9.95e-07	9.95e-07
1.530e-02	1.19e-06	1.19e-06	1.19e-06	1.19e-06	1.19e-06	1.19e-06
1.620e-02	1.42e-06	1.42e-06	1.42e-06	1.42e-06	1.42e-06	1.42e-06
1.710e-02	1.67e-06	1.67e-06	1.67e-06	1.67e-06	1.67e-06	1.67e-06
...	...	...	...	...	...	...
1.986e+00	2.61e+00	4.55e+00	6.87e+00	9.75e+00	5.93e+01	5.93e+01
1.987e+00	2.62e+00	4.56e+00	6.89e+00	9.80e+00	6.27e+01	6.27e+01
1.988e+00	2.62e+00	4.57e+00	6.92e+00	9.84e+00	6.64e+01	6.64e+01
1.989e+00	2.62e+00	4.58e+00	6.94e+00	9.89e+00	7.07e+01	7.07e+01
1.990e+00	2.63e+00	4.59e+00	6.96e+00	9.94e+00	7.55e+01	7.55e+01
1.991e+00	2.63e+00	4.60e+00	6.98e+00	9.98e+00	8.10e+01	8.10e+01
1.992e+00	2.63e+00	4.61e+00	7.01e+00	1.00e+01	8.73e+01	8.73e+01
1.993e+00	2.64e+00	4.62e+00	7.03e+00	1.01e+01	9.48e+01	9.48e+01
1.994e+00	2.64e+00	4.64e+00	7.07e+00	1.02e+01	1.14e+02	1.14e+02
1.995e+00	2.65e+00	4.65e+00	7.10e+00	1.02e+01	1.27e+02	1.27e+02
1.996e+00	2.65e+00	4.66e+00	7.12e+00	1.03e+01	1.44e+02	1.44e+02
1.997e+00	2.66e+00	4.67e+00	7.14e+00	1.03e+01	1.65e+02	1.65e+02
1.998e+00	2.66e+00	4.68e+00	7.17e+00	1.04e+01	1.94e+02	1.94e+02
1.999e+00	2.66e+00	4.69e+00	7.19e+00	1.04e+01	2.35e+02	2.35e+02
2.000e+00	2.67e+00	4.70e+00	7.21e+00	1.05e+01	2.98e+02	2.99e+02

Вопросы при защите лабораторной работы

1. Укажите интервалы значений аргумента, в которых можно считать решением заданного уравнения каждое из первых 4-х приближений Пикара. Точность результата оценивать до второй цифры после запятой. Объяснить свой ответ.

2. Пояснить, каким образом можно доказать правильность полученного результата при фиксированном значении аргумента в численных методах.

3. Каково значение функции при x=2, т.е. привести значение u(2).

Код программы

Листинг 1 — Лабораторная работа №1

```
from math import sqrt

def f(x, y):
  return x**2 + y**2

def picard_1(x_args):
  res = []
  for x in x_args:
    res.append(x**3 / 3)
  return res

def picard_2(x_args):
  res = []
  for x in x_args:
    res.append(x**3 / 3 + x**7 / 63)
  return res

def picard_3(x_args):
  res = []
  for x in x_args:
    res.append(x**3 / 3 + x**7 / 63 + x**15 / 59535 + 2*x**11 / 2079)
  return res

def picard_4(x_args):
  res = []
  for x in x_args:
    res.append(x**3/3 + x**7/63 + x**15/59535 + 2*x**11/2079 +
    x**31/109876902975 + 4*x**23/99411543 + 4*x**27/3341878155 + 2*x**23/86266215 + 2*x**19/3393495 + 4*x**19/2488563 + 4*x**15/93555)
  return res

def runge_kutta(x, y, h, num):
  alpha = 0.5
  res = []
  temp = h / (2 * alpha)

  for i in range(num):
    res.append(y)
```

```
    k1 = f(x, y)
    k2 = f(x + temp, y + temp * k1)

    y += h * ((1 - alpha) * k1 + alpha * k2)
    x += h

  return res


def euler_explicit(x, y, h, num):
  res = []

  for i in range(num):
    res.append(y)

  try:
    y += h * f(x, y)
    x += h
  except OverFlowError:
    for k in range(i, num):
      res.append('——')
    break
  return res

def count_x_args(x, x_max, h):
  x_args = []
  while x <= x_max:
    x_args.append(x)
    x += h
  return x_args


def print_head():
  print(' '*4+'x'+' '*4+'|'+' '*17+'Метод Пикара'+' '*18+'|'+' '*6+'Метод Эйлера'+' '*5+'|'+' '*3+'Метод РунгеКутта—'+' '*3+'\n'+' '*9+'|'+' '*5+'1'+' '*5+'|'+' '*5+'2'+' '*5+'|'+' '*5+'3'+' '*5+'|'+' '*5+'4'+' '*5+'|'+' '*3+'Явный'+' '*3+'\n'+'-'*55)
```

```
def main():
    print_head()

    x, x_max, y = 0, 2, 0
    h = 10 ** -6 / 2

    num = int((x_max - x) / h)
    x_args = count_x_args(x, x_max, h)

    res_runge_kutta = runge_kutta(x, y, h, num)
    res_euler_explicit = euler_explicit(x, y, h, num)
    res_euler_implicit = euler_implicit(x, y, h, num)
    res_picard_1 = picard_1(x_args)
    res_picard_2 = picard_2(x_args)
    res_picard_3 = picard_3(x_args)
    res_picard_4 = picard_4(x_args)

    for i in range(num):
        if not i % 1000:
            print('{:9.3e}|{:11.3e}|{:11.3e}|{:11.3e}|{:11.3e}|{:11.3e}|{:11.3e}'.format(x_args[i], res_picard_1[i], res_picard_2[i], res_picard_3[i], res_picard_4[i], res_euler_explicit[i], res_runge_kutta[i]))


if __name__ == '__main__':
    main()
```