

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А.И.
ГЕРЦЕНА»



Направление подготовки

09.03.01 – Информатика и вычислительная техника

Профиль «Технологии разработки программного обеспечения»

Лабораторная работа №4

«Численные методы решения дифференциальных уравнений»

Работу выполнили студенты 2 курса 2-1 группы:

Зухир Амира

Крючкова Анастасия

Стецук Максим

Каргаполов Денис

САНКТ-ПЕТЕРБУРГ

2022

СОДЕРЖАНИЕ

Отчет Зухир Амиры	3
Отчет Крючковой Анастасии	13
Отчет Стецук Максима	23
Отчет Каргополова Дениса	33

Лабораторная работа №3

Численные методы решения дифференциальных уравнений

Часть 1

Цель лабораторной работы: разработать программы решения дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутты.

Инструменты: ПК, PyCharm, Telegram.

В рамках данной лабораторной работы, был использован язык программирования Python 3.10

Постановка задачи: изучить численные методы Эйлера и Рунге-Кутты и предложенные варианты алгоритмов их реализации. Разработать программы решения дифференциальных уравнений, используя актуальный для студента язык программирования.

Решить дифференциальное уравнение $y' = y(1 - x)$ на отрезке $[0; 1]$ с начальными условиями $x_0 = 0, y_0 = 1$.

Математическая модель:

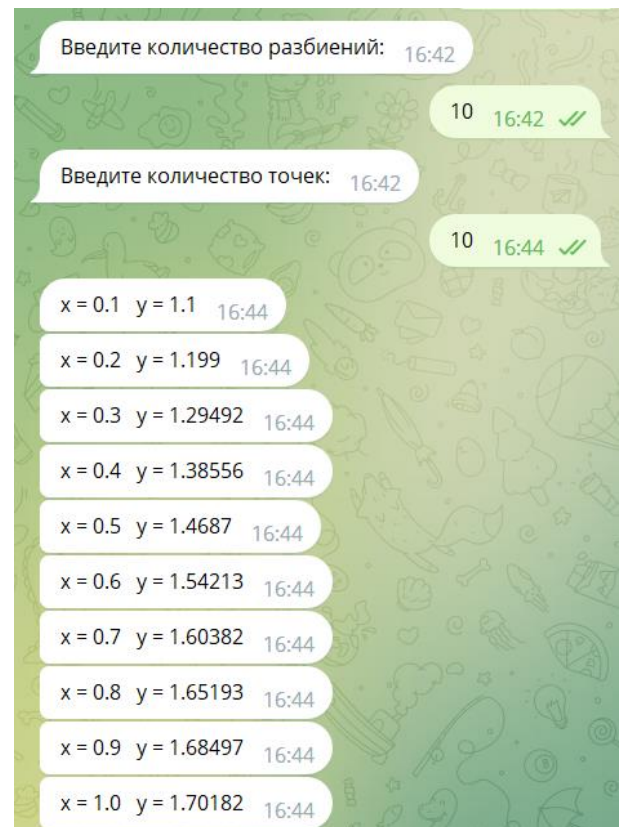
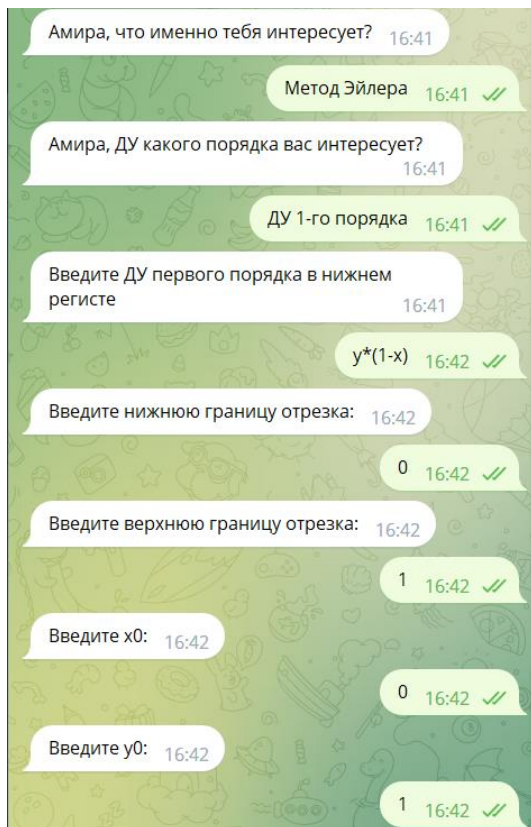
$$y' = y(1 - x)$$

Код:

Метод Эйлера:

```
def Euler_1_poryadok(x0, y0, a, b, n, f:str, colResh):
    count = 0
    h = (b-a) / n
    x = x0
    y = y0
    while (x < b) and (count < colResh):
        y += h * eval(f)
        x += h
        print('x = ', round(x, 5), ' y = ', round(y, 5))
        count += 1
```

Результат работы бота:



Код метода Рунге-Кутты:

```
def Runge_1_poryadok(x0, y0, a, b, n, f:str, colResh):
    count = 0
    h = (b - a) / n
    x = x0
    y = y0
    while (x < b) and (count < colResh):
        xbuffer = x
        ybuffer = y
        k1 = h * eval(f)
        x = xbuffer + h/2
        y = ybuffer + k1/2
        k2 = h * eval(f)
        x = xbuffer + h/2
        y = ybuffer + k2/2
        k3 = h * eval(f)
        x = xbuffer + h
        y = ybuffer + k3
        k4 = h * eval(f)
        F = (k1 + 2 * k2 + 2 * k3 + k4) / 6
        y = ybuffer + F
        x = xbuffer + h
        print('x = ', round(x, 5), ' y = ', round(y, 5))
        count += 1
```

Результат работы бота:

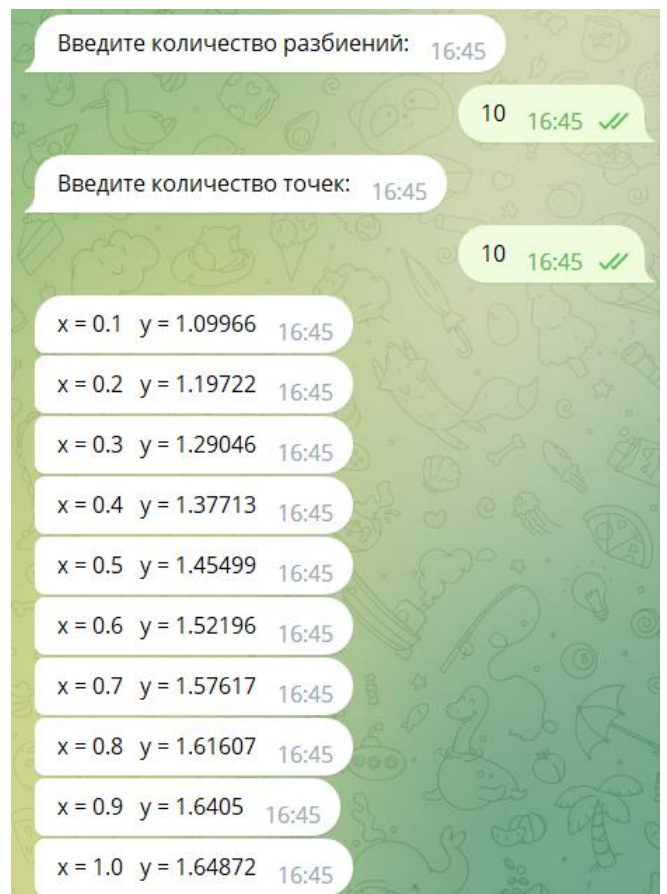
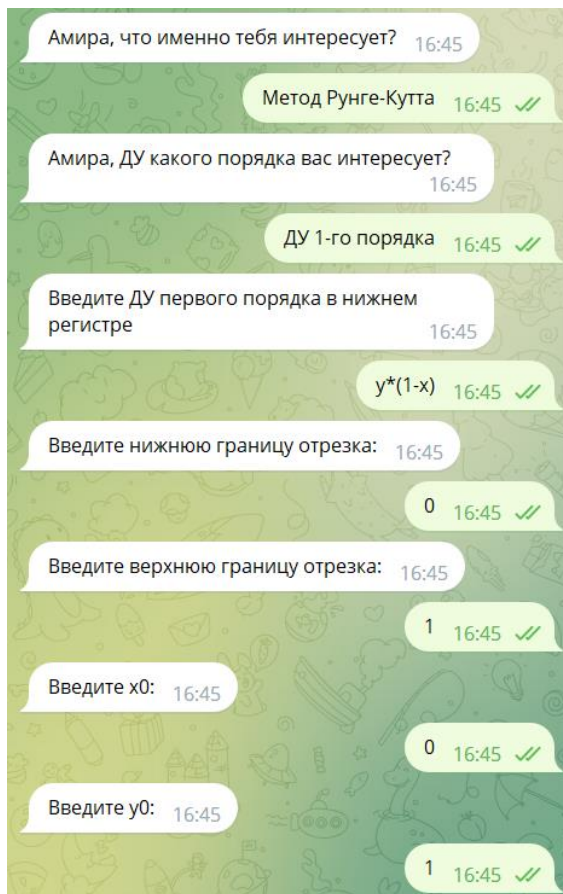


Таблица результатов при разбиении на 10 частей:

x =	Метод Эйлера: y =	Метод Рунге-Кутты: y =
0.1	1.1	1.09966
0.2	1.199	1.19722
0.3	1.29492	1.29046
0.4	1.38556	1.37713
0.5	1.4687	1.45499
0.6	1.54213	1.52196
0.7	1.60382	1.57617
0.8	1.65193	1.61607
0.9	1.68497	1.6405
1.0	1.70182	1.64872

Решим диф.уравнение и построим график с помощью Photomath:

$$y' = y \times (1 - x)$$



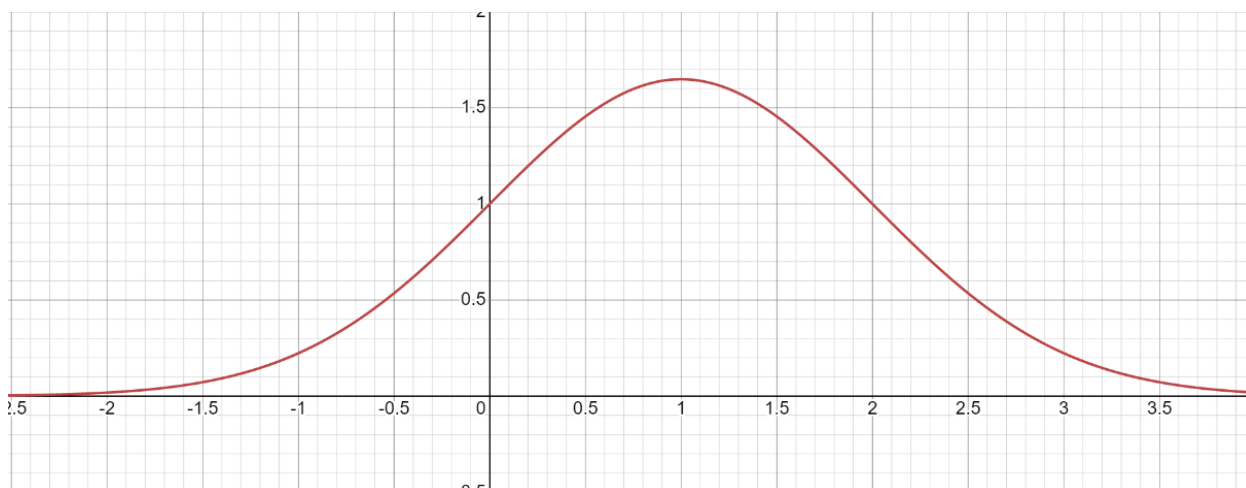
Решите методом разделения переменных

$$y = C e^{x - \frac{x^2}{2}}, C \in \mathbb{R}$$

Подставим y_0 и x_0 чтобы найти C и получим, что $C = 1$;

x	y
0	1
0.1	1.09966
0.5	1.45499
0.8	1.61607
1	1.64872

Получили данный график:



Часть 2

Цель лабораторной работы: разработать программы решения дифференциальных уравнений второго порядка и системы дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутты.

Инструменты: ПК, PyCharm, Telegram.

В рамках данной лабораторной работы, был использован язык программирования Python 3.10

Постановка задачи: изучить алгоритмы решения дифференциальных уравнений высших порядков (второго порядка) и системы дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутты. Разработать программы, используя актуальный для студента язык программирования.

Контрольный пример 2.

Применяя метод Эйлера (Рунге-Кутты) составить на отрезке $[1; 1,5]$ таблицу значений решения уравнения $y'' + y'/x + y = 0$ с начальными условиями:

$$y(1) = 0.77$$

$$y'(1) = -0.44.$$

Шаг вычисления $h = 0.1$.

Математическая модель:
$$y'' + \frac{y'}{x} + y = 0$$

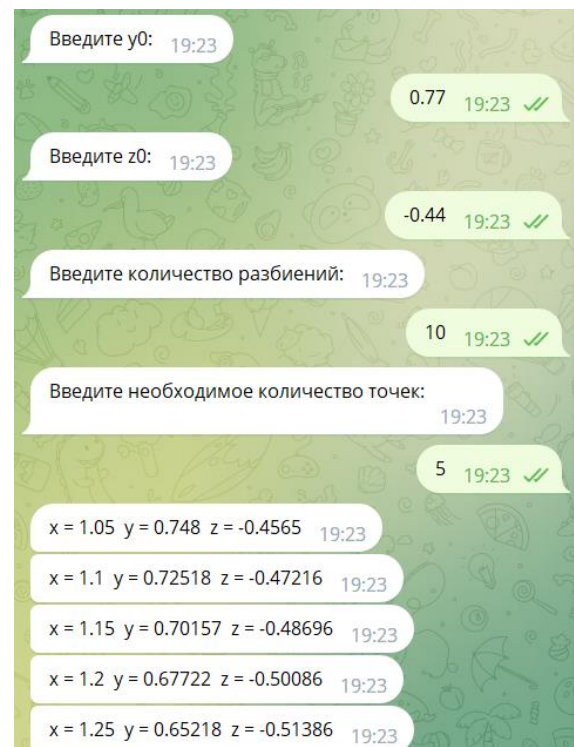
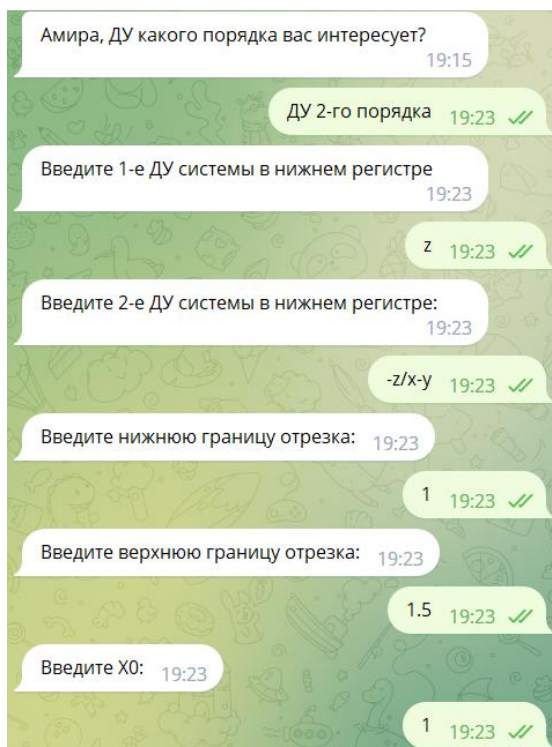
$$y' = z$$

$$z' = -z/x - y$$

Программа для метода Эйлера:

```
def Euler_2_poryadok(x0, y0, z0, a, b, n, f1:str, f2:str, colResh):
    count = 0
    h = (b-a) / n
    x = x0
    y = y0
    z = z0
    while (x < b) and (count < colResh):
        yBuf = y + h * eval(f1)
        zBuf = z + h * eval(f2)
        y = yBuf
        z = zBuf
        x += h
        print('x = ', round(x, 5), ' y = ', round(y, 5), ' z = ', round(z, 5))
        count += 1
```

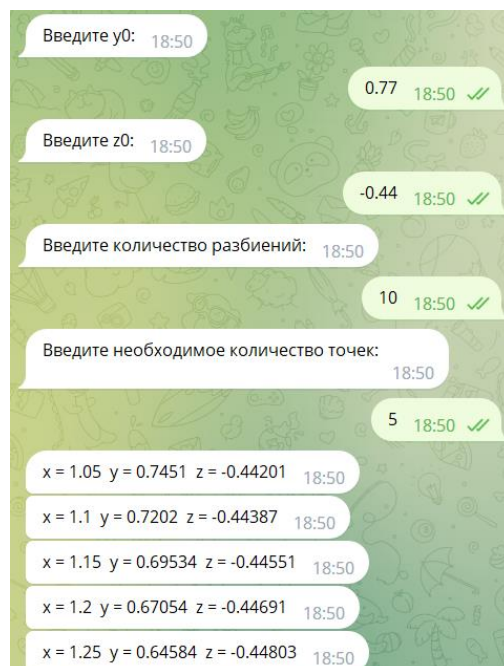
Результат работы бота:



Программа для метода Рунге-Кутты:

```
def Runge_2_poryadok(x0, y0, z0, a, b, n, f1:str, f2:str, colResh):
    count = 0
    h = (b - a) / n
    x = x0
    y = y0
    z = z0
    while (x < b) and (count < colResh):
        xbuffer = x
        ybuffer = y
        zbuffer = z
        q1 = eval(f2)
        k1 = eval(f1)
        x = xbuffer + h/2
        y = ybuffer + k1/2
        z = zbuffer + q1/2
        q2 = eval(f2)
        k2 = eval(f1)
        x = xbuffer + h / 2
        y = ybuffer + k2 / 2
        z = zbuffer + q2 / 2
        q3 = eval(f2)
        k3 = eval(f1)
        x = xbuffer + h
        y = ybuffer + k3
        z = zbuffer + q3
        q4 = eval(f2)
        k4 = eval(f1)
        x = xbuffer + h
        y = ybuffer + h / 6 * (k1 + 2 * k2 + 2 * k3 + k4)
        z = zbuffer + h / 6 * (q1 + 2 * q2 + 2 * q3 + q4)
        print('x = ', round(x, 5), ' y = ', round(y, 5), ' z = ', round(z, 5))
        count += 1
```

Результат работы бота:



Сравнительная таблица:

x=	Метод Эйлера	Метод Рунге-Кутта
1.05	y = 0.748 z = -0.4565	y = 0.7451 z = -0.44201
1.1	y = 0.72518 z = -0.47216	y = 0.7202 z = -0.44387
1.15	y = 0.70157 z = -0.48696	y = 0.69534 z = -0.44551
1.2	y = 0.67722 z = -0.50086	y = 0.67054 z = -0.44691
1.25	y = 0.65218 z = -0.51386	y = 0.64584 z = -0.44803

Контрольный пример 3.

Для системы дифференциальных уравнений

$$dy/dt = -2x + 5z$$

$$dy/dt = \sin(t-1)x - y + 3z$$

$$dz/dt = -x + 2z$$

с начальными условиями $x(0) = 2$, $y(0) = 1$, $z(0) = 1$ составить таблицу значений функций $x(t)$, $y(t)$, $z(t)$ на отрезке $[0; 0.3]$ с шагом $h = 0.003$. Использовать метод Эйлера.

Код программы:

```
def Euler_3_poryadok(x0, y0, z0, t0, a, b, n, f1:str, f2:str, f3:str, colResh):
    count = 0
    h = (b-a) / n
    x = x0
    y = y0
    z = z0
    t = t0
    while (t < b) and (count < colResh):
        xBuf = x + h * eval(f1)
        yBuf = y + h * eval(f2)
        zBuf = z + h * eval(f3)
        x = xBuf
        y = yBuf
        z = zBuf
        t += h
        print('t = ', round(t, 5), ' x = ', round(x, 5), ' y = ', round(y, 5), ' z = ', round(z, 5))
        count += 1
```

Результат вывода консоли:

```
[1]Численное интегрирование
[2]Вычисление элементарной функции
[3]Численное решение ДУ
more will coming soon
3
Выберите:
[1] Метод Эйлера
[2] Метод Рунге-Кутты
Для выхода в меню введите '3'
Для возврата нажмите '4'
3
Выбран 'Метод Эйлера для ДУ 3-го порядка'
Введите 1-е ДУ системы
-2*x +5*z
Введите 2-е ДУ системы
sin(t-1)*x-y+3*z
Введите 3-е ДУ системы
-x+2*z
Введите нижнюю границу отрезка
0
Введите верхнюю границу отрезка
0.3
Введите t0
0
Введите X0
2
Введите Y0
1
Введите Z0
1
Введите количество разбиений
10
Введите необходимое количество точек
10
t = 0.03  x = 2.03  y = 1.05895  z = 1.0
t = 0.06  x = 2.0582  y = 1.11615  z = 0.9991
t = 0.09  x = 2.08457  y = 1.17158  z = 0.9973
t = 0.12  x = 2.10909  y = 1.22519  z = 0.9946
t = 0.15  x = 2.13174  y = 1.27698  z = 0.991
t = 0.18  x = 2.15248  y = 1.32691  z = 0.98651
t = 0.21  x = 2.17131  y = 1.37497  z = 0.98113
t = 0.24  x = 2.1882  y = 1.42112  z = 0.97486
t = 0.27  x = 2.20314  y = 1.46536  z = 0.9677
t = 0.3  x = 2.21611  y = 1.50765  z = 0.95967
Выберите:
[1] ДУ 1-го порядка
[2] ДУ 2-го порядка (система из 2-х ДУ первого порядка)
[3] ДУ 3-го порядка (система из 3-х ДУ первого порядка)
Для возврата нажмите '4'
```

P.S: Возникли неполадки с телеграм ботом из-за функции $\sin(t-1)$, мы не смогли понять в чем проблема, поэтому решили контрольный пример 3 показать в консолюке. Однако, если не использовать тригонометрические функции, то данный метод работает в телеграм-боте нормально.

Уменьшив шаг в 10 раз получим:

```

Выберите:
[1] Метод Эйлера
[2] Метод Рунге-Кутты
Для выхода в меню введите '3'
1
Выберите:
[1] ДУ 1-го порядка
[2] ДУ 2-го порядка (система из 2-х ДУ первого порядка)
[3] ДУ 3-го порядка (система из 3-х ДУ первого порядка)
Для возврата нажмите '4'
3
Выбран 'Метод Эйлера для ДУ 3-го порядка'
Введите 1-е ДУ системы
-2*x+5*z
Введите 2-е ДУ системы
sin(t-1)*x - y +3*z
Введите 3-е ДУ системы
-x+2*z
Введите нижнюю границу отрезка
0
Введите верхнюю границу отрезка
0.3
Введите t0
0
Введите X0
2
Введите Y0
1
Введите Z0
1
Введите количество разбиений
100
Введите необходимое количество точек
10
t = 0.003  x = 2.003  y = 1.0061  z = 1.0
t = 0.006  x = 2.00598  y = 1.01219  z = 0.99999
t = 0.009  x = 2.00895  y = 1.01826  z = 0.99997
t = 0.012  x = 2.01189  y = 1.02431  z = 0.99995
t = 0.015  x = 2.01482  y = 1.03034  z = 0.99991
t = 0.018  x = 2.01773  y = 1.03636  z = 0.99987
t = 0.021  x = 2.02062  y = 1.04236  z = 0.99981
t = 0.024  x = 2.02349  y = 1.04834  z = 0.99975
t = 0.027  x = 2.02635  y = 1.0543  z = 0.99968
t = 0.03  x = 2.02919  y = 1.06024  z = 0.9996
Выберите:
[1] ДУ 1-го порядка
[2] ДУ 2-го порядка (система из 2-х ДУ первого порядка)
[3] ДУ 3-го порядка (система из 3-х ДУ первого порядка)
Для возврата нажмите '4'

```

Таблица:

Шаг 0.003	Шаг 0.0003
t= 0.03 x= 2.03 y= 1.05895 z= 1.0	t= 0.003 x= 2.003 y= 1.0061 z= 1.0
t= 0.06 x= 2.0582 y= 1.11615 z= 0.9991	t= 0.006 x= 2.00598 y= 1.01219 z= 0.99999
t= 0.09 x= 2.08457 y= 1.17158 z= 0.9973	t= 0.009 x= 2.00895 y= 1.01826 z= 0.99997
t= 0.12 x= 2.10909 y= 1.22519 z= 0.9946	t= 0.012 x= 2.01189 y= 1.02431 z= 0.99995
t= 0.15 x= 2.13174 y= 1.27698 z= 0.991	t= 0.015 x= 2.01482 y= 1.03034 z= 0.99991
t= 0.18 x= 2.15248 y= 1.32691 z= 0.98651	t= 0.018 x= 2.01773 y= 1.03636 z= 0.99987
t= 0.21 x= 2.17131 y= 1.37497 z= 0.98113	t= 0.021 x= 2.02062 y= 1.04236 z= 0.99981
t= 0.24 x= 2.1882 y= 1.42112 z= 0.97486	t= 0.024 x= 2.02349 y= 1.04834 z= 0.99975
t= 0.27 x= 2.20314 y= 1.46536 z= 0.9677	t= 0.027 x= 2.02635 y= 1.0543 z= 0.99968
t= 0.3 x= 2.21611 y= 1.50765 z= 0.95967	t= 0.03 x= 2.02919 y= 1.06024 z= 0.9996

Ссылка на Телеграм-бота: [Ссылка](#).



Вывод: мы изучили численные методы Эйлера и Рунге-Кутта и предложенные варианты алгоритмов их реализации. Разработали программы решения дифференциальных уравнений, используя язык программирования Python. А также, изучили алгоритмы решения дифференциальных уравнений высших порядков (второго порядка) и системы дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутта.

Лабораторная работа №3

Численные методы решения дифференциальных уравнений

Часть 1

Цель лабораторной работы: разработать программы решения дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутты.

Инструменты: ПК, PyCharm, Telegram.

В рамках данной лабораторной работы, был использован язык программирования Python 3.10

Постановка задачи: изучить численные методы Эйлера и Рунге-Кутты и предложенные варианты алгоритмов их реализации. Разработать программы решения дифференциальных уравнений, используя актуальный для студента язык программирования.

Решить дифференциальное уравнение $y' = y(1 - x)$ на отрезке $[0; 1]$ с начальными условиями $x_0 = 0, y_0 = 1$.

Математическая модель:

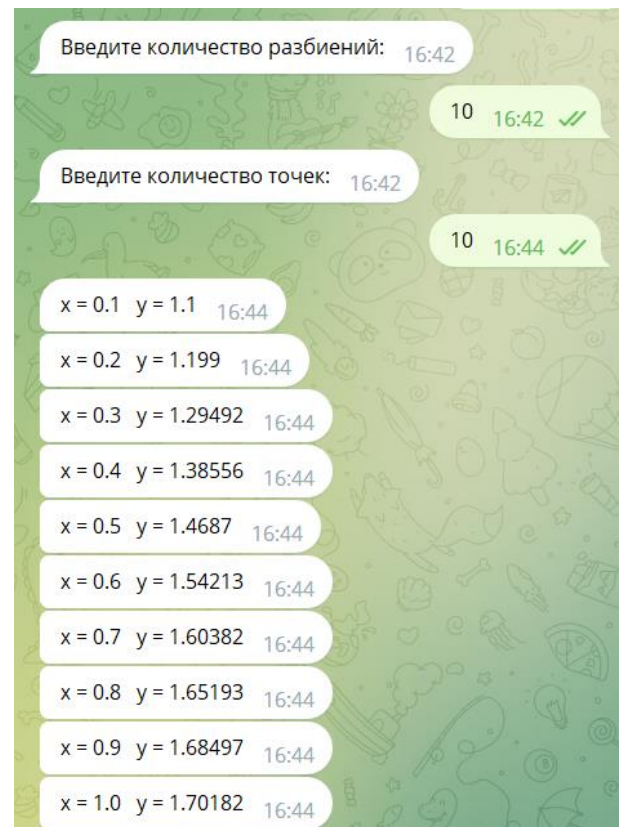
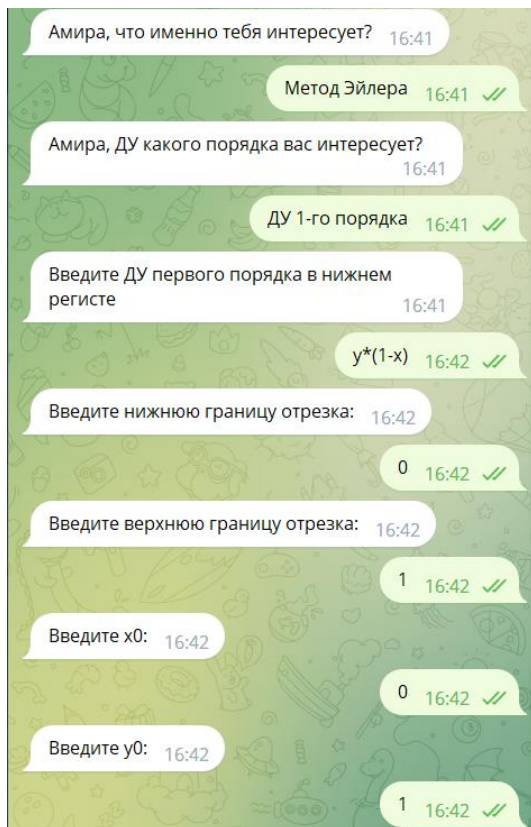
$$y' = y(1 - x)$$

Код:

Метод Эйлера:

```
def Euler_1_poryadok(x0, y0, a, b, n, f:str, colResh):
    count = 0
    h = (b-a) / n
    x = x0
    y = y0
    while (x < b) and (count < colResh):
        y += h * eval(f)
        x += h
        print('x = ', round(x, 5), ' y = ', round(y, 5))
        count += 1
```

Результат работы бота:



Код метода Рунге-Кутты:

```
def Runge_1_poryadok(x0, y0, a, b, n, f:str, colResh):
    count = 0
    h = (b - a) / n
    x = x0
    y = y0
    while (x < b) and (count < colResh):
        xbuffer = x
        ybuffer = y
        k1 = h * eval(f)
        x = xbuffer + h/2
        y = ybuffer + k1/2
        k2 = h * eval(f)
        x = xbuffer + h/2
        y = ybuffer + k2/2
        k3 = h * eval(f)
        x = xbuffer + h
        y = ybuffer + k3
        k4 = h * eval(f)
        F = (k1 + 2 * k2 + 2 * k3 + k4) / 6
        y = ybuffer + F
        x = xbuffer + h
        print('x = ', round(x, 5), ' y = ', round(y, 5))
        count += 1
```

Результат работы бота:

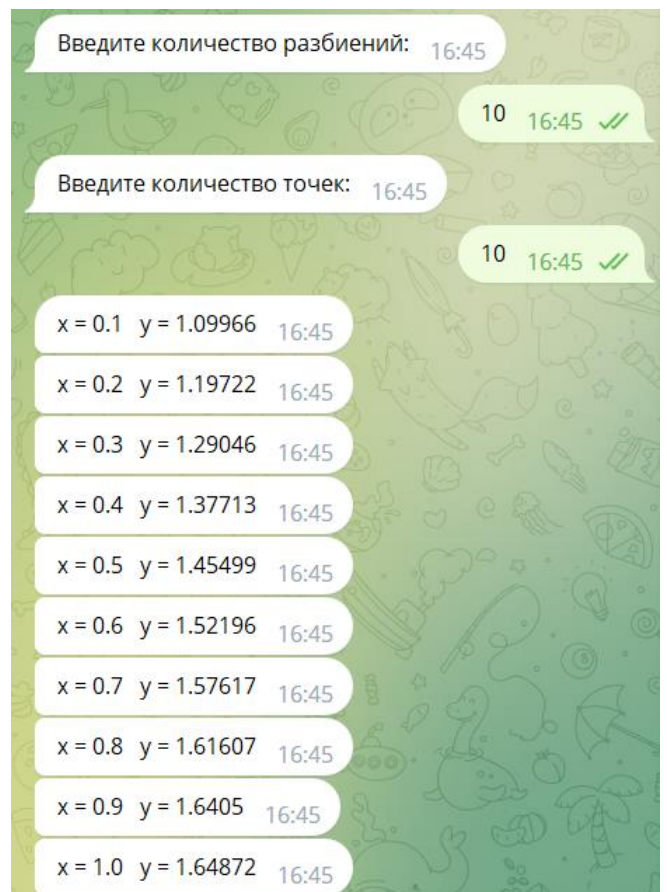
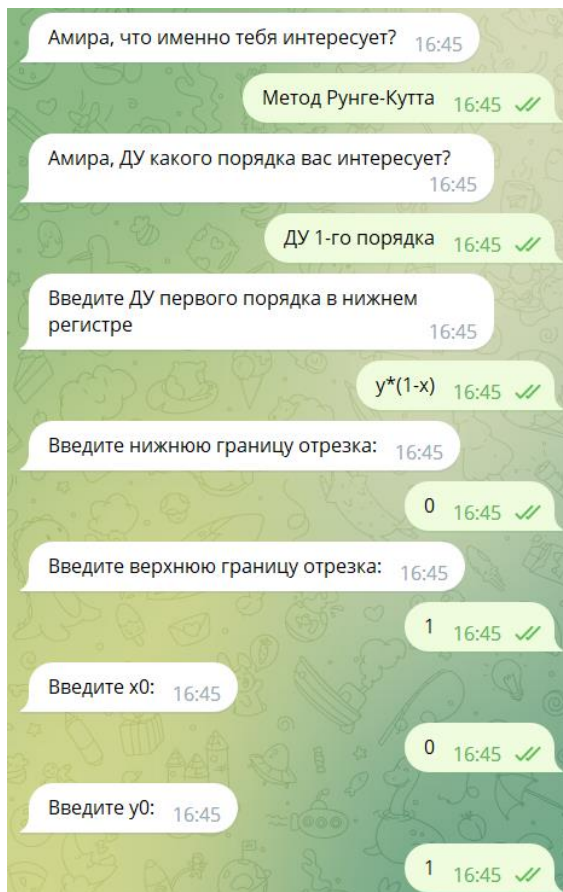


Таблица результатов при разбиении на 10 частей:

$x =$	Метод Эйлера: $y =$	Метод Рунге-Кутта: $y =$
0.1	1.1	1.09966
0.2	1.199	1.19722
0.3	1.29492	1.29046
0.4	1.38556	1.37713
0.5	1.4687	1.45499
0.6	1.54213	1.52196
0.7	1.60382	1.57617
0.8	1.65193	1.61607
0.9	1.68497	1.6405
1.0	1.70182	1.64872

Решим диф.уравнение и построим график с помощью Photomath:

$$y' = y \times (1 - x)$$



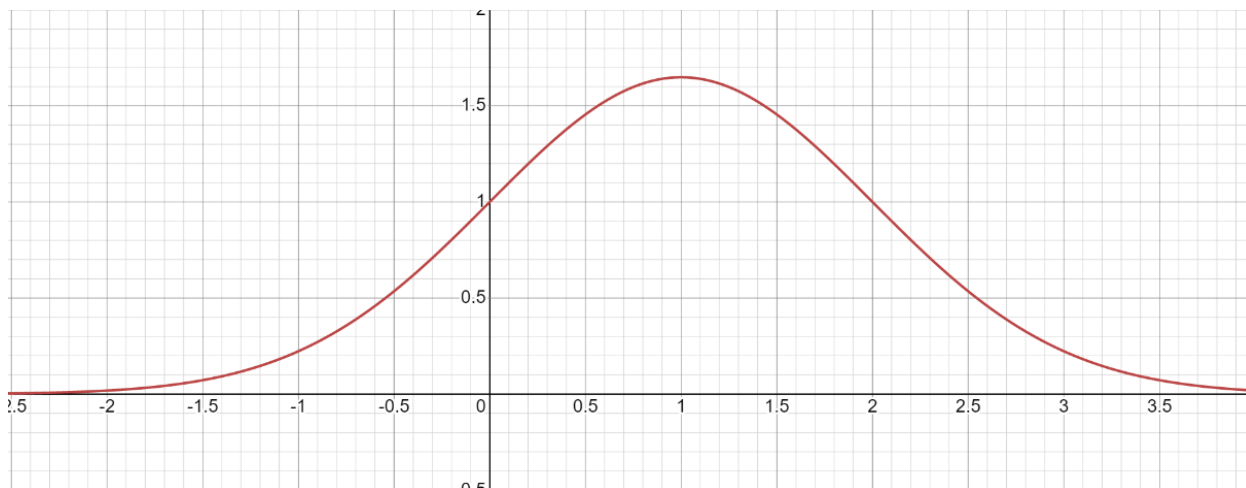
Решите методом разделения переменных

$$y = C e^{x - \frac{x^2}{2}}, C \in \mathbb{R}$$

Подставим y_0 и x_0 чтобы найти C и получим, что $C = 1$;

x	y
0	1
0.1	1.09966
0.5	1.45499
0.8	1.61607
1	1.64872

Получили данный график:



Часть 2

Цель лабораторной работы: разработать программы решения дифференциальных уравнений второго порядка и системы дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутты.

Инструменты: ПК, PyCharm, Telegram.

В рамках данной лабораторной работы, был использован язык программирования Python 3.10

Постановка задачи: изучить алгоритмы решения дифференциальных уравнений высших порядков (второго порядка) и системы дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутты. Разработать программы, используя актуальный для студента язык программирования.

Контрольный пример 2.

Применяя метод Эйлера (Рунге-Кутты) составить на отрезке $[1; 1.5]$ таблицу значений решения уравнения $y'' + y'/x + y = 0$ с начальными условиями:

$$y(1) = 0.77$$

$$y'(1) = -0.44.$$

Шаг вычисления $h = 0.1$.

Математическая модель:
$$y'' + \frac{y'}{x} + y = 0$$

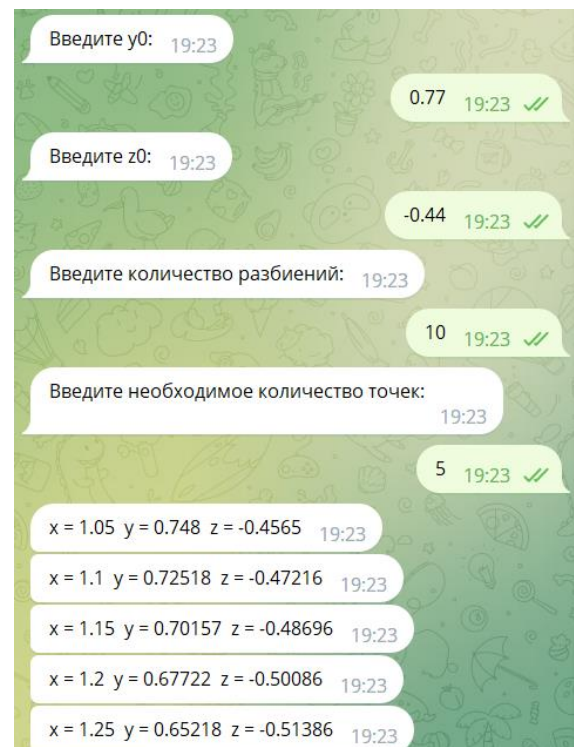
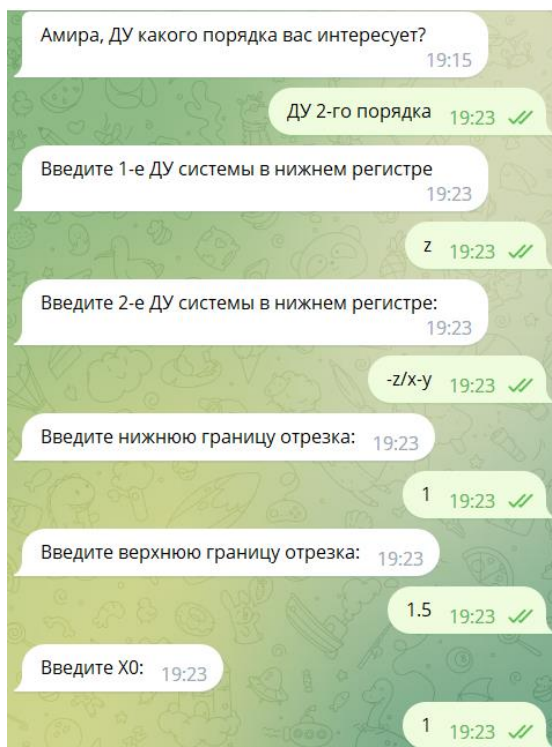
$$y' = z$$

$$z' = -z/x - y$$

Программа для метода Эйлера:

```
def Euler_2_poryadok(x0, y0, z0, a, b, n, f1:str, f2:str, colResh):
    count = 0
    h = (b-a) / n
    x = x0
    y = y0
    z = z0
    while (x < b) and (count < colResh):
        yBuf = y + h * eval(f1)
        zBuf = z + h * eval(f2)
        y = yBuf
        z = zBuf
        x += h
        print('x = ', round(x, 5), ' y = ', round(y, 5), ' z = ', round(z, 5))
        count += 1
```

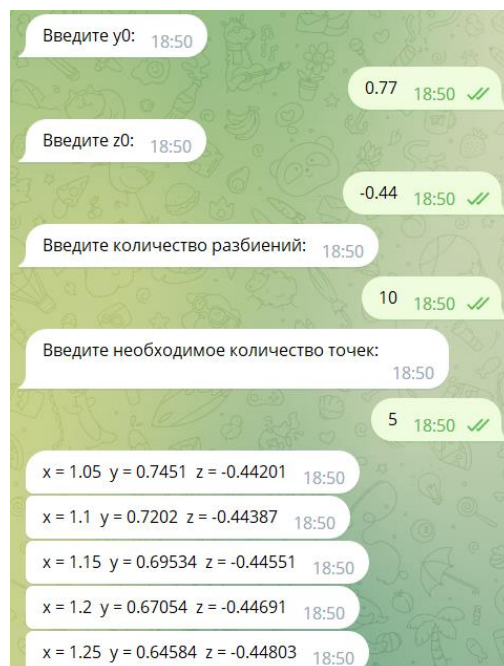
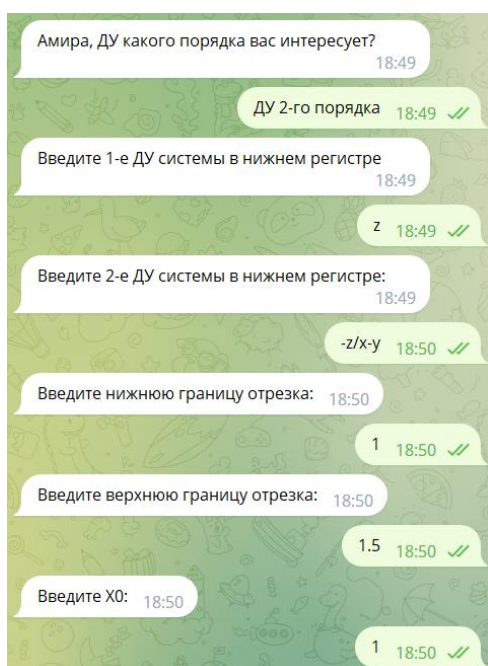
Результат работы бота:



Программа для метода Рунге-Кутты:

```
def Runge_2_poryadok(x0, y0, z0, a, b, n, f1:str, f2:str, colResh):
    count = 0
    h = (b - a) / n
    x = x0
    y = y0
    z = z0
    while (x < b) and (count < colResh):
        xbuffer = x
        ybuffer = y
        zbuffer = z
        q1 = eval(f2)
        k1 = eval(f1)
        x = xbuffer + h/2
        y = ybuffer + k1/2
        z = zbuffer + q1/2
        q2 = eval(f2)
        k2 = eval(f1)
        x = xbuffer + h / 2
        y = ybuffer + k2 / 2
        z = zbuffer + q2 / 2
        q3 = eval(f2)
        k3 = eval(f1)
        x = xbuffer + h
        y = ybuffer + k3
        z = zbuffer + q3
        q4 = eval(f2)
        k4 = eval(f1)
        x = xbuffer + h
        y = ybuffer + h / 6 * (k1 + 2 * k2 + 2 * k3 + k4)
        z = zbuffer + h / 6 * (q1 + 2 * q2 + 2 * q3 + q4)
        print('x = ', round(x, 5), ' y = ', round(y, 5), ' z = ', round(z, 5))
        count += 1
```

Результат работы бота:



Сравнительная таблица:

x=	Метод Эйлера	Метод Рунге-Кутта
1.05	y = 0.748 z = -0.4565	y = 0.7451 z = -0.44201
1.1	y = 0.72518 z = -0.47216	y = 0.7202 z = -0.44387
1.15	y = 0.70157 z = -0.48696	y = 0.69534 z = -0.44551
1.2	y = 0.67722 z = -0.50086	y = 0.67054 z = -0.44691
1.25	y = 0.65218 z = -0.51386	y = 0.64584 z = -0.44803

Контрольный пример 3.

Для системы дифференциальных уравнений

$$dy/dt = -2x + 5z$$

$$dy/dt = \sin(t-1)x - y + 3z$$

$$dz/dt = -x + 2z$$

с начальными условиями $x(0) = 2$, $y(0) = 1$, $z(0) = 1$ составить таблицу значений функций $x(t)$, $y(t)$, $z(t)$ на отрезке $[0; 0.3]$ с шагом $h = 0.003$. Использовать метод Эйлера.

Код программы:

```
def Euler_3_poryadok(x0, y0, z0, t0, a, b, n, f1:str, f2:str, f3:str, colResh):
    count = 0
    h = (b-a) / n
    x = x0
    y = y0
    z = z0
    t = t0
    while (t < b) and (count < colResh):
        xBuf = x + h * eval(f1)
        yBuf = y + h * eval(f2)
        zBuf = z + h * eval(f3)
        x = xBuf
        y = yBuf
        z = zBuf
        t += h
        print('t = ', round(t, 5), ' x = ', round(x, 5), ' y = ', round(y, 5), ' z = ', round(z, 5))
        count += 1
```

Результат вывода консоли:

```
[1]Численное интегрирование
[2]Вычисление элементарной функции
[3]Численное решение ДУ
more will coming soon
3
Выберите:
[1] Метод Эйлера
[2] Метод Рунге-Кутты
Для выхода в меню введите '3'
Для возврата нажмите '4'
3
Выбран 'Метод Эйлера для ДУ 3-го порядка'
Введите 1-е ДУ системы
-2*x +5*z
Введите 2-е ДУ системы
sin(t-1)*x-y+3*z
Введите 3-е ДУ системы
-x+2*z
Введите нижнюю границу отрезка
0
Введите верхнюю границу отрезка
0.3
Введите t0
0
Введите X0
2
Введите Y0
1
Введите Z0
1
Введите количество разбиений
10
Введите необходимое количество точек
10
t = 0.03  x = 2.03  y = 1.05895  z = 1.0
t = 0.06  x = 2.0582  y = 1.11615  z = 0.9991
t = 0.09  x = 2.08457  y = 1.17158  z = 0.9973
t = 0.12  x = 2.10909  y = 1.22519  z = 0.9946
t = 0.15  x = 2.13174  y = 1.27698  z = 0.991
t = 0.18  x = 2.15248  y = 1.32691  z = 0.98651
t = 0.21  x = 2.17131  y = 1.37497  z = 0.98113
t = 0.24  x = 2.1882  y = 1.42112  z = 0.97486
t = 0.27  x = 2.20314  y = 1.46536  z = 0.9677
t = 0.3  x = 2.21611  y = 1.50765  z = 0.95967
Выберите:
[1] ДУ 1-го порядка
[2] ДУ 2-го порядка (система из 2-х ДУ первого порядка)
[3] ДУ 3-го порядка (система из 3-х ДУ первого порядка)
Для возврата нажмите '4'
```

P.S: Возникли неполадки с телеграм ботом из-за функции $\sin(t-1)$, мы не смогли понять в чем проблема, поэтому решили контрольный пример 3 показать в консолюке. Однако, если не использовать тригонометрические функции, то данный метод работает в телеграм-боте нормально.

Уменьшив шаг в 10 раз получим:

```

Выберите:
[1] Метод Эйлера
[2] Метод Рунге-Кутты
Для выхода в меню введите '3'
1
Выберите:
[1] ДУ 1-го порядка
[2] ДУ 2-го порядка (система из 2-х ДУ первого порядка)
[3] ДУ 3-го порядка (система из 3-х ДУ первого порядка)
Для возврата нажмите '4'
3
Выбран 'Метод Эйлера для ДУ 3-го порядка'
Введите 1-е ДУ системы
-2*x+5*z
Введите 2-е ДУ системы
sin(t-1)*x - y +3*z
Введите 3-е ДУ системы
-x+2*z
Введите нижнюю границу отрезка
0
Введите верхнюю границу отрезка
0.3
Введите t0
0
Введите X0
2
Введите Y0
1
Введите Z0
1
Введите количество разбиений
100
Введите необходимое количество точек
10
t = 0.003  x = 2.003  y = 1.0061  z = 1.0
t = 0.006  x = 2.00598  y = 1.01219  z = 0.99999
t = 0.009  x = 2.00895  y = 1.01826  z = 0.99997
t = 0.012  x = 2.01189  y = 1.02431  z = 0.99995
t = 0.015  x = 2.01482  y = 1.03034  z = 0.99991
t = 0.018  x = 2.01773  y = 1.03636  z = 0.99987
t = 0.021  x = 2.02062  y = 1.04236  z = 0.99981
t = 0.024  x = 2.02349  y = 1.04834  z = 0.99975
t = 0.027  x = 2.02635  y = 1.0543  z = 0.99968
t = 0.03  x = 2.02919  y = 1.06024  z = 0.9996
Выберите:
[1] ДУ 1-го порядка
[2] ДУ 2-го порядка (система из 2-х ДУ первого порядка)
[3] ДУ 3-го порядка (система из 3-х ДУ первого порядка)
Для возврата нажмите '4'

```

Таблица:

Шаг 0.003	Шаг 0.0003
t= 0.03 x= 2.03 y= 1.05895 z= 1.0	t= 0.003 x= 2.003 y= 1.0061 z= 1.0
t= 0.06 x= 2.0582 y= 1.11615 z= 0.9991	t= 0.006 x= 2.00598 y= 1.01219 z= 0.99999
t= 0.09 x= 2.08457 y= 1.17158 z= 0.9973	t= 0.009 x= 2.00895 y= 1.01826 z= 0.99997
t= 0.12 x= 2.10909 y= 1.22519 z= 0.9946	t= 0.012 x= 2.01189 y= 1.02431 z= 0.99995
t= 0.15 x= 2.13174 y= 1.27698 z= 0.991	t= 0.015 x= 2.01482 y= 1.03034 z= 0.99991
t= 0.18 x= 2.15248 y= 1.32691 z= 0.98651	t= 0.018 x= 2.01773 y= 1.03636 z= 0.99987
t= 0.21 x= 2.17131 y= 1.37497 z= 0.98113	t= 0.021 x= 2.02062 y= 1.04236 z= 0.99981
t= 0.24 x= 2.1882 y= 1.42112 z= 0.97486	t= 0.024 x= 2.02349 y= 1.04834 z= 0.99975
t= 0.27 x= 2.20314 y= 1.46536 z= 0.9677	t= 0.027 x= 2.02635 y= 1.0543 z= 0.99968
t= 0.3 x= 2.21611 y= 1.50765 z= 0.95967	t= 0.03 x= 2.02919 y= 1.06024 z= 0.9996

Ссылка на Телеграм-бота: [Ссылка](#).



Вывод: мы изучили численные методы Эйлера и Рунге-Кутты и предложенные варианты алгоритмов их реализации. Разработали программы решения дифференциальных уравнений, используя язык программирования Python. А также, изучили алгоритмы решения дифференциальных уравнений высших порядков (второго порядка) и системы дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутты.

Лабораторная работа №3

Численные методы решения дифференциальных уравнений

Часть 1

Цель лабораторной работы: разработать программы решения дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутты.

Инструменты: ПК, PyCharm, Telegram.

В рамках данной лабораторной работы, был использован язык программирования Python 3.10

Постановка задачи: изучить численные методы Эйлера и Рунге-Кутты и предложенные варианты алгоритмов их реализации. Разработать программы решения дифференциальных уравнений, используя актуальный для студента язык программирования.

Решить дифференциальное уравнение $y' = y(1 - x)$ на отрезке $[0; 1]$ с начальными условиями $x_0 = 0, y_0 = 1$.

Математическая модель:

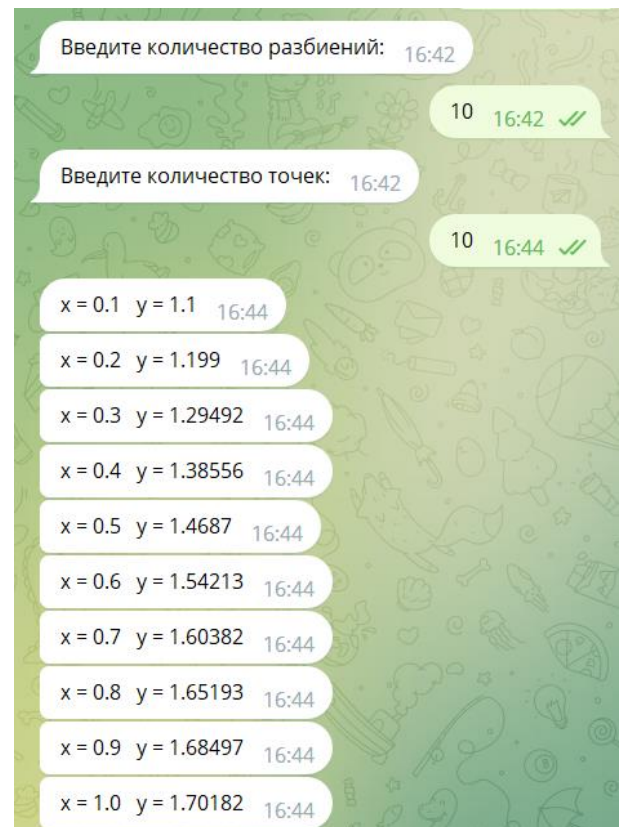
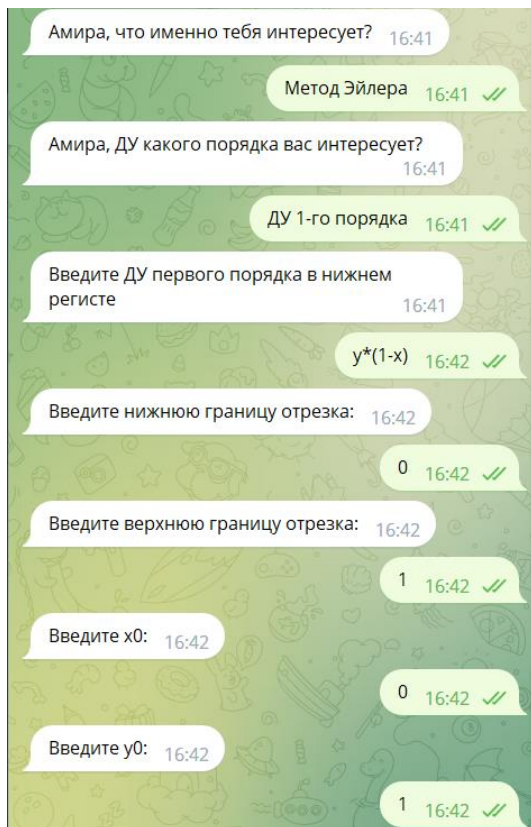
$$y' = y(1 - x)$$

Код:

Метод Эйлера:

```
def Euler_1_poryadok(x0, y0, a, b, n, f:str, colResh):
    count = 0
    h = (b-a) / n
    x = x0
    y = y0
    while (x < b) and (count < colResh):
        y += h * eval(f)
        x += h
        print('x = ', round(x, 5), ' y = ', round(y, 5))
        count += 1
```

Результат работы бота:



Код метода Рунге-Кутты:

```
def Runge_1_poryadok(x0, y0, a, b, n, f:str, colResh):
    count = 0
    h = (b - a) / n
    x = x0
    y = y0
    while (x < b) and (count < colResh):
        xbuffer = x
        ybuffer = y
        k1 = h * eval(f)
        x = xbuffer + h/2
        y = ybuffer + k1/2
        k2 = h * eval(f)
        x = xbuffer + h/2
        y = ybuffer + k2/2
        k3 = h * eval(f)
        x = xbuffer + h
        y = ybuffer + k3
        k4 = h * eval(f)
        F = (k1 + 2 * k2 + 2 * k3 + k4) / 6
        y = ybuffer + F
        x = xbuffer + h
        print('x = ', round(x, 5), ' y = ', round(y, 5))
        count += 1
```

Результат работы бота:

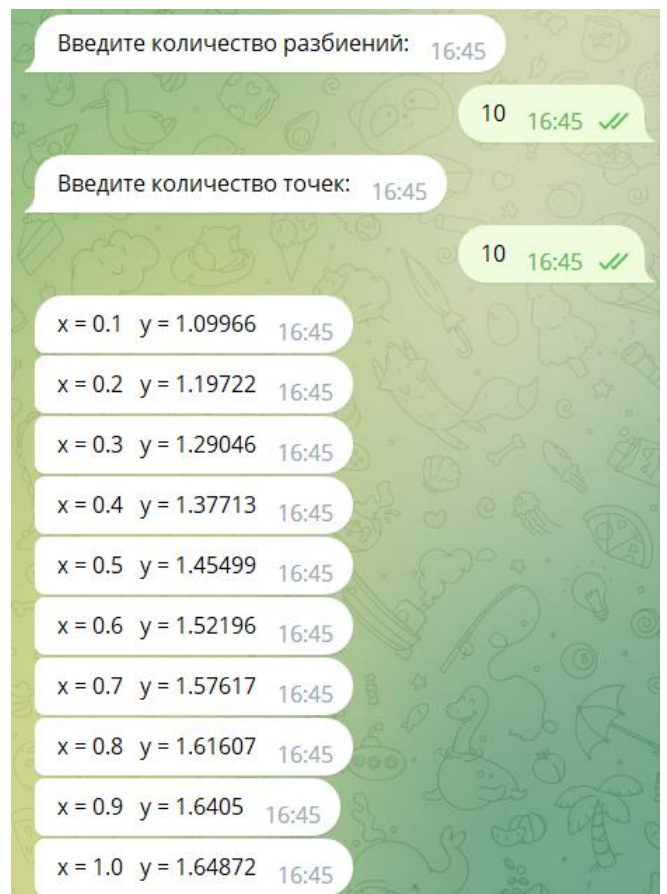
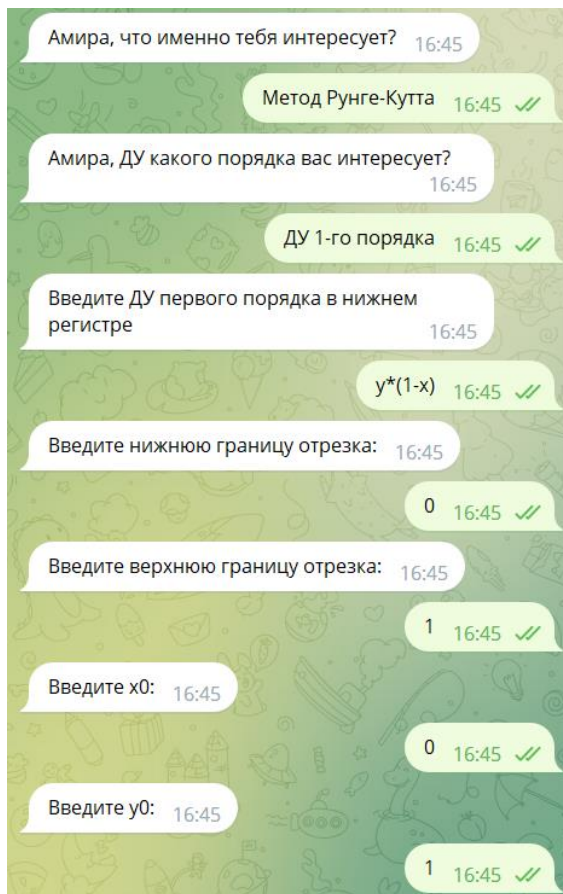


Таблица результатов при разбиении на 10 частей:

$x =$	Метод Эйлера: $y =$	Метод Рунге-Кутты: $y =$
0.1	1.1	1.09966
0.2	1.199	1.19722
0.3	1.29492	1.29046
0.4	1.38556	1.37713
0.5	1.4687	1.45499
0.6	1.54213	1.52196
0.7	1.60382	1.57617
0.8	1.65193	1.61607
0.9	1.68497	1.6405
1.0	1.70182	1.64872

Решим диф.уравнение и построим график с помощью Photomath:

$$y' = y \times (1 - x)$$



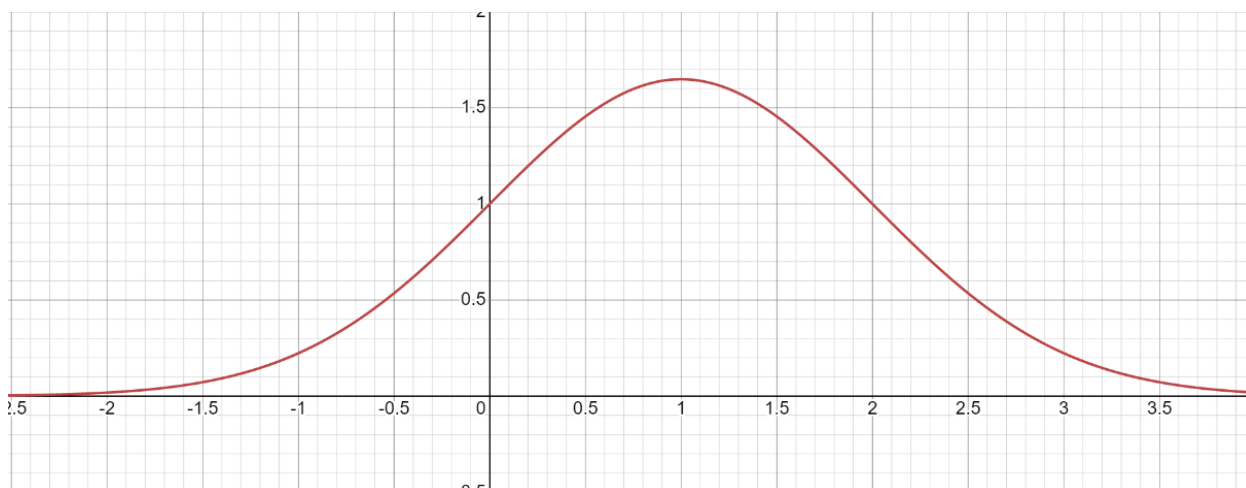
Решите методом разделения переменных

$$y = C e^{x - \frac{x^2}{2}}, C \in \mathbb{R}$$

Подставим y_0 и x_0 чтобы найти C и получим, что $C = 1$;

x	y
0	1
0.1	1.09966
0.5	1.45499
0.8	1.61607
1	1.64872

Получили данный график:



Часть 2

Цель лабораторной работы: разработать программы решения дифференциальных уравнений второго порядка и системы дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутты.

Инструменты: ПК, PyCharm, Telegram.

В рамках данной лабораторной работы, был использован язык программирования Python 3.10

Постановка задачи: изучить алгоритмы решения дифференциальных уравнений высших порядков (второго порядка) и системы дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутты. Разработать программы, используя актуальный для студента язык программирования.

Контрольный пример 2.

Применяя метод Эйлера (Рунге-Кутты) составить на отрезке $[1; 1,5]$ таблицу значений решения уравнения $y'' + y'/x + y = 0$ с начальными условиями:

$$y(1) = 0.77$$

$$y'(1) = -0.44.$$

Шаг вычисления $h = 0.1$.

Математическая модель:
$$y'' + \frac{y'}{x} + y = 0$$

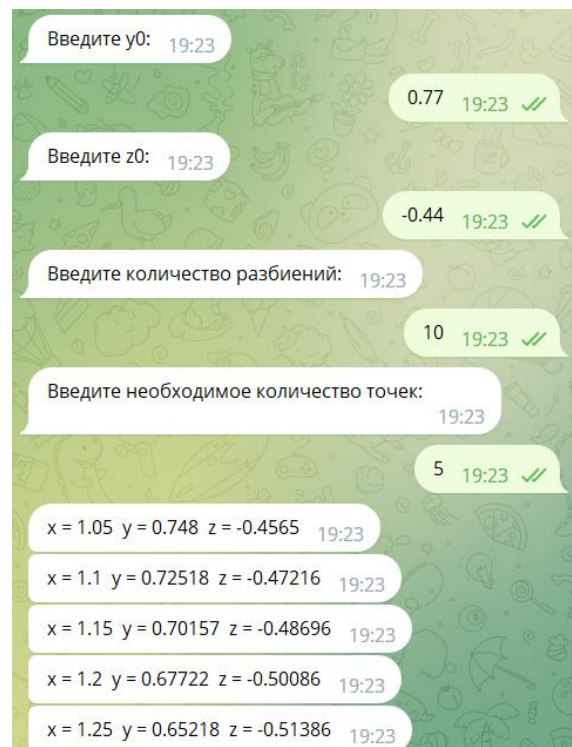
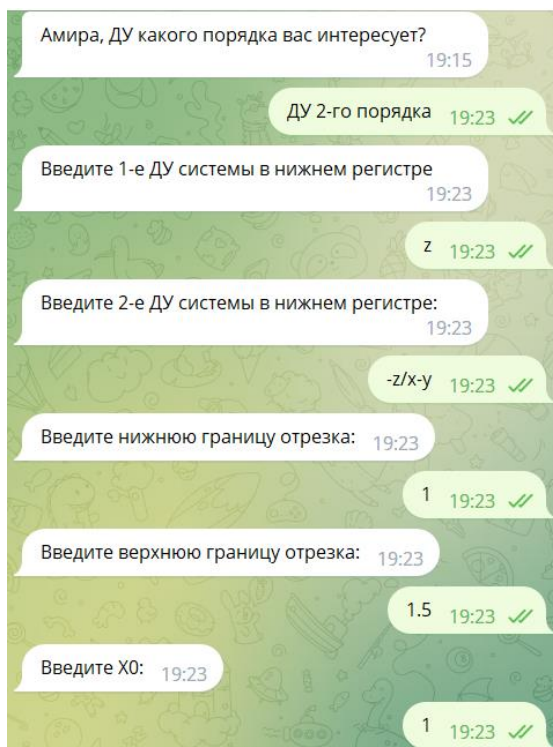
$$y' = z$$

$$z' = -z/x - y$$

Программа для метода Эйлера:

```
def Euler_2_poryadok(x0, y0, z0, a, b, n, f1:str, f2:str, colResh):
    count = 0
    h = (b-a) / n
    x = x0
    y = y0
    z = z0
    while (x < b) and (count < colResh):
        yBuf = y + h * eval(f1)
        zBuf = z + h * eval(f2)
        y = yBuf
        z = zBuf
        x += h
        print('x = ', round(x, 5), ' y = ', round(y, 5), ' z = ', round(z, 5))
        count += 1
```

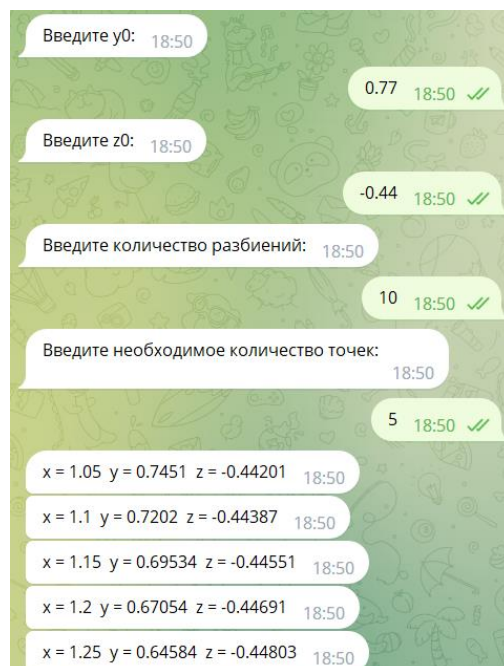
Результат работы бота:



Программа для метода Рунге-Кутты:

```
def Runge_2_poryadok(x0, y0, z0, a, b, n, f1:str, f2:str, colResh):
    count = 0
    h = (b - a) / n
    x = x0
    y = y0
    z = z0
    while (x < b) and (count < colResh):
        xbuffer = x
        ybuffer = y
        zbuffer = z
        q1 = eval(f2)
        k1 = eval(f1)
        x = xbuffer + h/2
        y = ybuffer + k1/2
        z = zbuffer + q1/2
        q2 = eval(f2)
        k2 = eval(f1)
        x = xbuffer + h / 2
        y = ybuffer + k2 / 2
        z = zbuffer + q2 / 2
        q3 = eval(f2)
        k3 = eval(f1)
        x = xbuffer + h
        y = ybuffer + k3
        z = zbuffer + q3
        q4 = eval(f2)
        k4 = eval(f1)
        x = xbuffer + h
        y = ybuffer + h / 6 * (k1 + 2 * k2 + 2 * k3 + k4)
        z = zbuffer + h / 6 * (q1 + 2 * q2 + 2 * q3 + q4)
        print('x = ', round(x, 5), ' y = ', round(y, 5), ' z = ', round(z, 5))
        count += 1
```

Результат работы бота:



Сравнительная таблица:

x=	Метод Эйлера	Метод Рунге-Кутта
1.05	y = 0.748 z = -0.4565	y = 0.7451 z = -0.44201
1.1	y = 0.72518 z = -0.47216	y = 0.7202 z = -0.44387
1.15	y = 0.70157 z = -0.48696	y = 0.69534 z = -0.44551
1.2	y = 0.67722 z = -0.50086	y = 0.67054 z = -0.44691
1.25	y = 0.65218 z = -0.51386	y = 0.64584 z = -0.44803

Контрольный пример 3.

Для системы дифференциальных уравнений

$$dy/dt = -2x + 5z$$

$$dy/dt = \sin(t-1)x - y + 3z$$

$$dz/dt = -x + 2z$$

с начальными условиями $x(0) = 2$, $y(0) = 1$, $z(0) = 1$ составить таблицу значений функций $x(t)$, $y(t)$, $z(t)$ на отрезке $[0; 0.3]$ с шагом $h = 0.003$. Использовать метод Эйлера.

Код программы:

```
def Euler_3_poryadok(x0, y0, z0, t0, a, b, n, f1:str, f2:str, f3:str, colResh):
    count = 0
    h = (b-a) / n
    x = x0
    y = y0
    z = z0
    t = t0
    while (t < b) and (count < colResh):
        xBuf = x + h * eval(f1)
        yBuf = y + h * eval(f2)
        zBuf = z + h * eval(f3)
        x = xBuf
        y = yBuf
        z = zBuf
        t += h
        print('t = ', round(t, 5), ' x = ', round(x, 5), ' y = ', round(y, 5), ' z = ', round(z, 5))
        count += 1
```

Результат вывода консоли:

```
[1]Численное интегрирование
[2]Вычисление элементарной функции
[3]Численное решение ДУ
more will coming soon
3
Выберите:
[1] Метод Эйлера
[2] Метод Рунге-Кутты
Для выхода в меню введите '3'
Для возврата нажмите '4'
3
Выбран 'Метод Эйлера для ДУ 3-го порядка'
Введите 1-е ДУ системы
-2*x +5*z
Введите 2-е ДУ системы
sin(t-1)*x-y+3*z
Введите 3-е ДУ системы
-x+2*z
Введите нижнюю границу отрезка
0
Введите верхнюю границу отрезка
0.3
Введите t0
0
Введите X0
2
Введите Y0
1
Введите Z0
1
Введите количество разбиений
10
Введите необходимое количество точек
10
t = 0.03  x = 2.03  y = 1.05895  z = 1.0
t = 0.06  x = 2.0582  y = 1.11615  z = 0.9991
t = 0.09  x = 2.08457  y = 1.17158  z = 0.9973
t = 0.12  x = 2.10909  y = 1.22519  z = 0.9946
t = 0.15  x = 2.13174  y = 1.27698  z = 0.991
t = 0.18  x = 2.15248  y = 1.32691  z = 0.98651
t = 0.21  x = 2.17131  y = 1.37497  z = 0.98113
t = 0.24  x = 2.1882  y = 1.42112  z = 0.97486
t = 0.27  x = 2.20314  y = 1.46536  z = 0.9677
t = 0.3  x = 2.21611  y = 1.50765  z = 0.95967
Выберите:
[1] ДУ 1-го порядка
[2] ДУ 2-го порядка (система из 2-х ДУ первого порядка)
[3] ДУ 3-го порядка (система из 3-х ДУ первого порядка)
Для возврата нажмите '4'
```

P.S: Возникли неполадки с телеграм ботом из-за функции $\sin(t-1)$, мы не смогли понять в чем проблема, поэтому решили контрольный пример 3 показать в консолюке. Однако, если не использовать тригонометрические функции, то данный метод работает в телеграм-боте нормально.

Уменьшив шаг в 10 раз получим:

```

Выберите:
[1] Метод Эйлера
[2] Метод Рунге-Кутты
Для выхода в меню введите '3'
1
Выберите:
[1] ДУ 1-го порядка
[2] ДУ 2-го порядка (система из 2-х ДУ первого порядка)
[3] ДУ 3-го порядка (система из 3-х ДУ первого порядка)
Для возврата нажмите '4'
3
Выбран 'Метод Эйлера для ДУ 3-го порядка'
Введите 1-е ДУ системы
-2*x+5*z
Введите 2-е ДУ системы
sin(t-1)*x - y +3*z
Введите 3-е ДУ системы
-x+2*z
Введите нижнюю границу отрезка
0
Введите верхнюю границу отрезка
0.3
Введите t0
0
Введите X0
2
Введите Y0
1
Введите Z0
1
Введите количество разбиений
100
Введите необходимое количество точек
10
t = 0.003  x = 2.003  y = 1.0061  z = 1.0
t = 0.006  x = 2.00598  y = 1.01219  z = 0.99999
t = 0.009  x = 2.00895  y = 1.01826  z = 0.99997
t = 0.012  x = 2.01189  y = 1.02431  z = 0.99995
t = 0.015  x = 2.01482  y = 1.03034  z = 0.99991
t = 0.018  x = 2.01773  y = 1.03636  z = 0.99987
t = 0.021  x = 2.02062  y = 1.04236  z = 0.99981
t = 0.024  x = 2.02349  y = 1.04834  z = 0.99975
t = 0.027  x = 2.02635  y = 1.0543  z = 0.99968
t = 0.03  x = 2.02919  y = 1.06024  z = 0.9996
Выберите:
[1] ДУ 1-го порядка
[2] ДУ 2-го порядка (система из 2-х ДУ первого порядка)
[3] ДУ 3-го порядка (система из 3-х ДУ первого порядка)
Для возврата нажмите '4'

```

Таблица:

Шаг 0.003	Шаг 0.0003
t= 0.03 x= 2.03 y= 1.05895 z= 1.0	t= 0.003 x= 2.003 y= 1.0061 z= 1.0
t= 0.06 x= 2.0582 y= 1.11615 z= 0.9991	t= 0.006 x= 2.00598 y= 1.01219 z= 0.99999
t= 0.09 x= 2.08457 y= 1.17158 z= 0.9973	t= 0.009 x= 2.00895 y= 1.01826 z= 0.99997
t= 0.12 x= 2.10909 y= 1.22519 z= 0.9946	t= 0.012 x= 2.01189 y= 1.02431 z= 0.99995
t= 0.15 x= 2.13174 y= 1.27698 z= 0.991	t= 0.015 x= 2.01482 y= 1.03034 z= 0.99991
t= 0.18 x= 2.15248 y= 1.32691 z= 0.98651	t= 0.018 x= 2.01773 y= 1.03636 z= 0.99987
t= 0.21 x= 2.17131 y= 1.37497 z= 0.98113	t= 0.021 x= 2.02062 y= 1.04236 z= 0.99981
t= 0.24 x= 2.1882 y= 1.42112 z= 0.97486	t= 0.024 x= 2.02349 y= 1.04834 z= 0.99975
t= 0.27 x= 2.20314 y= 1.46536 z= 0.9677	t= 0.027 x= 2.02635 y= 1.0543 z= 0.99968
t= 0.3 x= 2.21611 y= 1.50765 z= 0.95967	t= 0.03 x= 2.02919 y= 1.06024 z= 0.9996

Ссылка на Телеграм-бота: [Ссылка](#).



Вывод: мы изучили численные методы Эйлера и Рунге-Кутта и предложенные варианты алгоритмов их реализации. Разработали программы решения дифференциальных уравнений, используя язык программирования Python. А также, изучили алгоритмы решения дифференциальных уравнений высших порядков (второго порядка) и системы дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутта.

Лабораторная работа №3

Численные методы решения дифференциальных уравнений

Часть 1

Цель лабораторной работы: разработать программы решения дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутты.

Инструменты: ПК, PyCharm, Telegram.

В рамках данной лабораторной работы, был использован язык программирования Python 3.10

Постановка задачи: изучить численные методы Эйлера и Рунге-Кутты и предложенные варианты алгоритмов их реализации. Разработать программы решения дифференциальных уравнений, используя актуальный для студента язык программирования.

Решить дифференциальное уравнение $y' = y(1 - x)$ на отрезке $[0; 1]$ с начальными условиями $x_0 = 0, y_0 = 1$.

Математическая модель:

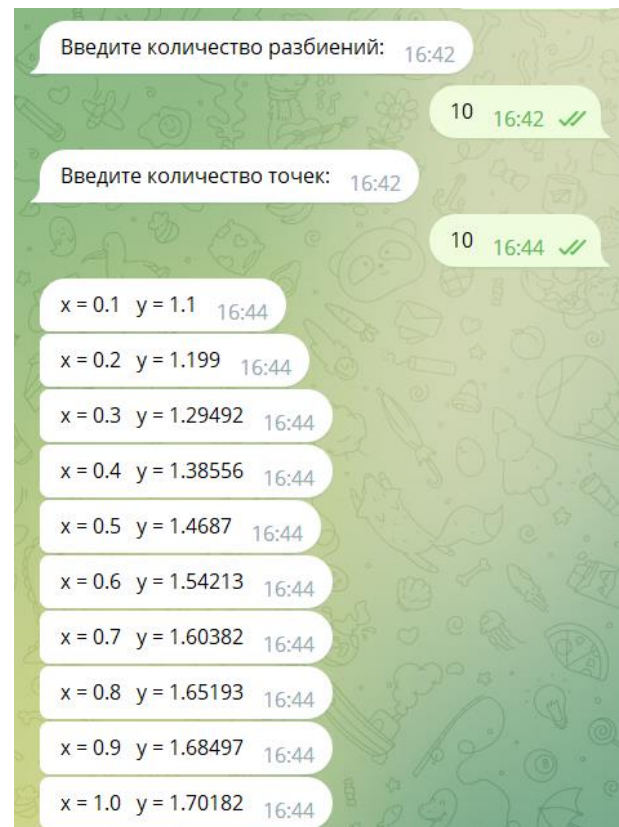
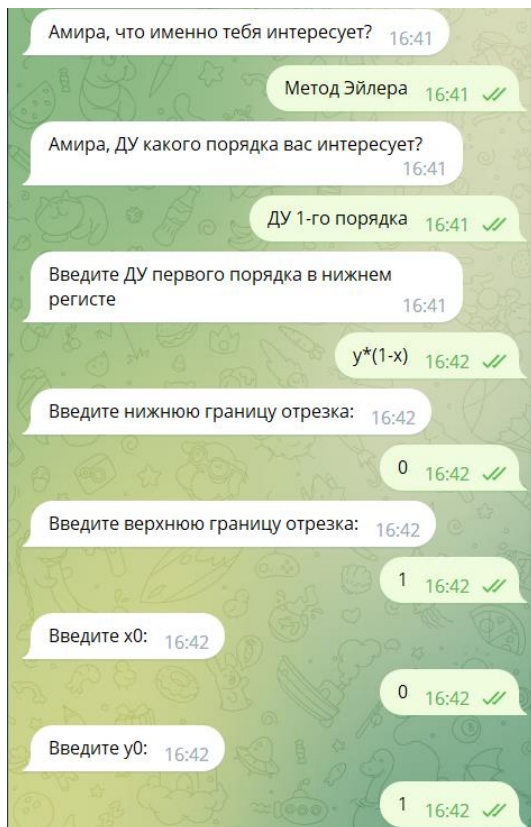
$$y' = y(1 - x)$$

Код:

Метод Эйлера:

```
def Euler_1_poryadok(x0, y0, a, b, n, f:str, colResh):
    count = 0
    h = (b-a) / n
    x = x0
    y = y0
    while (x < b) and (count < colResh):
        y += h * eval(f)
        x += h
        print('x = ', round(x, 5), ' y = ', round(y, 5))
        count += 1
```

Результат работы бота:



Код метода Рунге-Кутты:

```
def Runge_1_poryadok(x0, y0, a, b, n, f:str, colResh):
    count = 0
    h = (b - a) / n
    x = x0
    y = y0
    while (x < b) and (count < colResh):
        xbuffer = x
        ybuffer = y
        k1 = h * eval(f)
        x = xbuffer + h/2
        y = ybuffer + k1/2
        k2 = h * eval(f)
        x = xbuffer + h/2
        y = ybuffer + k2/2
        k3 = h * eval(f)
        x = xbuffer + h
        y = ybuffer + k3
        k4 = h * eval(f)
        F = (k1 + 2 * k2 + 2 * k3 + k4) / 6
        y = ybuffer + F
        x = xbuffer + h
        print('x = ', round(x, 5), ' y = ', round(y, 5))
        count += 1
```

Результат работы бота:

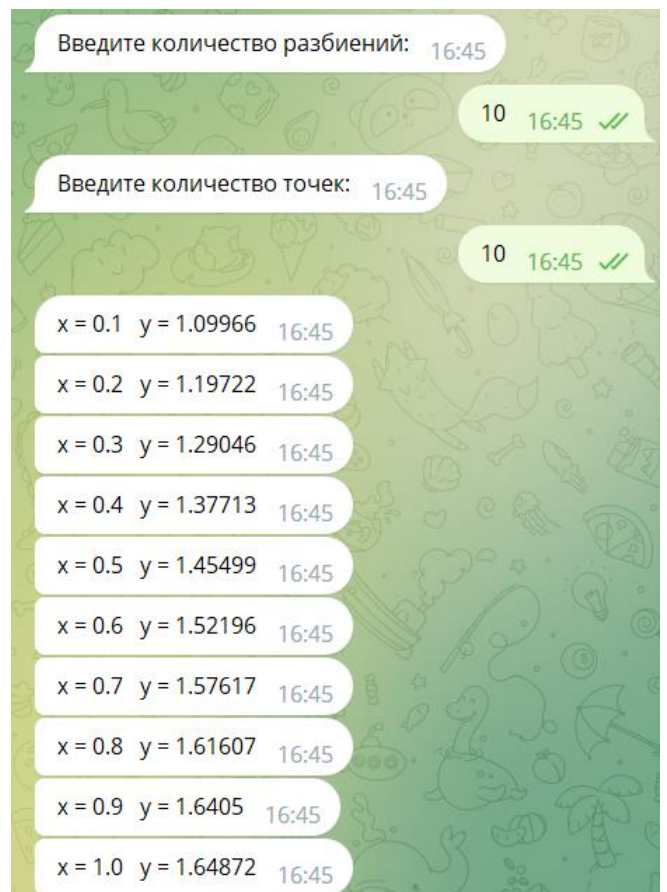
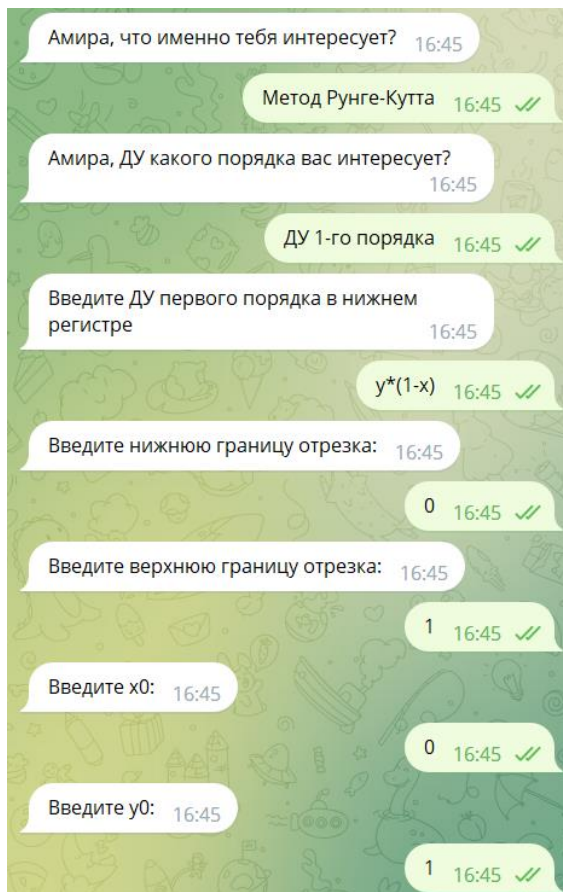


Таблица результатов при разбиении на 10 частей:

$x =$	Метод Эйлера: $y =$	Метод Рунге-Кутта: $y =$
0.1	1.1	1.09966
0.2	1.199	1.19722
0.3	1.29492	1.29046
0.4	1.38556	1.37713
0.5	1.4687	1.45499
0.6	1.54213	1.52196
0.7	1.60382	1.57617
0.8	1.65193	1.61607
0.9	1.68497	1.6405
1.0	1.70182	1.64872

Решим диф.уравнение и построим график с помощью Photomath:

$$y' = y \times (1 - x)$$



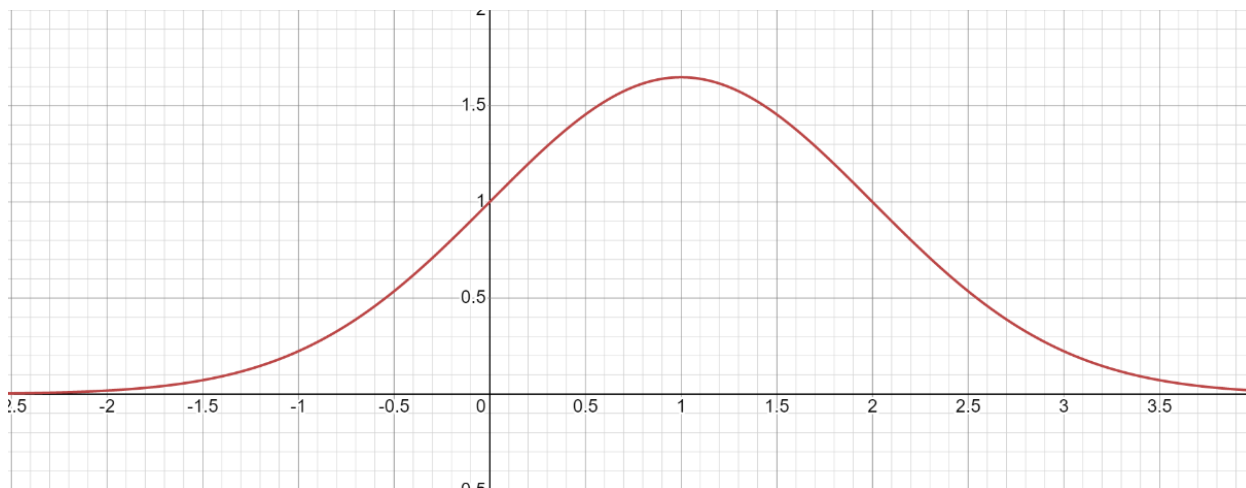
Решите методом разделения переменных

$$y = C e^{x - \frac{x^2}{2}}, C \in \mathbb{R}$$

Подставим y_0 и x_0 чтобы найти C и получим, что $C = 1$;

x	y
0	1
0.1	1.09966
0.5	1.45499
0.8	1.61607
1	1.64872

Получили данный график:



Часть 2

Цель лабораторной работы: разработать программы решения дифференциальных уравнений второго порядка и системы дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутты.

Инструменты: ПК, PyCharm, Telegram.

В рамках данной лабораторной работы, был использован язык программирования Python 3.10

Постановка задачи: изучить алгоритмы решения дифференциальных уравнений высших порядков (второго порядка) и системы дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутты. Разработать программы, используя актуальный для студента язык программирования.

Контрольный пример 2.

Применяя метод Эйлера (Рунге-Кутты) составить на отрезке $[1; 1,5]$ таблицу значений решения уравнения $y'' + y'/x + y = 0$ с начальными условиями:

$$y(1) = 0.77$$

$$y'(1) = -0.44.$$

Шаг вычисления $h = 0.1$.

Математическая модель:
$$y'' + \frac{y'}{x} + y = 0$$

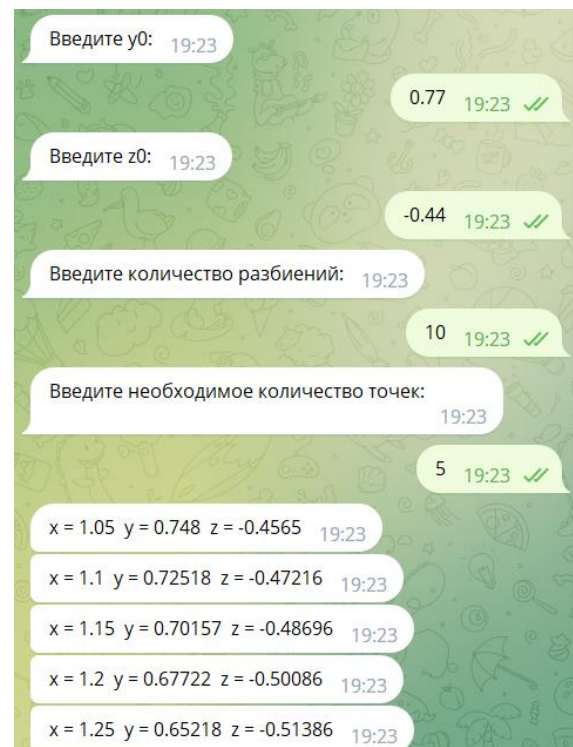
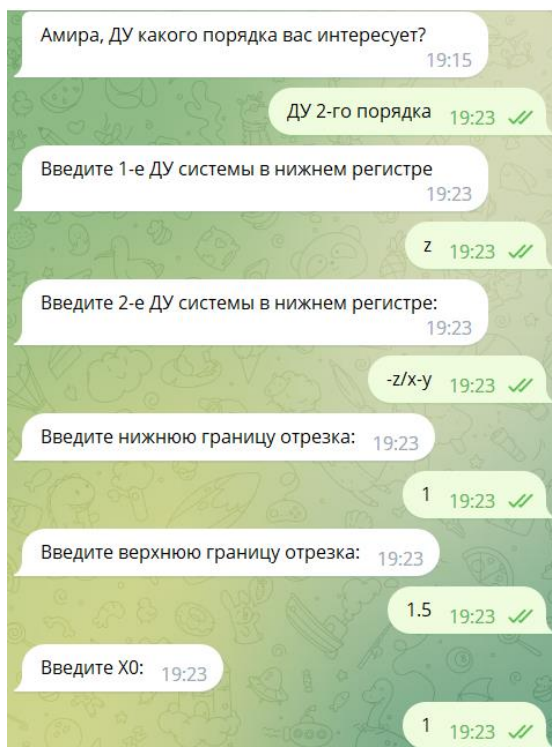
$$y' = z$$

$$z' = -z/x - y$$

Программа для метода Эйлера:

```
def Euler_2_poryadok(x0, y0, z0, a, b, n, f1:str, f2:str, colResh):
    count = 0
    h = (b-a) / n
    x = x0
    y = y0
    z = z0
    while (x < b) and (count < colResh):
        yBuf = y + h * eval(f1)
        zBuf = z + h * eval(f2)
        y = yBuf
        z = zBuf
        x += h
        print('x = ', round(x, 5), ' y = ', round(y, 5), ' z = ', round(z, 5))
        count += 1
```

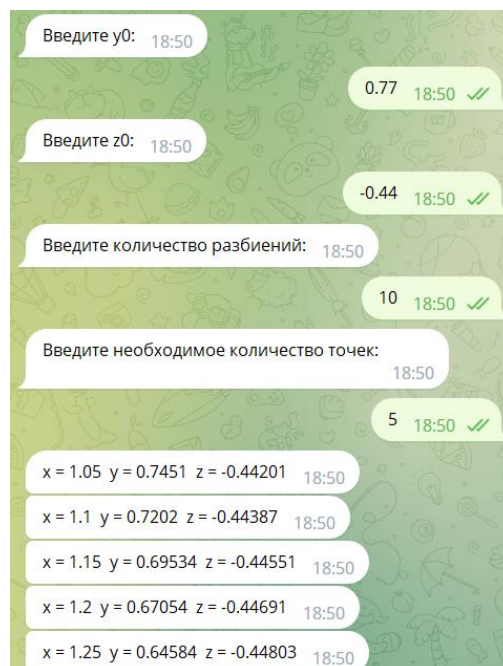
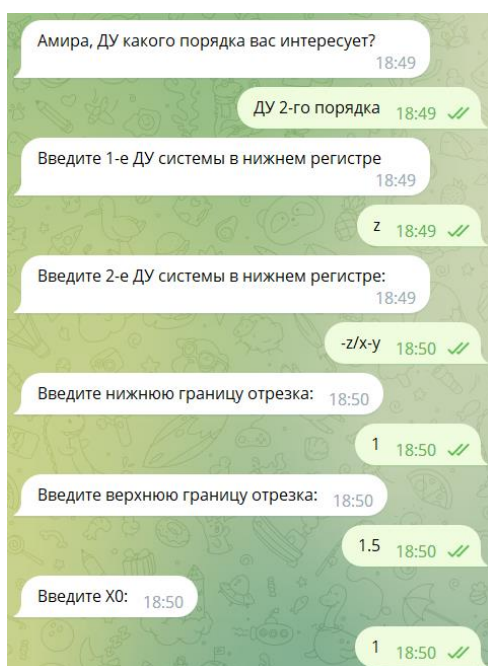
Результат работы бота:



Программа для метода Рунге-Кутты:


```
def Runge_2_poryadok(x0, y0, z0, a, b, n, f1:str, f2:str, colResh):
    count = 0
    h = (b - a) / n
    x = x0
    y = y0
    z = z0
    while (x < b) and (count < colResh):
        xbuffer = x
        ybuffer = y
        zbuffer = z
        q1 = eval(f2)
        k1 = eval(f1)
        x = xbuffer + h/2
        y = ybuffer + k1/2
        z = zbuffer + q1/2
        q2 = eval(f2)
        k2 = eval(f1)
        x = xbuffer + h / 2
        y = ybuffer + k2 / 2
        z = zbuffer + q2 / 2
        q3 = eval(f2)
        k3 = eval(f1)
        x = xbuffer + h
        y = ybuffer + k3
        z = zbuffer + q3
        q4 = eval(f2)
        k4 = eval(f1)
        x = xbuffer + h
        y = ybuffer + h / 6 * (k1 + 2 * k2 + 2 * k3 + k4)
        z = zbuffer + h / 6 * (q1 + 2 * q2 + 2 * q3 + q4)
        print('x = ', round(x, 5), ' y = ', round(y, 5), ' z = ', round(z, 5))
        count += 1
```

Результат работы бота:



Сравнительная таблица:

x=	Метод Эйлера	Метод Рунге-Кутта
1.05	y = 0.748 z = -0.4565	y = 0.7451 z = -0.44201
1.1	y = 0.72518 z = -0.47216	y = 0.7202 z = -0.44387
1.15	y = 0.70157 z = -0.48696	y = 0.69534 z = -0.44551
1.2	y = 0.67722 z = -0.50086	y = 0.67054 z = -0.44691
1.25	y = 0.65218 z = -0.51386	y = 0.64584 z = -0.44803

Контрольный пример 3.

Для системы дифференциальных уравнений

$$dy/dt = -2x + 5z$$

$$dy/dt = \sin(t-1)x - y + 3z$$

$$dz/dt = -x + 2z$$

с начальными условиями $x(0) = 2$, $y(0) = 1$, $z(0) = 1$ составить таблицу значений функций $x(t)$, $y(t)$, $z(t)$ на отрезке $[0; 0.3]$ с шагом $h = 0.003$. Использовать метод Эйлера.

Код программы:

```
def Euler_3_poryadok(x0, y0, z0, t0, a, b, n, f1:str, f2:str, f3:str, colResh):
    count = 0
    h = (b-a) / n
    x = x0
    y = y0
    z = z0
    t = t0
    while (t < b) and (count < colResh):
        xBuf = x + h * eval(f1)
        yBuf = y + h * eval(f2)
        zBuf = z + h * eval(f3)
        x = xBuf
        y = yBuf
        z = zBuf
        t += h
        print('t = ', round(t, 5), ' x = ', round(x, 5), ' y = ', round(y, 5), ' z = ', round(z, 5))
        count += 1
```

Результат вывода консоли:

```
[1]Численное интегрирование
[2]Вычисление элементарной функции
[3]Численное решение ДУ
more will coming soon
3
Выберите:
[1] Метод Эйлера
[2] Метод Рунге-Кутты
Для выхода в меню введите '3'
Для возврата нажмите '4'
3
Выбран 'Метод Эйлера для ДУ 3-го порядка'
Введите 1-е ДУ системы
-2*x +5*z
Введите 2-е ДУ системы
sin(t-1)*x-y+3*z
Введите 3-е ДУ системы
-x+2*z
Введите нижнюю границу отрезка
0
Введите верхнюю границу отрезка
0.3
Введите t0
0
Введите X0
2
Введите Y0
1
Введите Z0
1
Введите количество разбиений
10
Введите необходимое количество точек
10
t = 0.03  x = 2.03  y = 1.05895  z = 1.0
t = 0.06  x = 2.0582  y = 1.11615  z = 0.9991
t = 0.09  x = 2.08457  y = 1.17158  z = 0.9973
t = 0.12  x = 2.10909  y = 1.22519  z = 0.9946
t = 0.15  x = 2.13174  y = 1.27698  z = 0.991
t = 0.18  x = 2.15248  y = 1.32691  z = 0.98651
t = 0.21  x = 2.17131  y = 1.37497  z = 0.98113
t = 0.24  x = 2.1882  y = 1.42112  z = 0.97486
t = 0.27  x = 2.20314  y = 1.46536  z = 0.9677
t = 0.3  x = 2.21611  y = 1.50765  z = 0.95967
Выберите:
[1] ДУ 1-го порядка
[2] ДУ 2-го порядка (система из 2-х ДУ первого порядка)
[3] ДУ 3-го порядка (система из 3-х ДУ первого порядка)
Для возврата нажмите '4'
```

P.S: Возникли неполадки с телеграм ботом из-за функции $\sin(t-1)$, мы не смогли понять в чем проблема, поэтому решили контрольный пример 3 показать в консолюке. Однако, если не использовать тригонометрические функции, то данный метод работает в телеграм-боте нормально.

Уменьшив шаг в 10 раз получим:

```

Выберите:
[1] Метод Эйлера
[2] Метод Рунге-Кутты
Для выхода в меню введите '3'
1
Выберите:
[1] ДУ 1-го порядка
[2] ДУ 2-го порядка (система из 2-х ДУ первого порядка)
[3] ДУ 3-го порядка (система из 3-х ДУ первого порядка)
Для возврата нажмите '4'
3
Выбран 'Метод Эйлера для ДУ 3-го порядка'
Введите 1-е ДУ системы
-2*x+5*z
Введите 2-е ДУ системы
sin(t-1)*x - y +3*z
Введите 3-е ДУ системы
-x+2*z
Введите нижнюю границу отрезка
0
Введите верхнюю границу отрезка
0.3
Введите t0
0
Введите X0
2
Введите Y0
1
Введите Z0
1
Введите количество разбиений
100
Введите необходимое количество точек
10
t = 0.003  x = 2.003  y = 1.0061  z = 1.0
t = 0.006  x = 2.00598  y = 1.01219  z = 0.99999
t = 0.009  x = 2.00895  y = 1.01826  z = 0.99997
t = 0.012  x = 2.01189  y = 1.02431  z = 0.99995
t = 0.015  x = 2.01482  y = 1.03034  z = 0.99991
t = 0.018  x = 2.01773  y = 1.03636  z = 0.99987
t = 0.021  x = 2.02062  y = 1.04236  z = 0.99981
t = 0.024  x = 2.02349  y = 1.04834  z = 0.99975
t = 0.027  x = 2.02635  y = 1.0543  z = 0.99968
t = 0.03  x = 2.02919  y = 1.06024  z = 0.9996
Выберите:
[1] ДУ 1-го порядка
[2] ДУ 2-го порядка (система из 2-х ДУ первого порядка)
[3] ДУ 3-го порядка (система из 3-х ДУ первого порядка)
Для возврата нажмите '4'

```

Таблица:

Шаг 0.003	Шаг 0.0003
t= 0.03 x= 2.03 y= 1.05895 z= 1.0	t= 0.003 x= 2.003 y= 1.0061 z= 1.0
t= 0.06 x= 2.0582 y= 1.11615 z= 0.9991	t= 0.006 x= 2.00598 y= 1.01219 z= 0.99999
t= 0.09 x= 2.08457 y= 1.17158 z= 0.9973	t= 0.009 x= 2.00895 y= 1.01826 z= 0.99997
t= 0.12 x= 2.10909 y= 1.22519 z= 0.9946	t= 0.012 x= 2.01189 y= 1.02431 z= 0.99995
t= 0.15 x= 2.13174 y= 1.27698 z= 0.991	t= 0.015 x= 2.01482 y= 1.03034 z= 0.99991
t= 0.18 x= 2.15248 y= 1.32691 z= 0.98651	t= 0.018 x= 2.01773 y= 1.03636 z= 0.99987
t= 0.21 x= 2.17131 y= 1.37497 z= 0.98113	t= 0.021 x= 2.02062 y= 1.04236 z= 0.99981
t= 0.24 x= 2.1882 y= 1.42112 z= 0.97486	t= 0.024 x= 2.02349 y= 1.04834 z= 0.99975
t= 0.27 x= 2.20314 y= 1.46536 z= 0.9677	t= 0.027 x= 2.02635 y= 1.0543 z= 0.99968
t= 0.3 x= 2.21611 y= 1.50765 z= 0.95967	t= 0.03 x= 2.02919 y= 1.06024 z= 0.9996

Ссылка на Телеграм-бота: [Ссылка](#).



Вывод: мы изучили численные методы Эйлера и Рунге-Кутта и предложенные варианты алгоритмов их реализации. Разработали программы решения дифференциальных уравнений, используя язык программирования Python. А также, изучили алгоритмы решения дифференциальных уравнений высших порядков (второго порядка) и системы дифференциальных уравнений с использованием численных методов Эйлера и Рунге-Кутта.