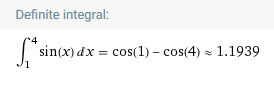
**Лабораторная работа №1**

**Тема:** Численное интегрирование

**Язык:** Python 3.6.5

**Постановка задачи:** Написать программу, которая реализует методы численного интегрирования с постоянным и переменным шагом для табличных интегралов

**Табличный интеграл:**



**Переменные:**

Модуль integrals1.py

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| i | Счетчик цикла | int |
| a, b, c, d | Границы интегрирования | float |
| n, nx, ny | Количество разбиений | int |
| e | Точность вычислений | float |
| listarg | Список элементов, передаваемых в методы | list |
| mainlist | Список с элементами главного меню | list |
| changelist | Список с элементами меню методов с постоянным шагом | list |
| postlist | Список с элементами меню методов с переменным шагом | list |
| switch | Переключатель главного меню | int |
| switch1 | Переключатель меню методов с постоянным шагом | int |
| switch2 | Переключатель меню методов с переменным шагом | int |
| printmain() | Печать главного меню | func |
| printchangelist() | Печать меню методов с постоянным шагом | func |
| printpostlist() | Печать меню методов с переменным шагом | func |
| readlist/1/2/3 | Ввод переменных в список | func |

Модуль metods.py

|  |  |  |
| --- | --- | --- |
| **Имя** | **Смысл** | **Тип** |
| i, j | Счетчик цикла | int |
| a, b, c, d | Границы интегрирования | float |
| x | x | int |
| h | Шаг вычисления | float |
| r | Остаточный член | float |
| n, nx, ny | Количество разбиений | int |
| res | Результат | float |
| f() | Интегральная функция | func |
| fpr1/2/3/4() | Производная первого/второго/третьего/четвертого порядка | func |
| trap() | Метод трапеций | func |
| pryaml | Метод прямоугольников левых частей | func |
| pryamr | Метод прямоугольников правых частей | func |
| par | Метод парабол | func |
| spectrap | Метод трапеций для использования в первом методе | func |
| met1 | Первый метод | func |
| specpryamr | Метод прямоугольников правых частей для использования во втором методе | func |
| met2 | Второй метод | func |
| double\_integral | Вычисление двойного интеграла | func |

**Код программы:**

Integrals1.py

import metods

def printmain():

for i in range(5):

print(mainlist[i])

print()

def printchangelist():

for i in range(6):

print(changelist[i])

print()

def printpostlist():

for i in range(4):

print(postlist[i])

print()

def readlist():

print("Введите значения a, b, n: ")

a=float(input("a = "))

b=float(input("b = "))

n=int(input("n = "))

listarg=[a, b, n]

return listarg

def readlist2():

print("Введите значения a, b, E: ")

a=float(input("a = "))

b=float(input("b = "))

e=float(input("E = "))

listarg=[a, b, e]

return listarg

def readlist3():

print("Введите значения a, b, c, d, nx, ny: ")

print("Введите первую область интегрирования:")

a=float(input("a = "))

b=float(input("b = "))

print("Введите вторую область интегрирования:")

c=float(input("c = "))

d=float(input("d = "))

print("Введите количество разбиений:")

nx=int(input("nx = "))

ny=int(input("ny = "))

listarg=[a, b, c, d, nx, ny]

return listarg

mainlist=["Главное меню", "1) Методы с постоянным шагом",

"2) Методы с переменным шагом", "3) Двойной интеграл","4) Выход из программы"]

changelist=["Методы с постоянным шагом", "1) Метод правых частей прямоугольников",

"2) Метод левых частей прямоугольников", "3) Метод трапеций",

"4) Метод парабол", "5) Возврат в главное меню"]

postlist=["Методы с переменным шагом", "1) Метод по 1 алгоритму",

"2) Метод по 2 алгоритму", "3) Возврат в главное меню"]

while True:

printmain()

switch=int(input())

if switch==1:

while True:

printchangelist()

switch1=int(input())

if switch1==1:

print("Результат: ", metods.pryamr(readlist()))

elif switch1==2:

print("Результат: ", metods.pryaml(readlist()))

elif switch1==3:

print("Результат: ", metods.trap(readlist()))

elif switch1==4:

print("Результат: ", metods.par(readlist()))

elif switch1==5:

break

elif switch==2:

while True:

printpostlist()

switch2=int(input())

if switch2==1:

print("Результат: ", metods.met1(readlist2()))

elif switch2==2:

print("Результат: ", metods.met2(readlist2()))

elif switch2==3:

break

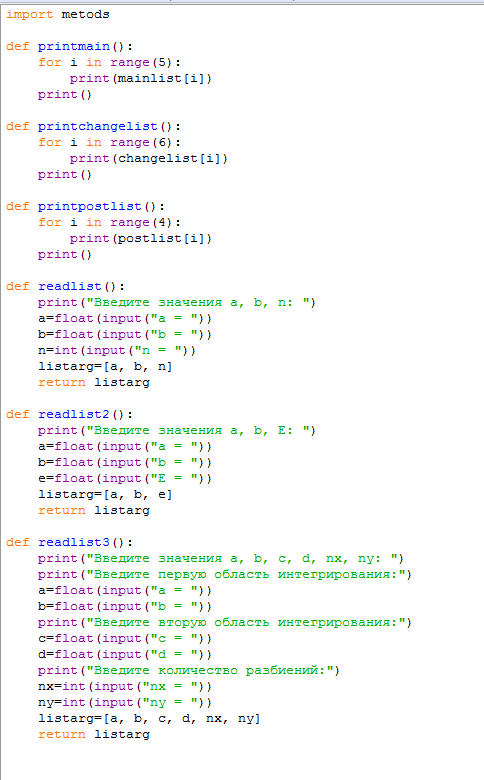
elif switch==3:

print("Результат: ",metods.double\_integral(readlist3()))

elif switch==4:

break

print("Программа завершена")





Metods.py

import math

def f(x):

return math.sin(x)

def fpr1(x):

return math.cos(x)

def fpr2(x):

return -(math.sin(x))

def fpr3(x):

return -(math.cos(x))

def fpr4(x):

return math.sin(x)

def trap(listarg):

a=listarg[0]

b=listarg[1]

n=listarg[2]

h=float(b-a)/n

res=0.5\*(f(a)+f(b))

for i in range(1,n):

res+=f(a+i\*h)

res\*=h

i=a

m=0

while i<=b:

if m < abs(fpr2(i)):

m=abs(fpr2(i))

i+=0.01

r=(((b-a)\*\*3)/(12\*n\*\*2))\*m

print("Остаточный член |R|: ", abs(r))

return res

def pryaml(listarg):

a=listarg[0]

b=listarg[1]

n=listarg[2]

h=float(b-a)/n

i=a

res=0

while i<=b-h:

res+=h\*f(i)

i+=h

i=a

m=0

while i<=b:

if m < abs(fpr1(i)):

m=abs(fpr1(i))

i+=0.01

r=(((b-a)\*\*2)/(2\*n))\*m

print("Остаточный член |R|: ", abs(r))

return res

def pryamr(listarg):

a=listarg[0]

b=listarg[1]

n=listarg[2]

h=float(b-a)/n

i=a+h

res=0

while i<=b:

res+=h\*f(i)

i+=h

i=a

m=0

while i<=b:

if m < abs(fpr1(i)):

m=abs(fpr1(i))

i+=0.01

r=(((b-a)\*\*2)/(2\*n))\*m

print("Остаточный член |R|: ", abs(r))

return res

def par(listarg):

a=listarg[0]

b=listarg[1]

n=listarg[2]

h=float(b-a)/n

i=a+h

nch=0

ch=0

while i<=b-h:

nch+=f(i)

i+=2\*h

i=a+2\*h

while i<=b-2\*h:

ch+=f(i)

i+=2\*h

res=h/3\*(f(a)+f(b)+4\*nch+2\*ch)

i=a

m=0

while i<=b:

if m < abs(fpr4(i)):

m=abs(fpr4(i))

i+=0.01

r=(((b-a)\*\*5)/(180\*2\*n\*\*2))\*m

print("Остаточный член |R|: ", abs(r))

return res

def spectrap(listarg, step):

a=listarg[0]

b=listarg[1]

h=step

i=a+h

res=0

while i<=b-h:

res+=f(i)

i+=h

res+=(f(a)+f(b))/2

res\*=h

return res

def met1(listarg):

e=listarg[2]

step=e\*\*1/2

res1=spectrap(listarg, step)

res2=spectrap(listarg, step/2)

while(abs(res2-res1)>e):

step=step/2

res1=spectrap(listarg, step)

res2=spectrap(listarg, step/2)

return res2

def specpryamr(listarg):

a=listarg[0]

b=listarg[1]

n=listarg[2]

h=float(b-a)/n

i=a+h

res=0

while i<=b:

res+=h\*f(i)

i+=h

return res

def met2(listarg):

a=listarg[0]

b=listarg[1]

hv=listarg[2]

hs=hv/2

n=(b-a)/hv

res1=specpryamr((a, b, n))

res2=0

while abs(res2-res1)>hv:

res1=specpryamr((a, b, n))

a+=hs

hd=hv/2

res2=specpryamr((a, b, n))

hv/=2

hs/=2

return res2

def double\_integral(listarg):

doublef = lambda x, y: 4\*x\*\*2+5\*y

a=listarg[0]

b=listarg[1]

c=listarg[2]

d=listarg[3]

nx=listarg[4]

ny=listarg[5]

hx=(b-a)/float(nx)

hy=(d-c)/float(ny)

res = 0

for i in range(nx):

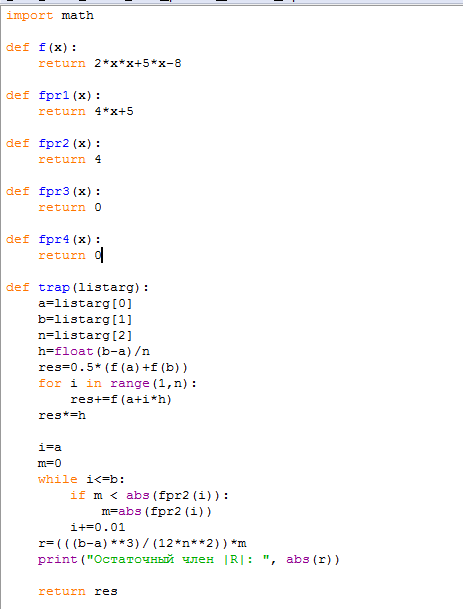
for j in range(ny):

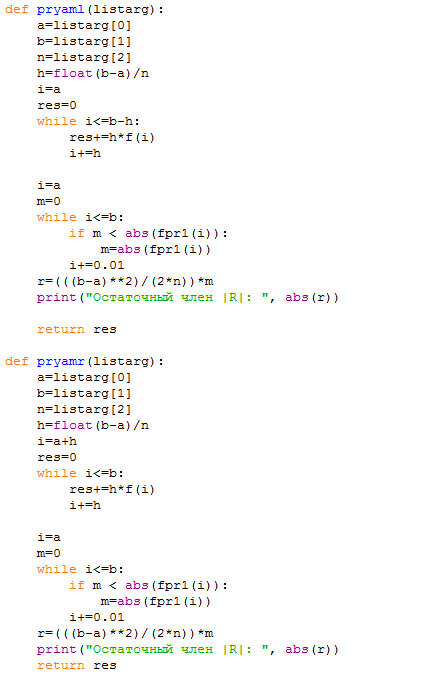
xi = a + hx/2 + i\*hx

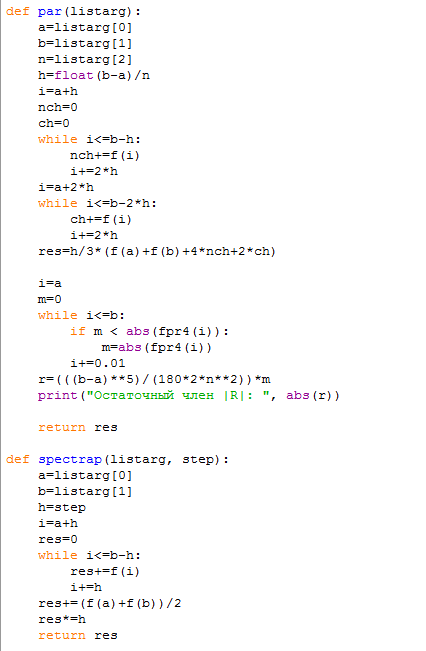
yj = c + hy/2 + j\*hy

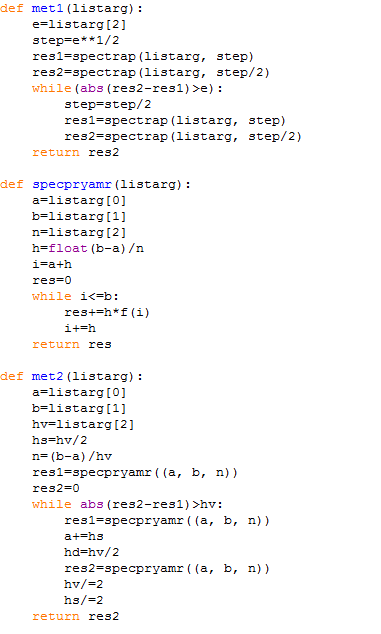
res += hx\*hy\*doublef(xi,yj)

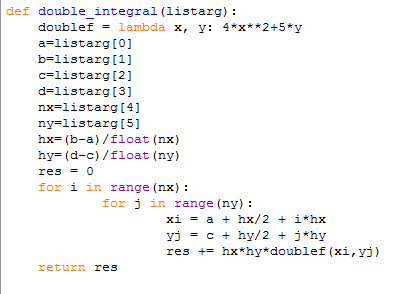
return res





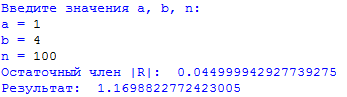




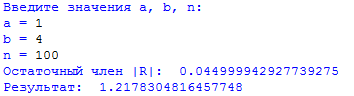


**Результат работы:**

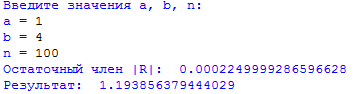
Метод прямоугольников правых частей:



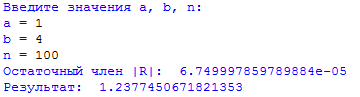
Метод прямоугольников левых частей:



Метод трапеций:



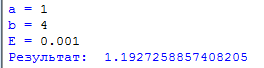
Метод парабол:



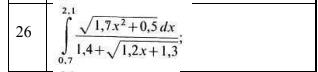
Метод по 1 алгоритму:

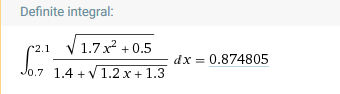


Метод по 2 алгоритму:



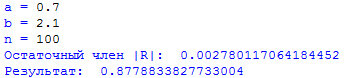
**Нетабличный интеграл:**



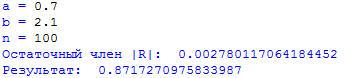


**Результат работы:**

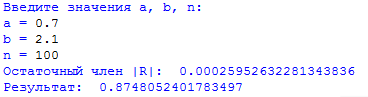
Метод прямоугольников правых частей:



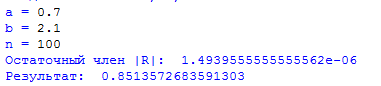
Метод прямоугольников левых частей:



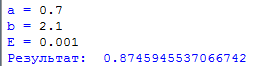
Метод трапеций:



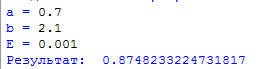
Метод парабол:



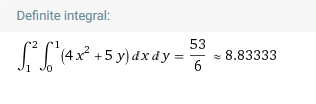
Метод по 1 алгоритму:

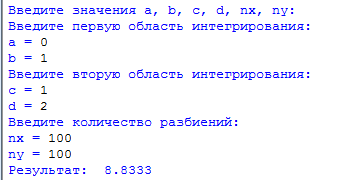


Метод по 2 алгоритму:



**Двойной интеграл:**





**Вывод:**

В ходе работы были реализованы различные методы решения определенного интеграла. Полученные, с помощью таких методов, значения зависят от таких параметров, как количество шагов (для методов с постоянным шагом), точности ε (для методов с постоянным шагом), а также самой функции. Также, для интегралов, решенных методами с постоянным шагом был получен остаточный член.