GLOBAL EDITION

Introduction to MATLAB®

THIRD EDITION

Delores M. Etter

ALWAYS LEARNING

PEARSON

Introduction to MATLAB®

```
-10.0000   1.0000
  0        1.5888
  .
  .
  .
100.0000   65.5257
110.0000   88.4608
120.0000   118.1931
```

The hand solution and the MATLAB solution match for $T = 0$°F. The Clausius–Clapeyron equation can be used for more than just humidity problems. By changing the value of ΔH and R, you could generalize the program to any condensing vapor.

3.3 TRIGONOMETRIC FUNCTIONS

The trigonometric functions **sin**, **cos**, and **tan** all assume that angles are represented in radians. The trigonometric functions **sind**, **cosd**, and **tand** all assume that angles are represented in degrees.

In trigonometric calculations, the value of π is often needed, so a constant, **pi,** is built into MATLAB. However, since π cannot be expressed as a floating-point number, the constant **pi** in MATLAB is only an approximation of the mathematical quantity π. Usually, this is not important; however, you may notice some surprising results—for example,

```
sin(pi)
ans =
      1.2246e-16
```

We expected an answer of zero; we got a very small answer, but not zero.

Using the Help browser described in Section 3.1, you can obtain a complete list of trigonometric functions available in MATLAB. Note that there are function for arguments in radians, and functions for arguments in degrees; some of these are listed here:

sin(x) Computes the sine of **x**, where **x** is in radians.

```
sin(pi/2)
ans =
     1
```

sind(x) Computes the sine of **x**, where **x** is in degrees.

```
sind(90)
ans =
     1
```

cos(x) Computes the cosine of **x**, where **x** is in radians.

```
cos(pi)
ans =
     -1
```

`tan(x)` Computes the tangent of **`x`**, where **`x`** is in radians.

```
tan(pi)
ans =
     -1.2246e-16
```

`asin(x)` Computes the arcsine, or inverse sine, of **`x`**, where **`x`** must be between -1 and 1. The function returns an angle in radians between $\pi/2$ and $-\frac{\pi}{2}$.

```
asin(-1)
ans =
     -1.5708
```

`sinh(x)` Computes the hyperbolic sine of **`x`**, where **`x`** is in radians.

```
sinh(pi)
ans =
     11.5487
```

EXAMPLE 3.2

COMPUTING DISTANCES USING GPS

The GPS (Global Positioning System) coordinates that specify a location on the Earth are the latitude and longitude values for the position. In this section, we develop an algorithm and MATLAB program to determine the distance between two objects given their latitudes and longitudes. Before developing the programs, we need to briefly discuss latitudes and longitudes and develop an equation to determine the distance between the two points using these coordinates.

Assume that Earth is represented by a sphere with a radius of 3960 miles. A **great circle** is formed by the intersection of this sphere and a plane that passes through the center of the sphere. If the plane does not pass through the center of the sphere, it will be a circle with a smaller circumference and hence is not a great circle. The **prime meridian** is a north–south great circle that passes through Greenwich, just outside London, and through the North Pole. The **equator** is an east–west great circle that is equidistant from the North Pole and the South Pole. Thus, we can define a **rectangular coordinate** system such that the origin is the center of the Earth, the z-axis goes from the center of the Earth through the North Pole, and the x-axis goes from the center of the Earth through the point where the prime meridian intersects the equator. (See Figure 3.4.) The **latitude** is an angular distance, is measured in degrees, and extends northward or southward from the

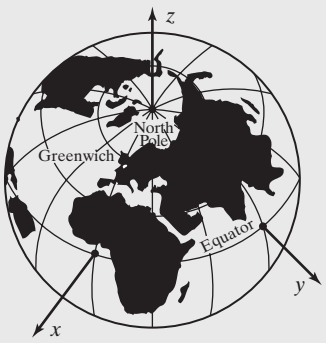


Figure 3.4
Rectangular Coordinate System for the Earth.

(*continued*)

equator (as in 25 N); and the **longitude** is an angular distance, is measured in degrees, and extends westward or eastward from the prime meridian (as in 120 W).

The **Global Positioning System (GPS)**, originally developed for military use, uses 24 satellites circling the Earth to pinpoint a location on the surface. Each satellite broadcasts a coded radio signal indicating the time and the satellite's exact position 11,000 miles above the Earth. The satellites are equipped with an atomic clock that is accurate to within one second every 70,000 years. A GPS receiver picks up the satellite signal and measures the time between the signal's transmission and its reception. By comparing signals from at least three satellites, the receiver can determine the latitude, longitude, and altitude of its position.

The shortest distance between two points on a sphere is known to be on the arc of the great circle containing them. If we know the angle between vectors from the center of the Earth to the two points defining the arc, we can then estimate the distance as a proportion of the Earth's circumference. To illustrate, suppose that the angle between two vectors from the center of the Earth is 45°. Then the angle is 45/360, or 1/8 of a complete revolution. Hence, the distance between the two points is 1/8 of the Earth's circumference (pi times twice the radius) or 3110 miles.

The best way to compute the shortest distance between two points that are specified in latitude (α) and longitude (β) is through a series of coordinate transformations. Recall that the **spherical coordinates** (ρ, φ, θ) of a point P in a rectangular coordinate system represents the length ρ (rho) of the vector connection the point to the origin, the angle φ (phi) between the positive z-axis and the vector, and the angle θ (theta) between the x-axis and the projection of the vector in the xy-plane. (See Figure 3.5.) We then convert the spherical coordinates to rectangular coordinates (x, y, z). Finally a simple trigonometric equation computes the angle between the points (or vectors) in rectangular coordinates. Once we know the angle between the points, we can then use the technique described in the previous paragraph to find the distance between the two points.

We need to use equations that relate latitude and longitude to spherical coordinates, that convert spherical coordinates to rectangular coordinates, and that compute the angle between two vectors. Figure 3.5 is useful in relating the notation to the following equations:

- Latitude/longitude and spherical coordinates:

$$\alpha = 90° - \varphi, \beta = 360° - \theta$$

- Spherical and rectangular coordinates:

$$\mathrm{x} = \rho \sin\varphi \cos\theta, \mathrm{y} = \rho \sin\varphi \sin\theta, \mathrm{z} = \rho \cos\theta$$

- Angle γ between two vectors a and b:

$$\cos\gamma = \mathrm{a} \cdot \mathrm{b} / (|\mathrm{a}||\mathrm{b}|)$$

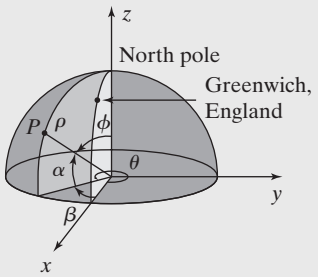


Figure 3.5
Spherical Coordinate System for the Earth.

where a · b is the dot product (defined below) of a and b and where |a| is the length of the vector a (also defined below)

- Dot product of two vectors (xa, ya, za) and (xb, yb, zb) in rectangular coordinates:

$$\mathrm{a}\cdot\mathrm{b} = \mathrm{xa}\cdot\mathrm{xb} + \mathrm{ya}\cdot\mathrm{yb} + \mathrm{za}\cdot\mathrm{zb}$$

- Length of a vector (xa, ya, za) in rectangular coordinates:

$$\sqrt{(xa^2 + ya^2 + za^2)}$$

- Great circle distance:

$$\begin{aligned}\text{distance} &= (\gamma/(2\pi))(\text{Earth's circumference})\\ &= (\gamma/(2\pi))(\pi\cdot 2\ \text{radius}) = \gamma\cdot 3960.\end{aligned}$$

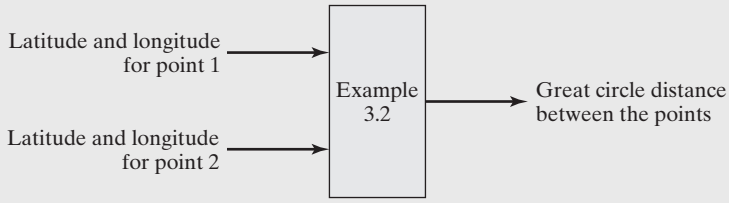
Pay close attention to the angle units in these equations. Unless otherwise specified, it is assumed that the angles are measured in radians.

Write a MATLAB program that asks the user to enter the latitude and longitude coordinates for two points in the Northern Hemisphere. Then compute and print the shortest distance between the two points.

1. Problem Statement

Compute the shortest distance between two points in the Northern Hemisphere.

2. Input/Output Description



3. Hand Example

For a hand example, we will compute the great circle distance between New York and London. The latitude and longitude of New York is 40.75°N and 74°W, respectively, and the latitude and longitude of London is 51.5°N and 0°W, respectively.

The spherical coordinates for New York are

$$\varphi = (90 - 40.75)^\circ = 49.25(\pi/180) = 0.8596 \text{ radians}$$

$$\theta = (360 - 74)^\circ = 286(\pi/180) = 4.9916 \text{ radians}$$

$$\rho = 3960.$$

The rectangle coordinates for New York (to two decimal places) are

$$\mathrm{x} = \rho \sin\varphi \cos\theta = 826.90$$

$$\mathrm{y} = \rho \sin\varphi \sin\theta = -2883.74$$

$$\mathrm{z} = \rho \cos\theta = 2584.93.$$

Similarly, the rectangular coordinates for London can be computed to be

$$\mathrm{x} = 2465.16,\ \mathrm{y} = 0,\ \mathrm{z} = 3099.13.$$

(*continued*)

The cosine of the angle between the two vectors is equal to the dot product of the two vectors divided by the product of their lengths, or 0.6408. Using an inverse cosine function, γ can be determined to be 0.875 radians. Finally, the distance between New York and London is

$$0.875 \cdot 3960 = 3466 \text{ miles}.$$

4. MATLAB Program

```
%--------------------------------------------------------------
% Example 3_2 This program determines the distance between
% two points that are specified with latitude and longitude
% values that are in the Northern Hemisphere.
%
clear, clc
%
% Get locations of two points.
lat1 = input('Enter latitude north for point 1: ');
long1 = input('Enter longitude west for point 1: ');
lat2 = input('Enter latitude north for point 2: ');
long2 = input('Enter longitude west for point 2: ');
%
% Convert latitude and longitude to rectangular coordinates.
rho = 3960;
phi = (90 - lat1)*(pi/180);
theta = (360 - long1)*(pi/180);
x1 = rho*sin(phi)*cos(theta);
y1 = rho*sin(phi)*sin(theta);
z1 = rho*cos(phi);
phi = (90 - lat2)*(pi/180);
theta = (360 - long2)*(pi/180);
x2 = rho*sin(phi)*cos(theta);
y2 = rho*sin(phi)*sin(theta);
z2 = rho*cos(phi);
%
% Compute the angle between vectors.
dot = x1*x2 + y1*y2 + z1*z2;
dist1 = sqrt(x1*x1 + y1*y1 + z1*z1);
dist2 = sqrt(x2*x2 + y2*y2 + z2*z2);
gamma = acos(dot/(dist1*dist2));
%
% Compute and print the great circle distance.
display('Great Circle Distance in miles:');
fprintf('%8.0f \n',gamma*rho)
%--------------------------------------------------------------
```

5. Testing

We start testing with the hand example, which gives the following interaction:

```
Enter latitude north for point 1: 40.75
Enter longitude west for point 1: 74
```

```
Enter latitude north for point 2: 51.5
Enter longitude west for point 2: 0
Great Circle Distance in miles:
   3466
```

This matches our hand example. Try this with some other locations, but remember that you need to choose points that are in the northern latitude and western longitude. The equations need to be slightly modified for points in other parts of the world.

3.4 DATA ANALYSIS FUNCTIONS

Analyzing data is an important part of evaluating test results. MATLAB contains a number of functions that make it easier to evaluate and analyze data. We first present a number of simple analysis functions, and then functions that compute more complicated measures or metrics related to a data set.

3.4.1 Simple Analysis

The following groups of functions are frequently used in evaluating a set of test data: maximum and minimum, mean and median, sums and products, and sorting. We now cover each group separately.

The **max** and **min** functions can be used in a number of ways to determine not only the maximum and minimum values, but also their locations in a matrix. Here are a set of examples to illustrate the various ways to use these functions.

max(x) Returns the largest value in the vector **x**.

```
x = [1,5,3];
max(x)
ans =
     5
```

max(x) Returns a row vector containing the maximum value from each column of the matrix **x**.

```
x = [1,5,3;
     2,4,6];
max(x)
ans =
     2   5   6
```

[a,b] = max(x) Returns a vector containing the largest value in a vector **x** and its location in the vector **x**.

```
x = [1,5,3];
[a,b] = max(x)
a =
   5
b =
   2
```

[a,b] = max(x) Returns a row vector **a** containing the maximum element from each column of the matrix **x**, and returns a row vector **b**