



Scientific Computer Software (*MatLab*)

Topic 8. Elements of differential and integral calculus

Курбатова Наталья Викторовна, к.ф.-м.н.,
доцент кафедры математического
моделирования, мехмат, ЮФУ



Contents:

1) Interpolation

2) Differentiation

- a) Analytical differentiation
- b) Numerical differentiation
- c) Taylor and Fourier series
- d) Solving applied problems (i.e. pendulum)

3) Integration:

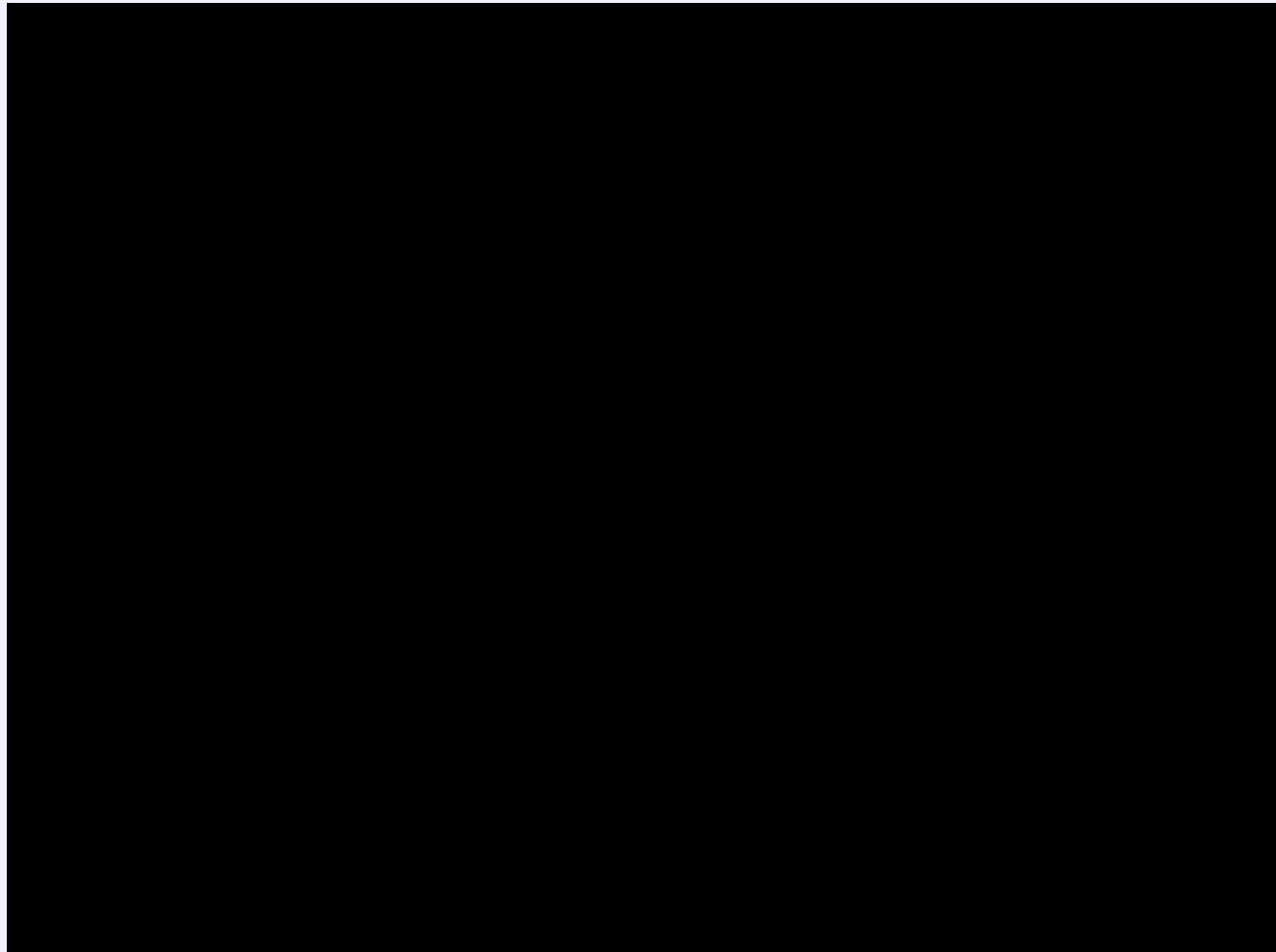
- a) Analytical integration Numerical integration.
- b) Trapezoid, Newton's methods, etc.
- c) Calculating the areas under the curve, inside the intersection of curves

4) Special operators



Почему целесообразны подгруппы?

Принцип МЕТРОНОМА в социуме. Мирзакарим Норбеков





Derivative and numeric differentiation in ML

Let function $y=f(x)$ is calculated in the points $\{x_i\}$: $x_{i+1} = x_i + h$ (h – constant) chosen on interval $[a, b]$ and $a = x_1$, $b = x_n$.

Numeric increment of a function on each subinterval $[x_i, x_{i+1}]$ is equal to $f(x_{i+1}) - f(x_i)$.

Derivative of function:

continuous case:

$$f' = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h},$$

numeric case:

$$y = \left\{ \frac{f(x_i+h) - f(x_i)}{h} \right\}_{i=1}^n, \quad h \text{ -small}$$

ML code: functions increment you can calculate using function **diff**

```
X=a:h:b; F=f(X);
```

```
Y=diff(F) % increment calculation
```

```
G=diff(Y) /h % numeric analogue of the first order derivative f'
```

```
R=diff(G) /h % analogue of the second order derivative f''
```

What about length of vectors Y,G,R? Continuous numeric



Case of Y - matrix

$[m,n]=\text{size}(Y) \rightarrow [m-1,n]=\text{size}(\text{diff}(Y))$

$f(x), g(x), p(x)$

$X=[x_1; x_2; \dots; x_n]$ - column

$Y=[f(x_i), g(x_i), p(x_i)];$

The first order derivatives for three functions:

$\text{diff}(Y) / h$



High order derivative. Size control!

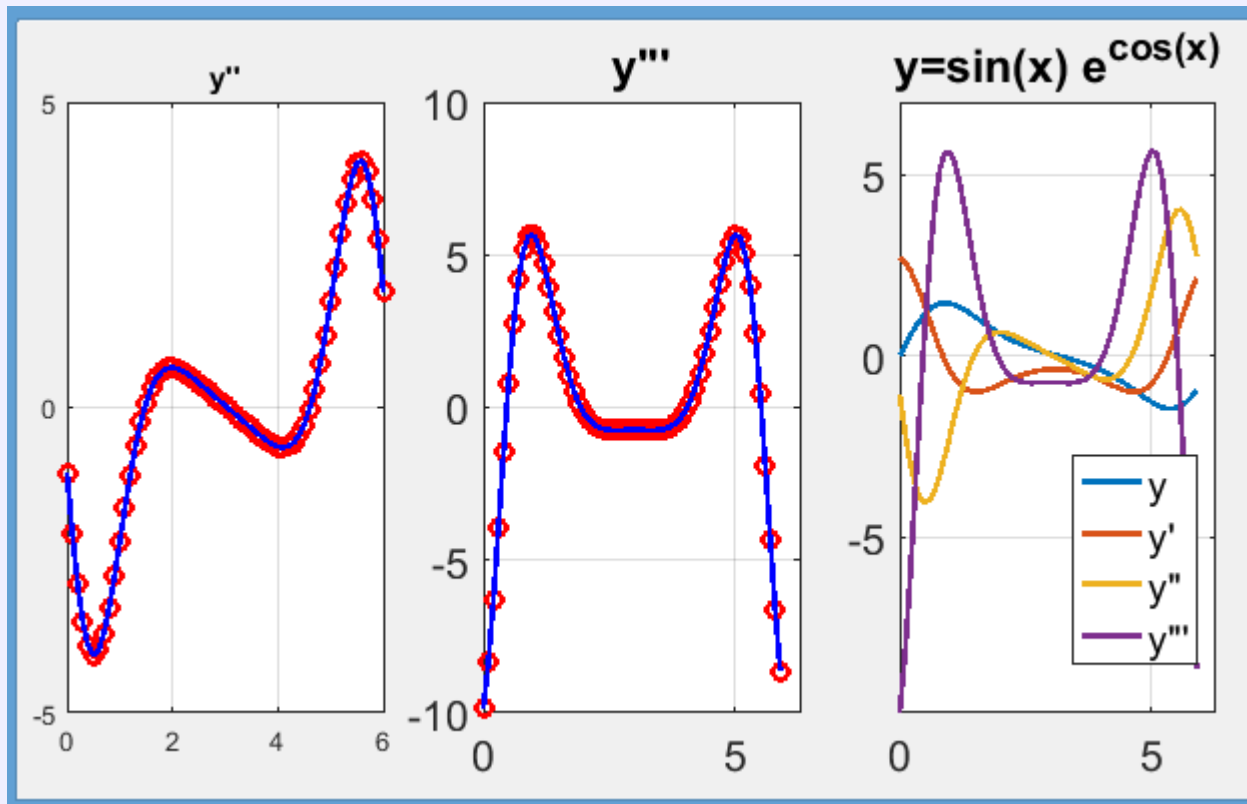
`diff(Y,n) – high order function increment`

`dim<n!`

`result= diff(Y,n), isempty(result)`



Numerical differentiation in pictures

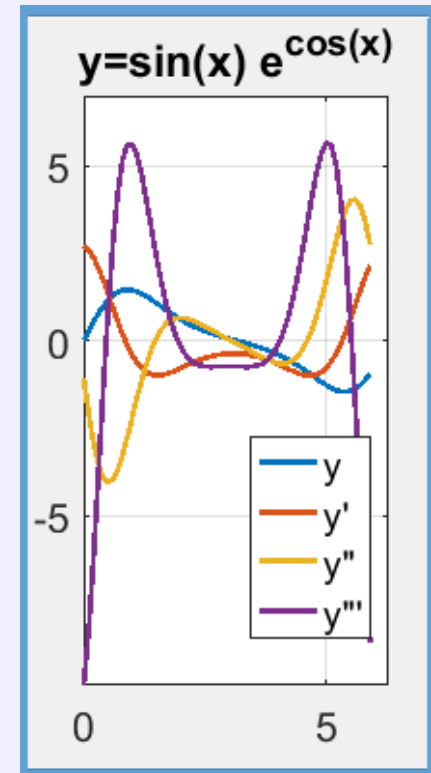


- Explain how the changes in the design of the contents of the windows, as well as the axes, were obtained?
- How to build all the graphs at the same time, window 3?
- What is the relative size of the vectors y , y' , y'' , y''' and which one should be chosen when simultaneously constructing y , y' , y'' , y''' ?
- Let the vector x be of length n .



An example of numerical differentiation

```
1) x=0:0.1:2*pi; n=length(x); h=0.1;
2) f=sin(x).*exp(cos(x))
3) fprime=diff(f,1)/h; fprime2=diff(f,2)/h^2;
4) fprime3=diff(f,3)/h^3;
5) subplot(1,3,3)
6) xnew=x(end-3:-1:1)' % row or column?
7) fNew=f(end-3:-1:1)' % explain index!
8) fprimeNew=fprime(end-2:-1:1)'
9) fprime2New=fprime2(end-1:-1:1)' % Why so?
10) fprime3New=fprime3(end:-1:1)'
11) YY=[fNew,fprimeNew,fprime2New,fprime3New]
12) pl=plot(xnew,YY)
13) set(pl,'linewidth',2)
14) set(gca,'fontsize',16,'xlim',[0, 2*pi],...
    'ylim',[min(min(YY)), 7])
15) leg=legend('y','y''','y''''','y''''''')
16) set(leg,'fontsize',14,'location','best')
17) title('y=sin(x) e^{cos(x)}'), grid on
```



Explain: line 12 и plot(YY)!



Analytical differentiation

- Analytical differentiation is performed in the **SYMBOL** class
- All variables are declared, i.e.

`syms var1 var2 var3;`

- When declaring variables, they are separated by spaces (not commas)!
- The function is constructed according to the rules of **MatLab** (external, subfunction, anonymous)
- The function can be specified in the syntax **Maple:**

`Namefunction(args)= expression(args)`

```
syms x y, z; % What is equal to r?  
f=cos(x)*y^2*exp(z), r=diff(f,2,z)
```



Class SYMBOL. An example of differentiation

```
clear; syms x y
```

I. **f** is created according Maple rules:

```
%f(x,y)=sin(x)*y*exp(cos(x))
```

```
g=diff(f,x,3)
```

```
fn=subs(g,y,2) % ~ subs in Maple
```

figure

% the range of the argument is specified:

```
fplot(fn, [0,2*pi] )
```

```
lg=legend('f'''''); set(lg,'fontsize',14)
```

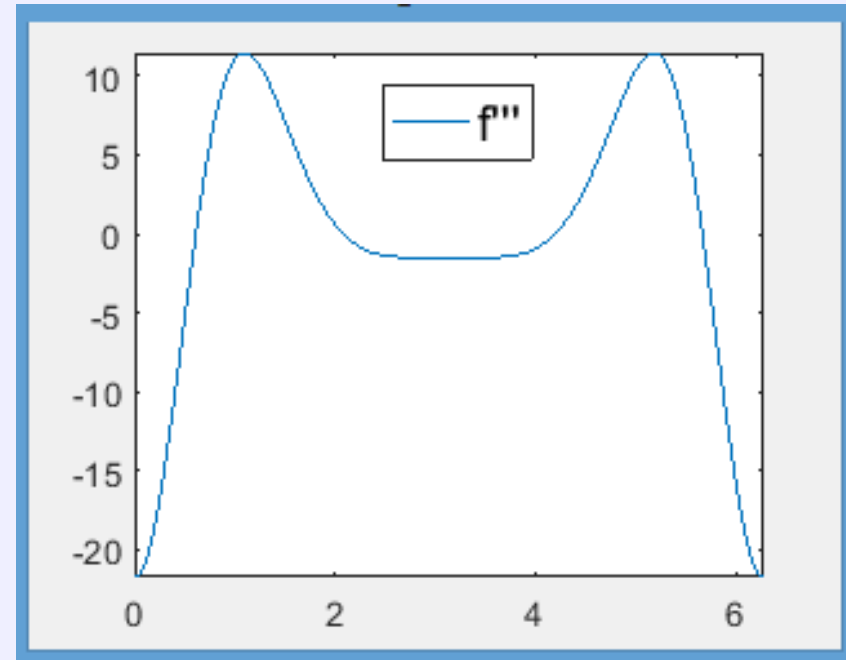
II. **f** is set as a subfunction :

```
function g=f(x,y)
```

```
syms x y
```

```
g=sin(x)*y*exp(cos(x))
```

```
end
```



Syntax:

NewFunc=diff(Func,Args,order)

Func – symbol fuction

Args – variable of differentiation

order – order of differentiation



Taylor series

nvkurbatova@sfedu.ru

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \dots$$
$$\dots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n + R_n(x)$$
$$R_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1}, \quad \xi \in (x_0, x)$$

- In 1715, it was again received **by Brooke Taylor** (already!!! it was used in the XIV century in India, as well as in the XVII century, including by Newton).
- In the period (1698-1746), **Colin Maclaurin**, a Scottish mathematician, applied and popularized the case $x_0=0$
- **Joseph Louis Lagrange** — French mathematician (1736-1813) proposed the representation of the residual term.
- MAtLab use **Maclaurin** decomposition **default** $x_0=0$ and **Taylor** — $x_0=a$

Why do we need function decomposition as a series?



Syntax taylor decomposition in MatLab

nvkurbatova@sfedu.ru

taylor(fun,var,Name,Value)

fun = symbolic function | symbolic expression

var= (default) if you do not specify a variable, the system will find the first one by itself `symvar(fun,1)` and it will spread it out at a point, otherwise we set it explicitly

'ExpansionPoint' `x0= 0` (default) | number | symbolic number | symbolic variable | symbolic function | symbolic expression

1) 'Order' = 6 (default) | positive integer | symbolic positive integer

2) 'OrderMode' = 'absolute' (default) | 'relative'
for polynomial approximation

Value according to the need and the value options described in 1)-3)

ML functions of Symbolic class you can find using

>>methods(sym)



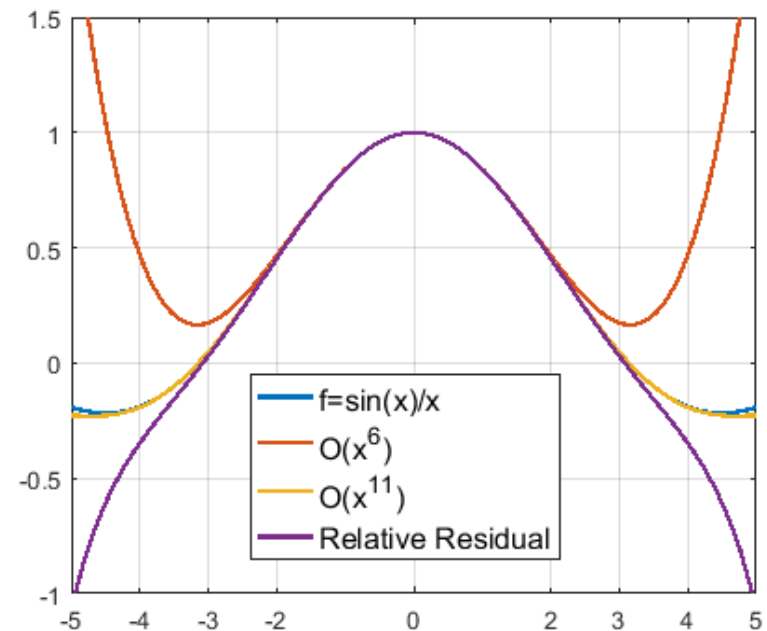
An example of a strategy for choosing Value

- 1) `clear; syms x`
- 2) `f = sin(x)/x`
- 3) `t6 = taylor(f, x) , % t9=... ???`
- 4) `tr=taylor(f, x, 'OrderMode', 'relative','order',7)`
- 5) `group=fplot([f t6 t9 tr ],'Linewidth',1.7)`
- 6) `xlim([-5 5]), ylim([-1 1.5]),`
- 7) `set(gca,'xtick',[-5 -4 -3 -2 0 2 3 4 5])`
- 8) `lg=legend('f=sin(x)/x','O(x^6)',...
'O(x^{11})','RelativeSpec')`
- 9) `set(lg,'fontsize',12), grid on`

**Can we use it for integration
during pre-decomposition?**

**What exactly is the
strategy?**

**We analyze the code and
graphs!**





>>odeexamples(SECTION)

SECTION = 'ode' | 'dae' | 'ide' | 'dde' | 'bvp' | 'pde'

Differential-Algebraic Equations

Ordinary Differential Equations

Differential-Algebraic Equations

Implicit Differential Equations

Delay Differential Equations

Boundary Value Problems

Partial Differential Equations

Differential Equations Examples

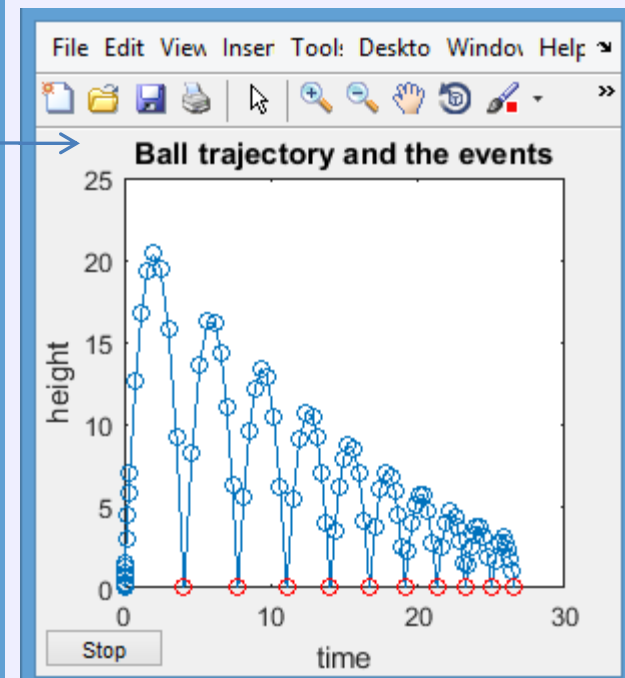
Examples of **Ordinary Differential Equations**

ode

- ballode - Simple event location (trajectory of a bouncing ball)
- batonode - ODE with time- and state-dependent mass matrix
- brussode - Stiff large problem (diffusion in a chemical reaction)
- burgersode - ODE with strongly state-dependent mass matrix
- femlode - Stiff problem with a time-dependent mass matrix
- fem2ode - Stiff problem with a constant mass matrix
- hblode - Stiff problem solved on a very long interval
- kneecode - The "knee problem" with non-negativity constraints
- orbitode - Advanced event location (restricted three body problem)
- rigidode - Nonstiff problem (Euler equations of motion)
- vdpcode - Stiff problem (van der Pol equation)

View Code Run Example

Close





Assignment of sections of thematic groups in the resource **odeexamples**

- 'dae' – differential algebraic equations (DAE)
- 'ide' – implicit methods for solving the DAE problem
- 'dde' – differential equations with time delay
- 'bvp' – DAE under specified boundary conditions
- 'pde' – DAE in a partial differential equation

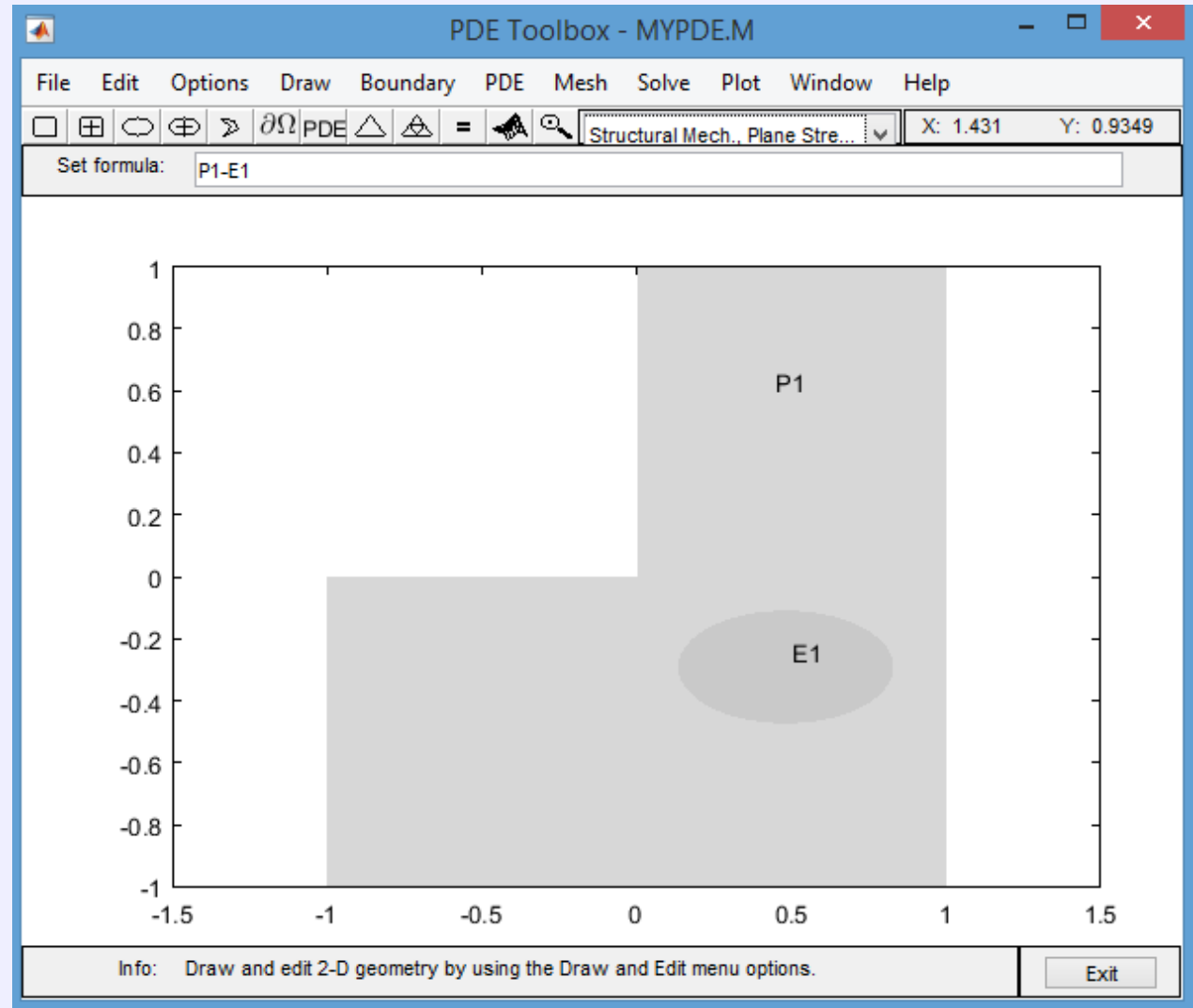


>> pdetool

nvkurbatova@sfedu.ru

PDE of 2order:

- Choosing the scope of application
- Constructing the PDE by setting the parameters
- Creating a geometry
- Boundary conditions
- Triangulate
- Improving triangulation as needed
- Solving
- Visualize the result





Class SYMBOL. Integration

nvkurbatova@sfedu.ru

Function given as subfunction

```
arg=symvar(func); res=int(func)
res2=int(int(func,arg(1)),arg(3))
resDef=
...int(int(func,arg(1),0,pi),arg(3),1,2)
function f=func
syms x y z
f=exp(y)*sin(x)/z
end
```

Func – according to Maple rules

```
f=exp(y)*sin(x)/z
arg=symvar(f)
res=int(f)
res2=int(int(f,arg(1)),arg(3))
resDef=int(int(f,arg(1),0,pi),arg(3),1,2)
```

Syntax:

Res=int(Func,Args)

Res=int(Func,Args,a,b)

Args=symvar(f)

Func – symbol function

Args – integration variable

a,b – limits

Note:

If no integration variable is specified, then integration is only based on the first variable

Res=int(Func)



Numeric integration

Syntax:

Z = trapz(X,Y,DIM) -

the trapezoid method

X is the value of the arguments (vector)

Y is a matrix or vector of values

DIM indicator (1 ∨ 2) what does it mean?

Q = quad(FUN,a,b,tol,trace) –

Simpson method

FUN – *external function or anonymous we build according to the rules of vector operations!*

a,b – **explain?**

trace>0 – output of intermediate values

The accuracy (tol) of the Simpson method in ML default: 1.e-6

Universal Function:

Res=integral(Func,Args)

Res=int(Func,Args,a,b) % may be old

Res=integral2(Func,Args,lims)

Res=integral3(Func,Args,lims)

Args=symvar(Func)

Func – symbol function

Args – no integration variable

a,b ∨ lims – limits of integration

Note

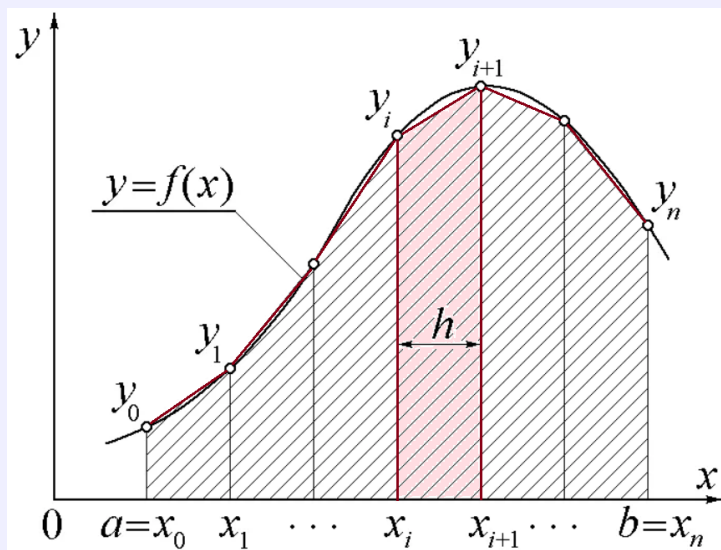
If the integration variable is not specified, then integration is based on the first (in the sense of the ASCII code) of the variable

Res=integral(Func)



Trapezoid and Simpson formulas

Trapezoid:



$$\begin{aligned} \int_a^b f(x) dx &\approx \frac{b-a}{2N} \sum_{n=1}^N (f(x_n) + f(x_{n+1})) \\ &= \frac{b-a}{2N} [f(x_1) + 2f(x_2) + \dots + 2f(x_N) + f(x_{N+1})] \\ h &= \frac{b-a}{N} \end{aligned}$$

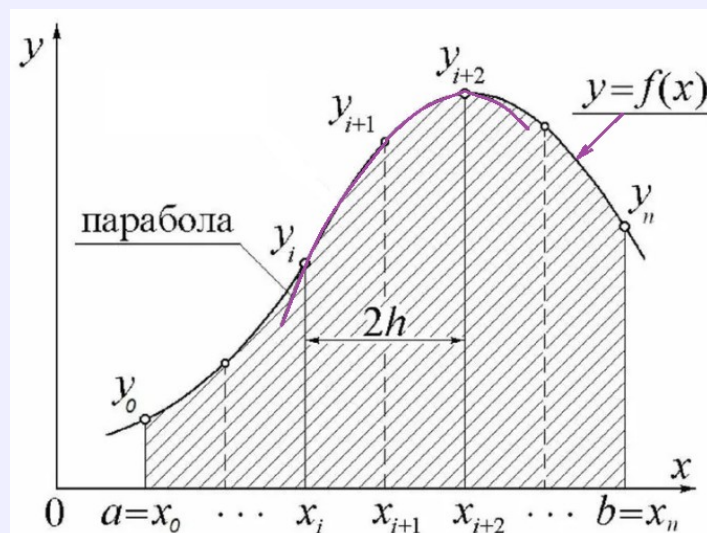
Simpson

$$I \approx \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right).$$

если $2N$ равных частей на $[a, b]$:

$$I \approx \frac{b-a}{6N} (f_0 + 4(f_1 + f_3 + \dots + f_{2N-1}) + \dots + 2(f_2 + f_4 + \dots + f_{2N-2}) + f_{2N}),$$

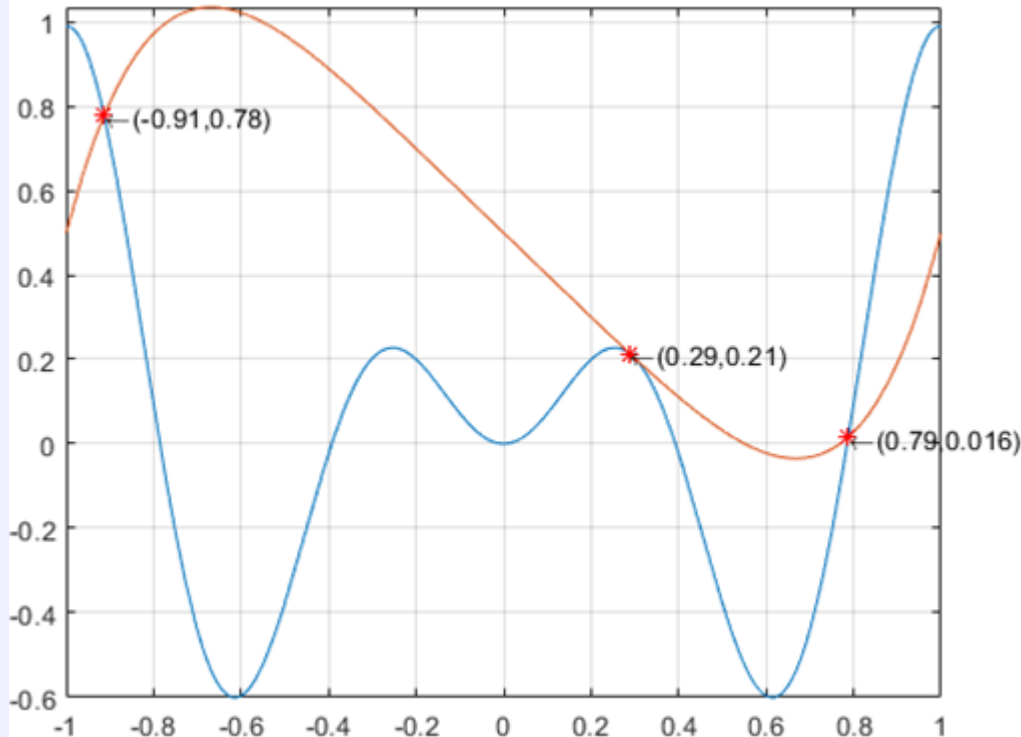
$$f_i = f\left(a + \frac{(b-a)i}{2N}\right).$$





Application. Calculate the area enclosed between the curves

nvkurbatova@sfedu.ru



Think about the solution algorithm!

pause(10)

Have you thought about it?

Пример.

Example

```
Clear, figure
%% строим графики
fplot('x*sin(8*x)', [-1 1]), hold on,
fplot('(x^5-x+0.5)', [-1 1]), grid on;
%% решаем, добавляем точки пересечения
x0=[-0.8, 0.65; -0.2, 0; 0.6, 0] % нулевые
приближения
for i=1:length(x0')
    [x{i}, Fx]=fsolve(@F, x0(i, :))
    plot(x{i}(1), x{i}(2), 'r*')
    xy=[ '(', num2str(x{i}(1), 2), ', ', '...',
        num2str(x{i}(2), 2), ') ' ]
    text(x{i}(1), x{i}(2), ['\leftarrow' xy])
end % for
function f=F(x) %% подфункция в этом же файле
f(1)=x(2)-x(1)*sin(8*x(1))
f(2)=x(2)-(x(1)^5-x(1)+0.5)
end
```

Is the point (0.29,0.21) a touch point?

Why is this important?

Where do we start solving the problem?



Спасибо за внимание!

"What the vessel is filled with, it pours out of it!"

Russian folk proverb

And why suddenly **Russian** proverbs?