**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа природных ресурсов

Направление подготовки Химическая технология

Отделение химической инженерии

**Численные методы решения**

**обыкновенных дифференциальных уравнений**

**Лабораторная работа по дисциплине «Углубленный курс информатики»**

Выполнил студент гр. 2Д91 А.В. Радионов

(Подпись)

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Отчет принят:

Преподаватель

Доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

**Цель:** изучить и использовать для составления программ в Паскале численные методы решения обыкновенных дифференциальных уравнений.

**Теоретическая часть.**

Большинство балансовых уравнений в химии и химической технологии представлены системой дифференциальных уравнений, в результате решения которых могут быть получены зависимости, характеризующие протекание процесса. Обыкновенные дифференциальные уравнения широко используются для математического моделирования химико-технологических процессов. С помощью обыкновенных дифференциальных уравнений, например, исследуется кинетика химических реакций, процессы, протекающие в химических реакторах, массообменных и теплообменных аппаратах.

Дифференциальные уравнения устанавливают связь между независимыми переменными, искомыми функциями и их производными. Если искомая функция зависит от одной переменной, то дифференциальное уравнение называется обыкновенным.

Обыкновенными дифференциальными уравнениями (ОДУ) называются уравнения, которые содержат одну или несколько производных от искомой функции y = y(x).

К численному решению обыкновенных дифференциальных уравнений приходится обращаться, когда не удаётся построить аналитическое решение задачи через известные функции, хотя для некоторых задач численные методы оказываются более эффективными даже при наличии аналитических решений.

Простейшим из численных методов решения дифференциальных уравнений является **метод Эйлера.** Это один из самых старых и широко известных методов. Метод Эйлера является сравнительно грубым методом решения дифференциальных уравнений, однако идеи, положенные в его основу, являются, по существу, исходными для очень широкого класса численных методов.

*Формула Эйлера*

Основная идея методов ***Рунге – Кутты*** заключается в том, что производные аппроксимируются через значения функции f(x, y) в точках на интервале [x0, x0+h], которые выбираются из условия наибольшей близости алгоритма к ряду Тейлора. В зависимости от старшей степени h, с которой учитываются члены ряда, построены вычислительные схемы Рунге – Кутты разных порядков точности.

*Формула Рунге-Кутты*

**Практическая часть.**

**Задание**

Решите дифференциальное уравнение методами Эйлера и Рунге-Кутты. Определите погрешность расчетного значения переменной ***y*** для каждого из методов.

**1 пример.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дифференциальное уравнение** | **Отрезок, шаг** | **Начальные условия** | **Аналитическое решение** |
|  | [1; 2];  h = 0.1 | x0 = 1;  y0 =3.0307; |  |

**Программная реализация:**

**program** lab\_12\_1;

**type**

matrix = **array of array of** real;

**function** f(x, y: real): real;

**begin**

result := -(2 \* y + 1) \* cos(x) / sin(x);

**end**;

**function** g(x: real): real;

**begin**

result := -0.5 + 5 / (2 \* sqr(sin(x)));

**end**;

**function** eyler\_method(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 2);

x := start\_x;

y := start\_y;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

y := y + h \* f(x, y);

x := x + h

**end**;

**end**;

**function** eyler\_method\_pogreshnosti(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y, z, p: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 4);

x := start\_x;

y := start\_y;

z := start\_y;

p := 0;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

result[i, 2] := z;

result[i, 3] := p;

y := y + h \* f(x, y);

x := x + h;

z := g(x);

p := 100 \* abs(y - z) / z;

**end**;

**end**;

**function** runge\_kutt\_method(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y: real;

k1, k2, k3, k4: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 2);

x := start\_x;

y := start\_y;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

k1 := f(x, y);

k2 := f(x + h / 2, y + h / 2 \* k1);

k3 := f(x + h / 2, y + h / 2 \* k2);

k4 := f(x + h, y + h \* k3);

y := y + h / 6 \* (k1 + 2 \* k2 + 2 \* k3 + k4);

x := x + h

**end**;

**end**;

**function** runge\_kutt\_method\_pogreshnosti(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y, z, p: real;

k1, k2, k3, k4: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 4);

x := start\_x;

y := start\_y;

z := start\_y;

p := 0;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

result[i, 2] := z;

result[i, 3] := p;

k1 := f(x, y);

k2 := f(x + h / 2, y + h / 2 \* k1);

k3 := f(x + h / 2, y + h / 2 \* k2);

k4 := f(x + h, y + h \* k3);

y := y + h / 6 \* (k1 + 2 \* k2 + 2 \* k3 + k4);

x := x + h;

z := g(x);

p := 100 \* abs(y - z) / z;

**end**;

**end**;

**procedure** print\_results(y: matrix);

**var**

i, j: integer;

**begin**

**for** i := 0 **to** High(y) **do**

**begin**

**for** j := 0 **to** High(y[i]) **do**

write(y[i, j]:8:4);

writeln

**end**

**end**;

**begin**

writeln('Решение методом Эйлера:');

writeln('X':5, 'Ответ':11, 'Ответ':7, 'Погрешность %':15);

writeln(' ':5, 'методом':11, 'аналит.':9);

print\_results(eyler\_method\_pogreshnosti(1, 2, 0.1, 3.0307));

writeln;

writeln('Решение методом Рунге-Кутты:');

writeln('X':5, 'Ответ':11, 'Ответ':7, 'Погрешность %':15);

writeln(' ':5, 'методом':11, 'аналит.':9);

print\_results(runge\_kutt\_method\_pogreshnosti(1, 2, 0.1, 3.0307))

**end**.

**Ответ:**

Решение методом Эйлера:

X Ответ Ответ Погрешность %

методом аналит.

1.0000 3.0307 3.0307 0.0000

1.1000 2.5773 2.6476 2.6563

1.2000 2.2640 2.3779 4.7870

1.3000 2.0491 2.1927 6.5469

1.4000 1.9076 2.0744 8.0402

1.5000 1.8245 2.0126 9.3430

1.6000 1.7916 2.0021 10.5170

1.7000 1.8050 2.0422 11.6172

1.8000 1.8649 2.1361 12.6973

1.9000 1.9752 2.2918 13.8141

2.0000 2.1443 2.5236 15.0302

Решение методом Рунге-Кутты:

X Ответ Ответ Погрешность %

методом аналит.

1.0000 3.0307 3.0307 0.0000

1.1000 2.6476 2.6476 0.0003

1.2000 2.3779 2.3779 0.0003

1.3000 2.1927 2.1927 0.0003

1.4000 2.0744 2.0744 0.0003

1.5000 2.0126 2.0126 0.0003

1.6000 2.0021 2.0021 0.0003

1.7000 2.0422 2.0422 0.0003

1.8000 2.1361 2.1361 0.0003

1.9000 2.2918 2.2918 0.0003

2.0000 2.5236 2.5236 0.0003

**2 пример.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дифференциальное уравнение** | **Отрезок, шаг** | **Начальные условия** | **Аналитическое решение** |
|  | [1; 2];  h = 0.2 | x0 = 1;  y0 = e; |  |

**Программная реализация:**

**program** lab\_12\_2;

**type**

matrix = **array of array of** real;

**function** f(x, y: real): real;

**begin**

result := -y \* ln(y) / x;

**end**;

**function** g(x: real): real;

**begin**

result := exp(1 / x);

**end**;

**function** eyler\_method(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 2);

x := start\_x;

y := start\_y;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

y := y + h \* f(x, y);

x := x + h

**end**;

**end**;

**function** eyler\_method\_pogreshnosti(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y, z, p: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 4);

x := start\_x;

y := start\_y;

z := start\_y;

p := 0;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

result[i, 2] := z;

result[i, 3] := p;

y := y + h \* f(x, y);

x := x + h;

z := g(x);

p := 100 \* abs(y - z) / z;

**end**;

**end**;

**function** runge\_kutt\_method(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y: real;

k1, k2, k3, k4: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 2);

x := start\_x;

y := start\_y;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

k1 := f(x, y);

k2 := f(x + h / 2, y + h / 2 \* k1);

k3 := f(x + h / 2, y + h / 2 \* k2);

k4 := f(x + h, y + h \* k3);

y := y + h / 6 \* (k1 + 2 \* k2 + 2 \* k3 + k4);

x := x + h

**end**;

**end**;

**function** runge\_kutt\_method\_pogreshnosti(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y, z, p: real;

k1, k2, k3, k4: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 4);

x := start\_x;

y := start\_y;

z := start\_y;

p := 0;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

result[i, 2] := z;

result[i, 3] := p;

k1 := f(x, y);

k2 := f(x + h / 2, y + h / 2 \* k1);

k3 := f(x + h / 2, y + h / 2 \* k2);

k4 := f(x + h, y + h \* k3);

y := y + h / 6 \* (k1 + 2 \* k2 + 2 \* k3 + k4);

x := x + h;

z := g(x);

p := 100 \* abs(y - z) / z;

**end**;

**end**;

**procedure** print\_results(y: matrix);

**var**

i, j: integer;

**begin**

**for** i := 0 **to** High(y) **do**

**begin**

**for** j := 0 **to** High(y[i]) **do**

write(y[i, j]:8:4);

writeln

**end**

**end**;

**begin**

writeln('Решение методом Эйлера:');

writeln('X':5, 'Ответ':11, 'Ответ':7, 'Погрешность %':15);

writeln(' ':5, 'методом':11, 'аналит.':9);

print\_results(eyler\_method\_pogreshnosti(1, 2, 0.2, e));

writeln;

writeln('Решение методом Рунге-Кутты:');

writeln('X':5, 'Ответ':11, 'Ответ':7, 'Погрешность %':15);

writeln(' ':5, 'методом':11, 'аналит.':9);

print\_results(runge\_kutt\_method\_pogreshnosti(1, 2, 0.2, e))

**end**.

**Ответ:**

Решение методом Эйлера:

X Ответ Ответ Погрешность %

методом аналит.

1.0000 2.7183 2.7183 0.0000

1.2000 2.1746 2.3010 5.4912

1.4000 1.8931 2.0427 7.3267

1.6000 1.7205 1.8682 7.9098

1.8000 1.6038 1.7429 7.9825

2.0000 1.5196 1.6487 7.8312

Решение методом Рунге-Кутты:

X Ответ Ответ Погрешность %

методом аналит.

1.0000 2.7183 2.7183 0.0000

1.2000 2.3010 2.3010 0.0003

1.4000 2.0427 2.0427 0.0004

1.6000 1.8683 1.8682 0.0004

1.8000 1.7429 1.7429 0.0003

2.0000 1.6487 1.6487 0.0003

**3 пример.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дифференциальное уравнение** | **Отрезок, шаг** | **Начальные условия** | **Аналитическое решение** |
|  | [0; 2];  h = 0.2 | x0 = 0;  y0 = ln(2); |  |

**Программная реализация:**

**program** lab\_12\_3;

**type**

matrix = **array of array of** real;

**function** f(x, y: real): real;

**begin**

result := 2\*x/exp(y-sqr(x));

**end**;

**function** g(x: real): real;

**begin**

result := ln(exp(sqr(x))+1);

**end**;

**function** eyler\_method(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 2);

x := start\_x;

y := start\_y;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

y := y + h \* f(x, y);

x := x + h

**end**;

**end**;

**function** eyler\_method\_pogreshnosti(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y, z, p: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 4);

x := start\_x;

y := start\_y;

z := start\_y;

p := 0;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

result[i, 2] := z;

result[i, 3] := p;

y := y + h \* f(x, y);

x := x + h;

z := g(x);

p := 100 \* abs(y - z) / z;

**end**;

**end**;

**function** runge\_kutt\_method(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y: real;

k1, k2, k3, k4: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 2);

x := start\_x;

y := start\_y;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

k1 := f(x, y);

k2 := f(x + h / 2, y + h / 2 \* k1);

k3 := f(x + h / 2, y + h / 2 \* k2);

k4 := f(x + h, y + h \* k3);

y := y + h / 6 \* (k1 + 2 \* k2 + 2 \* k3 + k4);

x := x + h

**end**;

**end**;

**function** runge\_kutt\_method\_pogreshnosti(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y, z, p: real;

k1, k2, k3, k4: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 4);

x := start\_x;

y := start\_y;

z := start\_y;

p := 0;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

result[i, 2] := z;

result[i, 3] := p;

k1 := f(x, y);

k2 := f(x + h / 2, y + h / 2 \* k1);

k3 := f(x + h / 2, y + h / 2 \* k2);

k4 := f(x + h, y + h \* k3);

y := y + h / 6 \* (k1 + 2 \* k2 + 2 \* k3 + k4);

x := x + h;

z := g(x);

p := 100 \* abs(y - z) / z;

**end**;

**end**;

**procedure** print\_results(y: matrix);

**var**

i, j: integer;

**begin**

**for** i := 0 **to** High(y) **do**

**begin**

**for** j := 0 **to** High(y[i]) **do**

write(y[i, j]:8:4);

writeln

**end**

**end**;

**begin**

writeln('Решение методом Эйлера:');

writeln('X':5, 'Ответ':11, 'Ответ':7, 'Погрешность %':15);

writeln(' ':5, 'методом':11, 'аналит.':9);

print\_results(eyler\_method\_pogreshnosti(0, 2, 0.2, ln(2)));

writeln;

writeln('Решение методом Рунге-Кутты:');

writeln('X':5, 'Ответ':11, 'Ответ':7, 'Погрешность %':15);

writeln(' ':5, 'методом':11, 'аналит.':9);

print\_results(runge\_kutt\_method\_pogreshnosti(0, 2, 0.2, ln(2)))

**end**.

**Ответ:**

Решение методом Эйлера:

X Ответ Ответ Погрешность %

методом аналит.

0.0000 0.6931 0.6931 0.0000

0.2000 0.6931 0.7133 2.8317

0.4000 0.7348 0.7763 5.3538

0.6000 0.8248 0.8893 7.2451

0.8000 0.9756 1.0635 8.2639

1.0000 1.2044 1.3133 8.2910

1.2000 1.5304 1.6526 7.3937

1.4000 1.9689 2.0918 5.8728

1.6000 2.5240 2.6345 4.1947

1.8000 3.1874 3.2784 2.7749

2.0000 3.9463 4.0181 1.7882

Решение методом Рунге-Кутты:

X Ответ Ответ Погрешность %

методом аналит.

0.0000 0.6931 0.6931 0.0000

0.2000 0.7133 0.7133 0.0001

0.4000 0.7763 0.7763 0.0006

0.6000 0.8893 0.8893 0.0015

0.8000 1.0635 1.0635 0.0031

1.0000 1.3133 1.3133 0.0055

1.2000 1.6528 1.6526 0.0089

1.4000 2.0920 2.0918 0.0128

1.6000 2.6349 2.6345 0.0164

1.8000 3.2790 3.2784 0.0190

2.0000 4.0190 4.0181 0.0204

**4 пример.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Дифференциальное уравнение** | **Отрезок, шаг** | **Начальные условия** | **Аналитическое решение** |
|  | [2; 3];  h = 0.2 | x0 = 2;  y0 = 4; |  |

**Программная реализация:**

**program** lab\_12\_4;

**type**

matrix = **array of array of** real;

**function** f(x, y: real): real;

**begin**

result := (y-3)/(3\*x\*x+x);

**end**;

**function** g(x: real): real;

**begin**

result := 3+3.5\*x/(3\*x+1);

**end**;

**function** eyler\_method(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 2);

x := start\_x;

y := start\_y;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

y := y + h \* f(x, y);

x := x + h

**end**;

**end**;

**function** eyler\_method\_pogreshnosti(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y, z, p: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 4);

x := start\_x;

y := start\_y;

z := start\_y;

p := 0;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

result[i, 2] := z;

result[i, 3] := p;

y := y + h \* f(x, y);

x := x + h;

z := g(x);

p := 100 \* abs(y - z) / z;

**end**;

**end**;

**function** runge\_kutt\_method(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y: real;

k1, k2, k3, k4: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 2);

x := start\_x;

y := start\_y;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

k1 := f(x, y);

k2 := f(x + h / 2, y + h / 2 \* k1);

k3 := f(x + h / 2, y + h / 2 \* k2);

k4 := f(x + h, y + h \* k3);

y := y + h / 6 \* (k1 + 2 \* k2 + 2 \* k3 + k4);

x := x + h

**end**;

**end**;

**function** runge\_kutt\_method\_pogreshnosti(start\_x, stop\_x, h, start\_y: real): matrix;

**var**

x, y, z, p: real;

k1, k2, k3, k4: real;

i: integer;

**begin**

SetLength(result, Trunc((stop\_x - start\_x) / h) + 1);

**for** i := 0 **to** High(result) **do**

SetLength(result[i], 4);

x := start\_x;

y := start\_y;

z := start\_y;

p := 0;

**for** i := 0 **to** High(result) **do**

**begin**

result[i, 0] := x;

result[i, 1] := y;

result[i, 2] := z;

result[i, 3] := p;

k1 := f(x, y);

k2 := f(x + h / 2, y + h / 2 \* k1);

k3 := f(x + h / 2, y + h / 2 \* k2);

k4 := f(x + h, y + h \* k3);

y := y + h / 6 \* (k1 + 2 \* k2 + 2 \* k3 + k4);

x := x + h;

z := g(x);

p := 100 \* abs(y - z) / z;

**end**;

**end**;

**procedure** print\_results(y: matrix);

**var**

i, j: integer;

**begin**

**for** i := 0 **to** High(y) **do**

**begin**

**for** j := 0 **to** High(y[i]) **do**

write(y[i, j]:8:4);

writeln

**end**

**end**;

**begin**

writeln('Решение методом Эйлера:');

writeln('X':5, 'Ответ':11, 'Ответ':7, 'Погрешность %':15);

writeln(' ':5, 'методом':11, 'аналит.':9);

print\_results(eyler\_method\_pogreshnosti(2, 3, 0.2, 4));

writeln;

writeln('Решение методом Рунге-Кутты:');

writeln('X':5, 'Ответ':11, 'Ответ':7, 'Погрешность %':15);

writeln(' ':5, 'методом':11, 'аналит.':9);

print\_results(runge\_kutt\_method\_pogreshnosti(2, 3, 0.2, 4));

**end**.

**Ответ:**

Решение методом Эйлера:

X Ответ Ответ Погрешность %

методом аналит.

2.0000 4.0000 4.0000 0.0000

2.2000 4.0143 4.0132 0.0281

2.4000 4.0264 4.0244 0.0504

2.6000 4.0368 4.0341 0.0684

2.8000 4.0459 4.0426 0.0831

3.0000 4.0539 4.0500 0.0953

Решение методом Рунге-Кутты:

X Ответ Ответ Погрешность %

методом аналит.

2.0000 4.0000 4.0000 0.0000

2.2000 4.0132 4.0132 0.0000

2.4000 4.0244 4.0244 0.0000

2.6000 4.0341 4.0341 0.0000

2.8000 4.0426 4.0426 0.0000

3.0000 4.0500 4.0500 0.0000

**Выводы:**

В ходе работы мною изучены численные методы решения обыкновенных дифференциальных уравнений. Это отработано на практике – написаны программы на языке Паскаль.