**Московский Авиационный Институт**

**(Национальный Исследовательский Университет)**

Кафедра: 804 «Теория вероятностей и компьютерное моделирование»

Факультет: «Информационные технологии и прикладная математика»

Дисциплина: «Статистическое моделирование на ЭВМ»

**Курсовой проект**

Группа: 8О-404Б

Студент: Сорокин Д.М.

Преподаватель: Мирошкин В.Л.

Оценка:

Подпись:

Москва

2018

# Задание 3

Посчитать методом Монте-Карло значение интеграла:

Решение:

Формула для расчета:

где – пределы интегрирования;

;

– случайная величина, имеющая равномерное распределение.

## Реализация на компьютере

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** pandas **as** pd

**def** my\_func(x):

**return** (x\*\*2 + 1 + np.exp(x)) / (np.sin(x) + np.cos(x) + np.tan(x))

X=np.linspace(np.pi / 2.5,np.pi / 10,1000)

plt.plot(X,my\_func(X))

plt.savefig("func.png")

**def** integrate(x1,x2,func=my\_func,n=100000):

X=np.linspace(x1,x2,1000)

y1=0

y2=max((func(X)))+1

**print**(x1,x2,y1,y2)

area=(x2-x1)\*(y2-y1)

check=[]

xs=[]

ys=[]

**for** i **in** range(n):

x=np.random.uniform(x1,x2,1)

xs.append(x)

y=np.random.uniform(y1,y2,1)

ys.append(y)

**if** abs(y)>abs(func(x)) **or** y<0:

check.append(0)

**else**:

check.append(1)

**return**(np.mean(check)\*area,xs,ys,check)

**def** plotting(par):

ans,x,y,c=integrate(np.pi / 2.5,np.pi / 10, n=par)

**print**(f"N = {par}**\n**", ans)

df=pd.DataFrame()

df['x']=x

df['y']=y

df['c']=c

plt.plot(X,my\_func(X))

plt.scatter(df[df['c']==0]['x'],df[df['c']==0]['y'],color='red')

plt.scatter(df[df['c']==1]['x'],df[df['c']==1]['y'],color='blue')

plt.savefig(f"intFor{par}.png")

plotting(10);

plotting(100);

plotting(1000);

plotting(10000);

plotting(100000);

## Полученный результат

N = 10

-1.2285120405872234

N = 100

-1.695346616010368

N = 1000

-1.4422731356494003

N = 10000

-1.4850253546618357

N = 100000

-1.466229120440851

## Графики

