**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**



**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ «Информатика и системы управления»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Лабораторная работа № 5

**Тема**  Построение и программная реализация алгоритмов численного интегрирования

**Студент** Челядинов Илья

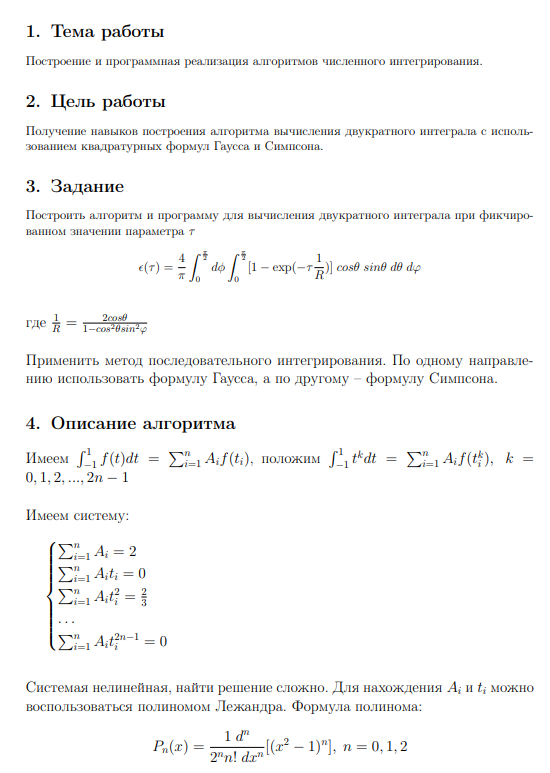
**Группа** ИУ7-43Б**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

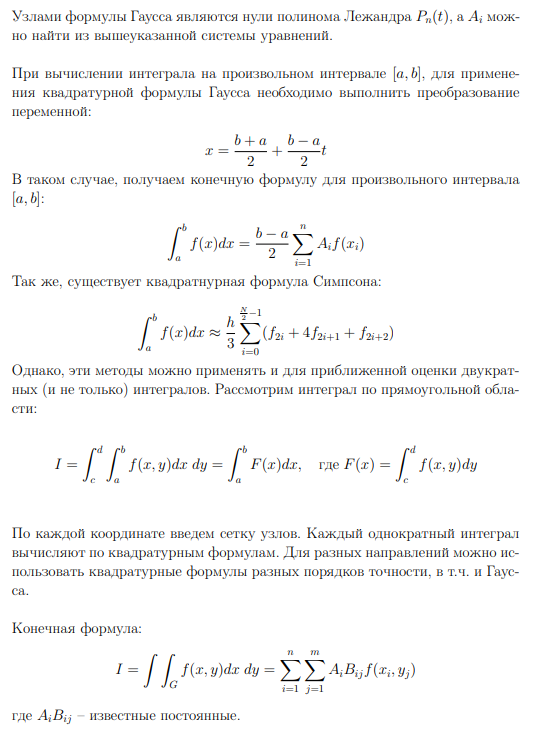
**Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Преподаватель** Градов Владимир Михайлович

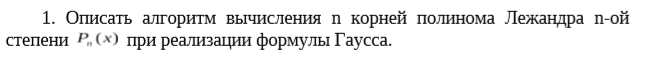
Москва.

2020 г.





**Результаты:**

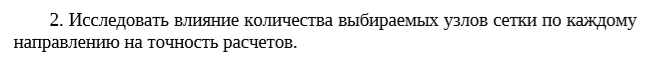


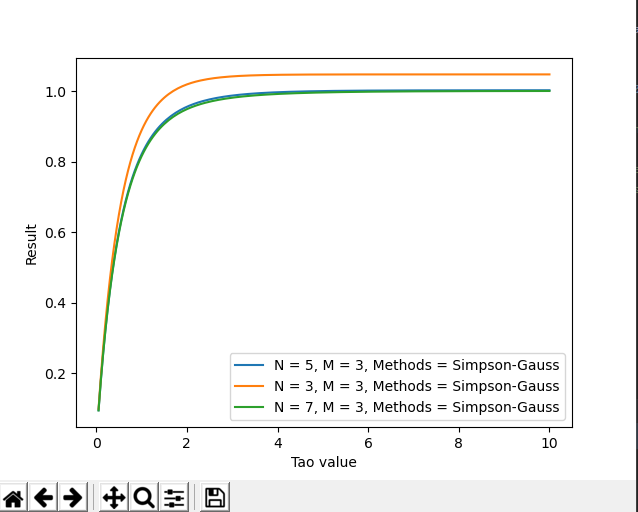
Так как мы знаем из математики, что производная четной функции – нечетна и наоборот, при составлении функции полинома Лежандра берется n-ная производная от четной функции, в результате получаем четную функцию, либо нечетную.

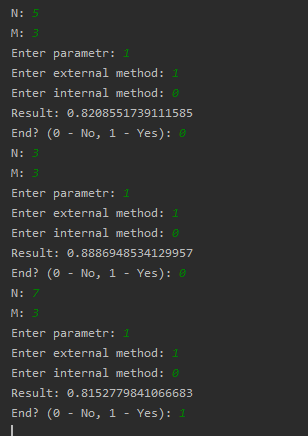
F(x) = 0 => f(-x) = 0

Нам известно, что у полинома Лежандра n-ной степени – n действительных различных корней на отрезке [-1;1], из каждого корня x следует корень -x. Тогда достаточно искать корни только на отрезке (0;1].

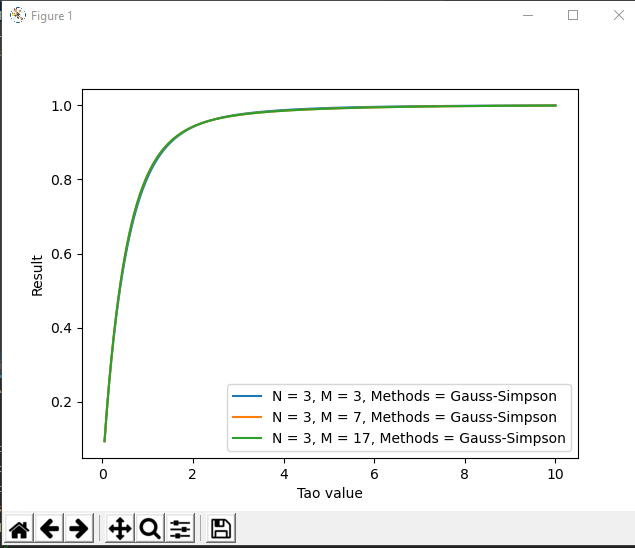
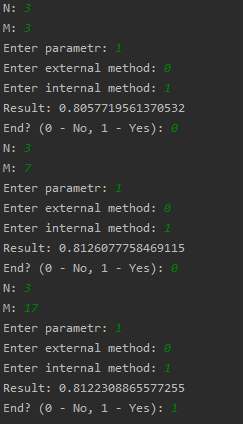
Мы ищем методом половинного деления на маленьких интервалах. Весь интервал разбивается на большое количество одинаковых отрезков и каждый из них проверяется. Если отрезок имеет разные знаки на концах – он содержит корень(мы ищем его методом половинного деления). Если один из концов отрезка равен 0, мы нашли корень.

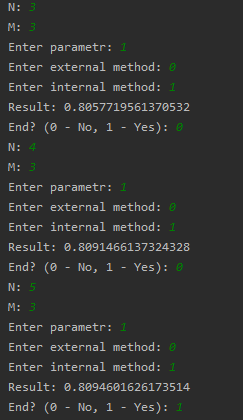
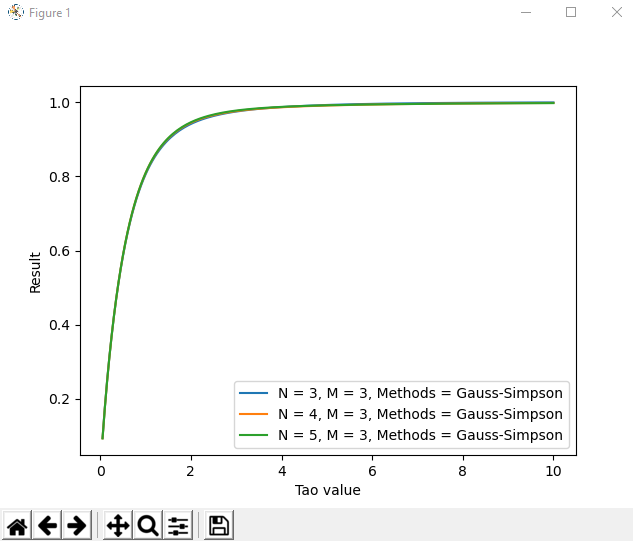


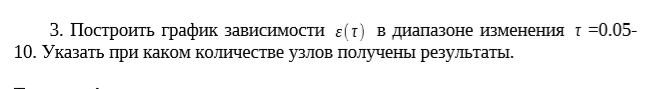


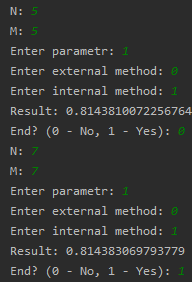
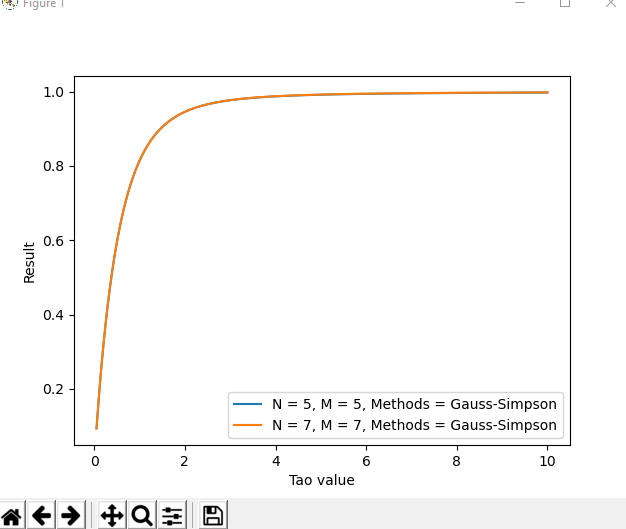




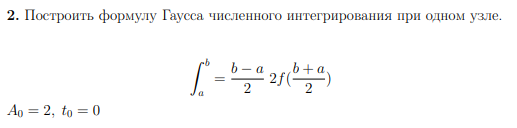


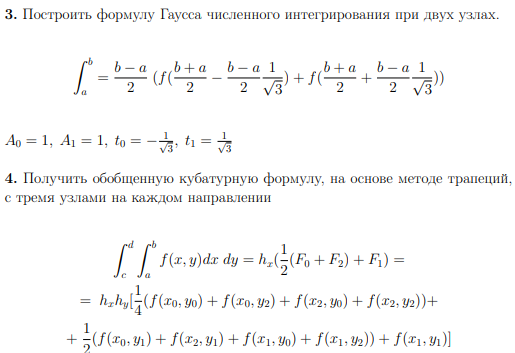


Графики наложились друг на друга.

**Ответы контрольные вопросы**

1. В каких ситуациях теоретический порядок квадратурных формул численного интегрирования не достигается? Если подынтегральная функция не имеет соответствующих производных. Например, если на отрезке интегрирования не существует 3-я и 4-я производные, то порядок тончности формула Симпсона будет только 2-ой.





Код программы:

def gauss(func, a, b, num\_of\_nodes):  
 args, coeffs = leggauss(num\_of\_nodes)  
 res = 0  
  
 for i in range(num\_of\_nodes):  
 res += (b - a) / 2 \* coeffs[i] \* func(t\_to\_x(args[i], a, b))  
  
 return res  
  
  
def simpson(func, a, b, num\_of\_nodes):  
 if (num\_of\_nodes < 3 or num\_of\_nodes & 1 == 0):  
 raise ValueError  
  
 h = (b - a) / (num\_of\_nodes - 1)  
 x = a  
 res = 0  
  
 for \_ in range((num\_of\_nodes - 1) // 2):  
 res += func(x) + 4 \* func(x + h) + func(x + 2 \* h)  
 x += 2 \* h  
  
 return res \* (h / 3)  
  
  
def main\_function(param):  
 subfunc = lambda x, y: 2 \* cos(x) / (1 - (sin(x) \*\* 2) \* (cos(y) \*\* 2))  
 func = lambda x, y: (4 / pi) \* (1 - exp(-param \* subfunc(x, y))) \* cos(x) \* sin(x)  
 return func  
  
  
def func\_2\_to\_1(func2, value):  
 return lambda y: func2(value, y)  
  
  
def t\_to\_x(t, a, b):  
 return (b + a) / 2 + (b - a) \* t / 2  
  
  
  
  
def integrate2(func, limits, num\_of\_nodes, integrators):  
 inner = lambda x: integrators[1](func\_2\_to\_1(func, x), limits[1][0], limits[1][1], num\_of\_nodes[1])  
 return integrators[0](inner, limits[0][0], limits[0][1], num\_of\_nodes[0])  
  
  
def tao\_graph(integrate\_func, ar\_params, label):  
 X = list()  
 Y = list()  
 for t in arange(ar\_params[0], ar\_params[1] + ar\_params[2], ar\_params[2]):  
 X.append(t)  
 Y.append(integrate\_func(t))  
 plt.plot(X, Y, label=label)  
  
  
def generate\_label(n, m, func1, func2):  
 res = "N = " + str(n) + ", M = " + str(m) + ", Methods = "  
 res += "Simpson" if func1 == simpson else "Gauss"  
 res += "-Simpson" if func2 == simpson else "-Gauss"  
 return res  
  
end = False  
  
  
while not end:  
 N = int(input("N: "))  
 M = int(input("M: "))  
  
 param = float(input("Enter parametr: "))  
 mode = bool(int(input("Enter external method: ")))  
 func1 = simpson if mode else gauss  
 mode = bool(int(input("Enter internal method: ")))  
 func2 = simpson if mode else gauss  
 param\_integrate = lambda tao: integrate2(main\_function(tao), [[0, pi / 2], [0, pi / 2]], [N, M], [func1, func2])  
 print("Result:", param\_integrate(param))  
  
 try:  
 tao\_graph(param\_integrate, [0.05, 10, 0.05], generate\_label(N, M, func1, func2))  
 except ValueError:  
 print("Be careful with simpson: argument should be > 2 and not even (3, 5...);")  
 except ZeroDivisionError:  
 print("Can't use 2 Simpsons, zero in denominator")  
 end = bool(int(input("End? (0 - No, 1 - Yes): ")))