Bil 108

Introduction to the Scientific and Engineering Computing with MATLAB

Lecture 5

F. Aylin Konuklar   Lale T. Ergene

Vector DOT Product

- In physics, analytical geometry, and engineering, the dot product has a geometric interpretation

$$\sigma = x \cdot y \quad \Longleftrightarrow \quad \sigma = \sum_{i=1}^{n} x_i y_i$$

$$x \cdot y = \|x\|_2 \, \|y\|_2 \cos\theta$$

DOT Vector dot product.
C = DOT(A,B) returns the scalar product of the vectors A and B.
A and B must be vectors of the same length.
When A and B are both column vectors, DOT(A,B) is the same as A'*B.

Example:

```
a = [1 2 3];  b = [4 5 6];
c = dot(a,b)
c = 32
```

Vector Inner Product

- The rules of linear algebra impose compatibility requirements on the inner product.
- The inner product of *x* and *y* *requires* that *x* be a row vector *y* be a column vector

$$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = x_1y_1 + x_2y_2 + x_3y_3 + x_4y_4$$

Computing the Inner Product in Matlab

- The * operator performs the inner product if two vectors are compatible.

```
>> u = (0:3)';        % u and v are
>> v = (3:-1:0)';     % column vectors
>> s = u'*v
s =
4
>> t = v'*u
t =
4
```

Vector Outer Product

- The inner product results in a scalar.
- The *outer product* creates a rank-one matrix:
- $A = uv^T \Longleftrightarrow a(i,j) = u(i)v(j)$

Outer product of two 4-element column vectors

$$uv^T = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \begin{bmatrix} v_1 & v_2 & v_3 & v_4 \end{bmatrix}$$

$$= \begin{bmatrix} u_1v_1 & u_1v_2 & u_1v_3 & u_1v_4 \\ u_2v_1 & u_2v_2 & u_2v_3 & u_2v_4 \\ u_3v_1 & u_3v_2 & u_3v_3 & u_3v_4 \\ u_4v_1 & u_4v_2 & u_4v_3 & u_4v_4 \end{bmatrix}$$

Computing the Outer Product in Matlab

- The * operator performs the outer product if two vectors are compatible.

```
u = (0:4)';
v = (4:-1:0)';
A = u*v'
A =
     0     0     0     0     0
     4     3     2     1     0
     8     6     4     2     0
    12     9     6     3     0
    16    12     8     4     0
```

Matrices

- Columns and Rows of a Matrix are Vectors
- Addition and Subtraction
- Multiplication by a scalar
- Transpose
- Linear Combinations of Vectors
- Matrix–Vector Product
- Matrix–Matrix Product

Matrix Operations

Addition and subtraction

$C = A + B$

or

$c(i,j) = a(i,j\ ) + b(i,j) \qquad i = 1, \ldots, m;\ j = 1, \ldots, n$

Multiplication by a Scalar

$B = \sigma A$

or

$b(i,j\ ) = \sigma a(i,j) \qquad i = 1, \ldots, m;\ j = 1, \ldots, n$

Note: Commas in subscripts are necessary when the subscripts are assigned numerical values. For example, $a(2,3)$ is the row 2, column 3 element of matrix A,whereas $a(23)$ is the 23rd element of vector a. When variables appear in indices, such as $a(ij)$ or $a(i,j,)$ the comma is optional

Examples

```
>>A=[1 4 2;7 7 3;9 1 6 ;4 2 8]
>>B=[6 1;2 5;7 3]
>>C=A*B
>>D=B*A
>>F=[13;5 7]
>>G=[4 2;1 6]
>>F*G
>>G*F
```

Matrix Transpose

- $B = A^T$ or $b(i,j) = a(j,i) \quad i = 1, . . . , m; \quad j = 1, . . . , n$

In MATLAB

```
>> A = [0 0 0; 0 0 0; 1 2 3; 0 0 0]
A =
        0       0       0
        0       0       0
        1       2       3
        0       0       0

>> B = A'
B =
        0       0       1       0
        0       0       2       0
        0       0       3       0
```

Matrix–Matrix Product Summary

- The **matrix–vector product** looks like:

$$\begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix} \begin{bmatrix} \bullet \\ \bullet \\ \bullet \end{bmatrix} = \begin{bmatrix} \bullet \\ \bullet \\ \bullet \\ \bullet \end{bmatrix}$$

- The **vector–matrix product** looks like:

$$\begin{bmatrix} \bullet & \bullet & \bullet & \bullet \end{bmatrix} \begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix} = \begin{bmatrix} \bullet & \bullet & \bullet \end{bmatrix}$$

- The **matrix–matrix product** looks like:

$$\begin{bmatrix} \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \end{bmatrix} \begin{bmatrix} \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{bmatrix} = \begin{bmatrix} \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \end{bmatrix}$$

Diagonal Matrices

- Diagonal matrices have non-zero elements only on the main diagonal.

$$C = \operatorname{diag}(c_1, c_2, \ldots, c_n) = \begin{bmatrix} c_1 & 0 & \cdots & 0 \\ 0 & c_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & c_n \end{bmatrix}$$

Example:

```
>> x = [1 -5  2  6];
>> A = diag(x)
A =
      1     0     0     0
      0    -5     0     0
      0     0     2     0
      0     0     0     6
```

Identity Matrices

An identity matrix is a square matrix with ones on the main diagonal.

Example: *The 3×3 identity matrix*

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

An identity matrix is special because

$AI = A$ and $IA = A$

for *any* compatible matrix A. This is like multiplying by one in scalar arithmetic.

Functions to Create Special Matrices

Matrix		Matlab function
Diagonal	→	diag
Identity	→	eye
Inverse	→	inv
Determinants	→	det
Left Division	→	\
Right Division	→	/

Left Division

- Used to solve the matrix equation AX=B

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

- So, the solution of AX=B is

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

- In Matlab **X=A\B**

Right Division

- The right divison is used to solve the matrix equation XC=D

$$\mathbf{XCC^{-1}=DC^{-1}}$$

$$\mathbf{X=DC^{-1}}$$

- In Matlab,

$$\mathbf{X=D/C}$$

Solving three linear equations

- Use matrix operations to solve the following systems of linear equations

$$4x-2y+6z=8$$

$$2x+8y+2z=4$$

$$6x+10y+3z=0$$

Solutions

```
%Solving by using left division
A=[4 -2 6;2 8 2;6 10 3];
B=[8;4;0];
X=A\B
%Solving by using the inverse
Xb=inv(A)*B
```

```
%Solving by using right division
C=[4 2 6;-2 8 10;6 2 3];
D=[8 4 0];
Xc=D/C
%Solving by using the inverse
Xd=D*inv(C)
```

Addressing matrix elements

- The following statements create a 3 by 3 matrix, print the (2,3) element and change the (3,2) element.

```
>> A = [1 2 3; 4 5 6; 7 8 9]
    A = 1 2 3
          4 5 6
          7 8 9
>> A(2,3)           %  ask MATLAB to print the (2,3) element
    ans =
        6
>> A(3,2) = -5      %  reassign the (2,3) element
    A =
        1   2   3
        4   5   6
        7  -5   9
```

Some features

size(M) returns number of rows, nr, and number of columns, nc, as a vector [nr , nc]

>> size(matrix)

ans =

3 3

trace(M) trace of M (sum of diagonal terms)

>> trace(matrix)

- ans =
- 14

det(M) determinant of M

- >> det(matrix)
- ans =
- -43

zeros(m,n) matrix of all zeros with m rows and n columns

```
>> zeros(3,3)
ans =
0 0 0
0 0 0
0 0 0
```

ones(m,n) matrix of all ones with m rows and n columns

```
>> ones(3,3)
ans =
1 1 1
1 1 1
1 1 1
```

eye(m,n) identity matrix with m rows and n columns

```
>> eye(3,3)
ans =
1 0 0
0 1 0
0 0 1
```

Some features

- >> matrix = [4 0 3; 0 3 5; 3 5 7]
- matrix =

  4 0 3

  0 3 5

  3 5 7
- Accessing individual components
- >> matrix(2,3)

  ans =

  5
- Accessing rows or columns

  >>matrix(: , 2)

  ans =

  0

  3

  5

  >>matrix(3, :)

  ans =

  3 5 7

Some features

```
>> matrix'
ans =
4 0 3
0 3 5
3 5 7 (no change since matrix is symmetric)

>> v = [ 1 2 3];
>> v*matrix
ans =
13 21 34

>> matrix* v'
ans =
13
21
34
```

Built-in Functions

- rand(n)-creates nxn matrix with randomly generated entries distributed uniformly between 0 and 1

- rand(m,n)-creates mxn matrix

- mean(A) If A is a vector returns the mean value of the elements of the vector

$$\mu = \frac{\sum_{k=1}^{N} x_k}{N}$$

Solving a system of linear equations Mx =b

$$1x_1 + 3x_2 + 5x_3 = 3$$
$$2x_1 + 1x_2 + 1x_3 = 2$$
$$4x_1 + 3x_2 + 6x_3 = 1$$

```
>> M =[ 1 3 5;
2 1 1;
4 3 6];
>> b =[3;2;1 ];
>> x =M\b
x =
-0.0909
3.9091
-1.7273
```

According to the Kirchhoff's Law, the sum of all currents flowing into a node is zero. Applying this rule to a certain circuit the following set of equations are obtained.

$$(R_1 + R_3 + R_4)I_1 + R_3I_2 + R_4I_3 = E_1$$
$$R_3I_1 + (R_2 + R_3 + R_5)I_2 - R_5I_3 = E_2$$
$$R_4I_1 - R_5I_2 + (R_4 + R_5 + R_6)I_3 = 0$$

If R1=1, R2=1, R3=2, R4=1, R5=2, R6=4 and E1=23, E2=29, solve the set of equations to find the values of currents (I1 , I2 and I3) passing through the circuit.