MATH.2720 Introduction to Programming with MATLAB
Vector and Matrix Algebra

A. Vectors

A *vector* is a quantity that has both magnitude and direction, like velocity. The location of a vector is irrelevant; all that matters are magnitude and direction. You can visualize a vector as an arrow, with the length of the arrow representing the magnitude of the vector and the direction of the arrow representing the direction of the vector.

A vector is usually denoted by a bold-face lower-case letter (e.g. $\mathbf{v}$) or by a lower-case letter with an arrow above it (e.g. $\vec{v}$). The *magnitude* (or *norm*) of a vector $\vec{v}$ is usually denoted either $\|\vec{v}\|$ or $|\vec{v}|$.

If you think of a vector as an arrow with its tail at the origin of a coordinate system, you can describe the vector analytically by specifying the location of the head of the vector. For example, $\vec{v} =< 1, 2 >$ is the vector in the xy plane that starts at the origin and ends at the point $(1, 2)$. A vector can have 2, 3, or more components. The magnitude of a vector is the distance from the tail to the head of the vector. For example, $\|< 1, 2 >\| = \sqrt{1^2 + 2^2} = \sqrt{5}$ by the distance formula.

MATLAB syntax: `>> norm([1 2])`

Vector Operations

1. Scalar Multiplication.

   If k is a real number (a *scalar*), then $k\vec{v}$ is the vector with magnitude $|k|\ \|\vec{v}\|$ and direction the same direction as $\vec{v}$ if $k > 0$ and the opposite direction of $\vec{v}$ if $k < 0$.

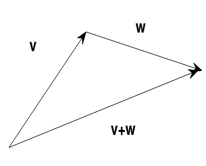
   Analytical definition: $k < v_1, v_2, v_3 >=< kv_1, kv_2, kv_3 >$.

   For example, $-2 < 1, 2, 3 >=< -2, -4, -6 >$

   MATLAB syntax: `>> -2*[1 2 3]`

2. Vector Addition.

   Geometric definition of $\vec{v} + \vec{w}$: Place the tail of $\vec{w}$ at the head of $\vec{v}$. The vector from the tail of $\vec{v}$ to the head of $\vec{w}$ is $\vec{v} + \vec{w}$. See the figure below.



   Analytical definition of vector addition:

   $< v_1, v_2, v_3 > + < w_1, w_2, w_3 >=< v_1 + w_1, v_2 + w_2, v_3 + w_3 >$.

   For example, $< 1, 2, 3 > + < 4, 5, 6 >=< 5, 7, 9 >$

   MATLAB syntax: `>> [1 2 3] + [4 5 6]`

3. Dot Product (or Inner Product).
   The dot product of two vectors of the same length is a *scalar.*
   Geometric definition: $\vec{v} \cdot \vec{w} = \|\vec{v}\| \, \|\vec{w}\| \cos(\theta)$, where θ is the angle between $\vec{v}$ and $\vec{w}$ when the vectors have their tails at the same point.
   Analytical definition: $\vec{v} \cdot \vec{w} = v_1 w_1 + v_2 w_2 + \cdots + v_n w_n$.
   For example, $< 1, 2, 3 > \cdot < 4, 5, 6 >= (1)(4) + (2)(5) + (3)(6) = 32$.
   MATLAB syntax: `>> dot([1 2 3], [4 5 6])`

4. Cross Product.
   The cross product of two 3-component vectors $\vec{v}$ and $\vec{w}$ is a vector with magnitude $\|\vec{v}\| \, \|\vec{w}\| \sin(\theta)$ and direction perpendicular to both $\vec{v}$ and $\vec{w}$ per the right-hand rule.
   Analytical definition:
   $< v_1, v_2, v_3 > \times < w_1, w_2, w_3 >=< v_2 w_3 - w_2 v_3, v_3 w_1 - w_3 v_1, v_1 w_2 - w_1 v_2 >$.
   For example, $< 1, 0, 3 > \times < 0, 2, -1 >=< 0(-1) - 2(3), 3(0) - (-1)(1), 1(2) - (-1)(0) >$ $=< -6, 1, 2 >$.
   MATLAB syntax: `>> cross([1 0 3], [0 2 -1])`

B. Matrices

A *matrix* is a rectangular array of numbers. (The plural of matrix is *matrices.*)

An $m \times n$ matrix is a matrix with m rows and n columns. Here is an example of a 2×3 matrix:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

If A is a matrix, then A_{ij} denotes the element in row i and column j of matrix A. For example, if A is the matrix defined above, then $A_{21} = 4$.

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

Matrix Operations

1. Scalar Multiplication.
   If A is an $m \times n$ matrix and k is a scalar, then kA is the $m \times n$ matrix whose entries are k times the entries of A.
   For example,

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \Rightarrow 2A = \begin{bmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \end{bmatrix}$$

   MATLAB syntax: `>> 2*A`

2. Matrix Addition.
   If A and B are $m \times n$ matrices, then $A + B$ is the $m \times n$ matrix with $(A + B)_{ij} = A_{ij} + B_{ij}$.
   For example,

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ and } B = \begin{bmatrix} -2 & 4 \\ -6 & 8 \end{bmatrix} \Rightarrow A + B = \begin{bmatrix} -1 & 6 \\ -3 & 12 \end{bmatrix}$$

   MATLAB syntax: `>> A+B`

3. Matrix Multiplication.

   If A is an $m \times n$ matrix and B is an $n \times p$ matrix, then AB is the $m \times p$ whose ij entry equals the dot product of row i of A and column j of B. Note that for the product AB to be defined, the number of columns of A must equal the number of rows of B.

   For example,

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ and } B = \begin{bmatrix} -2 & 4 \\ -6 & 8 \end{bmatrix} \Rightarrow AB = \begin{bmatrix} 1(-2)+2(-6) & 1(4)+2(8) \\ 3(-2)+4(-6) & 3(4)+4(8) \end{bmatrix} = \begin{bmatrix} -14 & 20 \\ -30 & 44 \end{bmatrix}$$

   MATLAB syntax: `>> A*B`

   Note that even if A and B are both $n \times n$ matrices, in general $AB \neq BA$.

   For example,

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ and } B = \begin{bmatrix} -2 & 4 \\ -6 & 8 \end{bmatrix} \Rightarrow AB = \begin{bmatrix} -14 & 20 \\ -30 & 44 \end{bmatrix} \text{ but } BA = \begin{bmatrix} 10 & 12 \\ 18 & 20 \end{bmatrix}$$

   The *identity matrix* I_n is the $n \times n$ matrix with 1 along the diagonal and 0 everywhere else.

   For example,

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

   MATLAB syntax: `>> eye(3)`

   If A is an $m \times n$ matrix, then $I_m A = A$ and $AI_n = A$.

4. Inverse of a Matrix.

   For most $n \times n$ matrices A there exists an *inverse matrix* A^{-1} with the property that $AA^{-1} = A^{-1}A = I_n$.

   For example,

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \Rightarrow A^{-1} = \begin{bmatrix} -2 & 1 \\ 3/2 & -1/2 \end{bmatrix}$$

   (You should check this by calculating AA^{-1} and $A^{-1}A$.)

   MATLAB syntax: `>> inv(A)`

5. Determinant of an $n \times n$ matrix.

   The determinant of an $n \times n$ matrix A is a scalar, denoted $\det(A)$ or $|A|$.

   If $\det(A) \neq 0$, then A^{-1} exists and A is said to be *nonsingular.*

   If $\det(A) = 0$, then A^{-1} does not exist and A is said to be *singular.*

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = aei + bfg + cdh - gec - hfa - idb$$

   MATLAB syntax: `>> det(A)`

6. Row-echelon form of a matrix (for those of you who have studied linear algebra).

   The command

   `>>rref(A)`

   generates the reduced row echelon form of matrix A.

C. Systems of Linear Equations

Systems of linear equations can be expressed as matrix equations. For example, the system $x_1 + 2x_2 = 4,\ 3x_1 + 4x_2 = 10$ can be written as the matrix equation $Ax = b$ where $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, and $b = \begin{bmatrix} 4 \\ 10 \end{bmatrix}$.

You can solve this system using the following MATLAB commands.

```
>> A = [1, 2; 3, 4];
>> b = [4; 10];
>> x = A\b %Note: This is the backslash key (above the Enter key), not /
```

If the system $Ax = b$ is overdetermined (more equations than unknowns), then `A\b` is a least-squares solution of the system, i.e., a vector x that minimizes $\|Ax - b\|$. If the system $Ax = b$ is underdetermined (more unknowns than equations), then `A\b` produces a solution of the system, if there are any, or a least-squares solution if the system has no solution.

D. Operations on Arrays

The symbol `*` denotes matrix multiplication. If you want to multiply corresponding elements of arrays with the same dimensions, use `.*`

For example, `>>[1 2 3]*[4 5 6]` produces an error message in MATLAB, but `>>[1 2 3].*[4 5 6]` produces the array `[4 10 18]`.

Similarly, you can perform element-by-element division or exponentiation using `./` and `.^`

For example, `[1 2 3].^2` produces the array `[1 4 9]`

You can apply built-in MATLAB functions to arrays, just as you can to single numbers. For example, `sqrt([1 4 9])` produces the array `[1 2 3]`

E. MATLAB Array Functions

Here are some useful MATLAB functions for working with arrays.

MATLAB Command	Description
max(A)	Largest element of A, if A is a vector Row vector containing largest element in each column, if A is a matrix
min(A)	Same as max(A) but gives minimum instead
sum(A)	Sum of the elements of A if A is a vector
mean(A)	Average of the elements of A if A is a vector

Practice Problems

1. The unit vector $\vec{u}_n$ in the direction of vector $\vec{u}$ is given by $\left(\frac{1}{|\vec{u}|}\right)\vec{u}$. Find the unit vector in the direction of $\vec{u} =< -8, -14, 25 >$ using one MATLAB command.

2. Define the vector $v = [2, 4, 6, 8, 10]$. Then use v to create the following vectors:
   (a) $a = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{6} & \frac{1}{8} & \frac{1}{10} \end{bmatrix}$ (b) $b = \begin{bmatrix} \frac{1}{2^2} & \frac{1}{4^2} & \frac{1}{6^2} & \frac{1}{8^2} & \frac{1}{10^2} \end{bmatrix}$
   (c) $c = [1\ 2\ 3\ 4\ 5]$ (d) $d = [1\ 1\ 1\ 1\ 1]$

3. Define the vectors $\vec{u} =< -2, 6, 5 >$, $\vec{v} =< 5, -1, 3 >$, and $\vec{w} =< 4, 7, -2 >$. Use MATLAB's built-in functions `cross` and `dot` to verify the vector identity $\vec{u} \times (\vec{v} \times \vec{w}) = (\vec{u} \cdot \vec{w})\,\vec{v} - (\vec{u} \cdot \vec{v})\,\vec{w}$.

4. Solve the following system of linear equations.

$$\begin{cases} x + 2y - 3z &= -5 \\ 2x - y - z &= 0 \\ -x - y + z &= 1 \end{cases}$$

5. (a) Generate the row array v = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]. (Can you do this without typing in all 10 elements?)
   (b) Use a MATLAB function to find the average value of the entries of v.
   (c) Use a MATLAB function to find the sum of the entries of v.

Partial Answers to Practice Problems

1. $< -0.26892, -0.4706, 0.84037 >$
2.
3. Both sides of the equation should equal $< 124, -17, 70 >$
4. $x = 1,\ y = 0,\ z = 2$
5. b) 38.5 c) 385