



*Российский государственный университет
нефти и газа им. И.М. Губкина*

Кафедра Информатики

*Дисциплина: Программные комплексы
общего назначения*

Преподаватель:

К.Т.Н., ДОЦЕНТ

Коротаев

Александр Фёдорович

Вычисление определенных интегралов



Численное интегрирование заключается в приближенном вычислении определенного интеграла вида

$$\int_a^b y(x)dx$$

trapz(Y) — использует интегрирование **методом трапеций** с единичным шагом между отсчетами

В форме **trapz(x, Y)** — возвращает интеграл функции, заданной значениями **Y**, вычисленными по значениям переменной **x**, (пределы интегрирования в этом случае задаются начальным и конечным элементами вектора **x**)



Метод трапеций

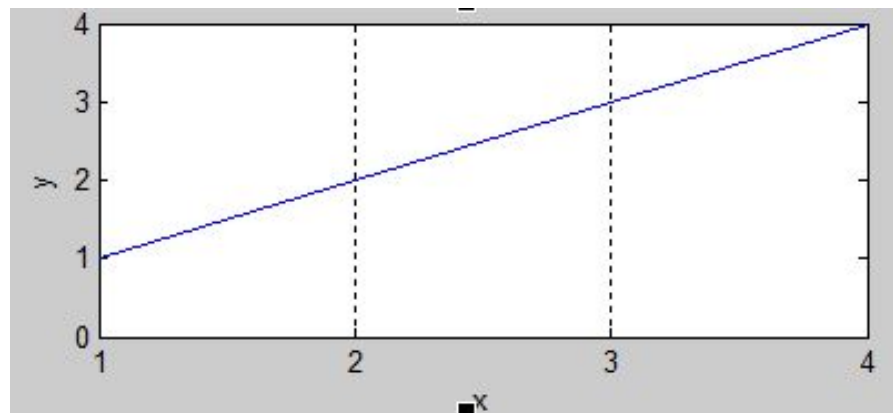
Пример 1

```
» Y = [1, 2, 3, 4]
```

```
» trapz(Y)
```

```
ans =
```

```
7.5000
```



Пример 2

π

Вычислить $\int_0^{\pi} \sin(x) dx$ с шагом $\pi/5$

0

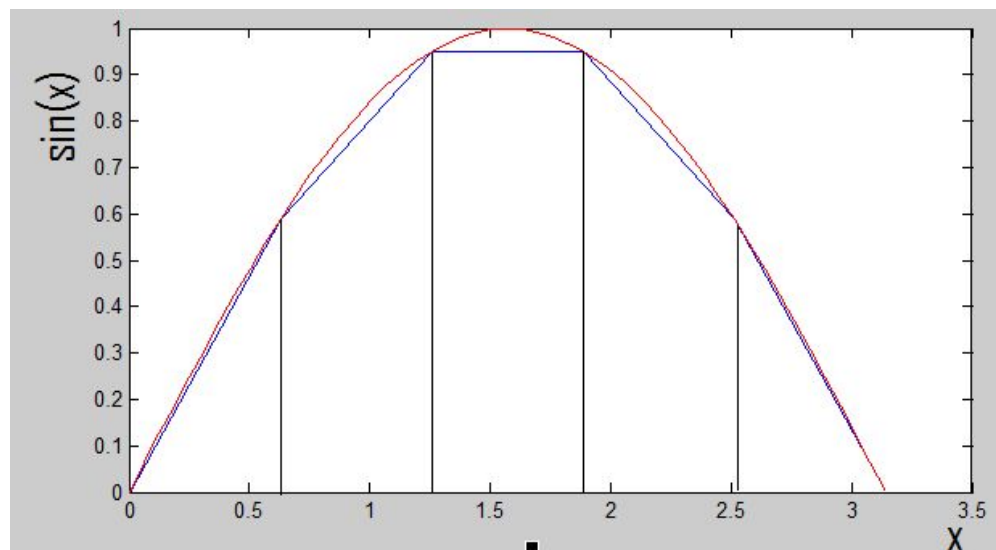
```
>> X = 0:pi/5:pi;
```

```
>> Y = sin(X);
```

```
>> Z = trapz(X, Y)
```

```
Z =
```

```
1.9338
```



Численное интегрирование методом квадратур



Квадратура — численный метод нахождения площади под графиком функции

quad(@fun,a,b,tol) выполняет интегрирование низкого порядка с использованием квадратурной формулы Симпсона. Эффективна при низкой требуемой точности вычислений

fun должна быть описана в m-файле

a, b – пределы интегрирования

tol - относительная погрешность (необязательный параметр)

quadl(@fun,a,b) - использует квадратуру Гаусса-Лобатто очень высокого порядка, что даёт более высокую точность вычислений

Двойные интегралы



Сводятся к вычислению повторных определенных интегралов
(внутренний интеграл является подынтегральной функцией для
внешнего)

```
dblquad(@fun,x0,x1,y0,y1)
```

Пример 1 **quad('exp(x)+x.^2+2*sin(x)-5',1,5,0.001)**

ans =

167.5415

Пример 2

```
function z=for2Var(x,y)
```

```
z=x.*sin(y) +y.*sin(x);
```

Записав этот текст в файл **for2Var.m** , находим
интеграл

```
int=dblquad(@for2Var,1,2,0,1)
```

```
int=1.1678
```

Аналитический метод вычисления интегралов



Применимы следующие варианты:

int(y) , если вычисляется неопределенный интеграл
int(y,a,b) , если вычисляется определенный
интеграл в пределах **[a,b]**

где **y** – подынтегральная функция,

a,b – пределы интегрирования

Порядок записи программы:

1. Символьные переменные описываются как **syms**
2. Вычисляется подынтегральное выражение **y=f(x)**
3. Обращение к функции **int**

Пример



$$\int \frac{x}{a + bx^2} dx$$

```
>> syms x a b;
```

```
>> y=x/(a+b*x^2);
```

```
>> int(y) % неопределённый интеграл
```

```
ans =
```

```
log(b*x^2 + a)/(2*b)
```

```
>> syms x
```

```
>> a=1; b=2; y=x/(a+b*x^2);
```

```
>> int(y,0,1) % определённый интеграл
```

```
ans =
```

```
log(3)/4
```

Разложение в степенной ряд



По формуле Тейлора

$$f(x) = f(a) + (x-a)f'(a)/1! + (x-a)^2 f''(a)/2! + \dots + (x-a)^n f^{(n)}(a)/n! + R_n$$

$f(a), f'(a), f''(a), \dots, f^{(n)}(a)$ – значения функции и её производных в точке a

Если $a=0$, получаем ряд Маклорена

$$f(x) = f(0) + x f'(0)/1! + x^2 f''(0)/2! + \dots + x^n f^{(n)}(0)/n! + R_n$$

`taylor(y)` – выдаёт **5** членов разложения в ряд

Маклорена функции, заданной в y

```
>> syms x;
```

```
>> y=sin(x);
```

```
>> MacSin=taylor(y)
```

```
MacSin =
```

```
x^5/120 - x^3/6 + x
```




Разложение в степенной ряд

Более общий вид функции

taylor(y, 'ExpansionPoint', val1, 'Order', val2)

даёт разложение функции **y** в точке, заданной в **val1**, с числом членов ряда, заданным в **val2**

```
>> syms x;
```

```
>> y=log(x);
```

```
>> TayLog1=taylor(y, 'ExpansionPoint', 1, 'Order', 6)
```

```
TayLog1 =
```

```
x - (x - 1)^2/2 + (x - 1)^3/3 - (x - 1)^4/4 + (x - 1)^5/5 - 1
```

В более ранних версиях была другая форма функции

taylor(y,x,x0,n) — выдаёт **n** членов разложения в ряд Тейлора функции, заданной в **y**, в точке **x0**



Примеры к лабораторной работе №4

Задание 1-1 $\int (2 - 5x)e^{3x} dx$

Задание 1-2 Замена переменной !!!

Задание 2 $\int_{-0.7}^{0.8} \frac{dx}{1-x^2}$

Задание 4 разложить в степенной ряд
 $\ln(-12x^2 - x + 1)$



Решение системы с помощью функции solve

```
>> syms x y z;
```

```
>> Y=solve('3*x+y-z=3','-5*x+3*y+4*z=1', 'x+y+z=0.5')
```

```
Y =
```

```
  x: [1x1 sym]
```

```
  y: [1x1 sym]
```

```
  z: [1x1 sym]
```

```
>> Y.x
```

```
ans =
```

```
-0.10714285714285714285714285714286
```

Можно воспользоваться функцией

vpa(Y.x, n) , где **x** – неизвестное, **n** – число значащих цифр в ответе

```
>> vpa(Y.x,5)
```

```
ans =
```

```
-.10714
```

Решение систем нелинейных уравнений



fsolve (FUN, x0, options) ,

где **FUN** – система уравнений, сохраненная в m-файле

x0 – начальное приближение

Пример: $x_1 x_2 + x_3 = 6.5$; $x_1 x_2^4 + x_3 = 167$; $x_1 x_2^6 + x_3 = 1470$

function F=myfun(x)

F=[x(1)*x(2)+x(3)-6.5 x(1)*x(2)^4+x(3)-167 x(1)*x(2)^6+x(3)-1470];

>> X=fsolve(@myfun,[1 1 1])

X =

2.1512 2.9678 0.1157

Эту же систему можно решить с помощью функции

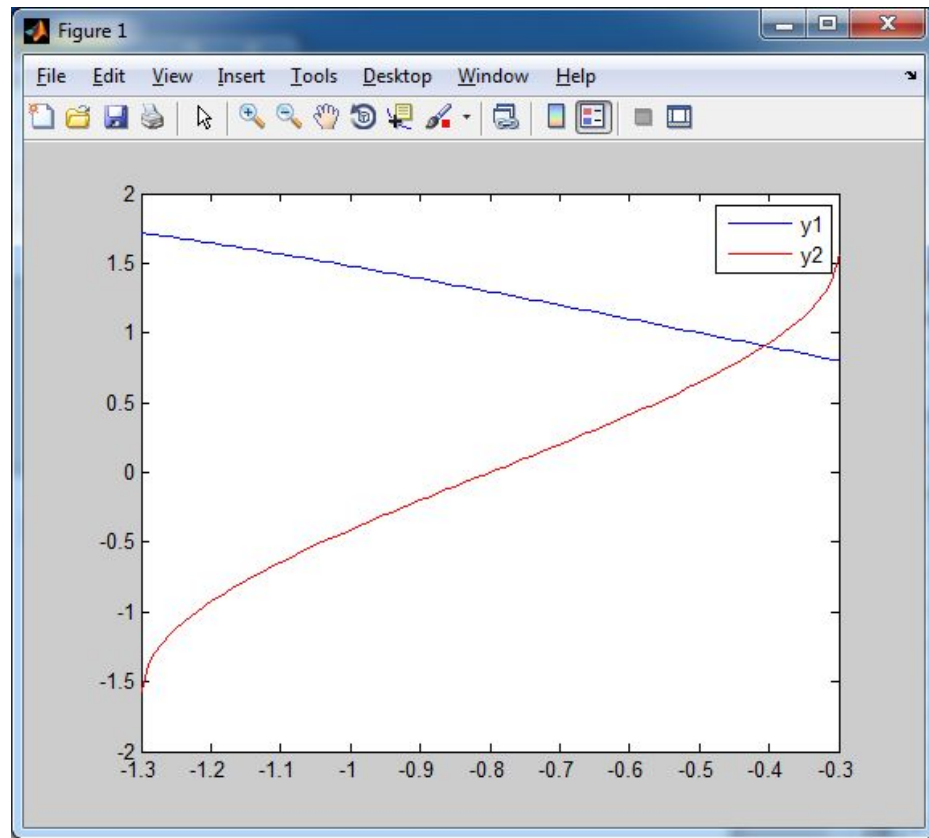
solve

Решение систем нелинейных уравнений



$$\begin{cases} \sin(x+0.5)+y=1 \\ \sin(y)-2*x=1.6 \end{cases}$$

1. Строим графики
 $y1=1-\sin(x+.5)$
 $y2=\arcsin(1.6+2*x)$



2. Используем **fsolve**, задав **m**-функцию для системы и начальное приближение **[-1 1]**
3. Решаем систему с помощью **solve**