

# MATLAB Review

Symbolic Math & more

# Symbolic Math in MATLAB



## License Center

[Licenses](#) | [Trials](#) | [Prereleases](#)[Contact support](#)

40941043   MATLAB  
**MATLAB (Individual)**

Annual   Academic   Total Headcount

Master License 30551759

[Manage Products](#)[Install and Activate](#)[Contact Administrator\(s\)](#)[Access MATLAB Online](#)[Download](#)[Simulink Test](#)[SoC Blockset](#)[Spreadsheet Link](#)[Stateflow](#)[Statistics and Machine Learning Toolbox](#)[Symbolic Math Toolbox](#)[System Composer](#)[System Identification Toolbox](#)

<https://www.youtube.com/watch?v=G2hzWTgVcJE>

# Symbolic Math in MATLAB

- Defining symbolic variables:

```
syms x a b c
```

- Defining an expression:

```
expr1 = x^2 + x + 1;
```

```
expr2 = a*x^2 + b*x + c;
```

- Substituting values in expressions:

```
subs(expr1,x,2)
```

```
subs(expr2,[a b c],[1 2 3])
```

# Symbolic Math in MATLAB

- Solve an equation symbolically:

```
syms x
eq=x-25==0;
sol=solve(eq,x)
```

- Solve a system of equations (for two or more unknowns):

```
syms x y
eqn1 = x^2 + y^2 == 1;
eqn2 = x - y == 0;
solutions= solve([eqn1 eqn2],[x y]);
solutions.x (%this is a structure, a special data type!)
solutions.y
```

# Symbolic Math in MATLAB

- Define a symbolic function:

```
syms a b
h(a,b) = pi*a*b
double(h(2,4))
subs(h,[a b],[2 4])
```

- Visualize a symbolic plot:

```
syms x y
g(x) = x^2
fplot(g)
f(x,y) = x^2 + y^2
fsurf(f)
```

## Introduction to Cell Arrays

A cell array is a special array of arrays. Each element of the cell array may point to a scalar, an array (matrix or vector), or another cell array.

```
c = {42, rand(5,5), [1 2 3]}
```

Cell array indexing:

```
c{2}(2,3)
```

```
c{1}
```

Very powerful tool if you want to add another dimension to a 2D matrix!

<https://www.youtube.com/watch?v=zz0iQbAzeR0>

Additional resources for plotting

# Attributes for Plot

- Hold on/off

hold on - Retains current plots when making new plots (useful for adding plots to an existing plot)

hold off - Sets the hold state to off so that new plots added to the axes clear existing plots and reset all axes properties.

```
x = linspace(0, 2*pi, 100);  
y1 = sin(x);  
y2 = cos(x);  
plot(x, y1, 'r-', 'LineWidth', 2);  
hold on;  
plot(x, y2, 'b--', 'LineWidth', 2);  
hold off;
```

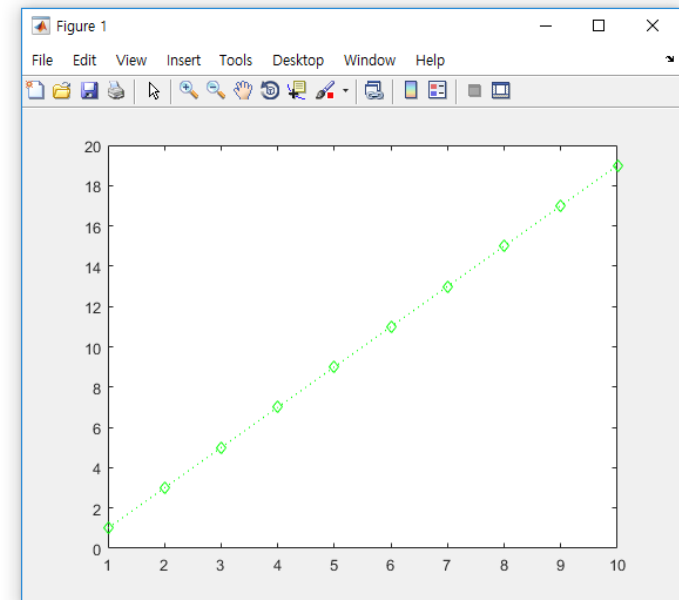


### Attributes for Plot

- Line specifiers can be added in the plot command
  - `>> plot(x,y, 'line specifiers')`

SYMBOL	COLOR	SYMBOL	LINE STYLE	SYMBOL	MARKER
k	Black	—	Solid	+	Plus sign
r	Red	--	Dashed	o	Circle
b	Blue	:	Dotted	*	Asterisk
g	Green	—.	Dash-dot	.	Point
c	Cyan	none	No line	×	Cross
m	Magenta			s	Square
y	Yellow			d	Diamond

- `>> plot(x,y,'g:d')`
  - A **green** dotted line connecting the points which are marked with diamond markers.



```
>> x=1:10; y=1:2:20;
>> plot(x,y,'g:d')
fx >>
```

## Useful 3D plot functions

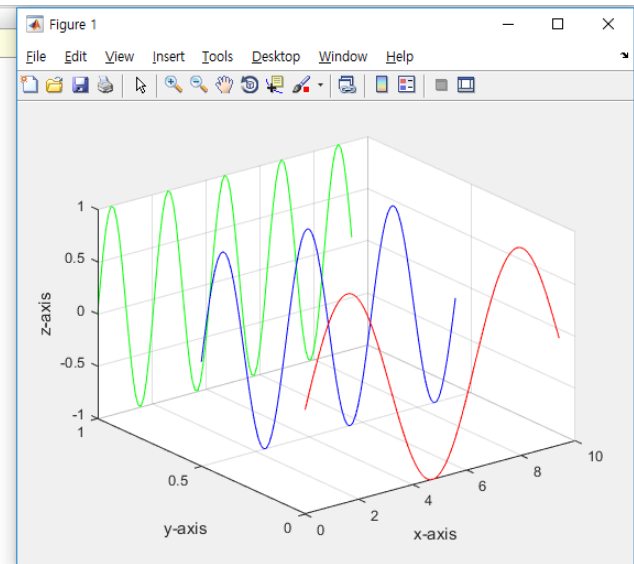
- **plot3()**: x, y, z 3D line plot
- **mesh()**: mesh plot
- **surf()**: surface plot
- **contour()**: contour plot
- **surfc()**: combination of surface with contour

**surf**(x,y,Z) and **surf**(x,y,Z,C), with two vector arguments replacing the first two matrix arguments, must have `length(x) = n` and `length(y) = m` where `[m,n] = size(Z)`. In this case, the vertices of the surface patches are the triples `(x(j), y(i), Z(i,j))`. Note that x corresponds to the columns of Z and y corresponds to the rows.

### Examples for plot3(x, y, z)

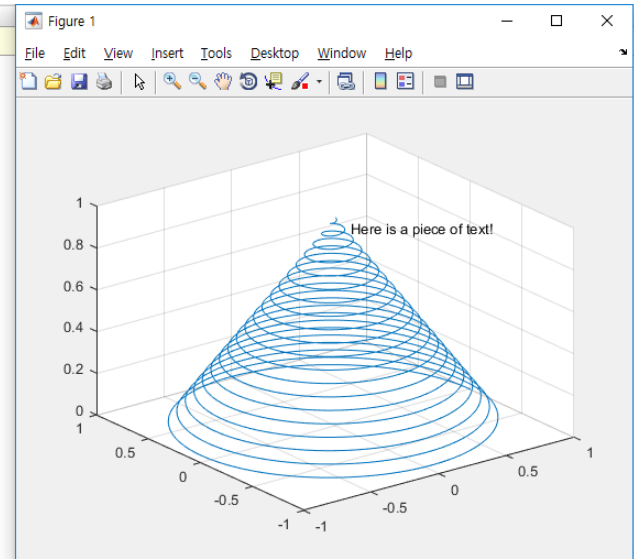
```
Command Window
New to MATLAB? See resources for Getting Started.

>> x=0:0.1:3.*pi;
z1=sin(x);
z2=sin(2.*x);
z3=sin(3.*x);
y1=zeros(size(x));
y3=ones(size(x));
y2=y3./2;
plot3(x,y1,z1,'r',x,y2,z2,'b',x,y3,z3,'g')
grid on
xlabel('x-axis'), ylabel('y-axis'), zlabel('z-axis')
fx >>
```

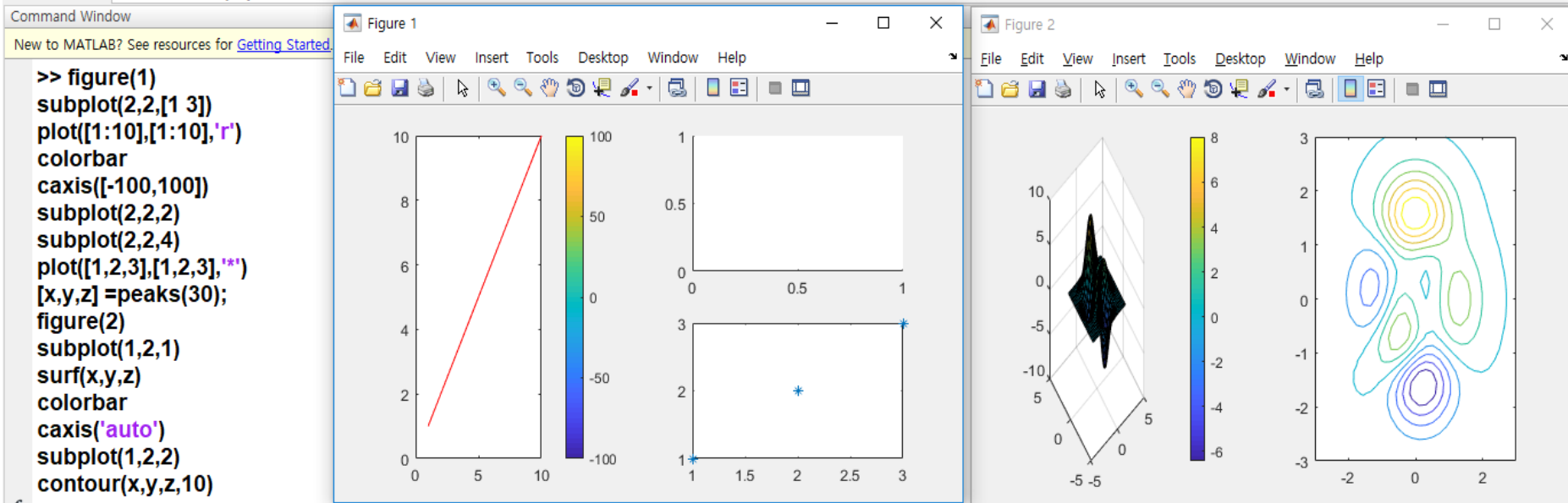


```
Command Window
New to MATLAB? See resources for Getting Started.

>> turns=40.*pi;
theta=linspace(0,turns,4000);
x=cos(theta).*(turns-theta)./turns;
y=sin(theta).*(turns-theta)./turns;
z=theta./turns;
plot3(x,y,z)
grid on
text(0.5,0.5,0.75,'Here is a piece of text!');
fx >>
```



# Additional Command for Plotting



- **figure():** creates a new figure window (recommended to use this before any plot)
- **subplot():** divides the current figure into rectangular panes that are numbered row-wise.
- **colorbar:** adds a color bar that defines the color used in the plot.
- **axis():** sets manual scaling of color bar

# Formatting Commands

- Formatting commands are entered after the **plot** command
- `>> title('string')`
  - Adds the string as a title at the top of the plot
- `>> xlabel('string') / ylabel('string')`
  - Adds the string as a label to the x-/y-axis
- `>> axis([xmin xmax ymin ymax])`
  - Sets the minimum and maximum limits of the x- and y-axes
- `>> legend('string1', 'string2', 'string3')`
  - Creates a legend using the string to label various curves.
- `>> text(x, y, 'string')`
  - Place the string(text) on the plot at coordinate x, y relative to the plot axes.
- `>> gtext('string')`
  - Place the string(text) on the plot by clicked with the mouse.

# Northwestern Engineering

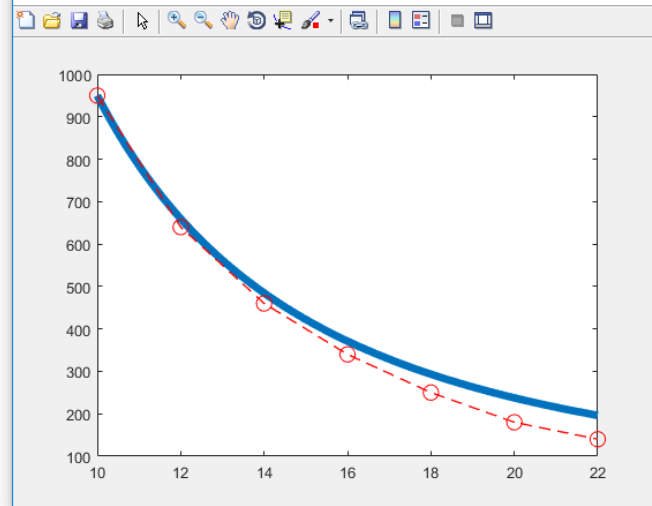
Command Window

New to MATLAB? See resources for [Getting Started](#).

```
>> x=[10:0.1:22];
y=95000./x.^2;
xd=[10:2:22];
yd=[950 640 460 340 250 180 140];
plot(x,y,'-', 'linewidth',5.0)
hold on
plot(xd,yd,'ro--', 'linewidth',1.0, 'markersize',10)
hold off
fx >>
```

Figure 1

File Edit View Insert Tools Desktop Window Help



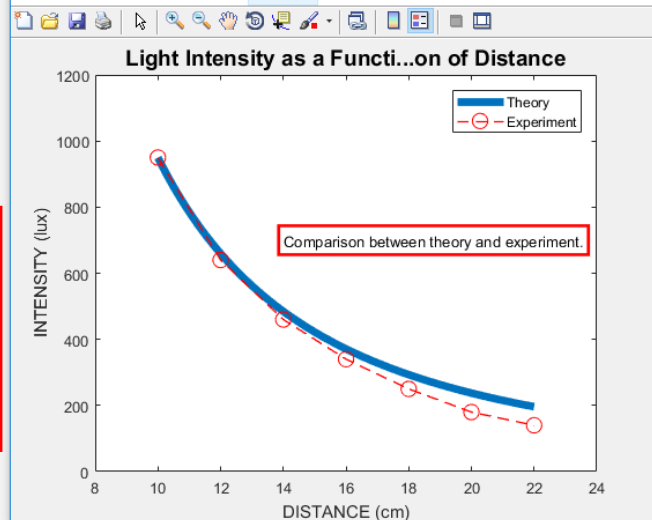
Command Window

New to MATLAB? See resources for [Getting Started](#).

```
>> x=[10:0.1:22];
y=95000./x.^2;
xd=[10:2:22];
yd=[950 640 460 340 250 180 140];
plot(x,y,'-', 'linewidth',5.0)
hold on
plot(xd,yd,'ro--', 'linewidth',1.0, 'markersize',10)
hold off
xlabel('DISTANCE (cm)')
ylabel('INTENSITY (lux)')
title('fontname(Arial)Light Intensity as a Functi...on of Distance',...
'FontSize',14)
axis([8 24 0 1200])
text(14,700,'Comparison between theory and experiment.',...
'EdgeColor','r','LineWidth',2)
legend('Theory','Experiment')
fx >>
```

Figure 1

File Edit View Insert Tools Desktop Window Help



## Additional Tutorial Materials for MATLAB

1. Introduction To Matlab For Engineering Students,  
by David Houcque at Northwestern University.
2. Experiments with MATLAB,  
by Cleve Moler, the inventor of MATLAB.  
(<https://www.mathworks.com/moler/exm.html>)

References

D. Houcque, Introduction to MATLAB for Engineering Students.

K. Tseng, Introduction to MATLAB.

R. Larsen and S. Hunt, Using MATLAB for Statics and Dynamics.

Prepared by

Department of Civil & Environmental Engineering, Northwestern University