Performing Calculations

This Live Script provides an example of how calculations are performed in MATLAB using scalars, mathematical operators, and mathematical functions.

Table of Contents

Calculate the distance between Chicago and Boston

The Haversine formula is implemented in the following sections to calculate the distance between Boston and Chicago as an example of performing mathematical computations in MATLAB.

Introduction

The distance between two locations on the Earth's surface at sea level can be estimated using the Haversine formula. The formula uses locations specified by two latitude and longitude pairs to compute their *great circle distance*

$$d = 2R \arcsin(\sqrt{h}),$$

where R is the radius of the Earth, and

$$h = \sin^2\left(\frac{lat_2 - lat_1}{2}\right) + \cos(lat_2) \cdot \cos(lat_1) \cdot \sin^2\left(\frac{lon_2 - lon_1}{2}\right).$$

Note that the great circle distance is only an estimate because Earth is not a perfect sphere and elevations between two locations may vary from sea level.

Create latitude and longitude variables

The latitude and longitude of Chicago and Boston are given below:

- Chicago: (41.8781, -87.6298)
- Boston: (42.3601, -71.0589)

Run the code in this section to create the corresponding latitude and longitude variables.

```
lat1 = 41.8781;
```

```
lon1 = -87.6298;
lat2 = 42.3601;
lon2 = -71.0589;
```

Calculate the intermediate quantity, *h*

While it's possible to implement the entire Haversine formula in one line of MATLAB code, it's often easier to break complex calculations into two or more simpler steps. In this example, it's natural to first compute the quantity *h* before computing the final distance.

Mathematical operators

The formula for *h* given in the introduction involves both basic mathematical operations and the use of mathematical functions. The equivalent mathematical operators in MATLAB are given by:

- Addition: +
- Subtraction: -
- Multiplication: *
- Division: \
- Exponentiation: ^

Mathematical functions

MATLAB also provides equivalents for many mathematical functions including:

- Trigonometric and inverse trigonometric functions: `sin, cos, tan, sind, asin, ...`
- Exponential and logarithmic functions: `exp, log, log2, log10, pow2, ...`
- Hyperbolic functions: `sinh, cosh, tanh, asinh, acosh, ...`
- Specialized mathematical functions: `gamma, airy, besselj, besselk, erf, ...`

See the documentation for a list of mathematical functions and operators. Additionally, just like with mathematical expressions, parentheses can also be used to group operations both for clarity and to ensure operations are carried out in the desired order.

Calculate *h*

The value of *h* is calculated below. First compare the MATLAB code to the mathematical formula, then run the code to compute this value and store it in the variable h.

```
h = sind((lat2-lat1)/2)^2 + cosd(lat2)*cosd(lat1)*sind((lon2-lon1)/2)^2
```

Note that the degree-based trigonometric functions are used above, as the latitude and longitude values are given in degrees.

Calculate the total distance, *d*

Now that the value of *h* has been computed, the next step is to compute *d*. Enter the MATLAB code corresponding to the formula for *d*, and assign the result to a new variable, d. The value of *R* has been provided for you. You may find the `asin` and `sqrt` functions helpful. Then, run your code and view the output. The value of d should be approximately 1,365.6 *km*.

```
% Mean radius of Earth
R = 6371; %km
% Enter your formula for d below:
```

Calculate the distance between two other cities

By using *variables* instead of numbers when computing h and d, you can easily reuse the code with different values of latitude and longitude. Add the code to compute h and d beneath the edit boxes below. Then enter the latitude and longitude of two new locations and run this section to compute their distance:

```
lat1 = 0;
lon1 = 180;
lat2 = 90;
lon2 = 180;

% Add the code to compute h and d below:
```

More practice with calculations

Now that you have seen how to perform calculations using mathematical operators and functions, use the examples below to practice further.

Calculate the circumference of the Earth

The circumference of the Earth can be computed using the formula for the circumference of a circle: $C = 2\pi R$. Use the existing variable R to calculate C below. Use the MATLAB command pi in your code (so don't have to enter a numeric value).

```
% Compute the circumference of Earth:
```

Calculate the volume of the Earth

The volume of a sphere is given by the formula $V = \frac{4}{3}\pi R^3$. Add code below to calculate the volume of the Earth.

```
% Compute the volume of the Earth:
```

Compute the 'straight-line' distance for two points on the surface

Earlier you computed the great circle distance between two points along the earth's surface using the Haversine formula. The (shorter) distance, *D*, along the line connecting the points passing through the Earth's surface can be expressed in terms of the quantity *h* given earlier as:

$$D = R\sqrt{2(1 - \cos(\theta))},$$

where

3

$$\theta = 2\arcsin\left(\sqrt{h}\right).$$

Note that the quantity θ is the 'great circle angle' between the two points (in radians). Add the MATLAB code to compute h below, then add additional code to compute the variables `theta` and `D` and run this section to compute the straight-line distance between Boston and Chicago (or enter your own coordinates).

```
% Latude and longitude
lat1 = 41.8781;
lon1 = -87.6298;
lat2 = 42.3601;
lon2 = -71.0589;
% Add code to calculate the quantities below:
```

Perform your own calculations

Now that you understand how MATLAB can be used to perform calculations, practice further with your own calculations.

```
% Your calculations below:
```