# Лабораторная работа №8 по теме: «Приближенные методы решения задач математической физики»

## Подготовил студент Михайлов Денис

## группы Б8117-02.03.01

**Постановка задачи:**

Получить решение краевой задачи дифференциального уравнения методом коллокаций и методом Ритца, а также сравнить методы.

**Дано:**

С краевыми условиями:

В качестве базисных функций, удовлетворяющие краевым условиям, возьмем следующие полиномы:

* ,

**Ход решения:**

Ограничиваясь тремя базисными функциями, положим:

Далее, для нахождения решения (2) используется один из нескольких методов: метод коллокаций, метод Ритца, метод Галеркина и метод конечных элементов. Рассмотрим метод коллокации и метод Ритца.

**Метод коллокации.**

За точки коллокации возьмем точки .

Продифференцируем дважды уравнение (2) и осуществим подстановку в дифференциальное уравнение (1), получив уравнение В точках коллокации имеем . Отсюда, получаем для определения коэффициентов и линейную систему уравнений.

Решаем эту систему уравнений и получаем следующее:

Тем самым, получив решение дифференциального уравнения (1):

**Метод Ритца.**

В методе Ритца в общем случае линейное дифференциальное уравнение имеет вид:

Где:

Следовательно:

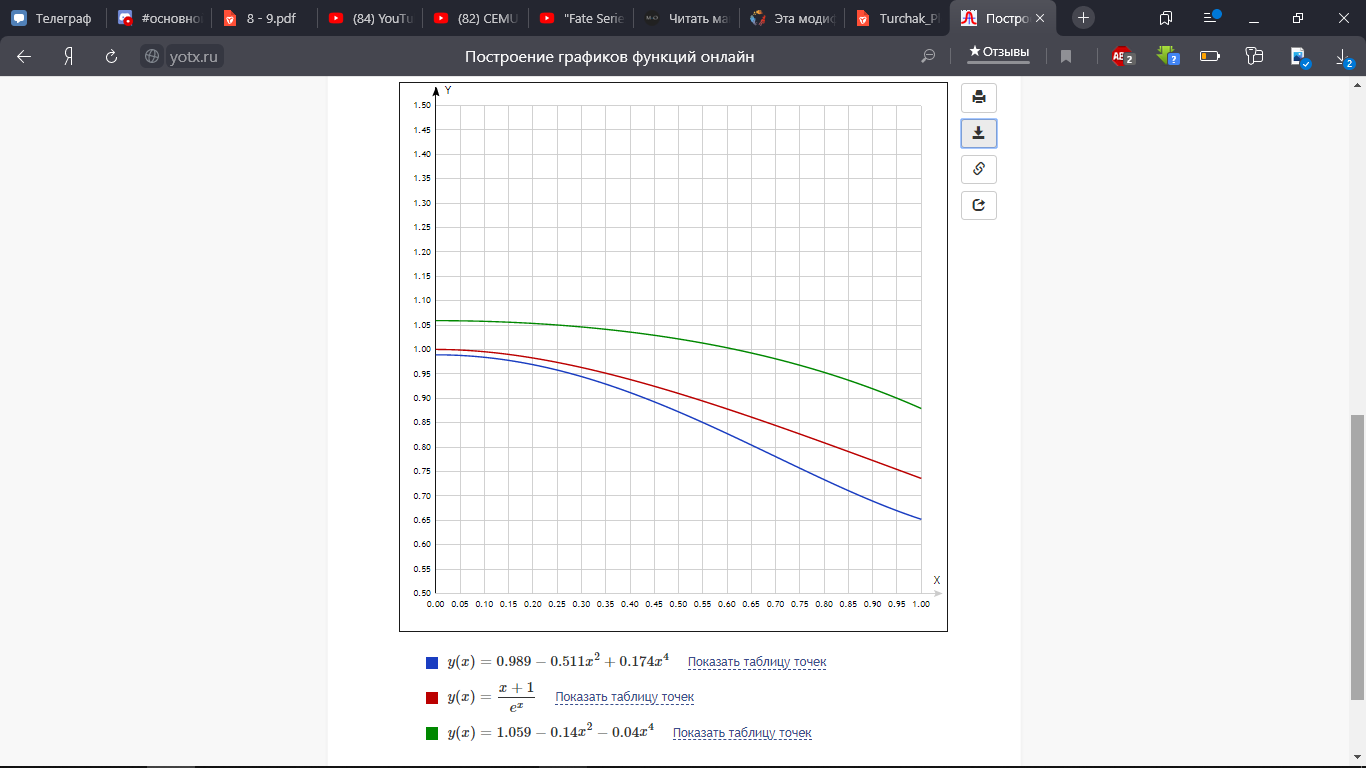
Составляем для уравнения (3) соответствующий функционал:

Продифференцируем уравнение (3) и подставим значения в (6). Найдя частные производные и , где квадратичная функция переменных , проинтегрируем по . Отсюда, получаем для определения коэффициентов и линейную систему уравнений.

Решаем эту систему уравнений и получаем следующее:

Тем самым, получив решение дифференциального уравнения (1):

Сравним полученные решения (4) и (7) с точным решением :



Как можно видно из графика, метод коллокаций обеспечивает большую точность, чем метод Ритца.

**Код программы**

from sympy import symbols, exp, diff, integrate

from sympy.solvers import solve

from numpy import zeros

def Collocation():

    x, y, z = symbols('x y z')

    u = [y\*(5 - x\*\*2)+z\*(9 - x\*\*4)]

    form = u[0]

    for \_ in range(2):

        form = diff(form, x)

        u.append(form)

    func = (u[2])-(u[1])-2\*(u[0])+3\*exp(-x)

    ans = [func.subs(x, 0.1), func.subs(x, 0.5)]

    return solve(ans, [y, z])

def Ritz():

    x, y, z = symbols('x y z')

    u = [y\*(5 - x\*\*2)+z\*(9 - x\*\*4), diff(y\*(5 - x\*\*2)+z\*(9 - x\*\*4), x)]

    F = -3\*exp(-x)

    Q = -2

    P = -1

    px = exp(integrate(P, (x, 0, x)))

    qx = px \* Q

    fx = px \* F

    J = (px)\*(u[1])\*\*2 - (qx)\*(u[0])\*\*2 + 2\*fx\*u[0]

    ans = []

    ans.append(integrate(diff(J, y), (x, 0, 1)))

    ans.append(integrate(diff(J, z), (x, 0, 1)))

    return solve(ans, [y, z])