



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

Моделирование

Студент: Нечитайло Д.В.

Группа: ИУ7-66Б

Преподаватель: Градов В.М.

Москва, 2021 г.

1 Условия и задачи лабораторной работы

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

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

Задача Коши:

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

Результатом работы программы должна быть таблица, содержащая значение аргумента и значения полученные методом Пикара, явным методом Эйлера и методом Рунге-Кутта.

Вопросы по лабораторной работе

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 $u(2)$.

2 Численные и явные методы

2.1 Метод Пикара

Метод Пикара является представителем приближенных методов решения рассматриваемого класса задач. Идея метода сводится к процедуре последовательных приближений для решения интегрального уравнения, к которому приводится исходное дифференциальное уравнение.

Поставлена задача Коши:

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

Проинтегрируем выписанное уравнение

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

Процедура последовательных приближений метода Пикара реализуется согласно следующей схеме

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

причем $y_0(t) = u_0$, (i – номер итерации).

Заданная в лабораторной работе ОДУ, не имеющее аналитического решения

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

Правая часть непрерывна и удовлетворяет условию Липшица. Значит, решение существует, а метод Пикара сойдется. По схеме Пикара рассчитаем первые четыре приближения для заданного ОДУ.

$$y_1(x) = 0 + \int_0^x t^2 dt = \frac{x^3}{3}$$

$$y_2(x) = 0 + \int_0^x (t^2 + (\frac{t^3}{3})^2) dt = \frac{x^3}{3} + \frac{x^7}{63}$$

$$y_3(x) = 0 + \int_0^x (t^2 + (\frac{t^3}{3} + \frac{t^7}{63})^2) dt = \frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{x^{15}}{59535}$$

$$\begin{aligned} y_4(x) &= 0 + \int_0^x (t^2 + (\frac{t^3}{3} + \frac{t^7}{63} + \frac{2t^{11}}{2079} + \frac{t^{15}}{59535})^2) dt = \\ &= \frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{13x^{15}}{218295} + \frac{82x^{19}}{37328445} + \frac{662x^{23}}{10438212015} + \frac{4x^{27}}{3341878155} + \frac{1}{10} \end{aligned}$$

2.2 Метод Эйлера

Также задача может быть решена с помощью численных методов.

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

$$f(x_n, y_n) = y_n^2 + x_n^2$$

2.3 Метод Рунге-Кутты 2-ого порядка точности

$$y_{n+1} = y_n + h[(1 - \alpha)k_1 + \alpha k_2], \quad (2.1)$$

где

$$\begin{aligned} k_1 &= f(x_n, y_n), \\ k_2 &= f\left(x_n + \frac{h}{2\alpha}, y_n + \frac{h}{2\alpha}k_1\right), \end{aligned}$$

В практике расчетов используют формулу (2.1) при значениях $\alpha = 1$, $\alpha = \frac{1}{2}$. В лабораторной работе приняли $\alpha = \frac{1}{2}$

3 Листинг программы

На листинге 3.1 показан код программы, выводящий аргумент с результатами работы методов, которые необходимо было реализовать по условию лабораторной.

Листинг 3.1: код программы

```
1
2 static class Methods
3 {
4     public static double f(double x, double y)
5     {
6         return (Math.Pow(x, 2) + Math.Pow(y, 2));
7     }
8     public static double EulerOpened(double x, double h, double y)
9     {
10        return y + h * f(x, y);
11    }
12
13    public static double Pikar(double x, int n)
14    {
15
16        switch (n)
17        {
18            case 1:
19                return Math.Pow(x, 3) / 3;
20            case 2:
21                return Math.Pow(x, 3) / 3 + Math.Pow(x, 7) / 63;
22            case 3:
23                return Math.Pow(x, 3) / 3 + Math.Pow(x, 7) / 63 + 2 *
24                    Math.Pow(x, 11) / 2079 + Math.Pow(x, 15) / 59535;
25            case 4:
26                return Math.Pow(x, 3) / 3 + Math.Pow(x, 7) / 63 + 2 *
27                    Math.Pow(x, 11) / 2079 + 13 * Math.Pow(x, 15) /
28                    218295 + 82 * Math.Pow(x, 19) / 37328445 + 662 *
29                    Math.Pow(x, 23) / 10438212015 + 4 * Math.Pow(x, 27)
30                    / 3341878155 + Math.Pow(x, 31) / 109876902975;
```

```

26         default:
27             return Math.Pow(x, 3) / 3;
28     }
29 }
30
31 public static double Runge(double x, double y, double h)
32 {
33     var alpha = 0.5;
34     var k1 = Methods.f(x, y);
35     var k2 = f(x + h / 2 / alpha, y + k1 * h / 2 / alpha);
36     y = y + h * ((1 - alpha) * k1 + alpha * k2);
37     return y;
38 }
39
40 }
41
42 class Program
43 {
44     static void Main()
45     {
46         Console.Write("Enter xMax:");
47         var xMax = double.Parse(Console.ReadLine());
48
49         Console.Write("Enter step:");
50         var h = double.Parse(Console.ReadLine());
51
52         Console.Write("Enter Pikard approximation:");
53         var n = int.Parse(Console.ReadLine());
54
55         double EulerOpened_y = 0;
56         double Pikar_y;
57         double Runge_y = 0;
58
59         double x0 = 0;
60         Console.WriteLine($"X Euler Pikard {n} Runge");
61         while (x0 < xMax + h)
62         {
63             EulerOpened_y = Methods.EulerOpened(x0, h, EulerOpened_y);
64             Runge_y = Methods.Runge(x0, Runge_y, h);
65             Pikar_y = Methods.Pikar(x0, n);
66             Console.WriteLine("{0,6:F8}|{1,6:F8}|{2,6:F8}|{3,6:F8}|",
67                               x0, EulerOpened_y, Pikar_y, Runge_y);
68             x0 += h;
69         }
70     }
71 }

```


На рисунках 3.1 и 3.2 показан вывод программы со значениями аргументов и результатом работы методов с этими аргументами.

Enter xMax:2			
Enter step:0,01			
Enter Pikard approximation:3			
X	Euler	Pikard3	Runge
0,00000000	0,00000000	0,00000000	0,00000050
0,01000000	0,00000100	0,00000033	0,00000300
0,02000000	0,00000500	0,00000267	0,00000950
0,03000000	0,00001400	0,00000900	0,00002200
0,04000000	0,00003000	0,00002133	0,00004250
0,05000000	0,00005500	0,00004167	0,00007300
0,06000000	0,00009100	0,00007200	0,00011550
0,07000000	0,00014000	0,00011433	0,00017200
0,08000000	0,00020400	0,00017067	0,00024450
0,09000000	0,00028500	0,00024300	0,00033500
0,10000000	0,00038500	0,00033333	0,00044550
0,11000000	0,00050600	0,00044367	0,00057801
0,12000000	0,00065001	0,00057601	0,00073451
0,13000000	0,00081901	0,00073234	0,00091702
0,14000000	0,00101502	0,00091468	0,00112753
0,15000000	0,00124003	0,00112503	0,00136804
0,16000000	0,00149604	0,00136538	0,00164057
0,17000000	0,00178506	0,00163773	0,00194710
0,18000000	0,00210910	0,00194410	0,00228964
0,19000000	0,00247014	0,00228648	0,00267020
0,20000000	0,00287020	0,00266687	0,00309079
0,21000000	0,00331128	0,00308729	0,00355340
0,22000000	0,00379539	0,00354973	0,00406004
0,23000000	0,00432454	0,00405621	0,00461273
0,24000000	0,00490073	0,00460873	0,00521347
0,25000000	0,00552597	0,00520930	0,00586428
0,26000000	0,00620227	0,00585994	0,00656716
0,27000000	0,00693166	0,00656266	0,00732414
0,28000000	0,00771614	0,00731948	0,00813724
0,29000000	0,00855773	0,00813241	0,00900847
0,30000000	0,00945846	0,00900347	0,00993987
0,31000000	0,01042036	0,00993470	0,01093346
0,32000000	0,01144544	0,01092812	0,01199127
0,33000000	0,01253575	0,01198577	0,01311535
0,34000000	0,01369333	0,01310967	0,01430773
0,35000000	0,01492020	0,01430188	0,01557046
0,36000000	0,01621843	0,01556445	0,01690559

Рисунок 3.1: вывод программы

1,63000000	2,29219193	2,05823145	2,37913362
1,64000000	2,37162937	2,11587317	2,46481830
1,65000000	2,45510063	2,17530713	2,55516930
1,66000000	2,54293182	2,23660328	2,65059614
1,67000000	2,63548584	2,29983483	2,75155885
1,68000000	2,73316769	2,36507835	2,85857595
1,69000000	2,83643075	2,43241397	2,97223413
1,70000000	2,94578415	2,50192555	3,09319981
1,71000000	3,06180159	2,57370086	3,22223321
1,72000000	3,18513188	2,64783177	3,36020541
1,73000000	3,31651153	2,72441448	3,50811941
1,74000000	3,45678002	2,80354969	3,66713603
1,75000000	3,60689830	2,88534285	3,83860627
1,76000000	3,76797145	2,96990439	4,02411195
1,77000000	3,94127654	3,05734998	4,22551712
1,78000000	4,12829715	3,14780075	4,44503403
1,79000000	4,33076652	3,24138359	4,68530832
1,80000000	4,55072190	3,33823145	4,94953057
1,81000000	4,79057360	3,43848359	5,24158410
1,82000000	5,05319356	3,54228593	5,56624360
1,83000000	5,34203021	3,64979136	5,92944613
1,84000000	5,66125908	3,76116011	6,33866718
1,85000000	6,01598262	3,87656007	6,80345257
1,86000000	6,41249909	3,99616719	7,33618695
1,87000000	6,85866953	4,12016588	7,95323091
1,88000000	7,36442701	4,24874941	8,67665083
1,89000000	7,94249586	4,38212036	9,53693411
1,90000000	8,60942827	4,52049106	10,57741104
1,91000000	9,38713182	4,66408408	11,86177388
1,92000000	10,30517826	4,81313273	13,48754101
1,93000000	11,40439425	4,96788155	15,61173290
1,94000000	12,74263233	5,12858691	18,50382551
1,95000000	14,40440412	5,29551756	22,66647820
1,96000000	16,51768870	5,46895521	29,14998500
1,97000000	19,28483810	5,64919520	40,53877647
1,98000000	23,04309191	5,83654712	65,04692721
1,99000000	28,39253375	6,03133551	143,91341617
2,00000000	36,49389348	6,23390061	863,73908929

Рисунок 3.2: вывод программы

4 Ответы на вопросы

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.

Приближение можно считать решением пока его значение совпадает со значением следующего приближения (мы рассматриваем до второй запятой), когда их значения начнут расходиться, нужно рассматривать следующее приближение.

$$|y_n(x) - y_{n+1}(x)| < \epsilon,$$

Результат:

- Для Пикара 1 порядка: $(0, 0,937)$
- Для Пикара 2 порядка: $(0, 1,233)$
- Для Пикара 3 порядка: $(0, 1,419)$

Листинг 4.1: вычисление крайнего значения аргумента

```
1      double h = 0.001;
2      double eps = 0.01;
3      double x0 = 0;
4      int n = 3;
5
6      while (Math.Abs(Methods.Pikar(x0, n + 1) - Methods.Pikar(x0, n)) < eps)
7          {
8              x0 += h;
9          }
10     Console.WriteLine("{0,6:F8}", x0);
```

На листинге 4.1 код программы, результат которого является крайнее правое значение аргумента для метода Пикара порядка точности n

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

При фиксированном значении аргумента надо менять шаг и наблюдать, как ведёт себя решение. Если наступает такой момент с уменьшением шага, когда функция перестаёт меняться, это и есть истинное решение. Ибо сходимость определяется при фиксированном значении аргумента, как разность между точным значением и приближённым. По мере того, как шаг уменьшается, точность возрастает.

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

По схеме ответа на вопрос №2 рассчитаем значение $u(2)$

Листинг 4.2: вычисление значение функции при $u(2)$

```

1  double x0 = 0;
2  double y = 0;
3  double h = 0.1;
4  double xMax = 2;
5  int i = 0;
6  Console.WriteLine("|        h        |        x        |");
7  while (i < 8)
8  {
9      y = 0;
10     x0 = 0;
11     while (x0 < xMax)
12     {
13         y = Methods.EulerOpened(x0, h, y);
14         x0 += h;
15     }
16
17     Console.WriteLine("|{0,6:F8}|{1,6:F8}|", h, y);
18     h *= 0.1;
19     i++;
20 }

```

На листинге 4.2 показан код, который находит значение $u(2)$, а на рисунке 4.1, показан вывод программы.

h	x
0,10000000	5,85209961
0,01000000	28,39253375
0,00100000	142,62724970
0,00010000	277,36250013
0,00001000	312,06127640
0,00000100	317,24593495
0,00000010	317,66462963
0,00000001	317,71774176

Рисунок 4.1: вывод программы

Результат: $u(2) = 317,717$