

# Useful MATLAB Commands

## 1. Purpose

These brief notes are intended to provide you with some help on MATLAB in order to proceed quickly with the assigned projects.

## 2. Useful Functions

### 2.1 Sum and cumulative Sum

`sum(A)`: returns sums along different dimensions of an array.

`Cumsum(A)`: returns the cumulative sum along different dimensions of an array.

Example:

```
A=[1 2 3 4 5]; sum(A)
```

```
ans =
```

```
15
```

```
cumsum(A)
```

```
ans =
```

```
1    3    6   10   15
```

### 2.2 Basic Random Numbers

`rand(m,n)`: returns an m by n matrix each element of which is a pseudorandom, scalar value drawn from a uniform distribution on the unit interval.

`randn(m,n)`: returns an m by n matrix each element of which is a pseudorandom, scalar value drawn from a normal distribution with mean 0 and standard deviation 1.

The commands `rand` or `randn` used alone return a pseudorandom, scalar value drawn from a uniform or normal distribution, respectively.

#### Initialization of the state of random number generators:

The functions `rand` and `randn` use random number generator algorithms. The outcomes of these algorithms depend on the state of the generator. It is possible to initialize the state of the generator to the state `s` using

**`rand(method,s)`** or

**`randn(method,s)`**.

Depending on which random number generator algorithm is being used, `method` can be either 'state' or 'seed'. 'state' uses the Marsaglia's ziggurat algorithm (the default in MATLAB versions 5 and later) while 'seed' uses the polar algorithm (the default in MATLAB version 4). If `method` is set to 'state', then `s` must be a scalar integer value from 0 to  $2^{32}-1$ . If `method` is set to 'seed', then `s` must be a scalar integer value from 0 to  $2^{31}-2$ . To set the generator to its default initial state, set `s` equal to zero.

Note that the `rand` and `randn` generators each maintain their own internal state information. Initializing the state of one has no effect on the other.

## 3. Plotting

### 3.1 Plotting a single curve in a figure

**`plot(X,Y,LineSpec)`**: plots the Y vector versus the X vector. *LineSpec* is a line specification that determines line type, marker symbol, and color of the plotted lines.

**Line Style Specifiers:**

Specifier	Line Style	Specifier	Line Style
-	Solid Line (Default)	:	Dotted Line
--	Dashed Line	-.	Dash-dot Line

### Marker Specifiers:

Specifier	Marker Type	Specifier	Marker Type
+	Plus Sign	'diamond' or d	Diamond
O	Circle	^	Upward-pointing triangle
*	Asterisk	V	Downward-pointing triangle
.	Point	>	Right-pointing triangle
X	Cross	<	Left-pointing triangle
'square' or s	Square	'pentagram' or p	Pentagram

### Color Specifiers:

Specifier	Color	Specifier	Color
r	Red	m	Magenta
g	Green	y	Yellow
b	Blue	k	Black
c	Cyan	w	White

### Title and Labeling:

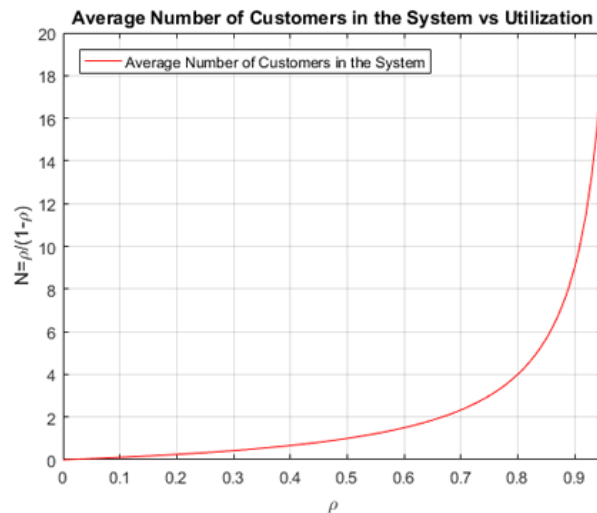
**title('string')**: Outputs the string at the top and in the center of the current axes.

**xlabel('string')**: Labels the x-axis of the current axes.

**ylabel('string')**: Labels the y-axis of the current axes.

*Example*: Plot the average number of the customers in the system ( $N = \frac{\rho}{1-\rho}$ ) versus the utilization factor  $\rho$ :

```
figure(1)
rho=0:.01:.95;
N=rho./(1-rho);
plot(rho,N,'-r')
axis([0,0.95,0,20]);
grid on;
xlabel('\rho')
ylabel('N=\rho/(1-\rho)')
legend('Average Number of Customers
in the
System','location','northwest')
title('Average Number of Customers
in the System vs Utilization')
```



### 3.2 Plotting multiple curves in a figure

**hold**: The hold function determines whether new graphics objects are added to the graph or replace objects in the graph.

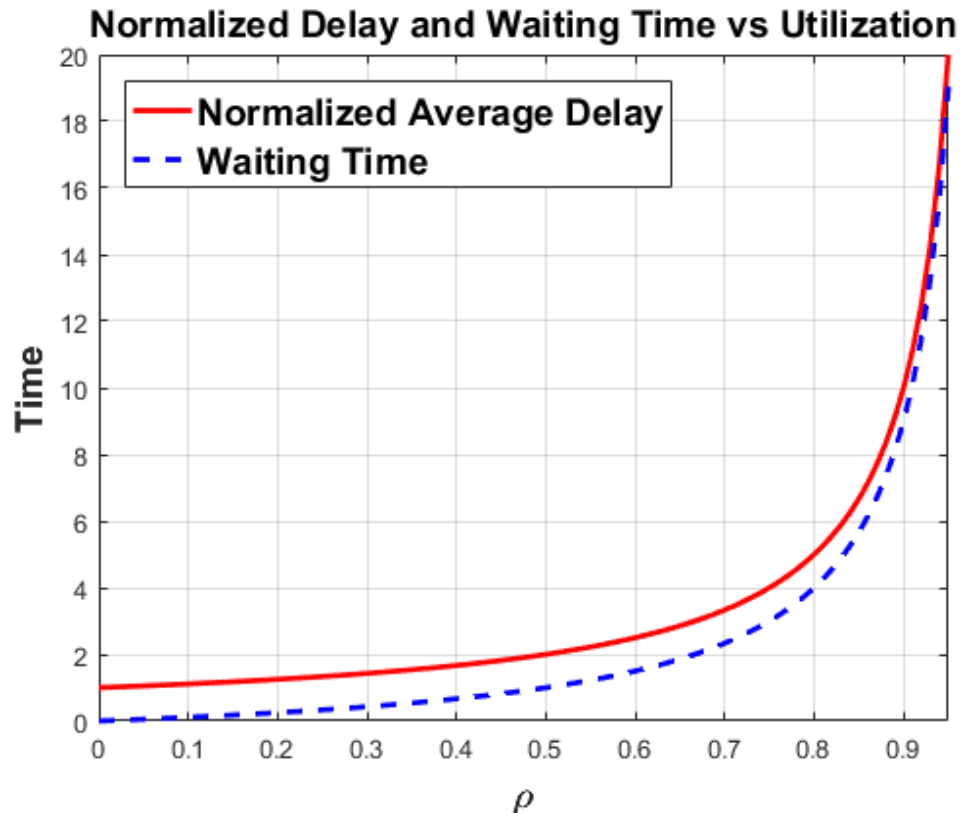
**hold on:** retains the current plot and certain axes properties so that subsequent graphing commands add to the existing graph.

**hold off:** resets axes properties to their defaults before drawing new plots. hold off is the default.

Legends:

**legend('string1','string2',...):** displays a legend in the current axes using the specified strings to label each set of data.

*Example:* Plot the normalized average delay versus the utilization factor  $\rho$  ( $T_n = \frac{1}{1-\rho}$ ) and the waiting time versus the utilization factor  $\rho$  ( $W_n = \frac{\rho}{1-\rho}$ ) in the same figure and use legends.

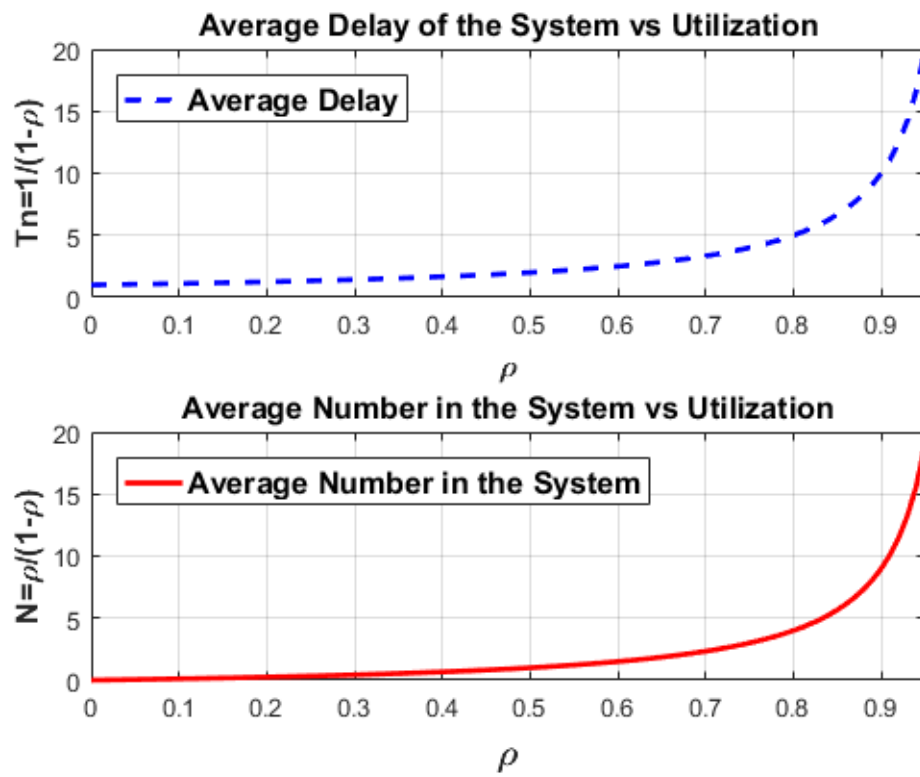


```
figure(2)
rho=0:.01:.99;
Tn=1./(1-rho);
Wn=rho./(1-rho);
plot(rho,Tn,'-r','linewidth',2,'MarkerSize',12)
hold on;
plot(rho,Wn,'--b','linewidth',2,'MarkerSize',12)
axis([0,0.95,0,20]);
grid on;
set(xlabel('\rho'),'fontSize',14,'fontWeight','bold')
set(ylabel('Time'),'fontSize',14,'fontWeight','bold')
set(legend('Normalized Average Delay','Normalized Average Waiting Time','location','northwest'),'fontSize',14,'fontWeight','bold');
title('Normalized Delay and Waiting Time vs Utilization','fontSize',14)
```

### 3.3 Plotting multiple graphs in a figure

subplot(m,n,p) or subplot(m,n,p): Breaks the figure window into an m-by-n matrix of small axes, selects the pth axes object for the current plot, and returns the axes handle. The axes are counted along the top row of the figure window, then the second row, etc.

*Example:* Plot the normalized average delay versus the utilization factor  $\rho$  ( $T_n = \frac{1}{1-\rho}$ ) and the average number of the customers in the system ( $N = \frac{\rho}{1-\rho}$ ) in the same figure and in two different graphs where the figure window is broken like a 2 by 1 matrix.



```

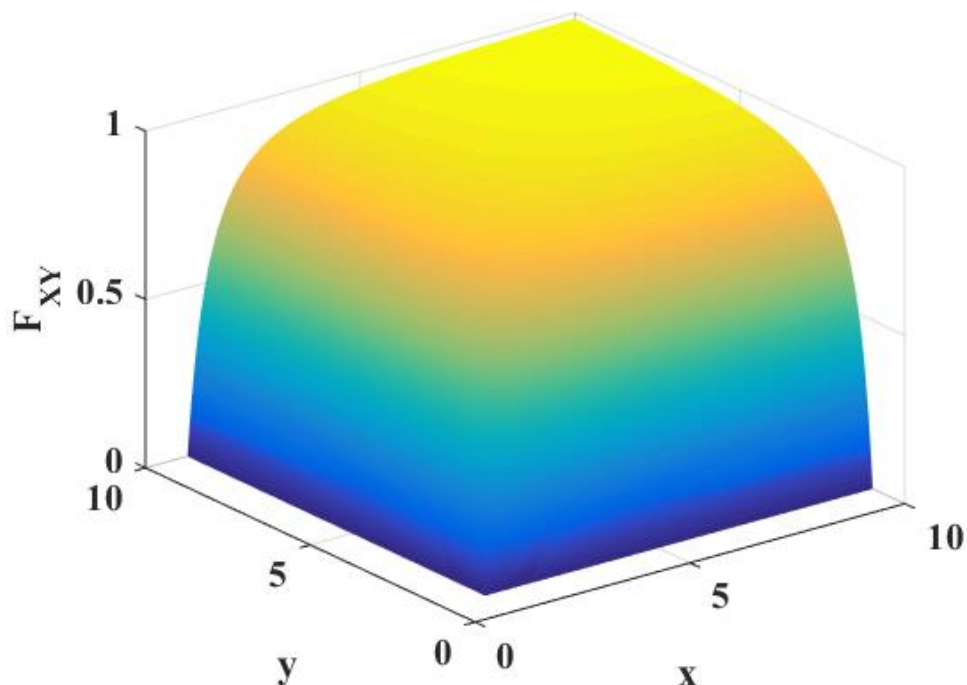
figure(3)
rho=0:.01:.95;
Tn=1./(1-rho);
N=rho./(1-rho);
subplot(2,1,1)
plot(rho,Tn,'--b','linewidth',2,'MarkerSize',12)
axis([0,0.95,0,20]);
grid on;
set(xlabel('\rho'),'fontSize',12,'fontWeight','bold')
set(ylabel('Tn=1/(1-\rho)'),'fontSize',12,'fontWeight','bold')
set(legend('Average
Delay','location','northwest'),'fontSize',12,'fontWeight','bold');
title('Average Delay of the System vs Utilization','fontSize',12)
subplot(2,1,2)
plot(rho,N,'-r','linewidth',2,'MarkerSize',12)
axis([0,0.95,0,20]);
grid on;
set(xlabel('\rho'),'fontSize',14,'fontWeight','bold')
set(ylabel('N=\rho/(1-\rho)'),'fontSize',12,'fontWeight','bold')
set(legend('Average Number in the
System','location','northwest'),'fontSize',12,'fontWeight','bold');
title('Average Number in the System vs Utilization','fontSize',12)

```

### 3.4 Plotting a figure of joint PDF

*Example:* Plot the joint PDF of  $F_{X,Y}$  below.

$$F_{X,Y}(x,y) = \begin{cases} (1-x^2)(1-y^2), & x > 1, y > 1, \\ 0, & \text{elsewhere.} \end{cases}$$



```
figure(4)
x1=[];y1=[];z=[];
x1=1:0.01:10
y1=1:0.01:10
[x,y]=meshgrid(x1,y1)
z=(1-x.^(-2)).*(1-y.^(-2))
mesh(x,y,z)
set(xlabel('x'),'fontsize',16,'fontweight','bold')
set(ylabel('y'),'fontsize',16,'fontweight','bold')
set(zlabel('F_{XY}'),'fontsize',16,'fontweight','bold')
set(gca,'fontsize',16,'fontweight','bold')
set(gca,'FontName','Times New Roman','FontSize',16)
```