**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики и информатики**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

**«Приближение функций с помощью кубического интерполяционного сплайна»**

**Студент**

Тев Никита Михайлович

2 курс, 2 группа

**Преподаватель**

Никифоров Иван Васильевич

**Минск 2019**

**1. Постановка задачи**

На отрезке [0; 1] задана функция . Приблизить функцию кубическим интерполяционным сплайном по n узлам. Построить график.

**2. Входные данные**

n1 = 3, n2 = 10.

**3. Листинг программы**

import numpy as nmp

import matplotlib.pyplot as pyplt

from scipy.interpolate import CubicSpline

def draw(x, y):

spline = CubicSpline(x, y)

x\_val = nmp.linspace(0.0, 1.0, 100)

pyplt.plot(x\_val, f(x\_val), 'g', label='function')

pyplt.plot(x\_val, spline(x\_val), 'r--', label='spline')

pyplt.legend()

pyplt.grid(True)

pyplt.title('n = {}'.format(n))

pyplt.show()

def f(x):

return nmp.exp(x)\*x + x\*\*2

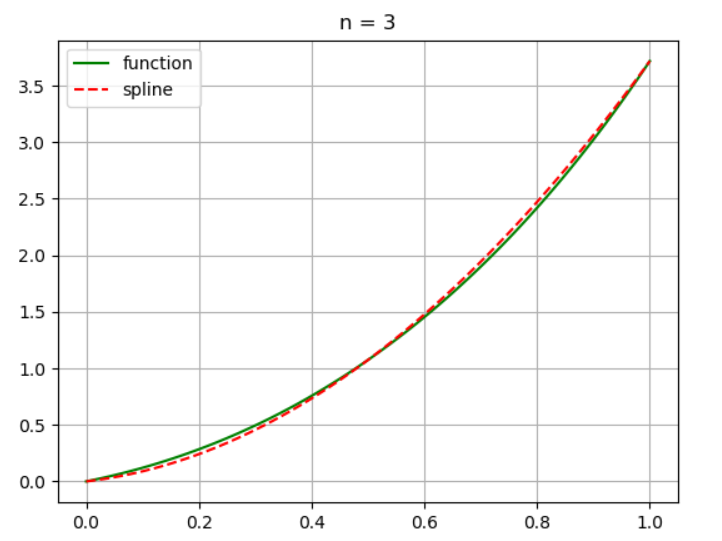
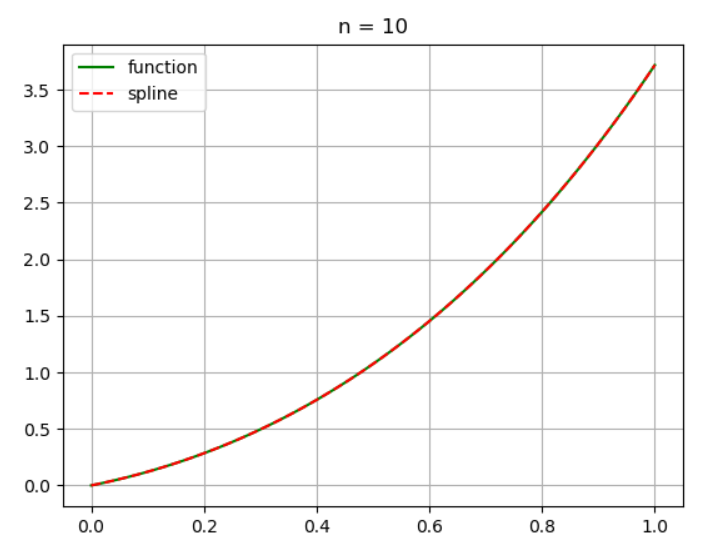
for n in [3, 10]:

x\_arr = nmp.linspace(0.0, 1.0, n)

y\_arr = f(x\_arr)

draw(x\_arr, y\_arr)

**4. Выходные данные**

Даже при делении отрезка на 10 частей, точность интерполяции настолько высока, что на графике не представляется возможным увидеть разницу между исходной функцией и интреполяционным сплайном.