An Introduction To MATLAB

Lecture 3

Basic MATLAB Commands

quit	quits MATLAB
exit	quits MATLAB
who	lists all of the variables in your MATLAB workspace
whos	list the variables and describe their matrix sizes

All variables are matrices in MATLAB

c=5.66 1x1 matrix (1 row, 1 column)

x=[3.4, 7, 2.2] 1x3 matrix (1 row, 3 columns)

y=[2; 5; 8; 11] 4x1 matrix (4 rows, 1 column)

A=[1, 3, 6; 2, 2, 2; 4, 4, 1] 3x3 matrix (3 rows, 3 columns)

$$A = \begin{matrix} 1 & 3 & 6 \\ 2 & 2 & 2 \\ 4 & 4 & 1 \end{matrix}$$

MATLAB is case sensitive

c=5.66 different from C=3.45

Keyboard Definition of Matrix

- x is a 1x3 row vector with elements x(1)=2, x(2)=4 and x(3)=-1
  - x=[2   4   -1]
  - x=[2 4 -1]
  - x=[2,4,-1]
- y is a 2x4 matrix with elements y(1,1)=0, y(1,2)=y(1,3)=2, y(1,4)=3, y(2,1)=5, y(2,2)=-3, y(2,3)=6 and y(2,4)=4
  - y=[0 2 2 3; 5 -3 6 4]
  - y=[0,2,2,3; 5,-3,6,4]
- Can use expressions for elements of matrix
  - a=[sin(pi/2), sqrt(2), 3+4, 6/3, exp(2)]
  - a=[1.0000, 1.4142, 7.0000, 2.0000, 7.3891]
- Can augment existing matrices to define new matrix
  - x1=[x 5 8]=[2, 4, -1, 5, 8]
  - x(5)=8 => x=[2, 4, -1, 0, 8] (Note value for x(4) which was not defined)
  - c=[4, 5, 6, 3]
  - z=[y; c]=[0, 2, 2, 3
5, -3, 6, 4
4, 5, 6, 3]

MATLAB Features

- MATLAB echoes ‘enter’ keystrokes at end of each line of code
- Utilize ; to cancel this echo
  - z=[y; c];
- Line continuation via … at end of line

```
4+5+3…
+7+2+9…
+5
```

MATLAB Practice Problems

- Determine the size and result for the following matrices:
  1. A=[1, 0, 0, 0, 0, 1]
  2. B=[2; 4; 6; 10]
  3. C=[5 3 5; 6 2 -3]
  4. D=[3 4
5 7
9 10]
  5. E=[3 5 10 0; 0 0 ...
0 3; 3 9 9 8]

A has 1 row, 6 columns

B has 4 rows, 1 column

C has 2 rows, 3 columns

D has 3 rows, 2 columns

E has 3 rows, 4 columns

MATLAB Practice Problems

- Determine the size and result for the following matrices:

6. T=[4 24 9] Q=[T 0 T] D=[3 4; 5 7; 9 10]
7. X=[3 6] Y=[D; X] Z=[X; D]
8. R=[C; X, 5] C=[5 3 5; 6 2 -3]
9. V=[C(2,1); B] B=[2; 4; 6; 10]
10. A=[1 0 0 0 0 1] A(2,1)=-3

T has 1 row, 3 columns; Q has 1 row, 7 columns, Q=[4 24 9 0 4 24 9], D has 3 rows, 2 columns
X has 1 row, 2 columns; Y has 4 rows, 2 columns; Z has 4 rows, 2 columns, Y=[3 4; 5 7; 9 10; 3 6], Z=[3 6; 3 4; 5 7; 9 10]
R has 3 rows, 3 columns, R=[5 3 5; 6 2 -3; 3 6 5]
V has 5 rows, 1 column, V=[6; 2; 4; 6; 10]
A has 2 rows, 6 columns, A=[1 0 0 0 0 1; -3 0 0 0 0 0]

File Commands

- **save** – saves all the matrices defined in current session into file matlab.mat, located in directory from which you executed MATLAB
- **load** – loads contents of matlab.mat into current workspace
- **save filename x y z** – saves the matrices x, y and z into the file titled filename.mat
- **load filename** – loads the contents of filename into current worksapce; the file can be a binary (.mat) file or an ascii file

File Commands

- Ascii files use editor to create text for entry or storage in MATLAB
  - using an ascii text editor we can create such files and save them as **filename.dat**
  - load files into MATLAB using command **load filename.dat**
  - ascii file **file1.dat** contains data 5CR 7CR 9; executing the command **load file1.dat** loads variable named file1 with contents [5; 7; 9] (1 row, 3 columns)
  - ascii file **file2.dat** contains data 5 7 9; executing the command **load file2.dat** loads variable named file2 with contents [5 7 9] (1 row, 3 columns)
  - ascii file **file3.dat** contains data 5 7 9; 6 8 10; 1 2 3; executing the command **load file3.dat** loads variable named file3 with contents [5 7 9;6 8 10; 1 2 3] (3 rows, 3 columns)
- .mat files – binary file format for load and store of MATLAB arrays
  - **save stuff x y z** stores matrices **x, y** and **z** in the file title **stuff.mat**
  - **load stuff** loads matrices **x, y** and **z** into the current directory
  - to save the matrices **a=[1 2 3;4 5 6]** and **b=[2 3;6 7;8 4;2 2]** we use the command **save mat1.mat a b**
  - to retrieve the matrices **a** and **b** we use the command **load mat1.mat** and that command extracts the matrices **a** and **b**

File Commands

- The colon operator
  - If two integers are separated by a colon, MATLAB will generate all integers between the two integers
  - a=1:8 gives a=[1 2 3 4 5 6 7 8]
  - If three integers are separated by two colons, the middle integer is a range and the first and third are limits
  - b=0.0:0.2:1.0 gives b=[0 0.2 0.4 0.6 0.8 1.0]
  - Colon operator can create a vector from a matrix
  - x=[2 6 8; 0 1 7; -2 5 6], y=x(:,1), y=[2; 0; -2] (first column)
  - yy=x(:,2), yy=[6; 1; 5] (second column)
  - z=x(1,:), z=[2 6 8] (first row)

File Commands

- Colon operator can extract sub-matrices
  - c=[-1 0 0; 1 1 0; 1 -1 0; 0 0 2]
  - d1=c(:,2:3), d1=[0 0; 1 0; -1 0; 0 2] (second and third columns)
  - d2=c(3:4,1:2), d2=[1 -1; 0 0] (third and fourth row, first and second column)
- **clear** command – erases all MATLAB matrices
- **clc** command – erases the screen only

Exercises

g=[0.6 1.5 2.3 -0.5
8.2 0.5 -0.1 -2.0
5.7 8.2 9.0 1.5
0.5 0.5 2.4 0.5
1.2 -2.3 -4.5 0.5]

Determine content and size of the following:

1. a=g(:,2)
2. b=g(4,:)
3. c=g[10:15]
4. d=[4:9;1:6]
5. e=[-5:5]
6. f=[1.0:-.2:0.0]
7. t1=g(4:5,1:3)
8. t2=g(1:2:5,:)

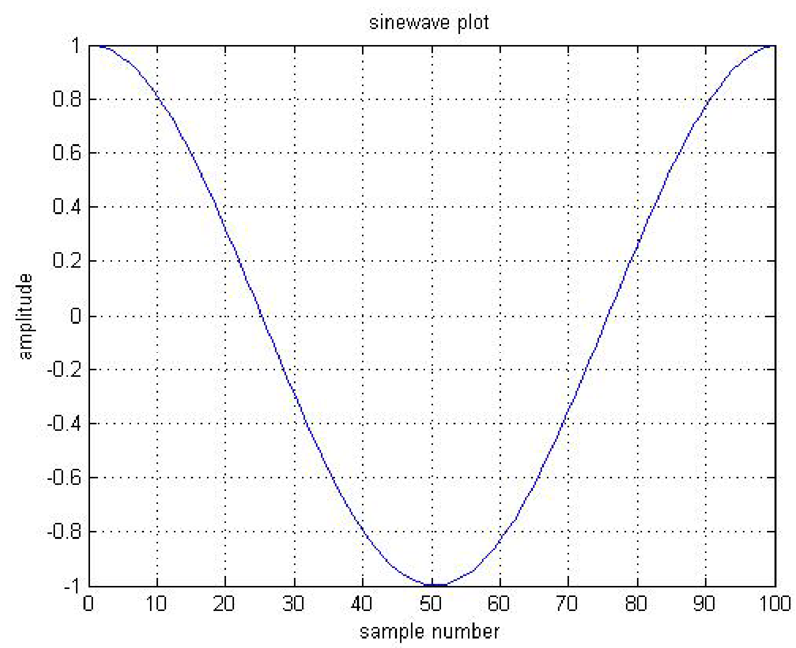
a=[1.5;0.5;8.2;0.5;-2.3] (second column)
b=[0.5 0.5 2.4 0.5] (fourth row)
c=[-2.3 2.3 -0.1 9.0 2.4 -4.5] (elements 10-15)
d=[4 5 6 7 8 9;1 2 3 4 5 6]
e=[-5 -4 -3 -2 -1 0 1 2 3 4 5]
f=[1 0.8 0.6 0.4 0.2 0]
t1=[0.5 0.5 2.4;1.2 -2.3 -4.5] (fourth and fifth row, first-to-third columns)
t2=[0.6 1.5 2.3 -0.5; 5.7 8.2 9.0 1.5; 1.2 -2.3 -4.5 0.5] (first, third and fifth rows, all columns)

Graphical Commands

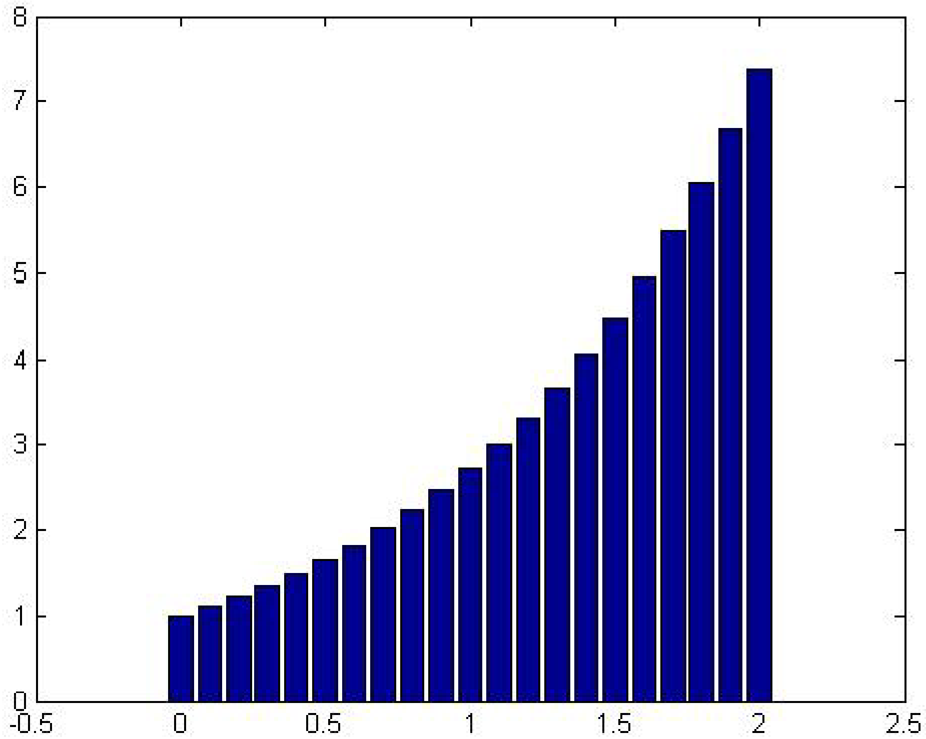
- plot(x,y) – creates a Cartesian plot of vectors x and y
- plot(y) – creates a plot of y vs numerical values of the elements of y
- semilogx(x,y) – plots log(x) vs y
- semilogy(x,y) – plots x vs log(y)
- loglog(x,y) – plots log(x) vs log(y)
- grid – creates a grid on plot
- title('text') – places a title at top of plot
- xlabel('text') – writes 'text' beneath the x-axis
- ylabel('text') – writes 'text' beside the y-axis
- text(x,y,'text') – writes 'text' at point x,y
- text(x,y,'text','sc') – writes 'text' at x,y in range (0-1,0-1) ('sc'=screen coordinates)
- bar(x,y) – creates a bar graph of vector x
- bar(x,y) – creates a bar graph of elements of y, locating the bars according to elements of x

Example of plot

- n=0:100;
- x=cos(2*pi*n/101);
- plot(n,x);
- grid;
- title(‘sinewave plot’);
- xlabel(‘sample number’);
- ylabel(‘amplitude’);

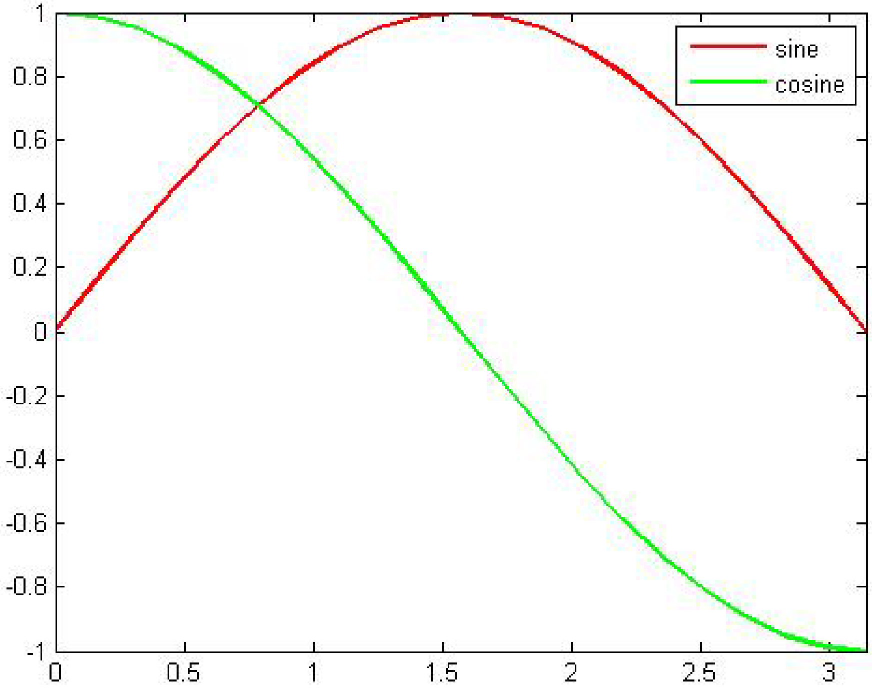


Example of bar plot




```
x=0:.1:2;
y=exp(x);
bar(x,y);
```

Multiple Plots

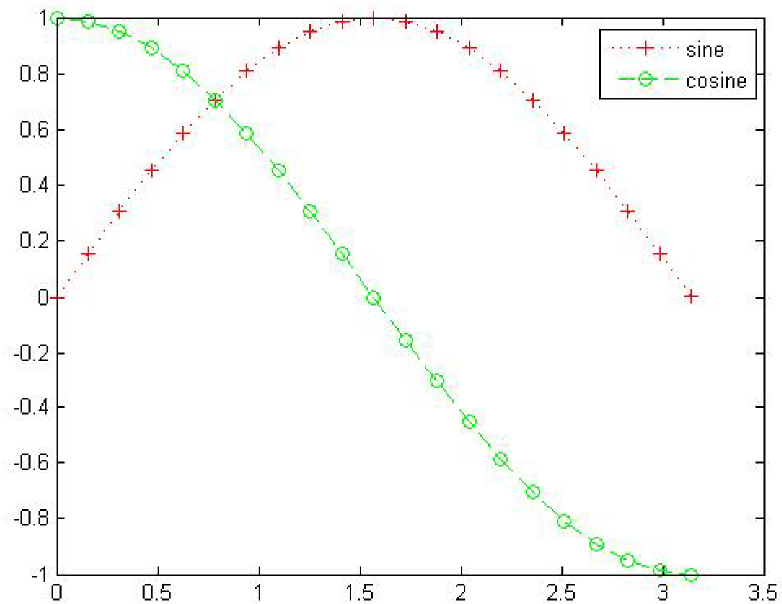



```
x1=0:0.05*pi:pi;
y1=sin(x1);
plot(x1,y1,'r','LineWidth',2);
hold on;
y2=cos(x1);
plot(x1,y2,'g','LineWidth',2);
legend('sine','cosine');
```

Fancy Plots

Various line types, plot symbols and colors may be obtained with PLOT(X,Y,S) where S is a character string made from one element from any or all the following 3 columns:

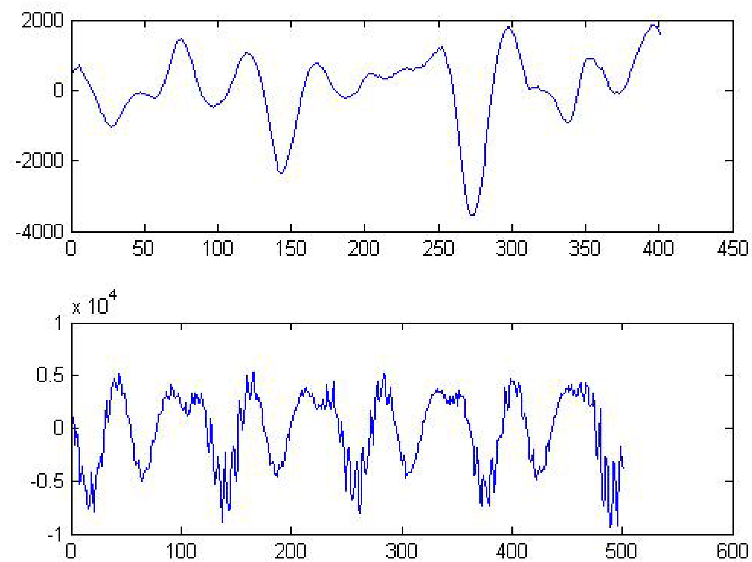
b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
w	white	v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		



For example, PLOT(X,Y,'c+:') plots a cyan dotted line with a plus at each data point; PLOT(X,Y,'bd') plots blue diamond at each data point but does not draw any line.

Multiple Plot Frames

```
subplot(nrows,ncols,current_plot)
    nrows=number of plot rows
    ncols=number of plot columns
    current_plot=current plot number
subplot(2,1,1),plot(x);
subplot(2,1,2),plot(y);
```



MATLAB Functions

max(x) – returns maximum value of elements in a vector, or if x is a matrix, returns a row vector whose elements are the maximum values from each column; max(max(f)) for maximum of two-dimensional array; max(f(:)) also works

min(x) – same as max but for minimums

mean(x) – same as max but for means

median(x) – same as max but for medians

sum(x) – same as max but for sums

prod(x) – same as max but for products

std(x) – same as max but for standard deviations

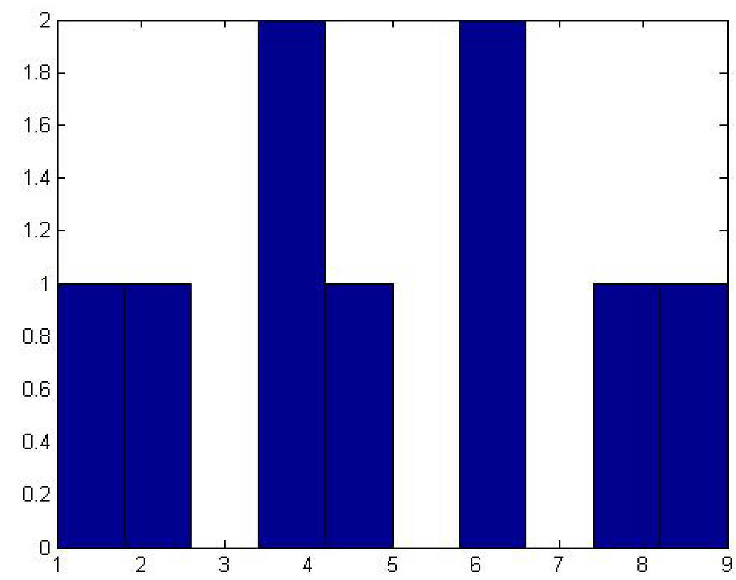
sort(x) – sorts the values in the vector x or the columns of a matrix and places them in ascending order

hist(x) – plots a histogram of the elements of vector, x. Ten bins are scaled based on the max and min values

hist(x,n) – plots a histogram with 'n' bins scales between the max and min values of the elements

MATLAB Functions

- f=[1 4 4;6 8 6;9 5 2];
- g=max(f)=[9 8 6]; % column maximums
- fmax=max(max(f))=max(f(:)) =9;
- fsort=sort(f)=[1 4 2;6 5 4;9 8 6]; % sort columns in increasing order
- hist(f(:)); % plot histogram of values of f



Examples

Time(sec)	Temp-T1(K)	Temp-T2(K)
0	306	125
1	305	121
2	312	123
3	309	122
4	308	124
5	299	129
6	311	122
7	303	122
8	306	123
9	303	127
10	306	124
11	304	123

timtemp(12,3)

Examples

M=max(timtemp)=[11 312 129] (maximum of each column)

M2=max(timtemp(:,2))=312 (maximum of second column)

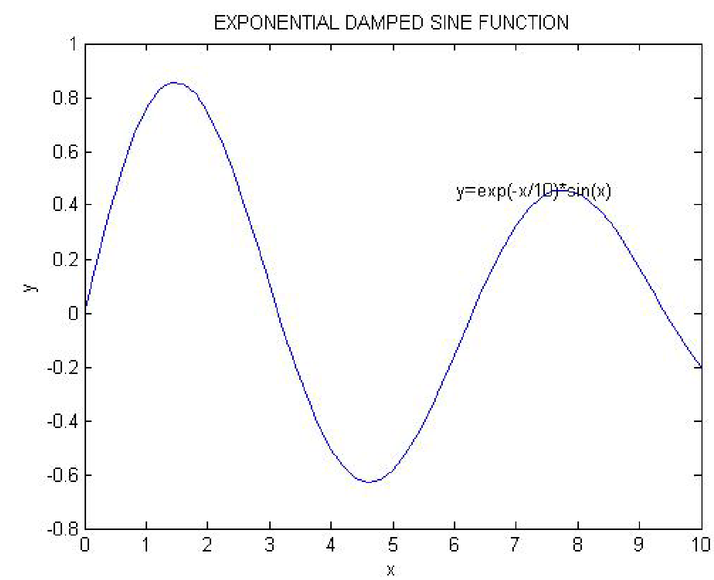
T1_var=(std(timtemp(:,2)))^2=13.2727 (variance is square of standard deviation)

STDDEV=std(timtemp)=[3.6056 3.6432 2.3012] (standard deviations of each column)

VAR=STDDEV.^2=[13.000 13.2727 5.2955] (note: have to use .^ notation)

Use of .m Files

```
% explot.m – m-file to plot exp(-x/10)sin(x)
x=[0:.2:10];
y=exp(-x/10) .* sin(x);
plot(x,y),…
title('EXPONENTIAL DAMPED SINE FUNCTION'),…
xlabel('x'),ylabel('y'),…
text(.6,.7,'y=exp(-x/10)*sin(x)','sc')
```



Function Files

```
function [y]=sind(x)
% This function calculates the sine when the
   argument is degrees
% Note that array multiplication and division
   allows this to operate on scalars, vectors and
   matrices.
y=sin(x .* pi ./ 180);
```

****Calling sequence: y=sind(x)

Algebraic Operations

+	addition
-	subtraction
*	multiplication
/	right division (a/b means a÷b)
\	left division (a\b means b÷a)
^	exponentiation

Algebraic Operations

- 3*4 = 12
- 4/5 = 0.8
- 4\5 = 1.25
- 3^2 = 9
- 3^2*2=18
- 3*2/5=1.2

Precedence

1 Parentheses
2 Exponentiation, left-to-right
3 Multiplication and division, left, right
4 Addition and subtraction, left, right

Matrix Operations

- A+B=B+A (A,B are matrices of the same order; i.e, the same number of rows and columns)
- A-B=-(B-A)
- x=[3 5 7], y=[4; -1; -3], x+y is undefined since x is a row vector (1x3 matrix) and y is a column vector (3x1 matrix)
- A*B is matrix multiplication => number of columns of first matrix (A) must be equal to the number of rows in the second matrix (B)

Matrix Operations

- A=[1 2 3]; B=[4; 5; 6];
- A*B=   32
- B*A=   [4 8 12; 5 10 15; 12 15 18]
- A.*B=   Undefined
- B.*A=   Undefined
- A.*B'= [4 10 18]
- B.*A'= [4; 10; 18]

Array Operations

- a .* b multiplies each element of a by the respective element of b
- a ./ b divides each element of a by the respective element of b
- a .\ b divides each element of b by the respective element of a
- a .^ b raises each element of a by the respective element of b

Array Operations

- ‘ Matrix transpose – interchange rows and columns

  G=[1 3 5; 2 4 6]; H=[-4 0 3; 1 9 8]

  G’=[1 2; 3 4; 5 6];

  G.*H=[-4 0 15; 2 36 48];

- Inner Product of two vectors

  G1=[1 3 5]; G2=[2 4 6]; G1*G2’=44

- Outer Product of two vectors

  G1’*G2=[2 4 6; 6 12 18; 10 20 30];

Special Matrices

- zeros(n,m) – matrix of zeros with n rows and m columns
- zeros(3,2) = [0 0; 0 0; 0 0];
- zeros(2,3) = [0 0 0; 0 0 0];
- ones(n,m) – matrix of ones with n rows and m columns
- ones(2,4)=[1 1 1 1; 1 1 1 1];
- eye(n) – identity matrix with 1’s on diagonals and 0’s off diagonal
- eye(4)=[1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];

Exercise

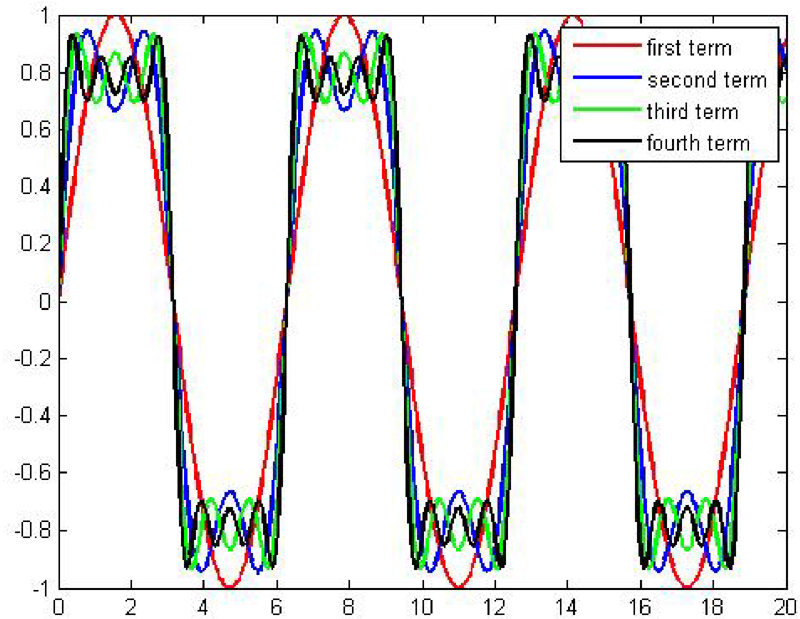
- The first four terms of the Fourier series for the square wave whose amplitude is 1 and whose duration is 2f are:

$$y = (10/f)[\sin(x) + (1/3)\sin(3x) + (1/5)\sin(5x) + (1/7)\sin(7x)]$$

- Calculate this series, term-by-term and plot the results for each partial sum.

Exercise Solution

```
% squarewave_series.m
% calculate first 4 terms of Fourier series of a square wave and plot them
%
   f=10;
   x=0:2*f/500:2*f;
   y=zeros(1,length(x));
   colors=['r';'b';'g';'k'];
   for i=1:4
      yp=sin((2*i-1)*x)/(2*i-1);
      y=y+yp;
      plot(x,y,colors(i),'LineWidth',2);
      hold on;
   end
   legend('first term','second term','third term','fourth term');
```



Relational Operators

- A==B 1 when $A(i,j)=B(i,j)$; 0 otherwise
- A~=B 1 when $A(i,j)\neq B(i,j)$; 0 otherwise
- A>=B 1 when $A(i,j)\geq B(i,j)$; 0 otherwise
- A>B 1 when $A(i,j)>B(i,j)$; 0 otherwise
- A<=B 1 when $A(i,j)\leq B(i,j)$; 0 otherwise
- A<B 1 when $A(i,j)<B(i,j)$; 0 otherwise

Logical Operators

- Logical 1 or non-zero term => logical true
- Logical 0 or numeric 0 => logical false

A=[1 2 0;0 4 5]; B=[1 -2 3;0 1 1]

- A&B both A and B true
- A|B either A or B true
- ~A not A true

A&B=[1 1 0; 0 1 1]; A|B=[1 1 1;0 1 1];

~A=[0 0 1;1 0 0]; ~B=[0 0 0;1 0 0]

Flow Control

```
for expression
    statements
end

while expression
    statements
end
```

```
if expression
        statements
elseif expression
        statements
else
        statements
end
```

Useful MATLAB Functions

- Read in filename

  filename=input('enter filename:','s');

- Read in speech file in .wav format

  [f,fs,nbits]=wavread(filename);

  f=speech wav file; fs=sampling rate

  [f,fs]=loadwav(filename); (course website)

- Save processed speech file

  fname=strcat(filename,'_processed.wav');

  savewav(speech,fname,fs);

- Read in parameters

  parameter=input('parameter value:');

Other MATLAB Functions

- path(path,'new directory') – lets you access files on other than the current MATLAB directory
- tic – start a clock ticking to time code
- time_spent=toc – measure time to execute region of code
- rand(m,n) – generate uniformly distributed random numbers over range (0,1); m rows, n columns
- randn(m,n) – generate Gaussian distributed random numbers with zero mean and variance 1

MATLAB General Functions

- abs(x) – |x|
- sqrt(x) – square root(x)
- round(x) – nearest integer=I[x+0.5]
- fix(x) – nearest lower integer → zero
- floor(x) – nearest lower integer
- ceil(x) – nearest higher integer
- sign(x) – 1 if x > 0, 0 if x-0, -1 if x < 0

MATLAB Trig Functions

- exp(x) – exponential (x)
- log(x) – logarithm (base e) (x)
- log10(x) – logarithm (base 10) (x)
- sin(x) – sine in radians
- cos(x) – cosine in radians
- tan(x) – tangent in radians
- asin(x) – arc sin(x)
- acos(x) – arc cosine(x)
- atan(x) – arc tangent(x)
- atan2(y,x) – arc tangent(x/y)

Signal Processing Toolbox

- conv(h,x) – convolve impulse response (h) with input signal (x)
- y=filter(b,a,x) – filter input signal (x) with digital system with numerator polynomial b and denominator polynomial a
  - b=[b0, b1, b2,…,bM]
  - a=[a0,a1,a2,…,aN]
- [h,w]=freqz(b,a,p,fs) – calculate frequency response from system function, at p frequencies at sampling rate fs; h is the resulting complex frequency response; w is the set of frequencies at which h is calculated

Play Speech/Audio Files

- sound(xin,fs) – xin must be normalized to range (-1,1)
- soundsc(xin,fs) – xin can be any range

MATLAB (cont.)

- Tips for Matlab programming of DSP
  - http://www.eedsp.gatech.edu/Information/MATLAB_User_Guide/index.shtml
  - http://www.eng.auburn.edu/~sjreeves/Classes/DSP/DSP.html