МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
(НИЯУ МИФИ)
Институт Финансовых Технологий и Экономической Безопасности
Кафедра Финансового Мониторинга

Лабораторная работа №2:
По курсу «Численные методы»

Работу выполнил: студент группы С18-712: Кольца И. В.
Проверил: Саманчук В.Н.

Москва 2020

Постановка задачи

2 Решить систему нелинейных уравнений методом Ньютона:

$$\begin{cases} \sin(x) = y + 1{,}32 \\ \cos(y) = x - 0{,}85 \end{cases}$$

Методика решения

Для решения поставленной задачи была написана программа на языке Python, в которой реализован метод Ньютона для решения системы нелинейных уравнений.

Теоретическая справка

Идея метода Ньютона для приближенного решения системы (2) заключается в следующем: имея некоторое приближение $\boldsymbol{x}^{(k)}$, мы находим следующее приближение $\boldsymbol{x}^{(k+1)}$, аппроксимируя $\boldsymbol{F}(\boldsymbol{x}^{(k+1)})$ линейным оператором и решая систему линейных алгебраических уравнений. Аппроксимируем нелинейную задачу $\boldsymbol{F}(\boldsymbol{x}^{(k+1)}) = 0$ линейной

$$\boldsymbol{F}(\boldsymbol{x}^{(k)}) + \boldsymbol{J}(\boldsymbol{x}^{(k)})(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}) = 0, \qquad (5)$$

где $\boldsymbol{J}(\boldsymbol{x}^{(k)})$ — *матрица Якоби (якобиан)*:

$$\nabla \boldsymbol{F}(\boldsymbol{x}^{(k)}) = \begin{bmatrix} \frac{\partial f_1(\boldsymbol{x}^{(k)})}{\partial x_1} & \frac{\partial f_1(\boldsymbol{x}^{(k)})}{\partial x_2} & \cdots & \frac{\partial f_1(\boldsymbol{x}^{(k)})}{\partial x_n} \\ \frac{\partial f_2(\boldsymbol{x}^{(k)})}{\partial x_1} & \frac{\partial f_2(\boldsymbol{x}^{(k)})}{\partial x_2} & \cdots & \frac{\partial f_2(\boldsymbol{x}^{(k)})}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_n(\boldsymbol{x}^{(k)})}{\partial x_1} & \frac{\partial f_n(\boldsymbol{x}^{(k)})}{\partial x_2} & \cdots & \frac{\partial f_n(\boldsymbol{x}^{(k)})}{\partial x_n} \end{bmatrix}$$

Уравнение (5) является линейной системой с матрицей коэффициентов $\boldsymbol{J}$ и вектором правой части $-\boldsymbol{F}(\boldsymbol{x}^{(k)})$. Систему можно переписать в виде

$$\boldsymbol{J}(\boldsymbol{x}^{(k)})\boldsymbol{\delta} = -\boldsymbol{F}(\boldsymbol{x}^{(k)}),$$

где $\boldsymbol{\delta} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$.

Таким образом, k-я итерация метода Ньютона состоит из двух стадий:

1. Решается система линейных уравнений (СЛАУ) $\boldsymbol{J}(\boldsymbol{x}^{(k)})\boldsymbol{\delta} = -\boldsymbol{F}(\boldsymbol{x}^{(k)})$ относительно $\boldsymbol{\delta}$.
2. Находится значение вектора на следующей итерации $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{\delta}$.

Решение задачи

```
import numpy as np

def J(x):
    return np.array([[np.cos(x[0]), -1], [-1, -np.sin(x[1])]])

def f(x):
    return np.array([np.sin(x[0]) - x[1] - 1.32, np.cos(x[1]) - x[0] + 0.85])

eps = 10**(-5)

k = 0
```

```
d_x = 1
x = np.array([7, 5])

while (np.linalg.norm(d_x) > eps):
    k += 1
    print(f'Итерация №{k}', end='; ')
    d_x = np.linalg.solve(J(x), -f(x))
    x = x + d_x
    print(f'x = {x}; f(x) = {f(x)}')

print(f'Ответ: x = {x[0]}; y = {x[1]}')
```

Результат работы

```
Итерация №1; x = [-33.77290839 -31.40180095]; f(x) = [29.37525114 35.62280862]
Итерация №2; x = [  1.79045663 -27.19343845]; f(x) = [26.84940998 -1.41099508]
Итерация №3; x = [20.47783099 -4.41597189]; f(x) = [  4.09432045 -19.91992642]
Итерация №4; x = [-4.74356789  1.12724481]; f(x) = [-1.44773083  6.02271791]
Итерация №5; x = [ 2.38603837 -0.09822865]; f(x) = [-0.53607911 -0.54085892]
Итерация №6; x = [ 1.83214534 -0.2311337 ]; f(x) = [-0.12282401 -0.00873803]
Итерация №7; x = [ 1.79733131 -0.34496232]; f(x) = [-0.00058719 -0.00624312]
Итерация №8; x = [ 1.79134435 -0.34420482]; f(x) = [-1.74719343e-05 -2.70024228e-07]
Итерация №9; x = [ 1.79133861 -0.34422104]; f(x) = [-1.60860214e-11 -1.23763777e-10]
Итерация №10; x = [ 1.79133861 -0.34422104]; f(x) = [-2.22044605e-16 -1.11022302e-16]
Ответ: x = 1.791338609963922; y = -0.3442210364067569
```

Заключение

В работе требовалось решить систему нелинейных уравнений методом Ньютона. Для решения задачи была написана программа на языке программирования Python.

Ответ: x = 1.791338609963922; y = -0.3442210364067569