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 _____метод Пикара____________________

Студент _Мокеев Даниил_____

Группа ____ИУ7-64Б_________

Оценка (баллы) ______________

Преподаватель __Градов В. М.___

Москва.
2020 г.

Оглавление

Введение **2**

1 Аналитическая часть **3**

1.1 Постановка задачи 3

1.2 Метод Пикара 3

1.3 Явный метод Эйлера 3

1.4 Неявный метод Эйлера 4

2 Технологическая часть **5**

2.1 Выбор ЯП 5

2.2 Листинг кода алгоритмов 5

Введение

Задачи данной лабораторной работы:

- Изучить метод Пикара;
- реализовать метод Пикарда, явный и неявный методы Эйлера для решения уравнения $y'(x) = x^2 + y^2$;
- сравнить методы между собой.

1 Аналитическая часть

В данной части будут рассмотрены теоретические основы методов Пикара, явного и неявного методов Эйлера.

1.1 Постановка задачи

Пусть поставлена задача Коши:

$$\begin{cases} u'(x) = f(x, u(x)) \\ u(x_0) = u_0 \\ x_0 \leq x \leq x_l \end{cases}$$

1.2 Метод Пикара

Данный метод является представителем класса приближенных методов решения. Идея метода чрезвычайно проста и сводится к процедуре последовательных приближений для решения интегрального уравнения, к которому приводится исходное дифференциальное уравнение.

Проинтегрируем выписанное уравнение

$$u(x) = u_0 + \int_{x_0}^{x} f(t, u(t))dt \tag{1.1}$$

Процедура последовательных приближений метода Пикара реализуется согласно следующей схеме.

$$y_s(x) = u_0 + \int_{x_0}^{x} f(t, y_{s-1}(t))dt \tag{1.2}$$

причем $y_0(t) = u_0$

1.3 Явный метод Эйлера

Самый простой метод решения уровнения - дискретизация расчетного интервала и замена производной в левой части $\frac{du(x)}{dt}$ разностным аналогом. Для некоторой i-ой точки сетки разностная производная определяется следующим образом:

$$\frac{du(t_i)}{dt} \approx \frac{y_{i+1} - y_i}{h} \tag{1.3}$$

Для того, чтобы схема имела простое решение, правую часть уравнения возьмем в той же точке t_i:

$$\frac{y_{i+1} - y_i}{h} = f(t_i, y_i) \tag{1.4}$$

Таким образом, мы сразу получаем рекуррентную формулу определения нового значения y в точке t_{i+1}, т.е. y_{i+1} по значению y в точке t_{i+1}. Это значение обозначим как y_i, а y_{i+1} запишем как:

$$y_{i+1} = y_i + h \cdot f(x_i, y_i) \tag{1.5}$$

1.4 Неявный метод Эйлера

$$y_{i+1} = y_i + h \cdot f(x_i + 1, y_i + 1) \tag{1.6}$$

Геометрическая интерпретация одного шага этого метода заключается в том, что решение на отрезке $[t_i; t_{i+1}]$ аппроксимируется касательной $y = y_{i+1} + y'(t_{i+1})(t - t_{i+1})$, проведенной в точке (t_{i+1}, y_{i+1}) к интегральной кривой, проходящей через эту точку.

2 Технологическая часть

2.1 Выбор ЯП

В качестве языка программирования был выбран golang. Время работы алгоритмов было замерено с помощью time.

2.2 Листинг кода алгоритмов

В данном разделе будут приведены листинги кода решения методом Пикара (Листинг 2.1) и Эйлера (Листинг 2.2)

Листинг 2.1: Метод Пикара

```

func picard(x float64, n int)float64{
  u0 := 0.0
  answer := 0.0
  poly := make(map[int]float64)
  curr := make(map[int]float64)
  poly[2] = 1.0
  var res float64
  for i:=0;i<n;i++{
    curr = poly_pow(curr)
    curr[2] = 1.0
    curr, res = integrate(curr, 0.0, x)
    answer = u0 + res
  }
  return answer
}
```

Листинг 2.2: Явный и неявный методы Эйлера

```
func euler_explicit(xn float64, n int)float64{
  h := xn / float64(n)
  y:=0.0
  x:=0.0
  for i:=0; i<=n;i++{
    y = y + h*f(x, y)
    x+=h
  }
  return y
```

```
}
func euler_implicit(xn float64, n int)float64{
  h := xn / float64(n)
  y:=0.0
  x:=0.0
  var a, b, c, dis, x1 float64
  for i:=0;i<=n;i++{
    a = 1; b = -1.0/h; c = 1.0/h*y+(x+h)*(x+h)
    dis = D(a, b, c)
    if dis>=0{
      x1 = (-b - math.Sqrt(dis))/2/a
    }
    y = x1
    x+=h
  }
  return y
}
```