**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 Е.А. Резинкина

(Подпись)

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

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

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

Томск 2020 г.

**Цель работы:** изучить и использовать метод Эйлера и Рунге-Кутты для решения обыкновенных дифференциальных уравнений.

**Теоретическая часть**

Суть метода Эйлера заключается в переходе от бесконечно малых приращений в уравнении к конечным: https://ok-t.ru/mydocxru/baza4/94753247916.files/image108.jpg (1)

т.е. в замене производной приближенным конечно-разностным отношением:

https://ok-t.ru/mydocxru/baza4/94753247916.files/image110.jpg

где h = ∆х - шаг интегрирования.

Отсюда https://ok-t.ru/mydocxru/baza4/94753247916.files/image112.jpg (3)

Рассматривая приближенное решение в точке https://ok-t.ru/mydocxru/baza4/94753247916.files/image114.jpg как новые начальные условия, можно по формуле (3) найти значение искомой функции ***у(х)*** в следующей точке. В общем случае формула Эйлера имеет вид: https://ok-t.ru/mydocxru/baza4/94753247916.files/image116.jpg (4)

Метод Эйлера может быть интерпретирован геометрически следующим образом: функцию ***у(х)*** заменяют ломаной, представляющей собой отрезки касательных к этой функции в узлах (рис. 5.1).

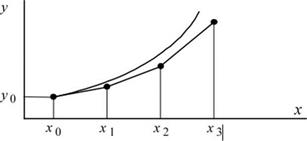


Рис. 5.1. Метод Эйлера

Достоинствами метода Эйлера являются его простота и наглядность, недостатками - относительно невысокая точность (он имеет первый порядок точности) и систематическое накопление ошибки. Точность и устойчивость решения в значительной степени зависят от величины шага интегрирования. Для оценки погрешности и выбора шага может быть применена формула Рунге https://ok-t.ru/mydocxru/baza4/94753247916.files/image119.jpg .

**Методы Рунге-Кутта второго порядка**

Методы Рунге-Кутта второго порядка основаны на разложении функции у(х) в ряд Тейлора и учете трех его первых членов (до второй производной включительно).

Метод Рунге-Кутта второго порядка с полным шагом реализуется по формуле:

https://ok-t.ru/mydocxru/baza4/94753247916.files/image121.jpg (6.1.)

Его геометрическая интерпретация (рис. 6.1.) заключается в следующем:

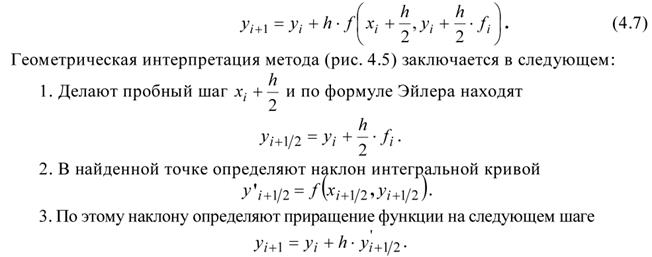
1. Приближенно вычисляют значение функции в точке xi+h по формуле Эйлера *https://ok-t.ru/mydocxru/baza4/94753247916.files/image123.jpg*и наклон интегральной кривой в этой точке https://ok-t.ru/mydocxru/baza4/94753247916.files/image125.jpg

2. Находят средний наклон на шаге h: https://ok-t.ru/mydocxru/baza4/94753247916.files/image127.jpg

3. По этому наклону уточняют значение *yi+1* по формуле (6.1.).

|  |
| --- |
| https://ok-t.ru/mydocxru/baza4/94753247916.files/image129.jpg Рис. 6.1. Метод Рунге-Кутта второго порядка с полным шагом |

Формула метода **Рунге-Кутта второго порядка с половинным шагом**имеет вид



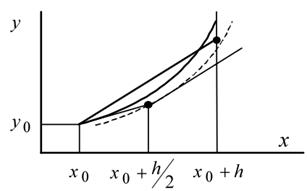


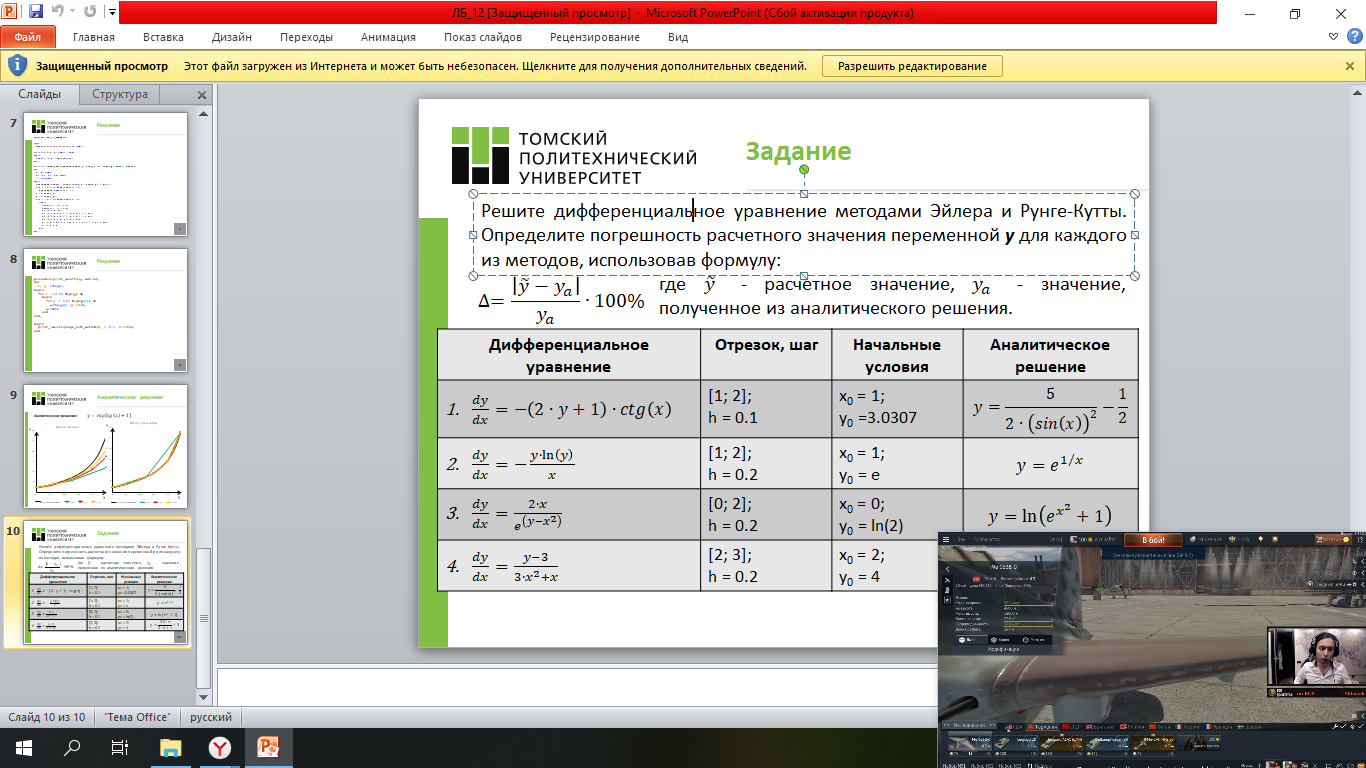
Рисунок 6.3. Метод Рунге-Кутта второго порядка с половинным шагом

**Практическая часть**

**Исходные данные:**

**Задание 1**

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



**Программная реализация**

Метод Эйлера

program lb12\_1\_1;

type

matrix = array of array of real;

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

begin

result := -(2\*y+1)/tan(x)

end;

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

var

x, y, ya, dy: 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;

ya := start\_y;

for i := 0 to High(result) do

begin

result[i, 0] := x;

result[i, 1] := y;

result[i, 2] := ya;

result[i, 3] := dy;

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

x := x + h;

ya :=5/(2\*sqr(sin(x)))-1/2;

dy := (abs(y-ya)/ya)\*100

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]:10:4);

write ('%');

writeln

end

end;

begin

writeln('Х':8,'Y':9,'Ya':11,'dY':10);

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

end.

**Ответ**

Х Y Ya dY

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%

Метод Рунге-Кутты

program lb12\_1\_2;

type

matrix = array of array of real;

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

begin

result := -(2\*y+1)/tan(x)

end;

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

var

x, y, ya, dy: 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;

ya := start\_y;

for i := 0 to High(result) do

begin

result[i, 0] := x;

result[i, 1] := y;

result[i, 2] := ya;

result[i, 3] := dy;

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;

ya :=5/(2\*sqr(sin(x)))-1/2;

dy := (abs(y-ya)/ya)\*100

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]:10:4);

write ('%');

writeln

end

end;

begin

writeln('Х':8,'Y':9,'Ya':11,'dY':10);

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

end.

**Ответ**

Х Y Ya dY

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%

**Задание 2**

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



**Программная реализация**

Метод Эйлера

program lb12\_2\_1;

type

matrix = array of array of real;

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

begin

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

end;

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

var

x, y, ya, dy: 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;

ya := start\_y;

for i := 0 to High(result) do

begin

result[i, 0] := x;

result[i, 1] := y;

result[i, 2] := ya;

result[i, 3] := dy;

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

x := x + h;

ya :=exp(1/x);

dy := (abs(y-ya)/ya)\*100

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]:10:4);

write ('%');

writeln

end

end;

begin

writeln('Х':8,'Y':9,'Ya':11,'dY':10);

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

end.

**Ответ**

Х Y Ya dY

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%

Метод Рунге-Кутты

program lb12\_2\_2;

type

matrix = array of array of real;

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

begin

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

end;

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

var

x, y, ya, dy: 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;

ya := start\_y;

for i := 0 to High(result) do

begin

result[i, 0] := x;

result[i, 1] := y;

result[i, 2] := ya;

result[i, 3] := dy;

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;

ya :=exp(1/x);

dy := (abs(y-ya)/ya)\*100

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]:10:4);

write ('%');

writeln

end

end;

begin

writeln('Х':8,'Y':9,'Ya':11,'dY':10);

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

end.

**Ответ**

Х Y Ya dY

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%

**Задание 3**

**Задание:** решите дифференциальное уравнение методами Эйлера и Рунге-Кутты. Определите погрешность расчетного значения переменной y для каждого из методов, использовав формулу:





**Программная реализация**

Метод Эйлера

program lb12\_3\_1;

type

matrix = array of array of real;

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

begin

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

end;

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

var

x, y, ya, dy: 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;

ya := start\_y;

for i := 0 to High(result) do

begin

result[i, 0] := x;

result[i, 1] := y;

result[i, 2] := ya;

result[i, 3] := dy;

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

x := x + h;

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

dy := (abs(y-ya)/ya)\*100

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]:10:4);

write ('%');

writeln

end

end;

begin

writeln('Х':8,'Y':9,'Ya':11,'dY':10);

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

end.

**Ответ**

Х Y Ya dY

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%

Метод Рунге-Кутты

program lb12\_3\_2;

type

matrix = array of array of real;

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

begin

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

end;

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

var

x, y, ya, dy: 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;

ya := start\_y;

for i := 0 to High(result) do

begin

result[i, 0] := x;

result[i, 1] := y;

result[i, 2] := ya;

result[i, 3] := dy;

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;

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

dy := (abs(y-ya)/ya)\*100

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]:10:4);

write ('%');

writeln

end

end;

begin

writeln('Х':8,'Y':9,'Ya':11,'dY':10);

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

end.

**Ответ**

Х Y Ya dY

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%

**Задание 4**

**Задание:** решите дифференциальное уравнение методами Эйлера и Рунге-Кутты. Определите погрешность расчетного значения переменной y для каждого из методов, использовав формулу:





**Программная реализация**

Метод Эйлера

program lb12\_4\_1;

type

matrix = array of array of real;

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

begin

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

end;

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

var

x, y, ya, dy: 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;

ya := start\_y;

for i := 0 to High(result) do

begin

result[i, 0] := x;

result[i, 1] := y;

result[i, 2] := ya;

result[i, 3] := dy;

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

x := x + h;

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

dy := (abs(y-ya)/ya)\*100

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]:10:4);

write ('%');

writeln

end

end;

begin

writeln('Х':8,'Y':9,'Ya':11,'dY':10);

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

end.

**Ответ**

Х Y Ya dY

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%

Метод Рунге-Кутты

program lb12\_3\_2;

type

matrix = array of array of real;

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

begin

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

end;

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

var

x, y, ya, dy: 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;

ya := start\_y;

for i := 0 to High(result) do

begin

result[i, 0] := x;

result[i, 1] := y;

result[i, 2] := ya;

result[i, 3] := dy;

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;

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

dy := (abs(y-ya)/ya)\*100

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]:10:4);

write ('%');

writeln

end

end;

begin

writeln('Х':8,'Y':9,'Ya':11,'dY':10);

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

end.

**Ответ**

Х Y Ya dY

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 были изучены и использованы метод Эйлера и Рунге-Кутты для решения обыкновенных дифференциальных уравнений.