Software Packages | Euler Method

Assignment_03

Name: Muhammad Arslan   Roll No: 201408

Date: 20-Nov-2023

Question: Write the matlab code for Euler Method and solve the first order ODE with given initial value by using euler method (numerical procedure).

$$n = 10$$

$$dy/dx = sin(xy), \qquad y(0) = \pi$$

in the given interval $x \epsilon [0, 1]$.

Solution:

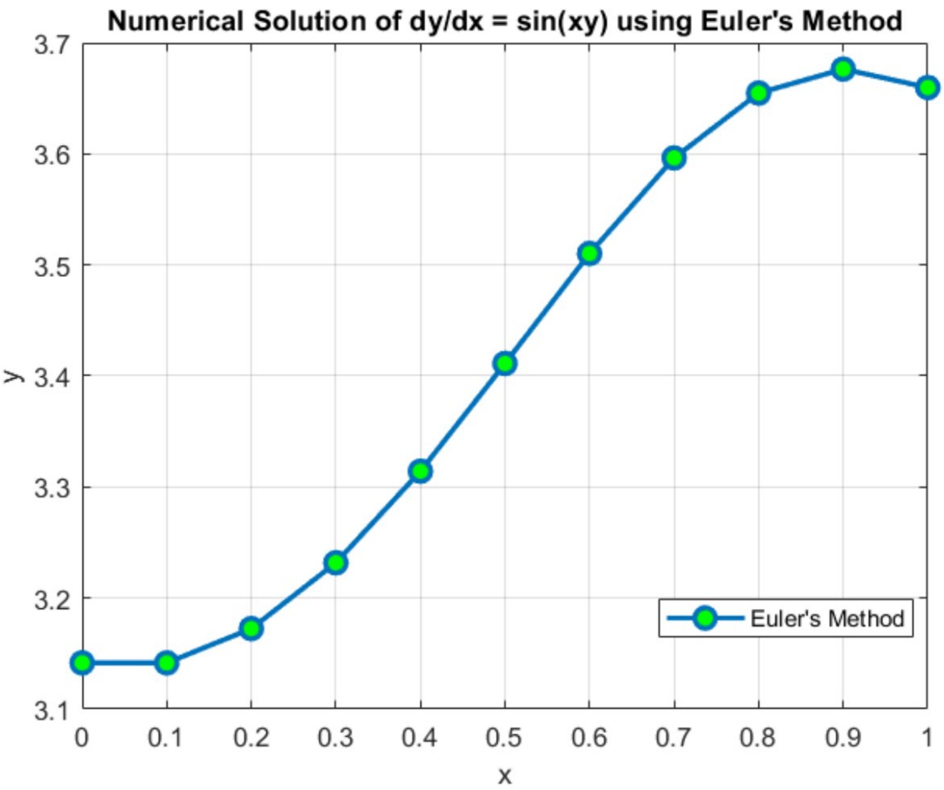
step size (h)   => $\Delta x = \frac{x_n - x_0}{n}$

```
h = (1-0)/10; % step size
x=0:h:1; % range of x
y=zeros(size(x)); % allocate the result y
y(1)=pi; % initial value of y
n=numel(y); % number of y values

for i = 1:n-1
    dydx = sin(x(i)*y(i));
    y(i+1) = y(i) + dydx*h;
    fprintf('="Y"\n\t   %0.01f', y(i));
end
```

```
="Y"
    3.1="Y"
    3.1="Y"
    3.2="Y"
    3.2="Y"
    3.3="Y"
    3.4="Y"
    3.5="Y"
    3.6="Y"
    3.7="Y"
    3.7
```

```
%%fprintf('="Y"\n\t   %0.01f',y);

% Plotting with labels, title, legend, and markers
figure;
plot(x, y, '-o', 'LineWidth', 2, 'MarkerSize', 8, 'MarkerFaceColor', 'g');
```

```
grid on;
xlabel('x');
ylabel('y');
title('Numerical Solution of dy/dx = sin(xy) using Euler''s Method');
legend('Euler''s Method', 'Location', 'Best');
```



Quiz_03 Boundary Value Problem (bvp4c)

Name: Muhammad Arslan Roll No: 201408

Date: 30_Nov_2023

Statement:

$$\frac{d^2y}{dx^2} + y = 0$$

$$y(0) = 1, \qquad y(\pi) = 0$$

Solution:

Defining a deviv function

Quiz_03!23_11_2023.mlx | deviv.m

```
function dydx = deviv( x, y)
% Reduciton
%  2nd order ODE to 1st order ODE
dydx(1) = y(2);
dydx(2) = -y(1);
end
```

Defining the boundary condition function

Quiz_03!23_11_2023.mlx | deviv.m | bcs.m

```
function res = bcs(ya, yb)
res = [ya(2)-1 yb(1)];
end
```

Using bvpinit for the initial guess on the interval

```
Solinit = bvpinit([0,pi],[0,0])
```

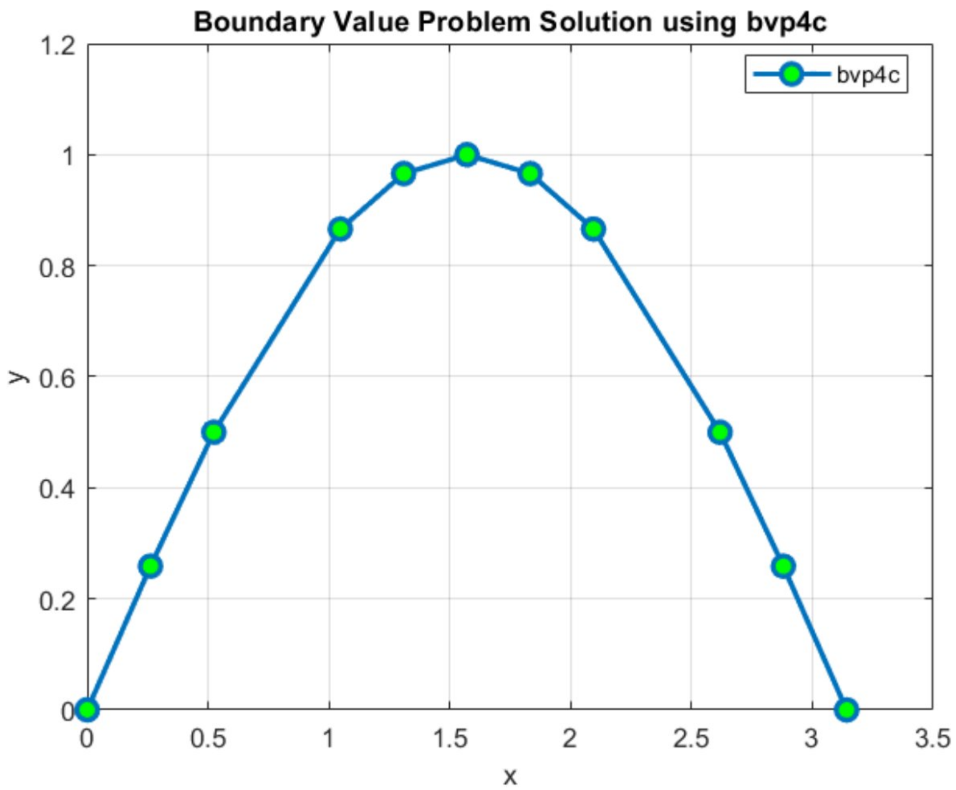
```
Solinit = struct with fields:
    solver: 'bvpinit'
         x: [0 3.1416]
         y: [2×2 double]
     yinit: [0 0]
```

bvp4c Iteration for the solution of non-linear system of equations

```
%bvp4c
Sol = bvp4c(@deviv, @bcs, Solinit);
```

```
figure ;
plot(Sol.x, Sol.y(1,:),'-o', 'LineWidth',2 ,'MarkerSize', 8, 'MarkerFaceColor','g')
grid on;
```

```
xlabel('x')
ylabel('y')
title('Boundary Value Problem Solution using bvp4c')
legend('bvp4c', 'Location', 'Best')
```



Notes_Linear Shooting Method:

- Linear
- For 2^{nd} Order Differential Equation

$$y'' = P(x)y' + q(x)y + r(x)\,, \quad a \le x \le b$$

$$y(a) = \alpha, \quad y(b) = \beta$$

Note: We've to construct two functions for solving this sutting method

Limitiations:

- Only for linear but there is another method for nonlinear shooting method
- 2nd Order Differential Equation

<u>`To solve this problem, convert in 2 initial value problem`</u>

IVP- i

$$y''_1 = p(x)y'_1 + q(x)y_1 + r(x), \qquad a \le x \le b$$

$$y_1(a) = \alpha, y'_1(a) = 0$$

IVP- ii

$$y''_2 = p(x)y'_2 + q(x)y_2\,, \qquad a \le x \le b$$

$$y_2(a) = 0, \quad y'_2(a) = 1$$

The the sol of bvp is:

ode45: back rk-method

$$y = y_1 + \frac{\beta - y_1(b)}{y_2(b)}(y_2)$$

Assignment_04 Linear Shooting Method | Software Packages

Name: Muhammad Arslan Roll No: 201408

Problem: Solve this problem using linear shooting method...

$$y'' = y' + 2y + \cos(x), 0 \le x \le \frac{\pi}{2}$$

$$y(0) = -0.3, \quad y\left(\frac{\pi}{2}\right) = -0.1$$

```
% y(a) = alpha | y(b) = beta
```

Exact Solution:

$$y = -0.1\sin(x) + 3\cos(x)$$

Step 1: Converting two 2nd order odes to initial value problems and then solve using ode45

Editor - C:\Users\Ghost \Downloads\New folder\deriv0.m

Ass4_30_Nov_23.mlx | deriv0.m | deriv1.m | +

```
function dy = deriv0(x, y)
% Conversion
% From 2nd order ode to 1st order ode
dy = zeros(2,1);
dy(1) = y(2);
dy(2) = y(2) + 2*y(1) + cos(x);
end
```

```
xspan = [0, pi/2];
y0 = [-0.3, 0];
[x, y01] = ode45(@deriv0, xspan, y0) ;
```

Editor - C:\Users\Ghost \Downloads\New folder\deriv1.m

Ass4_30_Nov_23.mlx | deriv0.m | deriv1.m | +

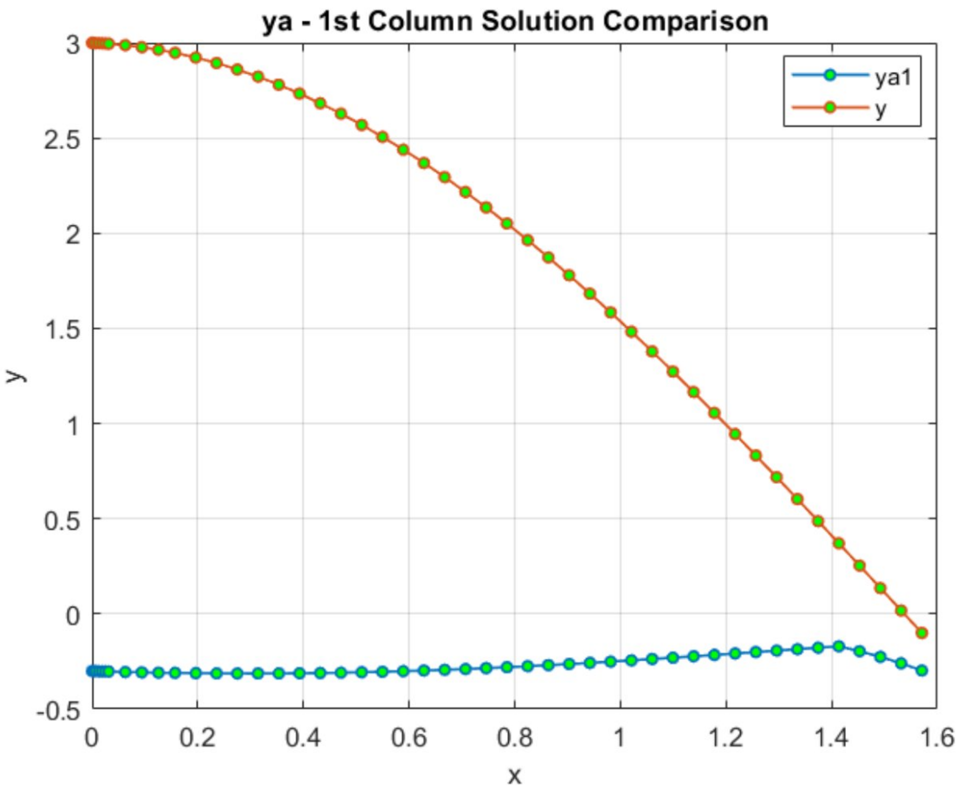
```
function dy2 = deriv1(x, y)
% Conversion
% From 2nd order ode to 1st order ode
dy2 = zeros(2,1);
dy2(1) = y(2);
dy2(2) = y(2) + 2*y(1);
end
```

```
y1 = [0 , 1];
[x, y02] = ode45(@deriv1, xspan, y1) ;

Beta =- 0.1;
ya = (y01) + (((Beta - y01(end))/y02 (end)) *y02) ;
```

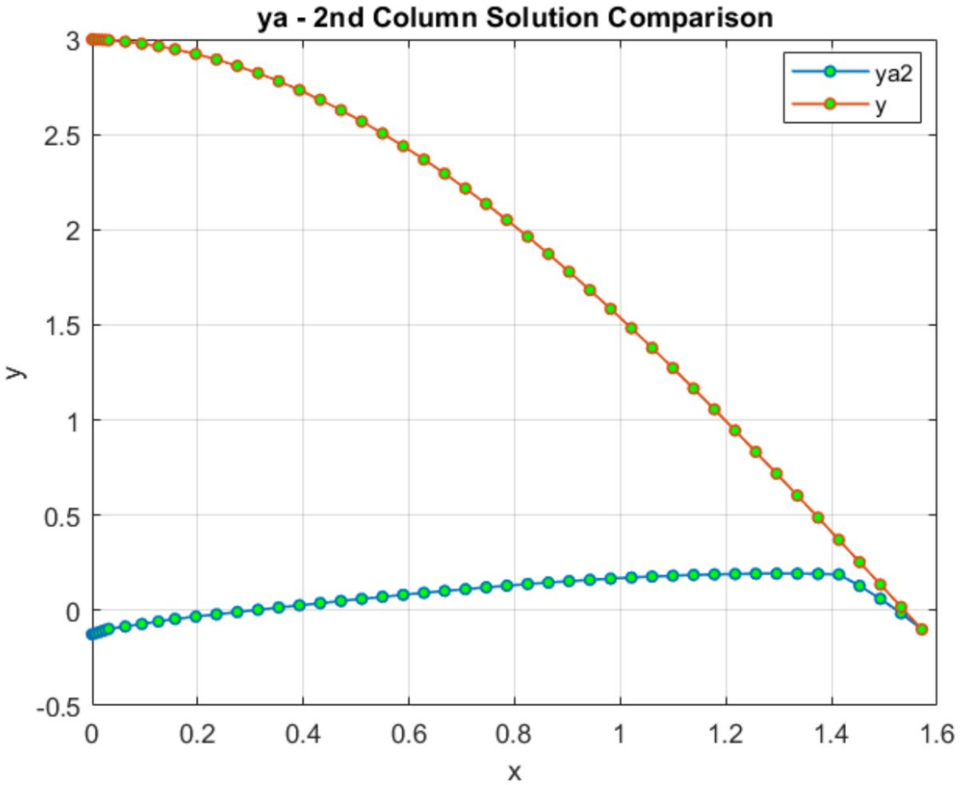
Step 2: Comparison of the Exact solution vs Evaluated Solution

```
y = -0.1*sin(x) + 3*cos(x);

plot (x, ya(:,1), '-o', 'LineWidth',1 ,'MarkerSize', 4, 'MarkerFaceColor','g')
hold on
plot (x, y, '-o', 'LineWidth',1 ,'MarkerSize', 4, 'MarkerFaceColor','g')

grid on;
```

```
xlabel('x')
ylabel('y')
title('ya - 1st Column Solution Comparison')
legend('ya1','y', 'Location', 'Best')
hold off
```




```
plot (x, ya(:,2), '-o', 'LineWidth',1 ,'MarkerSize', 4, 'MarkerFaceColor','g')
hold on
plot (x, y, '-o', 'LineWidth',1 ,'MarkerSize', 4, 'MarkerFaceColor','g')
grid on;
xlabel('x')
ylabel('y')
title('ya - 2nd Column Solution Comparison')
legend('ya2','y', 'Location', 'Best')
hold off
```


ya - 2nd Column Solution Comparison
ya2
y
y
x
3
2.5
2
1.5
1
0.5
0
-0.5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6

PDEs in 2 dim space:

(1) Elliptic: $\nabla(c\,\nabla u) + au - f$

$\nabla^2 u = 0$ Laplace eqn

$AU_{xx} + BU_{xy} + CU_{yy} + DU_x + EU_y + FU - G$

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$$

$\vec{\nabla}\cdot\vec{A}$ = Divergence of vector A

∇f = Gradient of scalar f.

(2) Parabolic: $dU_t - \nabla.(c\vec{\nabla}u) + au = f$

$\nabla^2 u - U_t = 0 \Rightarrow$ Heat eqn

(3) Hyperbolic: $dU_{tt} - \nabla.(c\nabla u) + au = f$

$\Rightarrow$ wave eqn

PDE : Matlab to built in ftn.

Example of elliptical PDE

Consider Poisson's eqn on a rectangle.

$u(x,y) \in [0,2]\times[0,1]$

$U_{xx} + U_{yy} = x^2 + y^2$

$u(x,0) = x \quad ; \quad u(x,1) = 1$

$u(0,y) = y \quad ; \quad u(2,y) = 1$

Dirichlet → on u.

N-bdd → norm

On command window

Type : pdetool

Turning grid on : Select options, Grid

Increase domain window : options, Axes limits on

x-axis [-0.5, 2.5]

thicks edges ⇒

y-axis [-0.5, 1.5] apply and close.

Specify to draw rectangle: click on rectangle icon at top left of menu options left click on the point (0,0) and keeping left mouse button pressed drag rectangle to point (2,1) and release it.

Convert it by double clicking, if wrong done then change by double clicking on that and write specific value of given order

Specify boundary Conditions:

Click on symbol $\partial\Omega$ or select boundary, Boundary mode or type control-b.

Color of boundary indicates type of condition i.e red = Dirichlet, blue = Neumann, green = mixed

Double click on button horizontal line = click on over written PDE and on left there is type of PDE i.e. elliptic

we get formula $hU = r$

our case ⇒ $h = 1$, $r = x$ } $u(x,0)$

Continue around boundary specifying remaining cond; then

Save as and save matlab file for future use.

For elliptic PDE:

$-\nabla\cdot(c\nabla u) + au = f$

our case $c = -1$, $a = 0$, $f(x,y) = x^2+y^2$

click on: PDE icon = elliptic

Create a triangular finite element mesh:

select Δ icon or choose mesh, Initialize mesh

we can refine mesh by choosing icon of Δ inside a triangle.

Finally solve: choose '=' icon or solve, solve PDE.

{click on plot and select ____ ?

Elliptic pdes no time involvement

Criterian for parabollic b^2 - 4ac > 0

Heat equation U_t =

Previous Task discussion Steady case no change in time eloptic pde

Boundary condition are dritchelet boundary conditions

no invelvement of derivatice

Dritchlet B.C's :(Matlab specify)

$$hu = r$$

ANeuman boundry condition

Mixed boundary condition (Combinition of both)

Noslip dritchlet = 0

flux = 0

neuman = 0

inflow = outflow

Pipe

boundary drich = 0

tab, drich = 1 neuman = 0, open (inflow = outflow)

Parabollic PDE: Lab_Task 14_Dec_23

Date: 14 Dec 2023 Thursday

Example: Consider the heat equation

$$U_t = U_{xx} + U_{yy} + \sin(t)$$

$$U(t, 0, y) = 0; U_x(t, \pi, y) = 1$$

$$U_y(t, x, 0) = 0; U(t, x, 2\pi) = x$$

$$U(0, x, y) = 0$$

Note: For unique solution because 2 derivative of space variable then we need 2 boundary conditions (left & right)

$$x = 0; x = \pi$$

Same goes for y

$$y = 0; y = 2\pi$$

For t we have initial condition

$$t = 0$$

Solution:

$$[0, \pi] x [0, 2\pi]$$

For Neumann B.C's

Matlab specify:

$$n * C * \text{grad}(U) + qU = g$$

$$\vec{n} \cdot (C(x, y)\nabla U) + q(x, y)U = g(x, y)$$

where $\vec{n}$ represents a unit vector normal to domain

For case $U_y(t, x, 0)$, we have,

$$\vec{n} = (0, -1), C(x, 0) = 1, q(x, 0) = 0, g(x, 0) = 0$$

For case $U_x(t, \pi, y) = 1$ we have

$$\vec{n} = (1, 0), C(\pi, y) = 1, q(\pi, y) = 0, g(x, 0) = 1$$

Note: Derechilt = Red , Neuman b.cs = Blue Color

General Parabolic Pde:

$$d\, U_t - \nabla \cdot (C \nabla U) + \text{ aU } = f$$

In our case,

$$d = 1, C(t, x, y) = 1, f(t, x, y) = \sin(t)$$

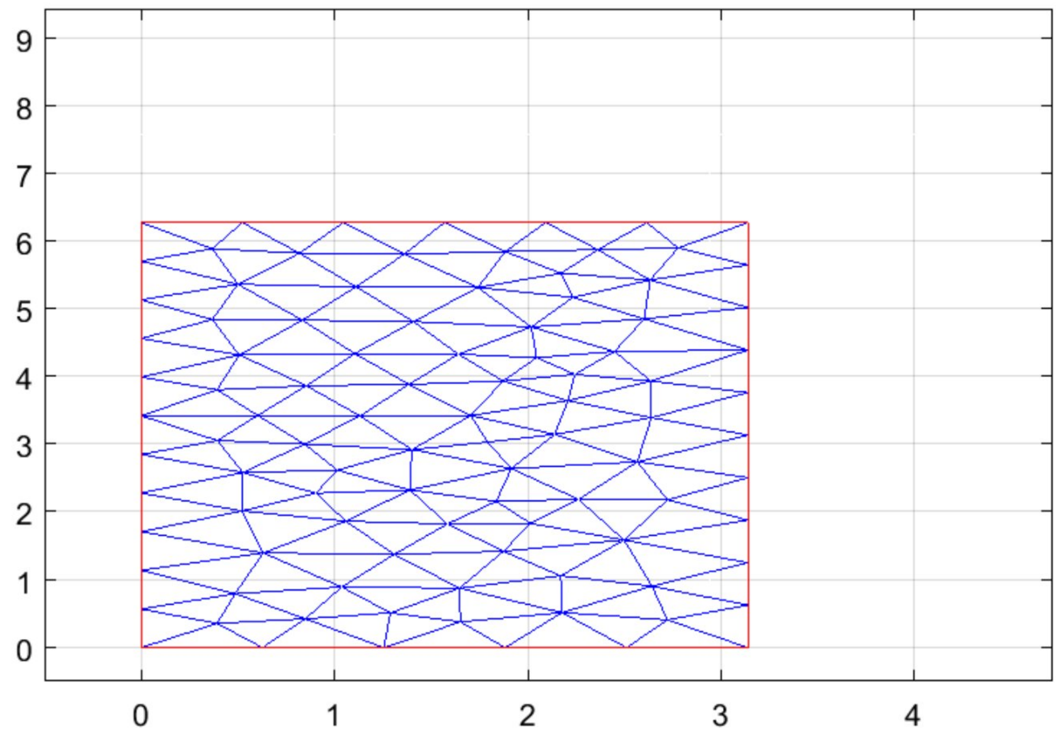
For Initial condition:

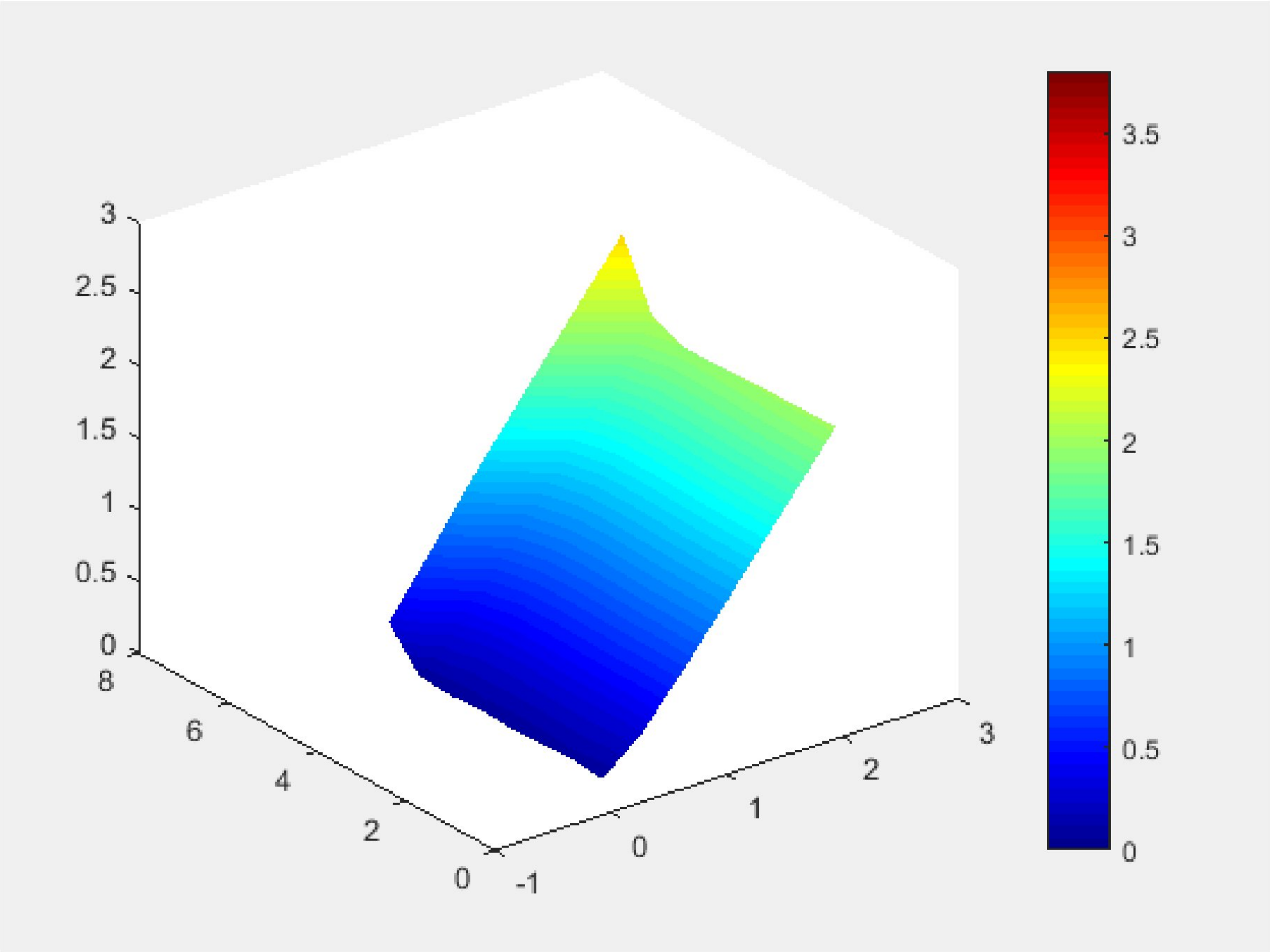
Solve, parameters, Time: U(t0)

Let type:

$$\text{linspace}(0, 5, 10)$$

for the time to run till 5 sec

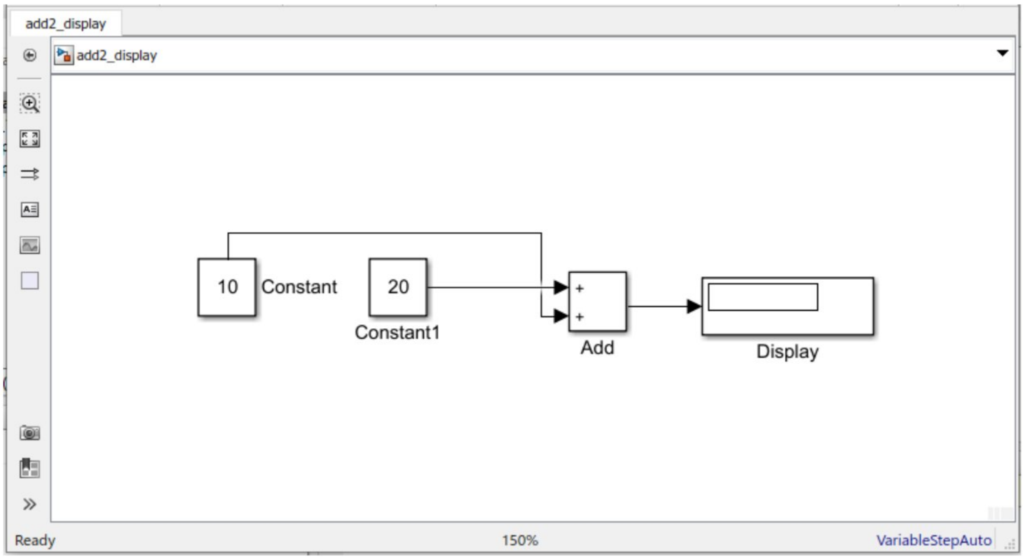





```
% write pdetool in command window
% make the domain
% option, grid on (for easy scaling)
% opetion axes limit
% x  = [-0.5 1.5*pi] for visualize actual [0, pi]
% y = [-0.5 3*pi] for visualization purpose actual [0, 2*pi]
% select the rectangle icon from the top(1) and drag
% make this exact by double clicking the rectangular domain
% alternative usign draw tab
% partial sign for boundary condition drichlet / neuman
% for pde click on PDE, select the type and give the general values
% give the initial condition by click solve, parameter and put values by comparison
% plot, parameters, check animations and plot
% check plot in x, y grid, 3d height
%(Autamitacally triangularization jet color is encouraged)
```

Simulink

Lets add two numbers



Exxample:

dy/dt = 4 sin 2t - 10 y

y(t) = \int(4 sin 2t - 10 y) dt

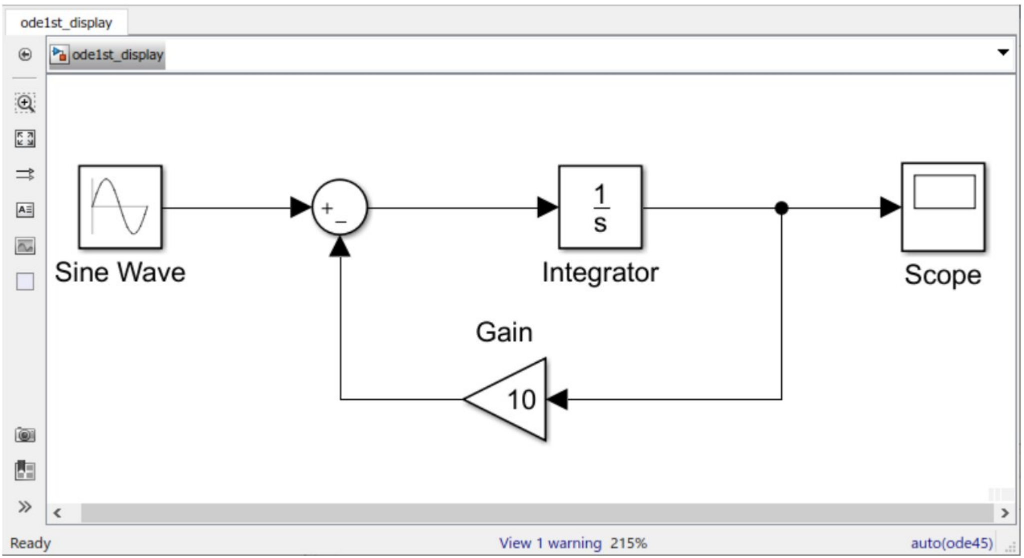
Note: For solving differential equaiton dintegrator block must

But for 2nd order double integrator block must be used

Simulink

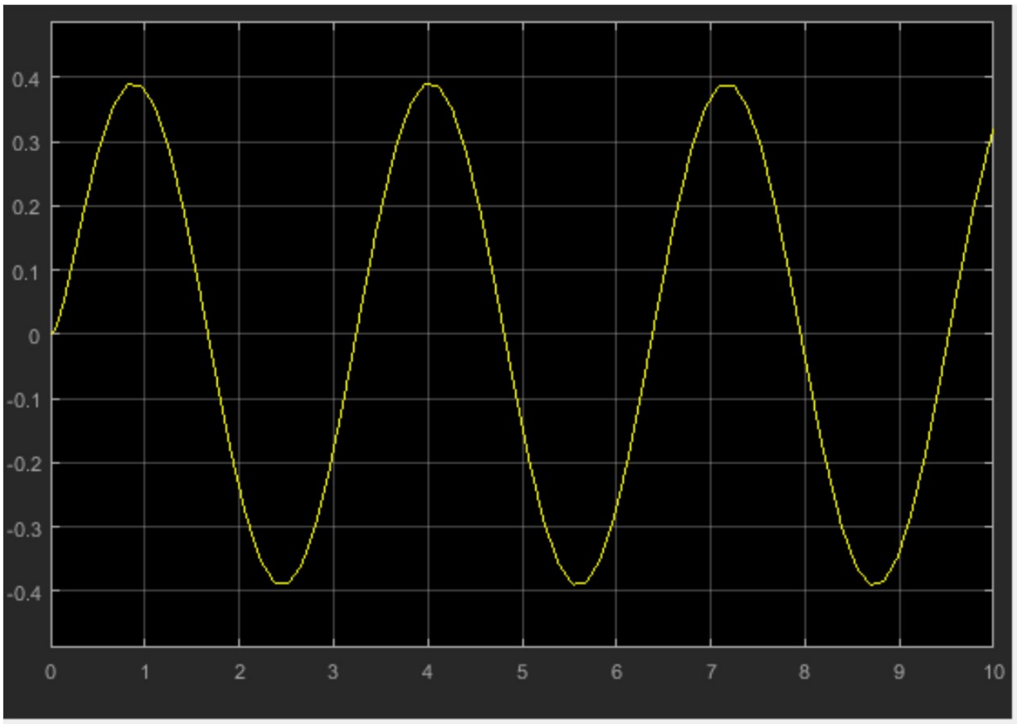
Library Browser

Sine wave aplitude 4 and frequency 2



Gain must be connected after integrator as input and add or in O input with 2 inputs and after that the source will show output

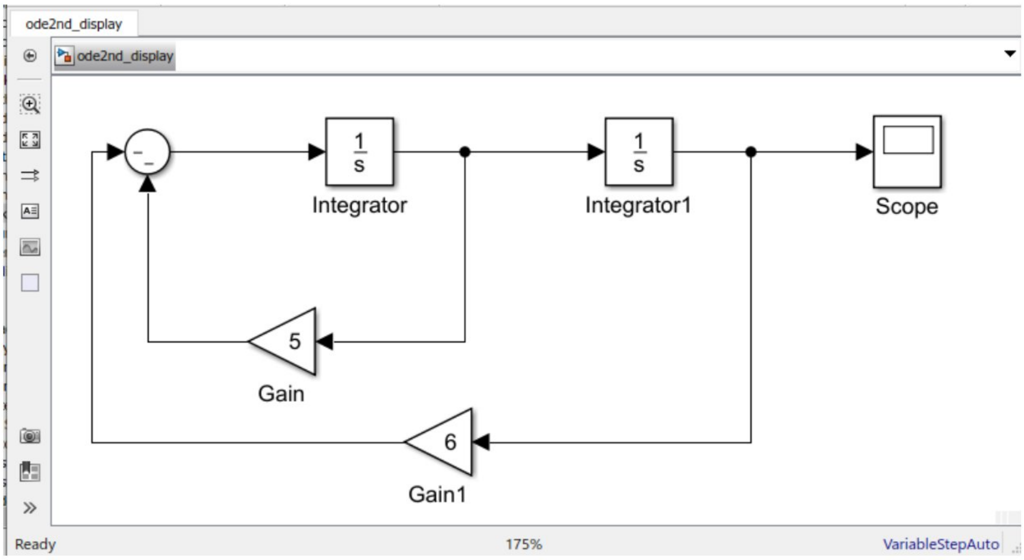
if variable change to t then sanerio change

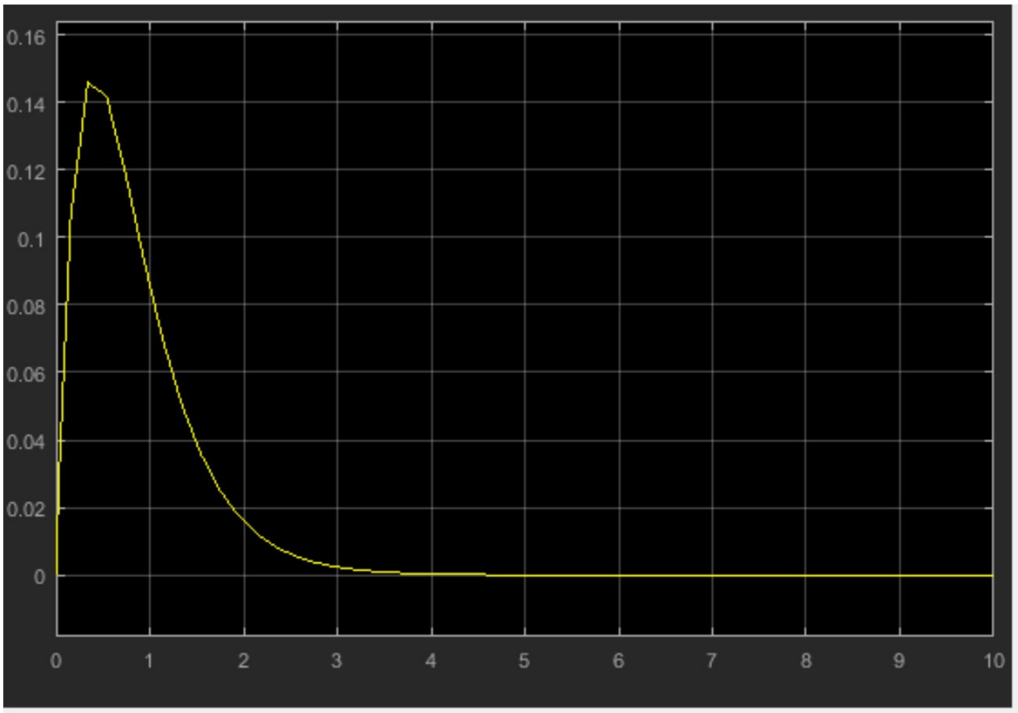


Example:

y'' + 5y' + 6y = 0, y(0) = 0, y(0) = 1

y" = - 5y' -6y





Note: if double integrator is only involved in the equaiton then only use integrator, sexcond-order

After mids only

ode45

bvp4c