

Отчёт по лабораторной работе №1
По дисциплине «Математическое моделирование»
Тема: «Приближенный аналитический метод Пикара в
сравнении с численными методами»

Студент: _____ Барсуков Н.М.

Группа: _____ ИУ7-66Б

Преподаватель: _____ Градов В.М

Москва, 2020

Содержание

1. Введение	3
2. Аналитическая часть	3
2.1. Задача Коши	3
2.2. Методы	3
2.2.1. Метод Пикара	4
2.3. Метод ломанных (явная схема)	5
2.4. Неявная схема	5
2.5. Метод Рунге-Кутты	6
3. Листинг	6
4. Результат работы	9
5. Ответы на вопросы	11

1. Введение

Класс дифференциальных уравнение, которые интегрируются в явном виде очень мал и найти общее решение в квадратурах удастся лишь для узкого класса функций. В связи с этим возникает потребность в приближенных методах интегрирования.

2. Аналитическая часть

Целью работы является получение навыков решения задачи Коши для ОДУ методами Пикара и явными и неявными методами Эйлера.

Исходные данные:

- 1) ОДУ, не имеющее аналитического решения:

$$u'(x) = u^2 + x^2;$$

$$u(0) = 0$$

Для выполнения поставленной цели необходимо выполнить следующие задачи:

- 1) Поставить задачу Коши для ОДУ
- 2) Понять какие методы существуют
- 3) Рассмотреть конкретные методы для решения поставленной задачи Коши:
 - а) Метод Пикара (Приближения 1 - 4)
 - б) Метод ломанных (Явный)
 - в) Метод Рунге-Кутта 2 порядка
 - г) Метод Рунге-Кутта 4 порядка

2.1. Задача Коши

Задача Коши - одна из основных задачи теории дифференциальных уравнений (обыкновенных и с частными производными); состоит в нахождении решения (интеграла) дифференциального уравнения, удовлетворяющего так называемым начальным условиям

2.2. Методы

Методы решения ДУ можно разбить на точные, приближенные и численные.

К точным относятся методы, позволяющие выразить решение дифференциального уравнения, через элементарные функции. Либо представить его в при помощи квадратур от элементарных функций. Однако классы, для которых разработаны

методы получения точных решений, сравнительно узки и охватывают только малую часть возникающих на практике задач. **В данной лабораторной работе рассматриваться не будут**

Приближенными будем называть методы, в которых решение получается как предел $u(x)$ некоторой последовательности $y_n(x)$, причем $y_n(x)$ выражается через элементарные функции или при помощи квадратур. Ограничиваясь конечным числом n , получаем приближенное выражение для $u(x)$. Эти методы удобны лишь в том случае, когда большую часть промежуточных выкладок удастся сделать точно. Примером может служить метод Пикара. Это выполнимо лишь для сравнительно простых задач, что сильно сужает область применения приближенных методов.

Численные методы - это алгоритмы вычисления приближенных (а иногда - точных) значений искомого решения $u(x)$ на некоторой выбранной сетке значений аргумента x_n . Численные методы не позволяют найти общего решения системы; они могут дать только какое-то частное решение. Зато данные методы применимы к очень широкому классу уравнений и всем типам задач для них.

Рассмотрим детально представителей данных методологий

2.2.1. Метод Пикара

Метод Пикара - это приближенный метод решения, являющийся обобщением метода последовательных приближений.

Рассмотрим пример из лекций:

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

Тогда:

$$\begin{aligned} y^{(1)}(x) &= 0 + \int_0^x t^2 dt = \frac{x^3}{3} \\ y^{(2)}(x) &= 0 + \int_0^x [t^2 + (\frac{t^3}{3})^2] dt = \frac{t^3}{3} + \frac{t^7}{63} \Big|_0^x = \frac{x^3}{3} + \frac{x^7}{63} \\ y^{(3)}(x) &= 0 + \int_0^x [t^2 + (\frac{x^3}{3} + \frac{x^7}{63})^2] dt = \frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{x^{15}}{59535} \\ y^{(4)}(x) &= 0 + \int_0^x [t^2 + (\frac{x^3}{3} + \frac{x^7}{63})^2] dt = \frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{x^{15}}{59535} \\ &+ \int_0^x [t^2 + (\frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{x^{15}}{59535})^2] dt = \frac{x^3}{3} + \frac{x^7}{63} + \frac{2x^{11}}{2079} + \frac{x^{15}}{59535} + \frac{2x^{15}}{3393495} + \frac{2x^{19}}{2488563} + \frac{2x^{23}}{86266215} + \frac{x^{23}}{99411543} + \frac{2x^{27}}{3341878155} + \\ &+ \frac{x^{31}}{109876902975} \end{aligned}$$

2.3. Метод ломанных (явная схема)

Это простейший численный метод. В практике вычислений он употребляется очень редко из-за невысокой точности. Но на его примере удобно пояснить способы построения и исследования численных методов.

Рассмотрим задачу Коши и выберем на отрезке $[\Xi, X]$ некоторую сетку $X_n, 0 \leq n \leq N$ значений аргумента так, чтобы выполнялись соотношения $\xi = x_0 < x_1 < x_2 < \dots < x_N$ (сетка может быть неравномерной). Разлагая решение $u(x)$ по формуле Тейлора на интервале сетки $x_n < x < x_{n+1}$ и обозначая $u(x_n) = u_n$ получим:

$$u_{n+1} = u_n + h_n u'_n + \frac{1}{2} h_n^2 u''_n + \dots, h_n = x_{n+1} - x_n. \quad (1)$$

Стоящие в правой части производные можно найти, дифференцируя уравнения:

$$u'(x) = f(x, u(x)), f'_x \leq x \leq X, f'_u(\xi) = \eta \quad (2)$$

требуемое число раз:

$$u' = f(x, u), u'' = \frac{d}{dx} f(x, u) = f_x + f f_u \quad (3)$$

и т.д.

Однако использовать для расчета формулу 1 с большим числом членов невыгодно. Во-первых, даже при сравнительно простой правой части выражения для производных могут оказаться громоздкими. Во-вторых, если правая часть известна лишь приближенно, то находить ее производные нежелательно. В простейшем случае, подставляя 3 в 1 и ограничиваясь только первым членом разложения, получим схему ломанных

$$y_{n+1} = y_n + h_n f(x_n, y_n), h_n = x_{n+1} - x_n \quad (4)$$

Поскольку при такой замене можно найти только приближенные значения искомой функции в узлах, то будем обозначать эти значения через y_n в отличие от точных значений $u_n = u(x_n)$. Для численного расчета по схеме ломанных достаточно задать начальное значение $y_0 = \eta$. Затем по формуле 4 последовательно вычисляем величины $y_1, y_2, y_3, \dots, y_N$

2.4. Неявная схема

$$y_{n+1} = y_n + h[f(x_{n+1}; y_{n+1})] \quad (5)$$

Решая это алгебраическое уравнение, можно определить y_{n+1} которое и будет приближенным значением искомого решения $u(x_n)$. Схема 5 имеет второй порядок точности, допускает счет неравномерным шагом, не требует специальных приемов для начала счета. Но у этой схемы есть серьезные недостатки. Во-первых, неизвестно, имеет ли уравнение 5 вещественный корень, т. е. разрешима ли задача.

Можно привести пример, когда при большом шаге корня нет. Пусть $f(x, u) = u^2$ и $u(0) = 1$; тогда на первом шаге

$$y_1 = 1 + \frac{1}{2} h(1 + y_1^2) \text{ и при } h > \sqrt{1 + \sqrt{2}} \text{ вещественного корня нет}$$

Во-вторых, даже если корень есть, то как его найти? Метод Ньютона применять нежелательно, так как для этого надо дифференцировать $f(x, u)$. Метод деления пополам не обобщается на системы уравнений. Остается метод последовательных приближений.

2.5. Метод Рунге-Кутта

Можно строить схемы различного порядка точности. Например, схема ломаных 3 есть схема Рунге—Кутта первого порядка точности. Наиболее употребительны схемы четвертого порядка точности, образующие семейство четырехчленных схем. Приведем без вывода ту из них, которая записана в большинстве стандартных программ ЭВМ:

$$\begin{aligned} y_{n+1} &= y_n + (k_1 + 4k_2 + k_3)/6 \\ k_1 &= h_n \phi(x_n, y_n), \\ k_2 &= h_n \phi(x_n + \frac{h_n}{2}, y_n + \frac{k_1}{2}), k_3 = h_n \phi(x_n + h_n, y_n - k_1 + 2k_2) \end{aligned}$$

Формулы более высокого порядка точности практически не употребляются. Пятичленные формулы имеют всего лишь четвертый порядок точности; шестичленные имеют шестой порядок, но слишком громоздки. Кроме того, высокий порядок реализуется лишь при наличии у правой части непрерывных производных соответствующего порядка.

3. Листинг

```

1  from polynomial import Polynomial
2  from prettytable import PrettyTable
3  import numpy as np
4
5
6  def test_func(x, u):
7      return x ** 2 + u ** 2
8
9
10 def picard(ns, x, func, pol=False):
11     """ Cauchy problem:
```

```

12     u'(x) = func(x, u) = x^2 + u^2
13     u(0) = 0
14
15     y(0) = u(0) """
16
17     y = Polynomial(0, [0.0])
18     t = Polynomial(1, [1.0])
19
20     us = []
21
22     for i in range(max(ns)):
23         under_int = func(t, y)
24         y = under_int.integral_variable_up(0.0)
25
26         if i+1 in ns:
27             if pol:
28                 print("Полином {0} итерации: ".format(i+1), y)
29                 us.append(round(y.get(x), 7))
30     return us
31
32
33 def broken_evident(x, func):
34     y = 0
35     t = 0
36
37     while t <= x:
38         f = func(t, y) * h
39         y += f
40         t += h
41
42     return y
43
44
45 def runge_kutta_second(x, alpha, h, func):
46
47     # assert(alpha == 0.5 or alpha == 1, "Alpha должна быть равно 1 или
48     0.5")
49
50     y = 0
51     t = 0
52
53     while t <= x:
54         f = h * ((1 - alpha) * func(t, y) +
55                 alpha * func(t + h / (2 * alpha),

```

```

55         y + (h / (2 * alpha)) * func(t, y)))
56
57     y += f
58     t += h
59
60     return y
61
62
63 def runge_kutta_fourth(x, h, func):
64     y = 0
65     t = 0
66
67     while t <= x:
68         k1 = func(t, y)
69         k2 = func(t + h / 2, y + h / 2 * k1)
70         k3 = func(t + h / 2, y + h / 2 * k2)
71         k4 = func(t + h, y + h * k3)
72
73         f = h / 6 * (k1 + 2 * k2 + 2 * k3 + k4)
74
75         y += f
76         t += h
77
78     return y
79
80
81 def calc(ns, a, b, h, alpha, func):
82     table_picard = PrettyTable()
83     title = ["X"]
84     for it in ns:
85         title.append("Пикар: я{0}- итерация".format(it))
86         title.append("Метод ломанной явный()")
87         title.append("Метод РунгеКутты- ого2- порядка неявный()")
88         title.append("Метод РунгеКутты- ого4- порядка")
89
90     table_picard.field_names = title
91
92     xs = np.arange(a, b, h)
93
94     for x in xs:
95         u = picard(ns, x, func)
96         u.insert(0, round(x, 3))
97         bevi = broken_evident(x, func)
98         u.append(bevi)

```



```

99     rks = runge_kutta_second(x, alpha, h, func)
100    u.append(rks)
101    rkf = runge_kutta_fourth(x, h, func)
102    u.append(rkf)
103    table_picard.add_row(u)
104
105    u = picard(ns, b, func, pol=True)
106    u.insert(0, round(b, 3))
107    bevi = broken_evident(b, func)
108    u.append(bevi)
109    rk = runge_kutta_second(b, alpha, h, func)
110    u.append(rk)
111    rkf = runge_kutta_fourth(b, h, func)
112    u.append(rkf)
113    table_picard.add_row(u)
114
115    print(table_picard)
116
117
118    if __name__ == "__main__":
119        x1 = int(input("Введите X: от "))
120        x2 = int(input(" " * 11 + "до "))
121        h = float(input(" " * 6 + "с шагом "))
122        nl = input("Введите через пробел интересующие итерации для метода
Пикара: ").split(" ")
123        a = float(input("Введите  $\alpha$  для метода РунгеКутты- второго порядка: "
))
124        nl = list(map(lambda x: int(x), nl))
125        calc(nl, x1, x2, h, a, test_func)
126

```

4. Результат работы

В данном разделе представлен результат работы программы на рисунках 1 2

X	Пикар: 1-я итерация	Пикар: 2-я итерация	Пикар: 3-я итерация	Пикар: 4-я итерация	Метод Локаньяи (анал.)	Метод Рунге-Кутты 2-ого порядка (неанал.)	Метод Рунге-Кутты 4-ого порядка
0.0	0.0	0.0	0.0	0.0	0.0	2.5000000000000004e-07	3.333333334595834e-07
0.01	3e-07	3e-07	3e-07	3e-07	1.0000000000000002e-06	2.5000000000000003e-06	2.466666667534722e-06
0.02	2.7e-06	2.7e-06	2.7e-06	2.7e-06	5.000000010000001e-06	8.750000208125005e-06	8.000000349635436e-06
0.03	9e-06	9e-06	9e-06	9e-06	1.4000000260000002e-05	2.1000001963750162e-05	2.133333594602541e-05
0.04	2.13e-05	2.13e-05	2.13e-05	2.13e-05	3.000000220000007e-05	4.1666601037375258e-05	4.166667908134154e-05
0.05	4.17e-05	4.17e-05	4.17e-05	4.17e-05	5.500001220001404e-05	7.150000326434978e-05	7.200004445893588e-05
0.06	7.2e-05	7.2e-05	7.2e-05	7.2e-05	8.500001220001404e-05	7.150000326434978e-05	7.200004445893588e-05
0.07	0.0001143	0.0001143	0.0001143	0.0001143	0.00014000012428008922	0.00017000031049814788	0.000170666959106638
0.08	0.0001707	0.0001707	0.0001707	0.0001707	0.0002040003202804372	0.0002422507185399861	0.00024300075829442776
0.09	0.000243	0.000243	0.000243	0.000243	0.000332501518002133375	0.000332501518002133375	0.00033333395017645215
0.1	0.0003333	0.0003333	0.0003333	0.0003333	0.00038500154869594165	0.0004427529810996755	0.0004343697600638993
0.11	0.0004437	0.0004437	0.0004437	0.0004437	0.00050600030309578666	0.0005750055137451706	0.0005760056878800406
0.12	0.000576	0.000576	0.000576	0.000576	0.00065000559134854	0.000732536999279108	0.000732343293694766
0.13	0.0007323	0.0007323	0.0007323	0.0007323	0.0008190086164212778	0.00091351633646104131	0.0009135333957133025
0.14	0.0009147	0.0009147	0.0009147	0.0009147	0.00105165241820217	0.001123776586484182	0.001125027112805113
0.15	0.001125	0.001125	0.001125	0.001125	0.0012400268267774655	0.001364041870302289	0.0013653799443617648
0.16	0.0013654	0.0013654	0.0013654	0.0013654	0.00149604220347777	0.001636314132469327	0.0016377318037289208
0.17	0.0016377	0.0016377	0.0016377	0.0016377	0.0017850645848655217	0.0019425858458458902	0.00194409718473121
0.18	0.0019441	0.0019441	0.0019441	0.0019441	0.0017850645848655217	0.0019425858458458902	0.00194409718473121
0.19	0.0022863	0.0022863	0.0022863	0.0022863	0.0021090964494212433	0.002284890139966846	0.002286475229466214
0.2	0.002669	0.002669	0.002669	0.002669	0.002470140932299873	0.00266520092739389	0.0026668960202987
0.21	0.0030873	0.0030873	0.0030873	0.0030873	0.0028702019482418273	0.003085330185378157	0.0030872889219963493
0.22	0.0035497	0.0035497	0.0035497	0.0035497	0.0033112843288540658	0.003547892320226487	0.003544792320785956
0.23	0.0040562	0.0040562	0.0040562	0.0040562	0.003795393974893113	0.00405425959783722	0.00405620720845016
0.24	0.0046007	0.0046007	0.0046007	0.0046007	0.00436453084947378	0.0046467224864947378	0.0046453084947378
0.25	0.0052093	0.0052093	0.0052093	0.0052093	0.00490072504133868	0.005207212077527273	0.005208302376838729
0.26	0.0058597	0.0058597	0.0058597	0.0058597	0.005528565212397988	0.005857766746770376	0.005858941912710343
0.27	0.0065627	0.0065627	0.0065627	0.0065627	0.006202270575313275	0.006560400641226499	0.006562460911212384
0.28	0.0073175	0.0073175	0.0073175	0.0073175	0.00713455213361459	0.00731713023381459	0.0073134707360591293
0.29	0.0081297	0.0081297	0.0081297	0.0081297	0.007716135735362176	0.00812967548516879	0.008132405951975667
0.3	0.0090035	0.0090035	0.0090035	0.0090035	0.008557731122869042	0.00900955744360347	0.0090003473137220844
0.31	0.0099303	0.0099347	0.0099347	0.0099347	0.00945464370480756	0.009932099026578509	0.00993470286465755
0.32	0.0109221	0.0109221	0.0109221	0.0109221	0.01038503501509954	0.01092453351509954	0.010925121060934703
0.33	0.011979	0.0119858	0.0119858	0.0119858	0.01144543934429431	0.01195289163623153	0.011958769702013
0.34	0.0131013	0.0131097	0.0131097	0.0131097	0.012535753916297993	0.013106811197180285	0.013109677136134066
0.35	0.0142917	0.0143019	0.0143019	0.0143019	0.013693323367560493	0.014288934172852209	0.014301889585972028
0.36	0.0155455	0.0155464	0.0155464	0.0155464	0.01485440789646113	0.01554440789646113	0.015545461084677826
0.37	0.0168943	0.0168994	0.0168994	0.0168994	0.016218426256296816	0.016896285828042045	0.016899419027737363
0.38	0.0182907	0.0183089	0.0183089	0.0183089	0.017580056936569943	0.018305627170269976	0.0183085085806314
0.39	0.0197773	0.0197948	0.0197948	0.0197948	0.01903751037600262	0.0197951499318363405	0.019794813292585616
0.4	0.0213594	0.0213594	0.0213594	0.0213594	0.02061775146954647	0.02135938946464646	0.02135938946464646
0.41	0.0229737	0.0230046	0.0230046	0.0230046	0.02216600303477747	0.023001132847194107	0.023004633087414787
0.42	0.024696	0.0247326	0.0247326	0.0247326	0.02385191635168285	0.02472906834317891	0.0247326626924603
0.43	0.0265023	0.0265455	0.0265455	0.0265455	0.02562160549081593	0.02654187917674713	0.02654568743331874
0.44	0.0283947	0.0284455	0.0284455	0.0284455	0.02747170157596603	0.02844170157596603	0.02844554646464646
0.45	0.030375	0.0304343	0.0304343	0.0304343	0.02942702160297246	0.030430576801696903	0.030434646046628205
0.46	0.03245145	0.0325145	0.0325145	0.0325145	0.0314547589401298	0.03251071638438279	0.03251469958991646
0.47	0.0346077	0.0346881	0.0346881	0.0346881	0.0335052696716413	0.034684258195993364	0.03468754951975267
0.48	0.036864	0.0369572	0.0369572	0.0369572	0.03569546016754003	0.03694330116384382	0.03695724951975267
0.49	0.0392163	0.039324	0.0393244	0.0393244	0.0381173628770498	0.03932076507365846	0.0393243652780493
0.5	0.0416667	0.0417907	0.0417907	0.0417907	0.040532892141179125	0.04178675194939268	0.04179114617220184
0.51	0.044217	0.0443594	0.0443594	0.0443594	0.04304932129463241	0.0443553107965547	0.04436003266801935

Рис. 1.

0.52	0.0466693	0.0470325	0.0470332	0.0470332	0.0466688537352717	0.04702863479479015	0.04703324574882013
0.53	0.0496257	0.0498121	0.049813	0.049813	0.04839371017286636	0.049808303235455184	0.04981302483249077
0.54	0.052489	0.0527005	0.0527016	0.0527016	0.05122612968913397	0.05269678257285689	0.0527016288299476
0.55	0.0557013	0.0557013	0.0557013	0.0557013	0.05469438629592326	0.05569438629592326	0.05570132756455451
0.56	0.0585387	0.0588128	0.0588144	0.0588145	0.05722371297677158	0.05880938134285878	0.05881445270906495
0.57	0.061731	0.0620413	0.0620433	0.0620433	0.06039145736557581	0.062038106509507474	0.062043299445934656
0.58	0.0650379	0.0653878	0.0653902	0.0653902	0.06367826646803319	0.06538949102899838	0.0653902771461395
0.59	0.0683597	0.0686576	0.0686576	0.0686576	0.06700147635922216	0.0686527668713661	0.0686576359650478
0.6	0.072	0.0724478	0.0724478	0.0724478	0.0706747540366591	0.07244228539095562	0.0724478612724697
0.61	0.0756603	0.0761592	0.0761634	0.0761634	0.07462132955948788	0.0761578769352877694	0.0761634036693196
0.62	0.0794427	0.0800017	0.0800067	0.0800067	0.07803347106942095	0.08000085717290951	0.08000670528351507
0.63	0.0833949	0.0839802	0.0839826	0.0839826	0.08159383262954238	0.08397427316017583	0.0839826333322934
0.64	0.0873813	0.0880794	0.0880866	0.0880866	0.08597450224528782	0.08808047285490311	0.0880866808053824
0.65	0.0915417	0.0923198	0.0923282	0.0923282	0.09014441839965795	0.09232202424931271	0.092328230104479921
0.66	0.095832	0.0966979	0.0967079	0.0967079	0.09445067056134408	0.09670153355108496	0.096707974621351592
0.67	0.1002543	0.1012164	0.1012281	0.1012282	0.09898589766815017	0.10122164464594948	0.10122824464594948
0.68	0.1048107	0.1058778	0.1058917	0.1058918	0.10348269183452337	0.10588505613152582	0.10589182174464317
0.69	0.1095053	0.110685	0.1107013	0.1107013	0.10821377850961657	0.1107034934992059	0.1107034934992059
0.7	0.1143333	0.1156405	0.1156596	0.1156599	0.11309189072820996	0.11565274163214266	0.11565985340601438
0.71	0.1193037	0.1207473	0.1207697	0.1207699	0.1111977846307651	0.12076263222167041	0.12076992594480557
0.72	0.124416	0.1260082	0.1260345	0.1260345	0.1233030128371798	0.12602704937309792	0.1260345133964979
0.73	0.1296723	0.1314259	0.1314562	0.1314566	0.1206363309268456	0.1314489323180649	0.13145660934315503
0.74	0.1350747	0.1370034	0.1370387	0.1370392	0.13413080398302735	0.13703127825219555	0.13703931734936903
0.75	0.140625	0.1427488	0.1427846	0.1427852	0.1397867147087987	0.14277714530453392	0.1427852339325694
0.76	0.1463253	0.14865	0.1486972	0.148698	0.1456071756488948	0.1486896565465753	0.1486979617365568
0.77	0.1521777	0.1547251	0.1547797	0.1547805	0.1515951322929099	0.15477199874196676	0.15478053046992485
0.78	0.1581894	0.1609722	0.1610351	0.1610362	0.1577394313425897	0.1610274347634998	0.1610362009478205
0.79	0.1643863	0.1673946	0.1674683	0.1674683	0.16408680620000304	0.16745829615530272	0.16746830303042423
0.8	0.1706467	0.1739555	0.1740787	0.1740802	0.1705970599965222	0.17407100137767376	0.17408024648749688
0.81	0.177147	0.1807782	0.1808737	0.1808755	0.17728080453779013	0.18086603883028432	0.18087555658537095
0.82	0.1837893	0.1877462	0.1878585	0.1878578	0.1841439518698094	0.1878479909652505	0.18785787925463217
0.83	0.1905957	0.1949027	0.1950279	0.1950306	0.1912265667402489	0.195005286031562	0.19503064618480666
0.84	0.197568	0.202252	0.2023945	0.2023977	0.1984812327083068	0.20238741746274191	0.2023978031203191
0.85	0.2047083	0.2097969	0.2099593	0.2099631	0.20593118070568092	0.2099522561956382	0.20996312951962568
0.86	0.2102017	0.2175413	0.2177262	0.2177307	0.21350025271754928	0.21771591449879453	0.21773003574143935
0.87	0.2157046	0.2236992	0.2239046	0.2239104	0.222569373560138	0.2236992561956382	0.2237003574143935
0.88	0.2217573	0.2336442	0.2338825	0.233889	0.2294917456575353	0.23387739409765798	0.23388910680790503
0.89	0.2349597	0.2421005	0.2422804	0.2422881	0.23726241027087474	0.24228827507946982	0.24228827507946982
0.9	0.243	0.250592	0.2508974	0.2509065	0.24624871990816247	0.2508942143517922	0.250894214472066
0.91	0.251394	0.2593927	0.2597879	0.2598065	0.250973635807726	0.259763035807726	0.25978791271217
0.92	0.2595627	0.2681474	0.2688066	0.2688193	0.26386863752204494	0.268806637522105344	0.2688196658705777
0.93	0.268119	0.2776989	0.2781085	0.2781234	0.2730446841516047	0.27811009090979106	0.27812336350463517
0.94	0.2768613	0.2871594	0.2876483	0.2876659	0.28240172779680025	0.2876222029396247	0.2876642765289194
0.95	0.2857197	0.2947438	0.2957453	0.295755	0.2920747934646355	0.2947438194646355	0.29475716646355
0.96	0.294512	0.3063897	0.3074628	0.307487	0.3018528339581581	0.30746210634675	0.3074878042570645
0.97	0.3042243	0.3177495	0.3177746	0.3177766	0.31200858099751174	0.317761753878212	0.317774874478395
0.98	0.3137267	0.32029231	0.3215046	0.321516	0.3204635320364246	0.320311097748464	0.32032766363021
0.99	0.323433	0.3382277	0.3391305	0.3391421	0.3351073593337327	0.339126566643023	0.3393434711461123
1.0	0.3333333	0.3492063	0.3501851	0.3502302	0.3404179644179383	0.350214334770679	0.350231845434498
1.01	0.3434337	0.3604517	0.3615945	0.3615968	0.3582514806061523	0.3615807350354668	0.3615987074506589
1.02	0.3537318	0.3713687	0.3732467	0.3732487	0.3666641207317616	0.3733233358773416	0.3733513596949416
1.03	0.3644233	0.3837641	0.3851219	0.3851258	0.37841255489066325	0.385175987731507	0.3851955828223665
1.04	0.3749547	0.3954285	0.3973337	0.3973435	0.3904353151953059	0.397418761740475	0.3974390786666423
1.05	0.385875	0.4082099	0.4098902	0.4099948	0.40279405447058965	0.4098608958214833	0.4098981586700491
1.06	0.3976883	0.4309125	0.4327385	0.4327483	0.4254561895050505	0.4326819058050505	0.4327385058050505

1.07	0.4083477	0.4338363	0.4359075	0.4360337	0.4284034014514166	0.4360171889292006	0.4360395933334228
1.08	0.418904	0.4471076	0.4494039	0.4495494	0.44168769619516807	0.44953320315706624	0.4495565967409818
1.09	0.4316763	0.4606928	0.4632364	0.4634039	0.45530257640487004	0.4633882400732336	0.4634125830309353
1.1	0.4436667	0.4745987	0.4774135	0.4776062	0.4682568076867917	0.4775912740961146	0.4776170222760482
1.11	0.4558977	0.4888319	0.4931943	0.4931455	0.48535859815159813	0.492151657463329	0.49217849790187996
1.12	0.4683093	0.5033995	0.5068378	0.5070917	0.49821788733006156	0.5070791419728095	0.5071071368585224
1.13	0.4809657	0.5183086	0.5221038	0.5223947	0.5132440979626178	0.5223839022110527	0.5224131168306761
1.14	0.493848	0.5335666	0.5377521	0.5380851	0.5286472930035525	0.538076560352808	0.5381070653820722
1.15	0.5069583	0.5491009	0.5537932	0.554174	0.5444379726075533	0.55414862126973858	0.554200038391277
1.16	0.5202987	0.5651593	0.5702377	0.5706727	0.5606270996677226	0.570670458063423	0.570703774877399
1.17	0.533871	0.5815098	0.5870971	0.5875935	0.5772261271165411	0.5875954282052519	0.5876302772789227
1.18	0.5476773	0.5982404	0.6043828	0.6049407	0.5942470271348007	0.6049558204217961	0.6049822940021345
1.19	0.5617197	0.6183596	0.6252107	0.6257515	0.6117023224273962	0.6227649325460812	0.62280301295845213
1.2	0.576	0.6328759	0.6402824	0.6410157	0.629605197400168	0.6410367005227662	0.6410767268108402
1.21	0.5905203	0.6507981	0.6589221	0.6597557	0.6479691458080452	0.6597857388016988	0.6598277084605002
1.22	0.6052827	0.6691352	0.6780098	0.6789864	0.6668087859472374	0.6790273837988582	0.679074190282587
1.23	0.620289	0.6878966	0.6976498	0.6987238	0.6861391255174016	0.69877774070020204	0.6988238725960285
1.24	0.6355413	0.7070916	0.7177669	0.7189844	0.7059759945130595	0.7190537339275258	0.7191023047848482
1.25	0.6510417	0.7267301	0.7384067	0.7397855	0.7263360155613465	0.7398731615670394	0.7399242233814008
1.26	0.666792	0.7462221	0.7595852	0.7611455	0.7472366556363619	0.7612647541996333	0.7613084712449435
1.27	0.6827948	0.7673778	0.7813193	0.7830035	0.7646962018316281	0.7832182384684344	0.7832747890003128
1.28	0.6990507	0.7884078	0.8036265	0.8056196	0.7907342215686458	0.8057844060030731	0.8058439793944641
1.29	0.715563	0.8099229	0.8265252	0.828775	0.8133708276602436	0.8289751878449599	0.8290379920164044
1.3	0.7323333	0.8319342	0.8500345	0.8525721	0.8366275486931307	0.8528137355420295	0.8528799943691043
1.31	0.7493637	0.854483	0.8741743	0.8770342	0.8608270052454536	0.8773245089619784	0.8773944645255323
1.32	0.766656	0.877491	0.8989654	0.9021861	0.8850390725130207	0.9025333718772903	0.9026072878872664
1.33	0.7842123	0.9010602	0.9244295	0.9280539	0.9103509698931261	0.9284676960287773	0.9285458574458243
1.34	0.8020347	0.9251729	0.9505893	0.9546449	0.936327388686184	0.9551364745504781	0.9552391914439274
1.35	0.820128	0.9498417	0.9774685	0.9820482	0.9630304480982772	0.9826304485775809	0.9827180584437266
1.36	0.8384853	0.9750794	1.0050917	1.0102343	0.9805501097541002	1.010822285476513	1.0110150983342747
1.37	0.8571177	1.0008993	1.0334549	1.0392555	1.0188580049534988	1.0400664703017408	1.0401650007535654
1.38	0.876024	1.027315	1.0626749	1.0691458	1.048007721296016	1.0701000838876584	1.0702046377641672
1.39	0.8952063	1.0543404	1.092649	1.099414	1.0780345923134976	1.1010462053042224	1.10111732574091901
1.4	0.9146667	1.0818897	1.1235596	1.1316803	1.108977516089963	1.132984373457613	1.133112676891074
1.41	0.934407	1.1102776	1.1553145	1.1644029	1.1408758274018997	1.1659415230134724	1.16606749916414
1.42	0.9544293	1.1392191	1.1879869	1.198152	1.1737723036373934	1.1998510720504934	1.2000083502184644
1.43	0.9747387	1.1688285	1.2216104	1.2329727	1.2077142298900239	1.2350738712740499	1.2352171432660519
1.44	0.995328	1.1991246	1.2562201	1.2689131	1.2427489665008125	1.2713643420694947	1.2715173694198272
1.45	1.0162083	1.2301205	1.2918529	1.3060239	1.278929216438201	1.3088807982206307	1.309044420813653
1.46	1.0373787	1.2618337	1.3285471	1.3448589	1.3143100158479931	1.3476858030738204	1.3478609474213283
1.47	1.058941	1.2942813	1.3663431	1.3839764	1.3549535574838931	1.3876465667502028	1.3880342585607253
1.48	1.0805973	1.3274805	1.4052828	1.424936	1.3849215489132758	1.429435890536722	1.429636746600403
1.49	1.1026497	1.3614491	1.4454102	1.467303	1.436283610189502	1.4725301538423594	1.4727464737064934
1.5	1.125	1.3962054	1.4867713	1.5111459	1.479113716278492	1.5172148802896388	1.5174475947697453

Рис. 3.

5. Ответы на вопросы

- 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.
 - 1ое приближение: $[0 - 0.9]$ Значение функции совпадает с приближением на данном интервале до 2ух знаков.
 - 2ое приближение: $[0 - 1.18]$ Значение функции совпадает с приближением на данном интервале до 2ух знаков.
 - 3ье приближение: $[0 - 1.36]$ Значение функции совпадает с приближением на данном интервале до 2ух знаков.
 - 4ое приближение: $[0 - 1.51]$ Значение функции совпадает с приближением на данном интервале до 2ух знаков.
- Пояснить, каким образом можно доказать правильность полученного результата при фиксированном значении аргумента в численных методах.
 - Численные методы зависят от шага, по этому мы сравниваем значение полученные Эйлером при разных его величинах. На пример: берем шаг равный 0.00001 получаем (для примера) 200, а при 0.0000001 получаем 300. Видим, что слишком большая разница которая нас не устраивает, продолжаем уменьшать шаг. И так до тех пор пока не получаем устраивающую нас точность. Правда есть вероятно того, что мы выйдем за разрядность и будет происходить округление.
- Из каких соображений выбирался корень уравнения в неявном методе?

1.997	2.6546847	94.17541680571122	159.9532568733473	162.66176119606615
1.998	2.6586747	103.04841394524065	189.7990399002563	194.26385243431724
1.999	2.6626687	113.67138156587032	232.98922440745535	241.10121320580254
2	2.6666667	142.62724969805748	420.28806125158275	465.4404228687651

Рис. 4.

- а) Выбирается наименьший из двух корней для сохранения непрерывности функции
- 4) Каково значение функции при $x=2$, т.е. привести значение $u(2)$.
 - а) Около 465

Список литературы

Список литературы

- [1] Градов В.М. Курс лекций по Моделированию - 2020
- [2] Калитикин Численные методы URL <https://studfile.net/preview/878067/>