THE UNIVERSITY OF RHODE ISLAND

EGR 106
Foundations of Engineering II

Lecture 3 – Part B
Arrays and Array Mathematics

THINK BIG WE DO℠

This Week's Topics

Review of last week's topics
Today's topics:
- Array addressing
- Special arrays
- Some array operators
- Character arrays
- Array mathematics
  - Vector operations
  - Element by element operations
    - addition, subtraction, multiplication, division
  - Matrix multiplication
  - Solving systems of equations

Review of Last Week

Arrays are the fundamental data units in MATLAB

- Rectangular collection of data
- All variables are considered to be arrays

$$\text{yield} = \begin{bmatrix} 4 & 5 & 3 & 9 \\ 10 & 4 & 66 & 20 \\ 18 & -3 & 2 & 0 \end{bmatrix}$$

Data values organized into rows and columns

Review of Last Week (cont.)

- Size of an array (R x C)
- Array Construction:
  - Brute force using brackets
  - Concatenation of other arrays – side-by-side and top-to-bottom
  - The colon operator
  - The linspace command
- MATLAB scripts
  - File editor
  - Useful commands: clc, clear, close, %, pause, disp

Array Addressing

We indicate a particular element within an array by it’s row/column position:

use parentheses after the array name

e.g.

yield(2,4)

$$\text{yield} = \begin{bmatrix} 4 & 5 & 3 & 9 \\ 10 & 4 & 66 & 20 \\ 18 & -3 & 2 & 0 \end{bmatrix}$$

Array Addressing (cont.)

Used to read a value from an array

```
test =

    0.4565    0.8214    0.6154
    0.0185    0.4447    0.7919

>> x = test(1,3)

x =

    0.6154
```

Array Addressing (cont.)

How about more than one entry?

Can specify a rectangular sub-array

again, use parenthesis after the array name

list desired rows, comma, desired columns

as separate vectors, typically in brackets

e.g.

$$\text{yield} = \begin{bmatrix} 4 & 5 & 3 & 9 \\ 10 & 4 & 66 & 20 \\ 18 & -3 & 2 & 0 \end{bmatrix}$$

yield([1 2],[3 4])

Special Arrays

Square versions

Special predefined arrays:

all zeros	zeros(R,C)	zeros(N)
all ones	ones(R,C)	ones(N)
zeros with ones on the diagonal		
	eye(R,C)	eye(N)
random numbers (within [0 1])		
	rand(R,C)	rand(N)

Examples

Command Window

```
>> eye(3,4)

ans =

     1     0     0     0
     0     1     0     0
     0     0     1     0

>> rand(2,5)

ans =

    0.9501    0.6068    0.8913    0.4565    0.8214
    0.2311    0.4860    0.7621    0.0185    0.4447
```

random on [0, 1]

Transpose Operation

Transpose (single quote symbol ')

switches rows and columns

```
test =

     1     2     3     4
     5     6     7     8
     9    10    11    12
```

```
>> test'

ans =

     1     5     9
     2     6    10
     3     7    11
     4     8    12
```

Array Size and Length

Size – the number of rows and columns

Length – the larger of these two

```
>> test=[4 5 3;10 4 66]
test =
     4     5     3
    10     4    66
>> size(test)
ans =
     2     3
>> length(test)
ans =
     3
```

```
>> bob=[5; 7; 3; 6]
bob =
     5
     7
     3
     6
>> size(bob)
ans =
     4     1
>> length(bob)
ans =
     4
```

Character Arrays

Rows of the array are strings of alphanumeric characters, one array entry per character

Enter using a single quotation mark (') at each end of the string

```
>> test = 'John'

test =

John

>> size(test)

ans =

     1     4
```

Character Arrays (cont.)

For multi-row alphanumeric arrays, each row must have the same number of characters

name = ['Marty' ; 'James' ; 'Bob ']

Note – need 2 spaces

Note that we have already used character arrays in plotting functions – recall Week 1 homework:

```
v=100; A=35*pi/180;
t=0:0.01:14;
x=v*cos(A)*t;
y=v*sin(A)*t-0.5*9.81*t.^2;
plot(x,y)
xlabel('x')
ylabel('y')
title('Trajectory Plot')
text(750,150,'Bill Smith, 1/29/2013')
```

"num2str" command

The built-in function, num2str(N), takes a number, N, and converts it to a character string.

Useful in displaying results

Ex.

```
%
N=254;
%
T=['The number is ' num2str(N)];
%
size(T)
%
disp(T)
```

Concatenates the strings 'The number is ' and '254'

```
ans =
     1    17
The number is 254
```

THE UNIVERSITY OF RHODE ISLAND

Demonstration Problem

Create a script which:

1. Creates the following array of characters:

```
This is a Matlab demo of
a character string array
```

2. Determines and displays the size of the array

3. Adds a row with the characters:

```
which has 24 columns
```

4. Determines and displays the new size of the array

Demonstration Problem (cont.)

Script

```
% Demo Problem
clear; clc
disp ('Demo Problem')
B=['This is a Matlab demo of'
   'a character string array']
pause
s=size(B);
disp(['The size of array B is ' num2str(s)])
pause
B(3,1:20)='which has 24 columns'
pause
s=size(B);
disp(['The size of array B is now ' num2str(s)])
```

Command Window Output

```
Demo Problem
B =
This is a Matlab demo of
a character string array
The size of array B is 2   24
B =
This is a Matlab demo of
a character string array
which has 24 columns
The size of array B is now 3   24
```

Vector Based Operations

Some operations analyze a vector to yield a single value.

For example:

```
A =

     6     3     5     1

>> sum(A)

ans =

    15
```

sums the elements

Vector Operations (cont.)

Some operators yield vector results

- size(A) we’ve already seen – gives rows and columns
- sort (A)

```
A =

     6     3     5     1

>> sort(A)

ans =

     1     3     5     6
```

Vector Operations (cont.)

Some operators give multiple vectors

```
A =

     6     3     5     1

>> [vals,locs] = sort(A)

vals =

     1     3     5     6


locs =

     4     2     3     1
```

Other Vector Operations

Minimum: min(A)

Maximum: max(A)

Median: median(A)

Mean or average: mean(A)

Standard deviation: std(A)

Product of the elements: prod(A)

Vector Operations (cont.)

Use help to discover how to use these work

```
>> help sum
 SUM Sum of elements.
    S = SUM(X) is the sum of the elements of the vector X. If
    X is a matrix, S is a row vector with the sum over each
    column. For N-D arrays, SUM(X) operates along the first
    non-singleton dimension.
    If X is floating point, that is double or single, S is
    accumulated natively, that is in the same class as X,
    and S has the same class as X. If X is not floating point,
    S is accumulated in double and S has class double.

    S = SUM(X,DIM) sums along the dimension DIM.
```

Element by Element Math Operations

For arrays of identical sizes, addition and subtraction is defined term by term:

the command F = A + B means

$$F(r,c) = A(r,c) + B(r,c)$$

for all row and column pairs r,c

"element-by-element" addition

Array Addition and Subtraction

Arrays must be of identical size

Term by term (or element by element) operation:

```
>> A=[1 2;3 4;5 6]
A =
     1     2
     3     4
     5     6
>> B=[7 8;9 10; 11 12]
B =
     7     8
     9    10
    11    12
>> C=A+B
C =
     8    10
    12    14
    16    18
```

Addition and Subtraction of Arrays (cont.)

Sizes must match:

```
>> A=[1 2;3 4;5 6]
A =
     1     2
     3     4
     5     6
>> B=[7 8;9 10]
B =
     7     8
     9    10
>> C=A+B
??? Error using ==> plus
Matrix dimensions must agree.
```

Array subtraction is identical:

```
>> A=[1 2;3 4;5 6]
A =
     1     2
     3     4
     5     6
>> B=[7 8;9 10; 11 12]
B =
     7     8
     9    10
    11    12
>> C=B-A
C =
     6     6
     6     6
     6     6
```

Built in Functions

Built-in functions also work element-by-element:

(sqrt, log, exp, sin, cos, etc.)

Example:

```
>> b = [ 4 9 25; 1 2 10 ]

b =

     4     9    25
     1     2    10

>> sqrt(b)

ans =

    2.0000    3.0000    5.0000
    1.0000    1.4142    3.1623
```

Matrix Multiplication

In linear algebra:

- Matrix multiplication of an M x N matrix times an N X P matrix yields an M X P matrix
- Each term is found by taking the dot product of rows of the first matrix with columns of the second

"Dot Product"

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & \\ & \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & 64 \\ & \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & 64 \\ 139 & 154 \end{bmatrix}$$

2 x 3   3 x 2   2 x 2

Matrix Multiplication (cont.)

Example from high school algebra text

EXAMPLE 2 ***Finding the Product of Two Matrices***

Find AB if $A = \begin{bmatrix} -2 & 3 \\ 1 & -4 \\ 6 & 0 \end{bmatrix}$ and $B = \begin{bmatrix} -1 & 3 \\ -2 & 4 \end{bmatrix}$.

SOLUTION

Because A is a 3×2 matrix and B is a 2×2 matrix, the product AB is defined and is a 3×2 matrix. To write the entry in the first row and first column of AB, multiply corresponding entries in the first row of A and the first column of B. Then add. Use a similar procedure to write the other entries of the product.

$$AB = \begin{bmatrix} -2 & 3 \\ 1 & -4 \\ 6 & 0 \end{bmatrix} \begin{bmatrix} -1 & 3 \\ -2 & 4 \end{bmatrix}$$

$$= \begin{bmatrix} (-2)(-1) + (3)(-2) & (-2)(3) + (3)(4) \\ (1)(-1) + (-4)(-2) & (1)(3) + (-4)(4) \\ (6)(-1) + (0)(-2) & (6)(3) + (0)(4) \end{bmatrix}$$

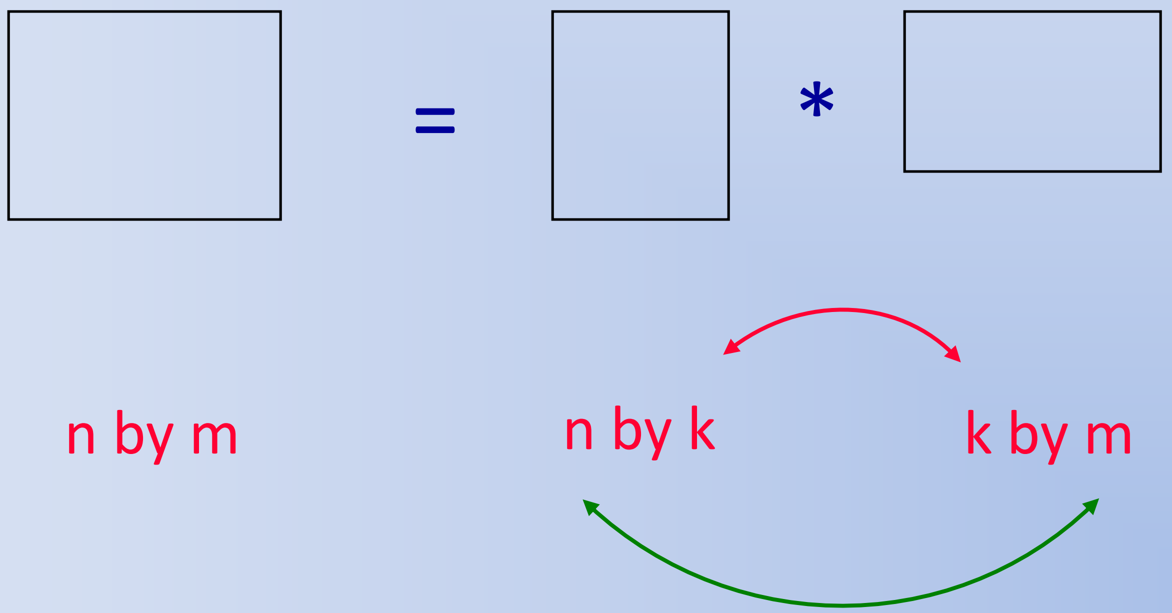
$$= \begin{bmatrix} -4 & 6 \\ 7 & -13 \\ -6 & 18 \end{bmatrix}$$

MATLAB

```
>> A=[-2 3; 1 -4; 6 0]
A =
    -2     3
     1    -4
     6     0
>> B=[-1 3;-2 4]
B =
    -1     3
    -2     4
>> A*B
ans =
    -4     6
     7   -13
    -6    18
```

Array Multiplication (cont.)

The number of columns of the 1st **must** match the number of rows of the 2nd



Array Multiplication vs. Element by Element Multiplication

Array Multiplication

```
>> A=[2 3;1 5]
A =
      2      3
      1      5
>> B=[1 4;2 6]
B =
      1      4
      2      6
>> A*B
ans =
      8     26
     11     34
```

A*B

Element by Element Multiplication

```
>> A=[2 3;1 5]
A =
      2      3
      1      5
>> B=[1 4;2 6]
B =
      1      4
      2      6
>> A.*B
ans =
      2     12
      2     30
```

A.*B

Other Element by Element Operations

The other basic math operations work element by element using the dot notation (with A,B the same sizes):

multiplication

F = A .* B → F(r,c) = A(r,c) * B(r,c)

division

F = A ./ B → F(r,c) = A(r,c) / B(r,c)

exponentiation:

F = A .^ B → F(r,c) = A(r,c) ^ B(r,c)

note periods!

Element by Element Operations - Examples

```
a =

     1     2     3
```

```
b =

     4     5     6
```

```
>> a.*b

ans =

     4    10    18
```

```
>> a./b

ans =

    0.2500    0.4000    0.5000
```

```
>> a.^b

ans =

     1    32   729
```

Solving System of Equations - Example

$$\begin{aligned} 2x_1 &+ 3x_2 &+ 3x_3 &= 7 \\ 4x_1 &+ 2x_2 &+ 9x_3 &= 5 \\ 6x_1 &- 7x_2 &+ 2x_3 &= 1 \end{aligned}$$

In matrix form this is

$$\mathbf{A * x = b}$$

where

$$\mathrm{A} = \begin{bmatrix} 2 & 3 & 3 \\ 4 & 2 & 9 \\ 6 & -7 & 2 \end{bmatrix} \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad b = \begin{bmatrix} 7 \\ 5 \\ 1 \end{bmatrix}$$

Solving Systems of Equations – Array Division

Recall the command eye(n)

This result is the array multiplication identity matrix **I**

For any array **A**

$$\mathbf{A} * \mathbf{I} = \mathbf{I} * \mathbf{A} = \mathbf{A}$$

```
>> eye(3)

ans =

     1     0     0
     0     1     0
     0     0     1
```

must be properly sized!

Solving Systems of Equations – Array Division (cont.)

Imagine that for square arrays **A** and **B** we have

$$\mathbf{A} * \mathbf{B} = \mathbf{B} * \mathbf{A} = \mathbf{I}$$

then we call them inverses

$$\mathbf{A} = \mathbf{B}^{-1} \qquad \mathbf{B} = \mathbf{A}^{-1}$$

In MATLAB: A ^ -1   or   inv(A)

When does $\mathbf{A}^{-1}$ exist?

A is square

A has a non-zero determinant (det(A) $\neq 0$)

Solving Systems of Equations – Array Division (cont.)

Example:

```
A =

     0     9     8
     7     4     5
     4     4     2

>> det(A)

ans =

   150
```

```
>> B = A^-1

B =

   -0.0800    0.0933    0.0867
    0.0400   -0.2133    0.3733
    0.0800    0.2400   -0.4200
```

```
>> A*B

ans =

    1.0000         0         0
    0.0000    1.0000   -0.0000
    0.0000   -0.0000    1.0000
```

```
>> B*A

ans =

    1.0000    0.0000    0.0000
         0    1.0000   -0.0000
         0    0.0000    1.0000
```

Solving Systems of Equations – Array Division (cont.)

Solving $\mathbf{A} * \mathbf{x} = \mathbf{b}$

Assume that **A** is square and det(**A**) ≠ 0

Multiply both sides by $\mathbf{A}^{-1}$ on the left

$$\underbrace{\underbrace{\mathbf{A}^{-1} * \mathbf{A}}_{=\mathbf{I}} * \mathbf{x}}_{=\mathbf{x}} = \mathbf{A}^{-1} * \mathbf{b}$$

so $\mathbf{x} = \mathbf{A}^{-1} * \mathbf{b}$

backwards slash

In MATLAB, x = A^-1*b, x = A \ b or x = inv(A)*b

Solving Systems of Equations – Array Division (cont.)

Example:

```
A =

     0     9     8
     7     4     5
     4     4     2
```

```
b =

     6
     8
     0
```

```
>> x = A\b

x =

    0.2667
   -1.4667
    2.4000
```

```
>> A*x

ans =

    6.0000
    8.0000
   -0.0000
```