Соколовський Євгеній 26 варіант 8 група

1)

from scipy import integrate

import math

eps = 0.001

def f1(x):

return 1/math.sqrt(x+1.2)

def left\_rec(f1,a,b,n):

h=(b-a)/n

sum=0

for i in range(0,n):

sum+=f1(a+i\*h)

return sum\*h

v,err = integrate.quad(f1,2,1.2)

if abs(left\_rec(f1,2,1.2,2\*10) - left\_rec(f1,2,1.2,10))/3. <=eps:

print("left rectangle:",round (left\_rec(f1,2,1.2,10), 5))

def right\_rec(f1,a,b,n):

h=(b-a)/n

sum=0

for i in range(1,n+1):

sum+=f1(a+i\*h)

return sum\*h

print("right rectangle:",round (right\_rec(f1,2,1.2,10), 5))

def aver\_rec(f1,a,b,n):

h=0.08

sum=0

for i in range(0,n):

sum+=f1(a+i\*h)

return sum\*h

print("average rectangle:",round (aver\_rec(f1,2,1.2,10), 5))

print("Check for the rectangle method= ",round (v, 5))

Изображение выглядит как текст

Автоматически созданное описание

2) from scipy import integrate

import math

eps = 0.001

def f1(x):

return (math.sqrt(x)+1)\*(math.tan(2\*x))

def left\_rec(f1,a,b,n):

h=(b-a)/n

sum=0

for i in range(0,n):

sum+=f1(a+i\*h)

return sum\*h

v,err = integrate.quad(f1,0.72,0.6)

if abs(left\_rec(f1,0.72,0.6,2\*10) - left\_rec(f1,0.72,0.6,10))/3. <=eps:

print("left rectangle:",round (left\_rec(f1,0.72,0.6,10), 5))

def right\_rec(f1,a,b,n):

h=(b-a)/n

sum=0

for i in range(1,n+1):

sum+=f1(a+i\*h)

return sum\*h

print("right rectangle:",round (right\_rec(f1,0.72,0.6,10), 5))

def aver\_rec(f1,a,b,n):

h=0.08

sum=0

for i in range(0,n):

sum+=f1(a+i\*h)

return sum\*h

print("average rectangle:",round (aver\_rec(f1,0.72,0.6,10), 5))

print("Check for the rectangle method= ",round (v, 5))

Изображение выглядит как текст

Автоматически созданное описание

3) from scipy import integrate

import math

eps = 0.001

def f1(x):

return 1/math.sqrt(2\*x\*\*2+3)

def left\_rec(f1,a,b,n):

h=(b-a)/n

sum=0

for i in range(0,n):

sum+=f1(a+i\*h)

return sum\*h

v,err = integrate.quad(f1,1.4,0.8)

if abs(left\_rec(f1,1.4,0.8,2\*10) - left\_rec(f1,1.4,0.8,10))/3. <=eps:

print("left rectangle:",round (left\_rec(f1,1.4,0.8,10), 5))

def right\_rec(f1,a,b,n):

h=(b-a)/n

sum=0

for i in range(1,n+1):

sum+=f1(a+i\*h)

return sum\*h

print("right rectangle:",round (right\_rec(f1,1.4,0.8,10), 5))

def aver\_rec(f1,a,b,n):

h=0.08

sum=0

for i in range(0,n):

sum+=f1(a+i\*h)

return sum\*h

print("average rectangle:",round (aver\_rec(f1,1.4,0.8,10), 5))

print("Check for the rectangle method= ",round (v, 5))

Изображение выглядит как текст

Автоматически созданное описание

1)

from scipy import integrate

import math

eps = 0.001

def f1(x):

return 1/math.sqrt(x+1.2)

def trap(f1,a,b,n):

h=(b-a)/n

sum=0.5\*(f1(a)+f1(b))

for i in range(1,n):

sum+=f1(a+i\*h)

return sum\*h

v,err = integrate.quad(f1,2,1.2)

if abs (trap(f1, 2,1.2, 2\*10) -trap(f1,2,1.2, 10))/3. <= eps:

print("Trapetzia method:",round (trap(f1,2,1.2,10), 5))

print("Check for the trapetzia method= ",round(v, 5))

Изображение выглядит как текст

Автоматически созданное описание

2)

from scipy import integrate

import math

eps = 0.001

def f1(x):

return (math.sqrt(x)+1)\*(math.tan(2\*x))

def trap(f1,a,b,n):

h=(b-a)/n

sum=0.5\*(f1(a)+f1(b))

for i in range(1,n):

sum+=f1(a+i\*h)

return sum\*h

v,err = integrate.quad(f1,0.72,0.6)

if abs (trap(f1,0.72,0.6, 2\*10) -trap(f1,0.72,0.6, 10))/3. <= eps:

print("Trapetzia method:",round (trap(f1,0.72,0.6,10), 5))

print("Check for the trapetzia method= ",round(v, 5))

Изображение выглядит как текст

Автоматически созданное описание

3)

from scipy import integrate

import math

eps = 0.001

def f1(x):

return 1/math.sqrt(2\*x\*\*2+3)

def trap(f1,a,b,n):

h=(b-a)/n

sum=0.5\*(f1(a)+f1(b))

for i in range(1,n):

sum+=f1(a+i\*h)

return sum\*h

v,err = integrate.quad(f1,1.4,0.8)

if abs (trap(f1,1.4,0.8, 2\*10) -trap(f1,1.4,0.8, 10))/3. <= eps:

print("Trapetzia method:",round (trap(f1,1.4,0.8,10), 5))

print("Check for the trapetzia method= ",round(v, 5))

Изображение выглядит как текст

Автоматически созданное описание

1)

from scipy import integrate

import math

eps = 0.001

def f1(x):

return 1/math.sqrt(x+1.2)

def simpson(a,b,n):

h = (b - a) / n

integr = f1(a) + f1(b)

for i in range(1,n):

k = a + i\*h

if i%2 == 0:

integr += 2 \* f1(k)

else:

integr += 4 \* f1(k)

integr \*= h/3

return integr

if abs(simpson(2,1.2,2\*8) -simpson(2,1.2,8))/ 15. <= eps:

print("Simpsone method:",round (simpson(2,1.2,8), 5))

v,err = integrate.quad(f1,2,1.2)

print("Check for the simpsone method= ",round(v, 5))

Изображение выглядит как текст

Автоматически созданное описание

2)

from scipy import integrate

import math

eps = 0.001

def f1(x):

return (math.sqrt(x)+1)\*(math.tan(2\*x))

def simpson(a,b,n):

h = (b - a) / n

integr = f1(a) + f1(b)

for i in range(1,n):

k = a + i\*h

if i%2 == 0:

integr += 2 \* f1(k)

else:

integr += 4 \* f1(k)

integr \*= h/3

return integr

if abs(simpson(0.72,0.6,2\*8) -simpson(0.72,0.6,8))/ 15. <= eps:

print("Simpsone method:",round (simpson(0.72,0.6,8), 5))

v,err = integrate.quad(f1,0.72,0.6)

print("Check for the simpsone method= ",round(v, 5))

Изображение выглядит как текст

Автоматически созданное описание

3)

from scipy import integrate

import math

eps = 0.001

def f1(x):

return 1/math.sqrt(2\*x\*\*2+3)

def simpson(a,b,n):

h = (b - a) / n

integr = f1(a) + f1(b)

for i in range(1,n):

k = a + i\*h

if i%2 == 0:

integr += 2 \* f1(k)

else:

integr += 4 \* f1(k)

integr \*= h/3

return integr

if abs(simpson(1.4,0.8,2\*8) -simpson(1.4,0.8,8))/ 15. <= eps:

print("Simpsone method:",round (simpson(1.4,0.8,8), 5))

v,err = integrate.quad(f1,1.4,0.8)

print("Check for the simpsone method= ",round(v, 5))

Изображение выглядит как текст

Автоматически созданное описание