|  |  |
| --- | --- |
| 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** Белоусова Ю.С.  **Группа** ИУ7-61Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Градов В.М. |  |

Москва.

2020 г.

***Цель работы:*** проанализировать и сравнить приближенный аналитический метод Пикара (3-е и 4-е приближения) и численные методы (явная и неявная схемы).

Указанные методы используются для решения задачи Коши, которая не имеет аналитического решения и задается следующей системой:

В таком случае выражение для i-го приближения в методе Пикара будет иметь вид:

Рассмотрим следующий пример:

Тогда первые четыре приближения будут вычисляться так:

Также эту задачу можно решить численными методами. Существует две схемы вычислений:

* явная схема
* неявная схема

При этом решение по неявной схеме сводится к решению уравнения относительно .

Рассмотрим неявную схему на примере нашей задачи:

В ходе лабораторной работы была написана программа, в которой были реализованы указанные выше методы. Далее приведен листинг кода программы.

*Листинг 1. Метод Пикара*

def pikar(xbeg, xend, step, etta):

p3 = []

p4 = []

x = xbeg

while (x < xend):

y3 = etta + 1/3 \* x\*\*3 + 1/63 \* x\*\*7 + 2/2079 \* x\*\*11 + 1/59535 \* x\*\*15

y4 = y3 + 4/93555 \* x\*\*15 + (2/2488563 + 2/3393495) \* x\*\*19 + (4/99411543 + 1/86266215) \* x\*\*23 + 4/371319795 \* x\*\*27 + 1/109876902975 \* x\*\*31

p3.append(y3)

p4.append(y4)

x += step

return p3, p4

*Листинг 2. Численный метод, явная схема*

def yavniy(xbeg, xend, realstep, writestep, y0):

yarr = [y0]

y = y0

summ = 0

x = xbeg

while (x < xend):

try:

y += realstep \* f(x, y)

if (abs(summ - writestep) < eps):

yarr.append(y)

summ = 0

summ += realstep

x += realstep

except:

n = trunc((xend - x)/realstep)

for i in range(n):

yarr.append('-')

break

return yarr

*Листинг 3. Численный метод, неявная схема*

def neyavniy(xbeg, xend, realstep, writestep, y0):

yarr = [y0]

y = y0

summ = 0

x = xbeg + realstep

while (x < xend + realstep):

d = 1 - 4\*realstep \* (realstep \* x\*\*2 + y)

if d < 0:

n = trunc((xend - x) / realstep)

for i in range(n):

yarr.append('-')

break

y = (1 - sqrt(d)) / (2 \* realstep)

if (abs(summ - writestep) < eps):

yarr.append(y)

summ = 0

summ += realstep

x += realstep

return yarr