|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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** Приближенный аналитический метод Пикара в сравнении с численными методами  **Студент** Громова В. П.  **Группа** ИУ7-61Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Градов В. М. |  |

Москва.

2020 г.

Цель лабораторной работы: изучить метод Пикара нахождения решения дифференциального уравнения с заданными начальными условиями и сравнить получившиеся результаты с численными методами решения задачи.

Рассмотрим задачу Коши:

()

Проинтегрируем обе части дифференциального уравнения:

()

Используя формулу Ньютона-Лейбница:

()

В выражении (3) s – степень приближение, чем больше s, тем меньше разность между точным и приближенным решениями при фиксированном значении x.

В лабораторной работе дифференциальное уравнение было задано в следующем виде:

()

Тогда выражение (3) принимает следующий вид:

()

При этом , , и т.д.

Помимо представленного способа, задачу Коши можно решить численными методами: явным и неявным.

Явная схема:

, где h – шаг разбиения, для данной задачи.

Неявная схема:

, где h – шаг разбиения, для данной задачи. Выражение можно упростить, в результате чего получается квадратное уравнение относительно , в качестве решения следует взять наименьший корень.

Программа, решающая поставленную в лабораторной работе задачу, была написана на языке программирования Python 3.6, для вывода на экран таблицы с результатами была использована сторонняя библиотека PrettyTable. Ниже будет представлен листинг программы.

Листинг 1:

1. from prettytable import PrettyTable
2. from numpy import arange
3. from math import sqrt
4. import warnings
5. warnings.filterwarnings("ignore")
6. # первое приближение
7. def Picard1(x):
8. return (x \* x \* x) / 3
9. # второе приближение
10. def Picard2(x):
11. return (x \* x \* x) / 3 \*(1 + x \*\* 4 / 21)
12. # третье приближение
13. def Picard3(x):
14. return (x \* x \* x) / 3 \*(1 + x \*\* 4 / 21 + 2 \* x \*\* 8 / 693 \
15. + x \*\* 12 / 19845)
16. # четвертое приближение
17. def Picard4(x):
18. return (x \* x \* x) / 3 + x \*\* 7 / 63 + 2 \* x \*\* 11 / 2079 \
19. + 13 \* x \*\* 15 / 218295 + 82 \* x \*\* 19 / 37328445 \
20. + 662 \* x \*\* 23 / 1.0438212e10 + 4 \* x \*\* 27 / 3.34187816e9 \
21. + x \*\* 31 / 1.09876903e11
22. # явная схема
23. def explicit(x0, y0, h):
24. return y0 + h \* (x0 \* x0 + y0 \* y0)
25. # неявная схема
26. def implicit(x1, y0, h):
27. inv = 1/ h
28. dinv = 0.5 \* inv
29. D = dinv \* dinv - inv \* y0 - x1 \* x1
30. if D < 0:
31. return float("NaN")
32. return dinv - sqrt(D)
33. def calc\_shemes(x\_values, step):
34. explicit\_y, implicit\_y = [], []
35. explicit\_y.append(0)
36. implicit\_y.append(0)
37. for i in range(1, len(x\_values)):
38. tmp1 = explicit(x\_values[i - 1], explicit\_y[i - 1], step)
39. explicit\_y.append(tmp1)
40. tmp2 = implicit(x\_values[i], implicit\_y[i - 1], step)
41. implicit\_y.append(tmp2)
42. return explicit\_y, implicit\_y
43. def calc\_picard(x\_values):
44. p1, p2, p3, p4 = [], [], [], []
45. for x in x\_values:
46. p1.append(Picard1(x))
47. p2.append(Picard2(x))
48. p3.append(Picard3(x))
49. p4.append(Picard4(x))
50. return p1, p2, p3, p4
51. def main():
52. step = 0.1
53. x\_end = 2.5
54. x\_values = arange(0, x\_end, step)
55. for i in range(len(x\_values)):
56. x\_values[i] = round(x\_values[i], 5)
57. expl\_y, impl\_y = calc\_shemes(x\_values, step)
58. p1, p2, p3, p4 = calc\_picard(x\_values)
59. table = PrettyTable()
60. table.add\_column('X', x\_values)
61. table.add\_column('1 приближение', p1)
62. table.add\_column('2 приближение', p2)
63. table.add\_column('3 приближение', p3)
64. table.add\_column('4 приближение', p4)
65. table.add\_column('Явная', expl\_y)
66. table.add\_column('Неявная', impl\_y)
67. print(table)