Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра информатики

Отчет по лабораторной работе №9

Методы Эйлера и Рунге-Кутта

Выполнил:

студент группы 153503

Вергасов В.М.

Руководитель:

доцент

Анисимов В.Я.

Минск 2021

**СОДЕРЖАНИЕ**

[1 Цель работы 3](#_Toc120555246)

[2 Теоретические сведения 3](#_Toc120555247)

[3 Программная реализация 8](#_Toc120555248)

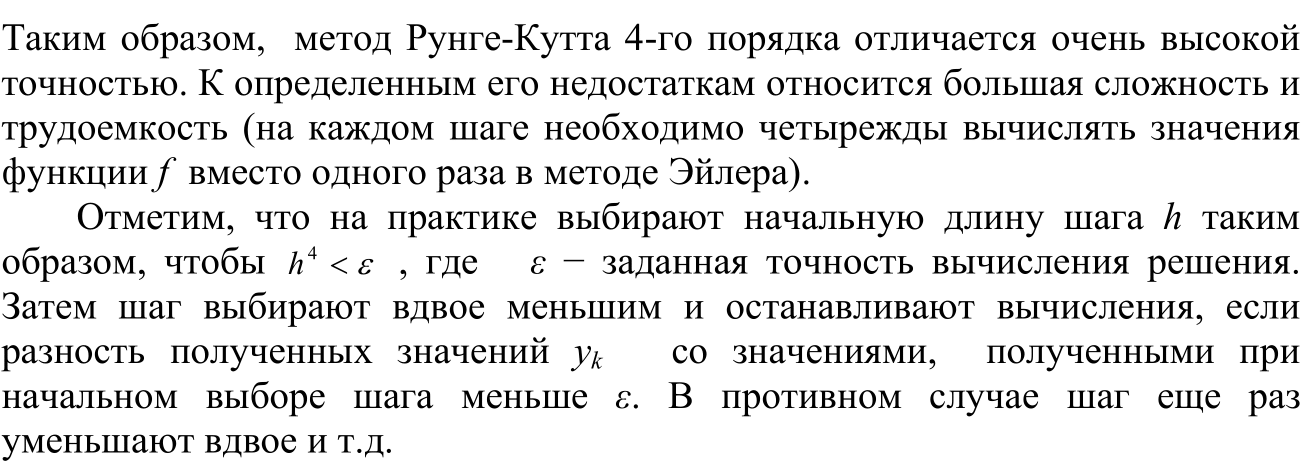
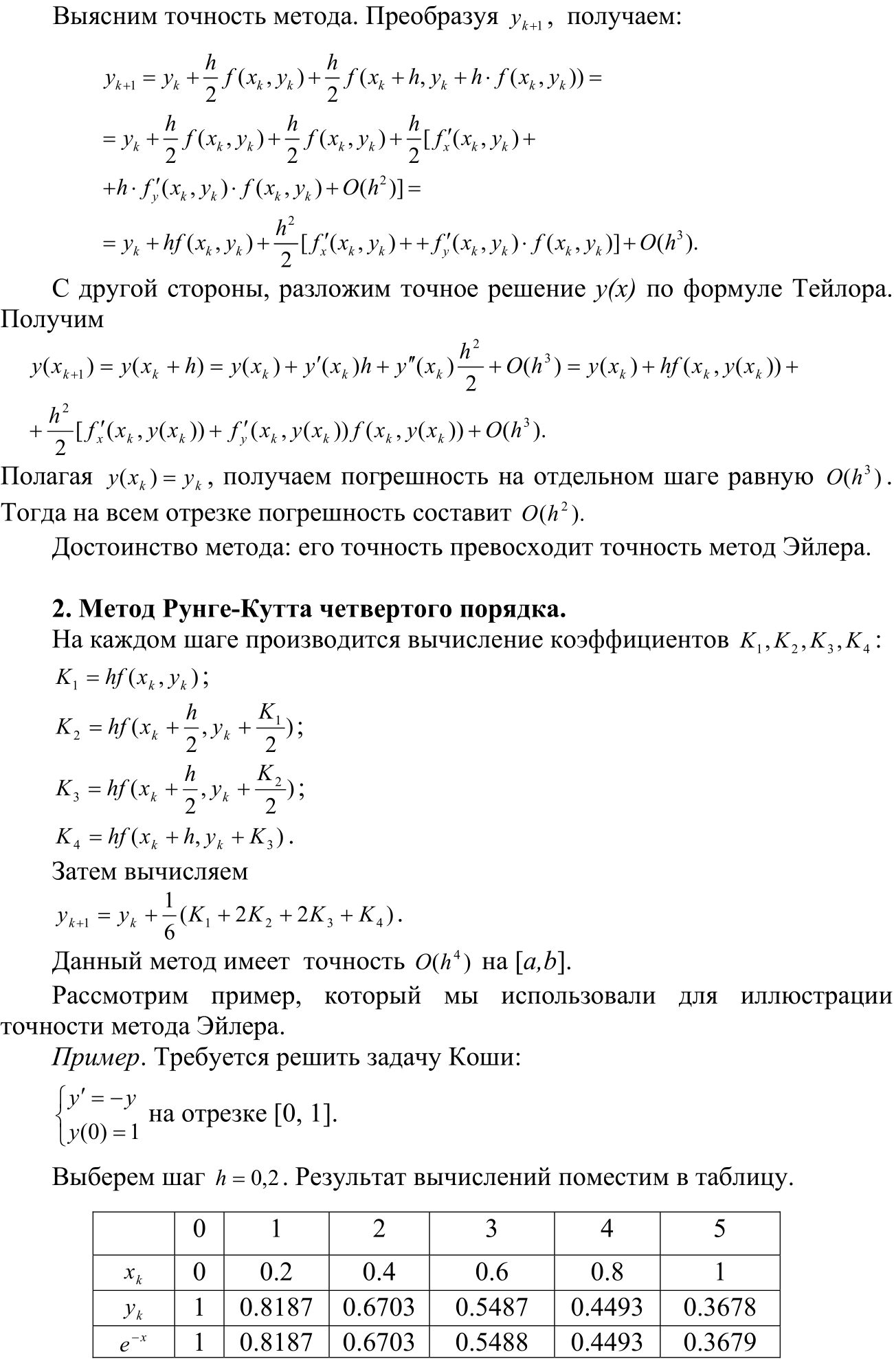
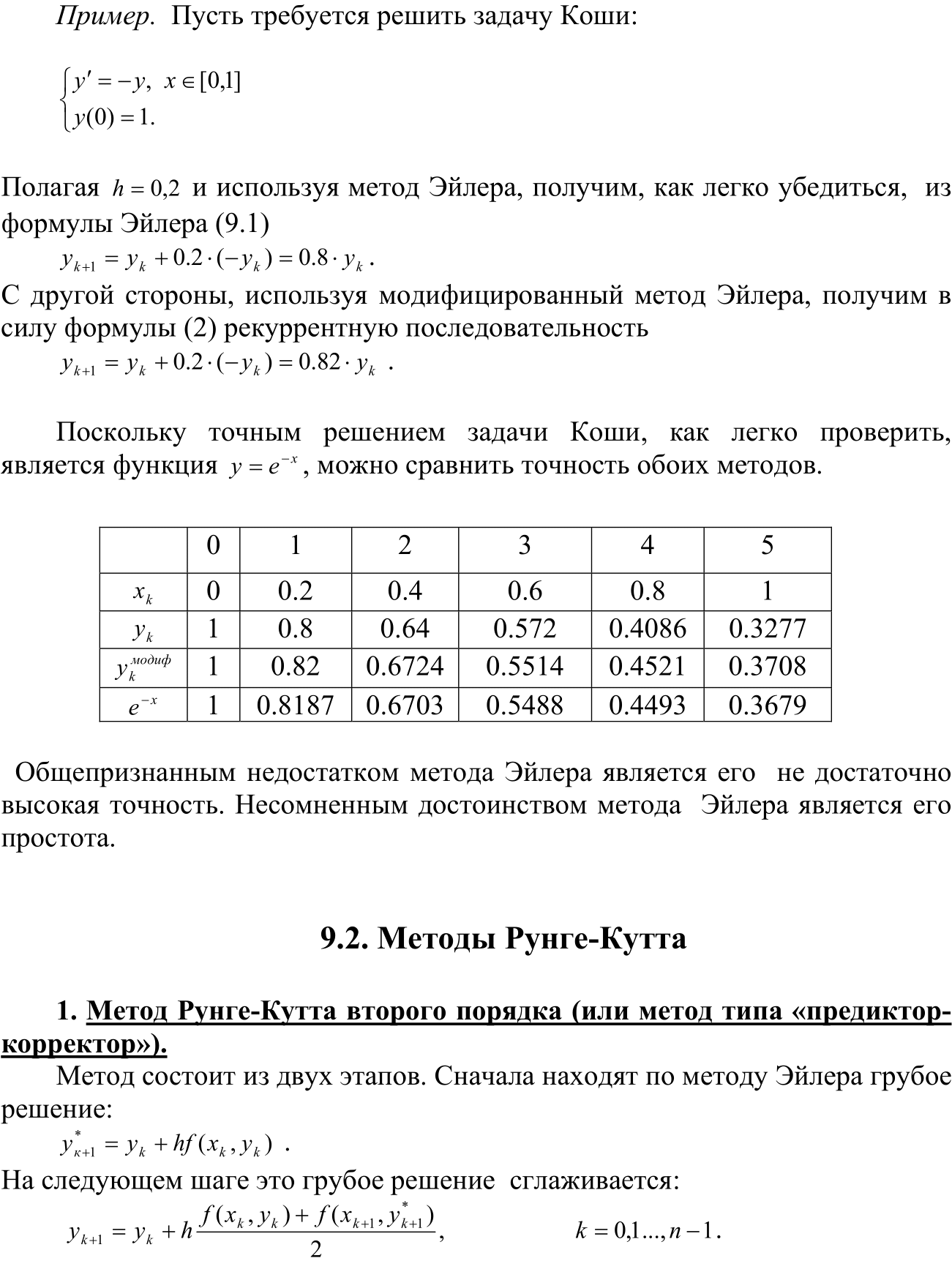
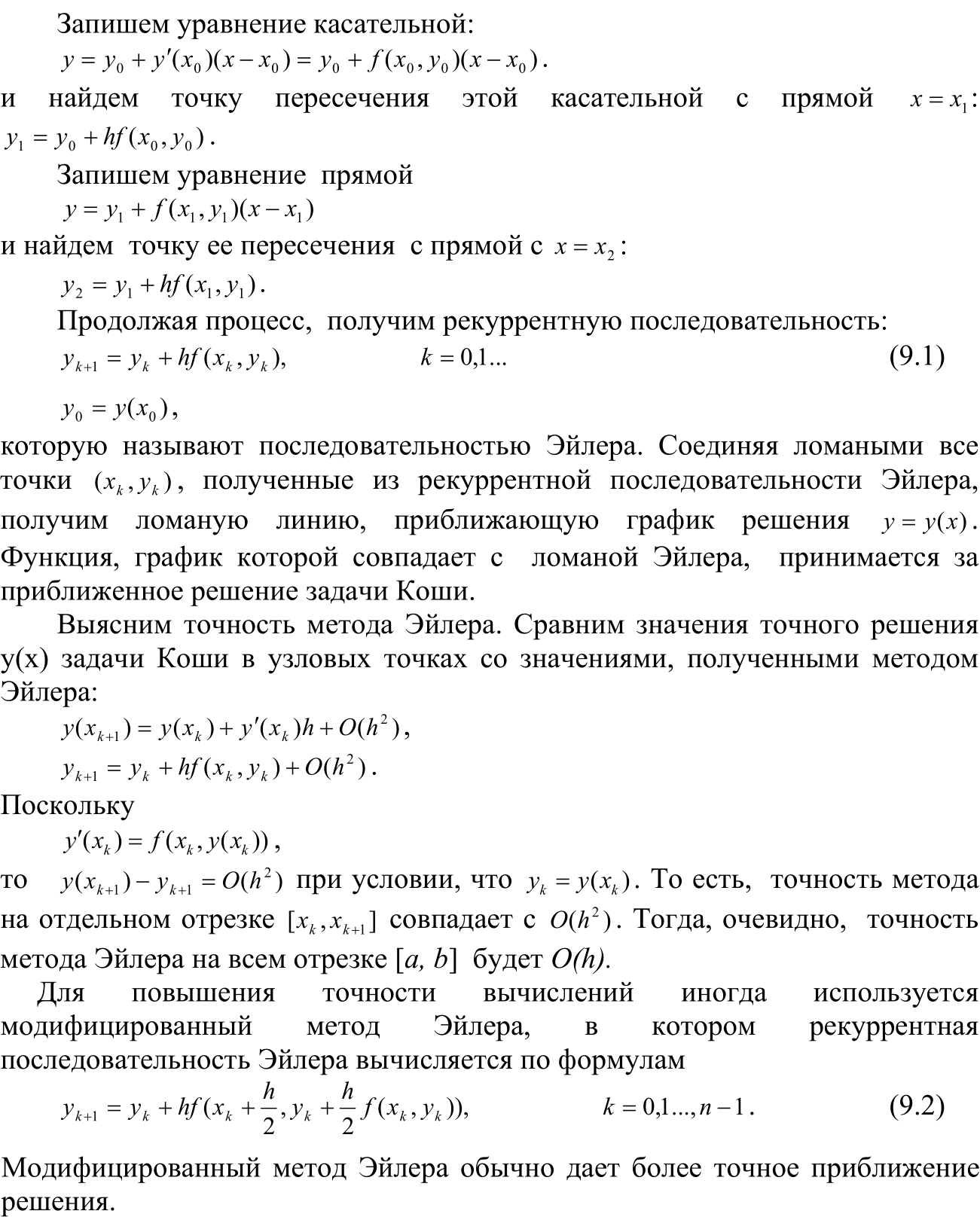
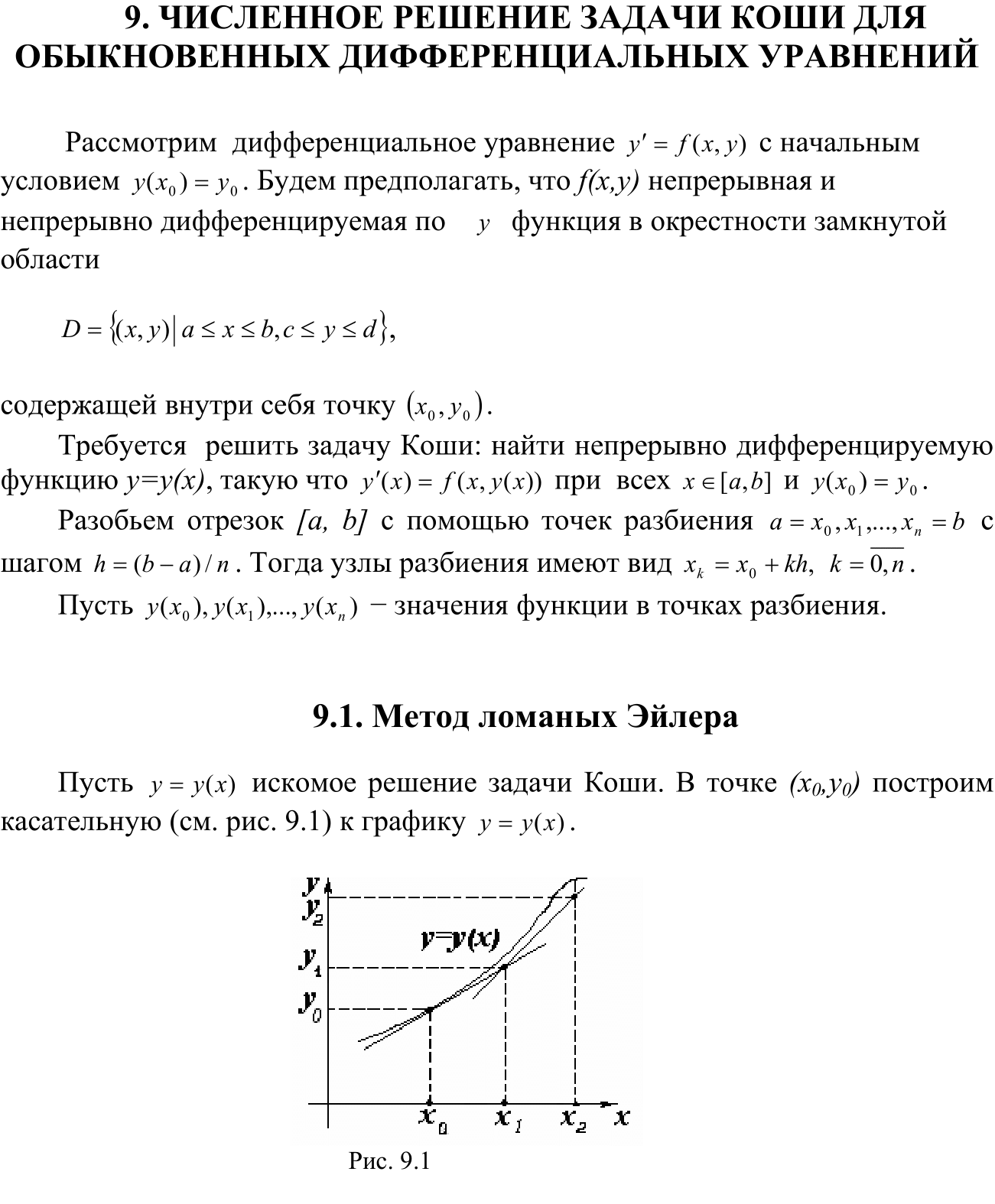
[4 Выводы 10](#_Toc120555249)

[Приложение А 11](#_Toc120555250)

1. **Цель работы**

Изучить решение задачи Коши для обыкновенных дифференциальных уравнений методом Эйлера и методом Рунге-Кутта.

1. **Теоретические сведения**



# **Программная реализация**

С помощью метода Эйлера, модифицированного метода Эйлера, метода Рунге-Кутта найти с заданной точностью решение заданного уравнения на заданном отрезке. Сравнить результаты.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Тестовый пример 1   |  |  |  |  | | --- | --- | --- | --- | | *Функция* |  | | *Отрезок* | |  |  | |  | | *Точность* | | *Кол-во точек* | | |  | |  | | | *Ср и макс кол-ва отрезков в методе* | | | | | *Эйлера* | *модиф. Эйлера* | | *Рунге-Кутта* | |  |  | |  | |  |  | |  | |  |
| Тестовый пример 2   |  |  |  |  | | --- | --- | --- | --- | | *Функция* |  | | *Отрезок* | |  |  | |  | | *Точность* | | *Кол-во точек* | | |  | |  | | | *Ср и макс кол-ва отрезков в методе* | | | | | *Эйлера* | *модиф. Эйлера* | | *Рунге-Кутта* | |  |  | |  | |  |  | |  | |  |
| Тестовый пример 3   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | *Функция* | |  | | *Отрезок* | | |  | |  | |  | | | *Точность* | | | *Кол-во точек* | | | |  | | |  | | | | *Ср и макс кол-ва отрезков в методе* | | | | | | | *Эйлера* | *модиф. Эйлера* | | | | *Рунге-Кутта* | |  |  | | | |  | |  |  | | | |  | |  |

ЗАДАНИЕ

Вариант 3

С помощью метода Эйлера, модифицированного метода Эйлера, метода Рунге-Кутта найти с точностью до 0.001 решение заданного уравнения на отрезке [0; 1]. Сравнить результаты.

Ответ:

|  |  |  |  |
| --- | --- | --- | --- |
| Количество точек для вычисления = 10 ^ 3 | | | |
| Количества отрезков в методе | | | |
|  | Эйлера | модиф. Эйлера | Рунге-Кутта |
| Среднее | 174.015984 | 8.697303 | 2.965035 |
| Максимальное | 512 | 16 | 4 |
|  | | | |

**Код программы представлен в Приложении А**

# **Выводы**

Таким образом, в ходе выполнения лабораторной работы были освоены метод Эйлера, модифицированный метод Эйлера, метод Рунге-Кутта четвёртого порядка для решения задачи Коши для обыкновенных дифференциальных уравнений. Составлена компьютерная программа, на тестовых примерах проверена правильность её работы, с заданной точностью построены графики решения дифференциального уравнения заданного варианта, по количеству необходимых для этого отрезков сравнена трудоёмкость методов. Получилось, что самым оптимальным будет использование метода Рунге-Кутта из-за малого количества отрезков.

**Приложение А**

print(f"Методы Эйлера и Рунге-Кутта\n")

import matplotlib.pyplot as plt

plotdots = 10\*\*3

eps = 10\*\*-3

m, a = 2.0, 0.9

def yder(x, y):

return (a \* (1 - y\*\*2)) / ((1 + m) \* x\*\*2 + y\*\*2 + 1)

y0 = 0

LEFT, RIGHT = 0, 1

def Euler(xdot, N):

ydots = [y0]

h = xdot / N

for i in range(N):

x = i \* h

y = ydots[-1]

ydots += [y + h \* yder(x, y)]

return ydots

def BetterEuler(xdot, N):

ydots = [y0]

h = xdot / N

for i in range(N):

x = i \* h

y = ydots[-1]

ydots += [y + h \* yder(x + h / 2, y + h / 2 \* yder(x, y))]

return ydots

def RungeKutta(xdot, N):

ydots = [y0]

h = xdot / N

for i in range(N):

x = i \* h

y = ydots[-1]

K1 = h \* yder(x, y)

K2 = h \* yder(x + h / 2, y + K1 / 2)

K3 = h \* yder(x + h / 2, y + K2 / 2)

K4 = h \* yder(x + h, y + K3)

ydots += [y + 1 / 6 \* (K1 + 2 \* K2 + 2 \* K3 + K4)]

return ydots

def GetValueAtPoint(method, x):

n = 1

while True:

olddots, newdots = method(x, n), method(x, 2 \* n)

if max(abs(newdots[2 \* i] - olddots[i]) for i in range(n + 1)) < eps:

return newdots[-1], 2 \* n

else:

n \*= 2

def CreateYdots(method, xdots):

ydots = []

maxn = 0

midn = []

for x in xdots:

y, n = GetValueAtPoint(method, x)

ydots.append(y)

maxn = max(maxn, n)

midn += [n]

midn = sum(midn) / len(xdots)

return ydots, midn, maxn

print(f"Количество точек для вычисления = {plotdots}")

print(f"Точность = {eps}")

xdots = [LEFT + (RIGHT - LEFT) / plotdots \* i for i in range(plotdots + 1)]

ydots, midn, maxn = CreateYdots(Euler, xdots)

print(f"\n MidN / MaxN точек в методе Эйлера = {midn:.6f} / {maxn}")

plt.plot(xdots, ydots, "y")

ydots, midn, maxn = CreateYdots(BetterEuler, xdots)

print(f"\n MidN / MaxN точек в модифицированном методе Эйлера = {midn:.6f} / {maxn}")

plt.plot(xdots, ydots, "b--")

ydots, midn, maxn = CreateYdots(RungeKutta, xdots)

print(f"\n MidN / MaxN точек в методе Рунге-Кутта = {midn:.6f} / {maxn}")

plt.plot(xdots, ydots, "r:")

plt.show()