Matrix Algebra
Chapter 10
Objectives

After studying this chapter you should be able to:

• Perform the basic operations of matrix algebra
• Solve simultaneous equations using MATLAB matrix operations
• Use some of MATLAB’s special matrices
The difference between an array and a matrix

- Most engineers use the two terms interchangeably
- The only time you need to be concerned about the difference is when you perform matrix algebra calculations
Arrays

• Technically an array is an orderly grouping of information
• Arrays can contain numeric information, but they can also contain character data, symbolic data etc.
Matrix

- The technical definition of a matrix is a two-dimensional numeric array used in linear algebra.
- Not even all numeric arrays can precisely be called matrices - only those upon which you intend to perform linear transformations meet the strict definition of a matrix.
10.1 Matrix Operations and Functions

- Matrix algebra is used extensively in engineering applications
- Matrix algebra is different from the array calculations we have performed thus far
Array Operators

- **A.* B** multiplies each element in array A times the corresponding element in array B
- **A./B** divides each element in array A by the corresponding element in array B
- **A.^B** raises each element in array A to the power in the corresponding element of array B
Operators used in Matrix Mathematics

- Transpose
- Multiplication
- Division
- Exponentiation
- Left Division
Some Matrix Algebra functions

- Dot products
- Cross products
- Inverse
- Determinants
Transpose

- In mathematics texts you will often see the transpose indicated with superscript T
  - $A^T$
- The MATLAB syntax for the transpose is
  - $A'$
The transpose switches the rows and columns

\[ A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix} \quad A^T = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix} \]
MATLAB for Engineers 3E, by Holly Moore. © 2011 Pearson Education, Inc., Upper Saddle River, NJ. All rights reserved.

This material is protected by Copyright and written permission should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permission(s), write to: Rights and Permissions Department, Pearson Education, Inc., Upper Saddle River, NJ 07458.
Using the transpose with complex numbers

```matlab
>> x = [-1:-1:-3]
   x =
   -1  -2  -3
>> y=sqrt(x)
   y =
   0 + 1.0000i  0 + 1.4142i  0 + 1.7321i
>> y'
   ans =
   0 - 1.0000i
   0 - 1.4142i
   0 - 1.7321i
>> l
```

When used with complex numbers, the transpose operator returns the complex conjugate.
Dot Products

• The dot product is sometimes called the scalar product
• the sum of the results when you multiply two vectors together, element by element.
Equivalent statements

MATLAB for Engineers 3E, by Holly Moore. © 2011 Pearson Education, Inc., Upper Saddle River, NJ. All rights reserved.
Example 10.1
Calculating the Center of Gravity

- Finding the center of gravity of a structure is important in a number of engineering applications
- The location of the center of gravity can be calculated by dividing the system up into small components.
\[
\overline{x}W = x_1W_1 + x_2W_2 + x_3W_3 + etc...
\]

\[
\overline{y}W = y_1W_1 + y_2W_2 + y_3W_3 + etc...
\]

\[
\overline{z}W = z_1W_1 + z_2W_2 + z_3W_3 + etc...
\]

- In a rectangular coordinate system
- \( \overline{x}, \overline{y}, \) and \( \overline{z} \) are the coordinates of the center of gravity
- \( W \) is the total mass of the system
- \( x_1, x_2, \) and \( x_3 \) etc are the x coordinates of each system component
- \( y_1, y_2, \) and \( y_3 \) etc are the y coordinates of each system component
- \( z_1, z_2, \) and \( z_3 \) etc are the z coordinates of each system component
- \( W_1, W_2, \) and \( W_3 \) etc are the weights of each system component
In this example...

- We’ll find the center of gravity of a small collection of the components used in a complex space vehicle.
Vehicle Component Locations and Mass

<table>
<thead>
<tr>
<th>Item</th>
<th>x, meters</th>
<th>y, meters</th>
<th>z meters</th>
<th>Mass</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bolt</td>
<td>0.1</td>
<td>2</td>
<td>3</td>
<td>3.50 gram</td>
</tr>
<tr>
<td>screw</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1.50 gram</td>
</tr>
<tr>
<td>nut</td>
<td>1.5</td>
<td>0.2</td>
<td>0.5</td>
<td>0.79 gram</td>
</tr>
<tr>
<td>bracket</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1.75 gram</td>
</tr>
</tbody>
</table>

Formulate the problem using a dot product
Input and Output

• Input
  • Location of each component in an x-y-z coordinate system – in meters
  • Mass of each component, in grams

• Output
  • Location of the center of gravity
Hand Example
Find the x coordinate of the center of gravity

<table>
<thead>
<tr>
<th>Item</th>
<th>x, meters</th>
<th>Mass, gram</th>
<th>x * m, gram meters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bolt</td>
<td>0.1</td>
<td>3.50</td>
<td>= 0.35</td>
</tr>
<tr>
<td>screw</td>
<td>1</td>
<td>1.50</td>
<td>= 1.50</td>
</tr>
<tr>
<td>nut</td>
<td>1.5</td>
<td>0.79</td>
<td>= 1.1850</td>
</tr>
<tr>
<td>bracket</td>
<td>2</td>
<td>1.75</td>
<td>= 3.5</td>
</tr>
<tr>
<td>sum</td>
<td></td>
<td>7.54</td>
<td>6.535</td>
</tr>
</tbody>
</table>
We know that...

- The x coordinate is equal to

\[ \bar{x} = \frac{\sum_{i=1}^{3} x_i m_i}{m_{Total}} = \frac{3}{\sum_{i=1}^{3} m_{i}} \]

This is a dot product

- So...

\[ \bar{x} = 6.535/7.54 = 0.8667 \text{ meters} \]
%% Example 10.1
% Calculating the Center of Gravity

mass = [3.5, 1.5, 0.79, 1.75];
x=[0.1, 1, 1.5, 2];
x_bar=dot(x,mass)/sum(mass)
y=[2, 1, 0.2, 2];
y_bar=dot(y,mass)/sum(mass)
z=[3, 1, 0.5, 4];
z_bar=dot(z,mass)/sum(mass)

x_bar = 0.87
y_bar = 1.61
z_bar = 2.57
%% Example 10.1

% Calculating the Center of Gravity

mass = [3.5, 1.5, 0.79, 1.75];
x=[0.1, 1, 1.5, 2];

x_bar = dot(x, mass) / sum(mass)
y=[2, 1, 0.2, 2];
y_bar = dot(y, mass) / sum(mass)
z=[3, 1, 0.5, 4];

z_bar = dot(z, mass) / sum(mass)

%% Plot the results

plot3(x, y, z, 'o', x_bar, y_bar, z_bar, 's')
grid on
xlabel('x-axis')
ylabel('y-axis')
zlabel('z-axis')
title('Center of Gravity')
axis([0, 2, 0, 2, 0, 4])

% The figure in the book was enhanced using the interactive plotting tools
We could use a plot to evaluate our results.

This plot was enhanced using the interactive plotting tools.
%% Example 10.2
%% Find the angle between two force vectors
%% Define the vectors
A = [5 6 3];
B = [1 3 2];

%% Calculate the magnitude of each vector
mag_A = sqrt(sum(A.^2));
mag_B = sqrt(sum(B.^2));

%% Calculate the cosine of theta
cos_theta = dot(A,B)/(mag_A*mag_B);

%% Find theta
theta = acos(cos_theta);

%% Send the results to the command window
fprintf('The angle between the vectors is %4.3f radians \n',theta)
fprintf('or %6.2f degrees \n',theta*180/pi)
Matrix Multiplication

• Similar to a dot product

```matlab
>> A = [1 2 3]
A =
   1   2   3
>> B=[3; 4; 5]
B =
   3
   4
   5
>> A*B
ans =
   26
>> dot(A,B)
ans =
   26
>>
```
Matrix Multiplication

• Matrix multiplication results in an array where each element is a dot product.

• In general, the results are found by taking the dot product of each row in matrix A with each column in Matrix B.
MATLAB for Engineers 3E, by Holly Moore. © 2011 Pearson Education, Inc., Upper Saddle River, NJ. All rights reserved.

This material is protected by Copyright and written permission should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permission(s), write to: Rights and Permissions Department, Pearson Education, Inc., Upper Saddle River, NJ 07458.

```
>> A=[1 2 3; 4 5 6]
A =
    1  2  3
    4  5  6

>> B=[10 20 30; 40 50 60; 70 80 90]
B =
    10  20  30
    40  50  60
    70  80  90

>> C=A*B
C =
    300  360  420
    660  810  960
```

\[ C_{i,j} = \sum_{k=1}^{N} A_{i,k} B_{k,j} \]
• Because matrix multiplication is a series of dot products

• the number of columns in matrix A must equal the number of rows in matrix B. **So, AxB \neq BxA**

• For an m x n matrix multiplied by an n x p matrix

\[
\begin{align*}
\text{m x n} & \quad \text{n x p} \\
\end{align*}
\]

These dimensions must match

The resulting matrix will have these dimensions
We could use matrix multiplication to solve the problem in Example 10.1, in a single step.

USING MATRIX MULTIPLICATION TO FIND THE CENTER OF GRAVITY
Matrix Powers

- Raising a matrix to a power is equivalent to multiplying it times itself the requisite number of times
  - $A^2$ is the same as $A*A$
  - $A^3$ is the same as $A*A*A$
- Raising a matrix to a power requires it to have the same number of rows and columns.
Matrix Inverse

• MATLAB offers two approaches
  • The matrix inverse function
    • inv(A)
  • Raising a matrix to the -1 power
    • A^{-1}
A matrix times its inverse is the identity matrix

Equivalent approaches to finding the inverse of a matrix

MATLAB for Engineers 3E, by Holly Moore. © 2011 Pearson Education, Inc., Upper Saddle River, NJ. All rights reserved.
This material is protected by Copyright and written permission should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permission(s), write to: Rights and Permissions Department, Pearson Education, Inc., Upper Saddle River, NJ 07458.
Not all matrices have an inverse

- Called
  - Singular
  - Ill-conditioned matrices

- Attempting to take the inverse of a singular matrix results in an error statement
Not all matrices have an inverse

3. Consider the following matrix:

\[ A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix} \]

Would you expect it to be singular or not? (Recall that singular matrices have a determinant of 0 and do not have an inverse.)
Determinants

- Related to the matrix inverse
- If the determinant is equal to 0, the matrix does not have an inverse
- The MATLAB function to find a determinant is
  - \( \text{det}(A) \)
```
>> A=[1 2;3 4]
A =
   1  2
   3  4
>> det(A)
ans =
   -2
>> inv(A)
```
```
>> A=[1 2;3 4;5 6;7 8;9]
A =
   1  2  3
   4  5  6
   7  8  9
>> det(A)
ans =
```
```
Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND = 1.541976e-018.
```
Matrix Determinant

• The determinant of a square matrix is a very useful value for finding if a system of equations has a solution or not.

• If it is equal to zero, there is no solution.

Notation: Determinant of A = |A| or det(A)

Formula for a 2x2 matrix:

\[ M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \]

\[ \text{det}(M) = m_{11} m_{22} - m_{21} m_{12} \]

IMPORTANT: the determinant of a matrix is a scalar
Matrix Inverse

- The inverse of a matrix is really important concept, for matrix algebra.
- Calculating a matrix inverse is very tedious for matrices bigger than 2x2. We will do that numerically with Matlab.

**Notation:** inverse of $A = A^{-1}$ or $\text{inv}(A)$

**Formula for a 2x2 matrix:**

$$M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \quad \text{and} \quad M^{-1} = \frac{1}{\det(M)} \begin{pmatrix} m_{22} & -m_{12} \\ -m_{21} & m_{11} \end{pmatrix}$$

**IMPORTANT:** the inverse of a matrix is a matrix.
Matrices properties

Property of inverse:

\[ A \times A^{-1} = I \]

and \[ A^{-1} \times A = I \]

Example:

\[
\begin{pmatrix}
1 & 1 & 2 \\
1 & 2 & 1 \\
2 & 0 & 1
\end{pmatrix}
\times
\begin{pmatrix}
-0.4 & 0.2 & 0.6 \\
-0.2 & 0.6 & -0.2 \\
0.8 & -0.4 & -0.2
\end{pmatrix}
=
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

Property of identity matrix:

\[ I \times A = A \]

and \[ A \times I = A \]
Cross Products

- sometimes called vector products
- the result of a cross product is a vector
- always at right angles (normal) to the plane defined by the two input vectors
- orthogonality
Consider two vectors

\[ A = A_x \vec{i} + A_y \vec{j} + A_z \vec{k} \]

\[ B = B_x \vec{i} + B_y \vec{j} + B_z \vec{k} \]

The cross product is equal to…

\[ A \times B = (A_y B_z - A_z B_y) \vec{i} + (A_z B_x - A_x B_z) \vec{j} + (A_x B_y - A_y B_x) \vec{k} \]
MATLAB for Engineers 3E, by Holly Moore. © 2011 Pearson Education, Inc., Upper Saddle River, NJ. All rights reserved.

This material is protected by Copyright and written permission should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permission(s), write to: Rights and Permissions Department, Pearson Education, Inc., Upper Saddle River, NJ 07458.
Cross Products are Widely Used

- Cross products find wide use in statics, dynamics, fluid mechanics and electrical engineering problems
Cross Products are Widely Used

MOMENT OF A FORCE ABOUT A POINT

The moment of a force about a point is found by computing the cross product of a vector that defines the position of the force with respect to a point, with the force vector:

\[ M_\theta = r \times F \]

Consider the force applied at the end of a lever, as shown in Figure 10.4. If you apply a force to the lever close to the pivot point, the effect is different than if you apply a force further out on the lever. That effect is called the moment.

Calculate the moment about the pivot point on a lever for a force described as the vector

\[ \vec{F} = -100\hat{i} + 20\hat{j} + 0\hat{k} \]

Assume that the lever is 12 inches long, at an angle of 45° from the horizontal. This means that the position vector can be represented as

\[ \vec{r} = \frac{12}{\sqrt{2}}\hat{i} + \frac{12}{\sqrt{2}}\hat{j} + 0\hat{k} \]

1. State the Problem
   Find the moment of a force vector about the pivot point of a lever.
Cross Products are Widely Used

%% Example 10.4
%Moment about a pivot point
%Define the position vector
r = [12/sqrt(2), 12/sqrt(2), 0];
%% Define the force vector
F = [-100, 20, 0];
%% Calculate the moment
moment=cross(r,F)
Cross Products are Widely Used

%%% More complicated Example
%Example 10.5
%Moment about a pivot point
%Define the position vector
%Define the force vector
%Calculate the moment
%Print the results
%% More complicated Example
%Example 10.5
%Moment about a pivot point
%Define the position vector
clear,clc
rx=input('Enter the x component of the position vector: ');
ry=input('Enter the y component of the position vector: ');
rz=input('Enter the z component of the position vector: ');
r = [rx, ry, rz];
disp('The position vector is')
fprintf('%8.2f i + %8.2f j  + %8.2f k  ft\n',r)
%Define the force vector
Fx=input('Enter the x component of the force vector: ');
Fy=input('Enter the y component of the force vector: ');
Fz=input('Enter the z component of the force vector: ');
F = [Fx, Fy, Fz];
disp('The force vector is')
fprintf('%8.2f i + %8.2f j  + %8.2f k lbf\n',F)
%Calculate the moment
moment=cross(r,F);
disp('The moment vector about the pivot point is \n')
fprintf('%8.2f i + %8.2f j  + %8.2f k ft-lbf\n',moment)
Solving systems of linear equations

Example: 3 equations and 3 unknown

\[
\begin{align*}
1x + 6y + 7z &= 0 \\
2x + 5y + 8z &= 1 \\
3x + 4y + 5z &= 2
\end{align*}
\]

Can be easily solved by hand, but what can we do if it we have 10 or 100 equations?
Solving systems of linear equations

1x + 6y + 7z = 0
2x + 5y + 8z = 1
3x + 4y + 5z = 2

First, write a matrix with all the (xyz) coefficients

\[
A = \begin{bmatrix}
1 & 6 & 7 \\
2 & 5 & 8 \\
3 & 4 & 5
\end{bmatrix}
\]

Write a matrix with all the constants

\[
B = \begin{bmatrix}
0 \\
1 \\
2
\end{bmatrix}
\]

Finally, consider the matrix of unknowns

\[
X = \begin{bmatrix}
x \\
y \\
z
\end{bmatrix}
\]
Solving systems of linear equations

\[ A \times X = B \]

\[ A^{-1} \times A \times X = A^{-1} \times B \]

\[ (A^{-1} \times A) \times X = A^{-1} \times B \]

\[ I \times X = A^{-1} \times B \]

\[ X = A^{-1} \times B \]
Solving systems of linear equations

The previous set of equations can be expressed in the following vector-matrix form:

\[ A \times X = B \]

\[
\begin{pmatrix}
1 & 6 & 7 \\
2 & 5 & 8 \\
3 & 4 & 5
\end{pmatrix}
\begin{pmatrix}
x \\
y \\
z
\end{pmatrix}
= 
\begin{pmatrix}
0 \\
1 \\
2
\end{pmatrix}
\]
Solving systems of equations in Matlab

\[ \begin{align*}
  x + 6y + 7z &= 0 \\
  2x + 5y + 8z &= 1 \\
  3x + 4y + 5z &= 2
\end{align*} \]

In Matlab:

\[
\begin{align*}
  A &= \begin{pmatrix} 1 & 6 & 7 \\ 2 & 5 & 8 \\ 3 & 4 & 5 \end{pmatrix} \\
  X &= \begin{pmatrix} x \\ y \\ z \end{pmatrix} \\
  B &= \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}
\end{align*}
\]

\[
\begin{align*}
  \text{Verification:} \\
  \text{ans} &= \text{det}(A) \\
  &= 28
\end{align*}
\]

\[
\begin{align*}
  X &= \begin{pmatrix} 0.8571 \\ -0.1429 \\ 0 \end{pmatrix}
\end{align*}
\]
Solving systems of equations in Matlab

\[ \begin{align*}
x + 6y + 7z &= 0 \\
2x + 5y + 8z &= 1 \\
3x + 4y + 9z &= 2
\end{align*} \]

In Matlab:

\[
\begin{align*}
&>> A=\begin{bmatrix} 1 & 6 & 7 \\ 2 & 5 & 8 \\ 3 & 4 & 9 \end{bmatrix} \\
&>> B=\begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} \\
&>> S=inv(A)*B
\end{align*}
\]

Warning: Matrix is singular to working precision.

\[
\begin{align*}
&>> S = \\
&\text{NaN} \\
&\text{NaN} \\
&\text{NaN}
\end{align*}
\]

Verification:

\[
\begin{align*}
&>> \text{det}(A) \\
&\text{ans} = \\
&0
\end{align*}
\]

NO Solution!!!!!!
10.2 Solutions to Systems of Linear Equations - Example

\[ \begin{align*}
3x + 2y - z &= 10 \\
-x + 3y + 2z &= 5 \\
x - y - z &= -1
\end{align*} \]
Using Matrix Nomenclature

\[
A = \begin{bmatrix}
3 & 2 & -1 \\
-1 & 3 & 2 \\
1 & -1 & -1 \\
\end{bmatrix} \quad X = \begin{bmatrix}
x \\
y \\
z \\
\end{bmatrix} \quad B = \begin{bmatrix}
10 \\
5 \\
-1 \\
\end{bmatrix}
\]

and

\[AX = B\]
We can solve this problem using the matrix inverse approach:

```matlab
>> A=[3 2 -1; -1 3 2; 1 -1 -1]
A =
    3     2    -1
   -1     3     2
    1    -1    -1
>> B=[10; 5; -1]
B =
    10
    5
   -1
>> X = inv(A)*B
X =
   -2.0000
    5.0000
   -6.0000
>>
```

This approach is easy to understand, but it's not the more efficient computationally.
Matrix left division uses Gaussian elimination, which is much more efficient, and less prone to round-off error.
Applications in Physics

Find the value of the forces F1 and F2
Applications in Physics

Projections on the X axis

\[ F_1 \cos(60) + F_2 \cos(80) - 7 \cos(20) - 5 \cos(30) = 0 \]
Applications in Physics

Projections on the Y axis

\[ F_1 \sin(60) - F_2 \sin(80) + 7 \sin(20) - 5 \sin(30) = 0 \]
Applications in Physics

F1 \cos(60) + F2 \cos(80) - 7 \cos(20) - 5 \cos(30) = 0
F1 \sin(60) - F2 \sin(80) + 7 \sin(20) - 5 \sin(30) = 0

In Matlab:
>> CF=pi/180;
>> A=[cos(60*CF), cos(80*CF); sin(60*CF), -sin(80*CF)];
>> B=[7*cos(20*CF)+5*cos(30*CF); -7*sin(20*CF)+5*sin(30*CF)];
>> F= inv(A)*B  or (A\B)

F =
16.7406  
14.6139

Solution:
F1= 16.7406 N
F2= 14.6139 N
10.3 Special Matrices

- We introduced some of MATLAB’s special matrices in previous chapters
  - ones
  - zeros
The identity matrix is another special matrix that is useful in Matrix Algebra.

It may be tempting to name an identity matrix \( i \), however \( i \) is already in-use for imaginary numbers.
Other matrices

- MATLAB includes a number of matrices that are useful for testing numerical techniques, computational algorithms, or that are just interesting
  - pascal
  - magic
  - rosser
- gallery – contains over 50 different test matrices
Summary

• Matrix algebra and array mathematics are significantly different

• The .* , ./ and .^ operators perform element-by-element computations

• The *, / and ^ operators transform entire matrices
Summary – Dot Product

- The dot product is the sum of the array multiplications of two equal size vectors.

\[ C = \sum_{i=1}^{N} A_i \ast B_i \]

- The MATLAB function for dot products is \texttt{dot(A,B)}
Summary – Matrix Multiplication

- Matrix multiplication is similar to the dot product
- Each element of the result array is a dot product

\[ C_{i,j} = \sum_{k=1}^{N} A_{i,k} B_{k,j} \]
Summary - Inverse

- A matrix times its inverse is equal to the identity matrix
- The MATLAB syntax to find a matrix inverse is $\text{inv}(A)$ or $A^{^-1}$
Summary - Determinants

- The matrix inverse is related to the determinant
- If a matrix has a determinant equal to zero it does not have an inverse
- The syntax for the determinant is \( \text{det}(A) \)
Summary – Cross Products

- Cross product is often called a vector product
- It produces a vector at right angles to the two input vectors
- The MATLAB syntax for cross products is cross(A,B)
Summary – Solving Linear Systems of Equations

• Use the matrix inverse approach
  • \( X = \text{inv}(A) \times B \)

• Or use the left division approach
  • \( X = A \backslash B \)

• Left division uses Gaussian elimination and is the preferred approach