

Commands and Functions

<code>fzero</code>	a function function that accepts a function handle or function definition as input and finds the function zero point nearest a specified value
<code>function</code>	identifies an M-file as a function
<code>global</code>	defines a variable that can be used in multiple sections of code
<code>meshgrid</code>	maps two input vectors onto two two-dimensional matrices
<code>nargin</code>	determines the number of input arguments in a function
<code>nargout</code>	determines the number of output arguments from a function
<code>pathtool</code>	opens the interactive path tool
<code>varargin</code>	indicates that a variable number of arguments may be input to a function

KEY TERMS

anonymous	folding	in-line
argument	function	input argument
comments	function function	local variable
directory	function handle	M-file
file name	function name	nesting
folder	global variable	toolbox

PROBLEMS

Function M-Files

As you create functions in this section, be sure to comment them appropriately. Remember that, although many of these problems could be solved without a function, the objective of this chapter is to learn to write and use functions. Each of these functions (except for the anonymous functions) must be created in its own M-file and then called from the command window or a script M-file program.

- 6.1** As described in Example 6.2, metals are actually crystalline materials. Metal crystals are called grains. When the average grain size is small, the metal is strong; when it is large, the metal is weaker. Since every crystal in a particular sample of metal is a different size, it isn't obvious how we should describe the average crystal size. The American Society for Testing and Materials (ASTM) has developed the following correlation for standardizing grain-size measurements:

$$N = 2^{n-1}$$

The ASTM grain size (n) is determined by looking at a sample of a metal under a microscope at a magnification of $100 \times$ (100 power). The number of grains in a 1-square-inch area (actual dimensions of 0.01 in \times 0.01 in) is estimated (N) and used in the preceding equation to find the ASTM grain size.

- (a) Write a MATLAB[®] function called `num_grains` to find the number of grains in a 1-square-inch area (N) at $100 \times$ magnification when the ASTM grain size is known.

- (b) Use your function to find the number of grains for ASTM grain sizes $n = 10$ to 100 .
- (c) Create a plot of your results.

6.2 Perhaps the most famous equation in physics is

$$E = mc^2$$

which relates energy E to mass m . The speed of light in a vacuum, c , is the property that links the two together. The speed of light in a vacuum is 2.9979×10^8 m/s.

- (a) Create a function called `energy` to find the energy corresponding to a given mass in kilograms. Your result will be in joules, since $1 \text{ kg m}^2/\text{s}^2 = 1 \text{ J}$.
 - (b) Use your function to find the energy corresponding to masses from 1 kg to 10^6 kg. Use the `logspace` function (consult `help logspace`) to create an appropriate mass vector.
 - (c) Create a plot of your results. Try using different logarithmic plotting approaches (e.g., `semilogy`, `semilogx`, and `loglog`) to determine the best way to graph your results.
- 6.3** The future-value-of-money formula relates how much a current investment will be worth in the future, assuming a constant interest rate.

$$\text{FV} = \text{PV} \times (1 + I)^n$$

where

FV is the future value

PV is the present value or investment

I is the interest rate expressed as a fractional amount per compounding period—i.e., 5% is expressed as .05

n is the number of compounding periods.

- (a) Create a MATLAB[®] function called `future_value` with three inputs: the investment (present value), the interest rate expressed as a fraction, and the number of compounding periods.
 - (b) Use your function to determine the value of a \$1000 investment in 10 years, assuming the interest rate is 0.5% per month, and the interest is compounded monthly.
- 6.4** In freshman chemistry, the relationship between moles and mass is introduced:

$$n = \frac{m}{\text{MW}}$$

where

n = number of moles of a substance

m = mass of the substance

MW = molecular weight (molar mass) of the substance.

- (a) Create a function M-file called `nmol` that requires two vector inputs—the mass and molecular weight—and returns the corresponding number of moles. Because you are providing vector input, it will be necessary to use the `meshgrid` function in your calculations.

- (b) Test your function for the compounds shown in the following table, for masses from 1 to 10 g:

Compound	Molecular Weight (Molar Mass)
Benzene	78.115 g/mol
Ethyl alcohol	46.07 g/mol
Refrigerant R134a (tetrafluoroethane)	102.3 g/mol

Your result should be a 10×3 matrix.

- 6.5 By rearranging the preceding relationship between moles and mass, you can find the mass if you know the number of moles of a compound:

$$m = n \times MW$$

- (a) Create a function M-file called `mass` that requires two vector inputs—the number of moles and the molecular weight—and returns the corresponding mass. Because you are providing vector input, it will be necessary to use the `meshgrid` function in your calculations.
- (b) Test your function with the compounds listed in the previous problem, for values of n from 1 to 10.
- 6.6 The distance to the horizon increases as you climb a mountain (or a hill). The expression

$$d = \sqrt{2rh + h^2}$$

where

d = distance to the horizon

r = radius of the earth

h = height of the hill

can be used to calculate that distance. The distance depends on how high the hill is and on the radius of the earth (or another planetary body).

- (a) Create a function M-file called `distance` to find the distance to the horizon. Your function should accept two vector inputs—radius and height—and should return the distance to the horizon. Don't forget that you'll need to use `meshgrid` because your inputs are vectors.
- (b) Create a MATLAB[®] program that uses your distance function to find the distance in miles to the horizon, both on the earth and on Mars, for hills from 0 to 10,000 feet. Remember to use consistent units in your calculations. Note that
- Earth's diameter = 7926 miles
 - Mars' diameter = 4217 miles

Report your results in a table. Each column should represent a different planet, and each row a different hill height.

- 6.7 A rocket is launched vertically. At time $t = 0$, the rocket's engine shuts down. At that time, the rocket has reached an altitude of 500 m and is rising

at a velocity of 125 m/s. Gravity then takes over. The height of the rocket as a function of time is

$$h(t) = -\frac{9.8}{2} t^2 + 125t + 500 \text{ for } t > 0$$

- (a) Create a function called `height` that accepts time as an input and returns the height of the rocket. Use your function in your solutions to parts b and c.
- (b) Plot `height` versus time for times from 0 to 30 seconds. Use an increment of 0.5 second in your time vector.
- (c) Find the time when the rocket starts to fall back to the ground. (The `max` function will be helpful in this exercise.)

6.8 The distance a freely falling object travels is

$$x = \frac{1}{2} g t^2$$

where

g = acceleration due to gravity, 9.8 m/s²

t = time in seconds

x = distance traveled in meters.

If you have taken calculus, you know that we can find the velocity of the object by taking the derivative of the preceding equation. That is,

$$\frac{dx}{dt} = v = gt$$

We can find the acceleration by taking the derivative again:

$$\frac{dv}{dt} = a = g$$

- (a) Create a function called `free_fall` with a single input vector `t` that returns values for distance `x`, velocity `v`, and acceleration `g`.
 - (b) Test your function with a time vector that ranges from 0 to 20 seconds.
- 6.9** Create a function called `polygon` that draws a polygon with any number of sides. Your function should require a single input: the number of sides desired. It should not return any value to the command window but should draw the requested polygon in polar coordinates.

Creating Your Own Toolbox

- 6.10** This problem requires you to generate temperature-conversion tables. Use the following equations, which describe the relationships between temperatures in degrees Fahrenheit (T_F), degrees Celsius (T_C), kelvins (T_K), and degrees Rankine (T_R), respectively:

$$T_F = T_R - 459.67^\circ\text{R}$$

$$T_F = \frac{9}{5} T_C + 32^\circ\text{F}$$

$$T_R = \frac{9}{5} T_K$$

You will need to rearrange these expressions to solve some of the problems.

- Create a function called `F_to_K` that converts temperatures in Fahrenheit to Kelvin. Use your function to generate a conversion table for values from 0°F to 200°F.
- Create a function called `C_to_R` that converts temperatures in Celsius to Rankine. Use your function to generate a conversion table from 0°C to 100°C. Print 25 lines in the table. (Use the `linspace` function to create your input vector.)
- Create a function called `C_to_F` that converts temperatures in Celsius to Fahrenheit. Use your function to generate a conversion table from 0°C to 100°C. Choose an appropriate spacing.
- Group your functions into a folder (directory) called `my_temp_conversions`. Adjust the MATLAB® search path so that it finds your folder. (Don't save any changes on a public computer!)

Anonymous Functions and Function Handles

- 6.11** Barometers have been used for almost 400 years to measure pressure changes in the atmosphere. The first known barometer was invented by Evangelista Torricelli (1608–1647), a student of Galileo during his final years in Florence, Italy. The height of a liquid in a barometer is directly proportional to the atmospheric pressure, or

$$P = \rho gh$$

where P is the pressure, ρ is the density of the barometer fluid, and h is the height of the liquid column. For mercury barometers, the density of the fluid is 13,560 kg/m³. On the surface of the earth, the acceleration due to gravity, g , is approximately 9.8 m/s². Thus, the only variable in the equation is the height of the fluid column, h , which should have the unit of meters.

- Create an anonymous function `P` that finds the pressure if the value of h is provided. The units of your answer will be

$$\frac{\text{kg m}}{\text{m}^3 \text{s}^2} \text{m} = \frac{\text{kg}}{\text{m s}^2} = \text{Pa}$$

- Create another anonymous function to convert pressure in Pa (Pascals) to pressure in atmospheres (atm). Call the function `Pa_to_atm`. Note that

$$1 \text{ atm} = 101,325 \text{ Pa}$$

- Use your anonymous functions to find the pressure for fluid heights from 0.5 m to 1.0 m of mercury.
- Save your anonymous functions as `.mat` files

- 6.12** The energy required to heat water at constant pressure is approximately equal to

$$E = mC_p \Delta T$$

where

- m = mass of the water, in grams
- C_p = heat capacity of water, 1 cal/g K
- ΔT = change in temperature, K.

- (a) Create an anonymous function called `heat` to find the energy required to heat 1 gram of water if the change in temperature is provided as the input.
- (b) Your result will be in calories:

$$\frac{\text{cal}}{\text{g K}} = \text{cal}$$

Joules are the unit of energy used most often in engineering. Create another anonymous function `cal_to_J` to convert your answer from part (a) into joules. (There are 4.2 J/cal.)

- (c) Save your anonymous functions as `.mat` files.

- 6.13.** (a) Create an anonymous function called `my_function`, equal to

$$-x^2 - 5x - 3 + e^x$$

- (b) Use the `fplot` function to create a plot from $x = -5$ to $x = +5$. Recall that the `fplot` function can accept a function handle as input.
- (c) Use the `fminbnd` function to find the minimum function value in this range. The `fminbnd` function is an example of a function function, since it requires a function or function handle as input. The syntax is

$$\text{fminbnd}(\text{function_handle}, \text{xmin}, \text{xmax})$$

Three inputs are required: the function handle, the minimum value of x , and the maximum value of x . The function searches between the minimum value of x and the maximum value of x for the point where the function value is a minimum.

- 6.14** In Problem 6.7, you created an M-file function called `height` to evaluate the height of a rocket as a function of time. The relationship between time, t , and height, $h(t)$, is:

$$h(t) = -\frac{9.8}{2}t^2 + 125t + 500 \text{ for } t > 0$$

- (a) Create a function handle to the `height` function called `height_handle`.
- (b) Use `height_handle` as input to the `fplot` function, and create a graph from 0 to 60 seconds.
- (c) Use the `fzero` function to find the time when the rocket hits the ground (i.e., when the function value is zero). The `fzero` function is an example of a function function, since it requires a function or function handle as input. The syntax is

$$\text{fzero}(\text{function_handle}, \text{x_guess})$$

The `fzero` function requires two inputs—a function handle and your guess as to the time value where the function is close to zero. You can select a reasonable `x_guess` value by inspecting the graph created in part (b).

Subfunctions

- 6.15** In Problem 6.10 you were asked to create and use three different temperature-conversion functions, based on the following conversion equations:

$$T_F = T_R - 459.67^\circ\text{R}$$

$$T_F = \frac{9}{5}T_C + 32^\circ\text{F}$$

$$T_R = \frac{9}{5}T_K$$

Recreate Problem 6.10 using nested subfunctions. The primary function should be called `temperature_conversions` and should include the subfunctions

```
F_to_K
C_to_R
C_to_F
```

Within the primary function use the subfunctions to:

- Generate a conversion table for values from 0°F to 200°F . Include a column for temperature in Fahrenheit and Kelvin.
- Generate a conversion table from 0°C to 100°C . Print 25 lines in the table. (Use the `linspace` function to create your input vector.) Your table should include a column for temperature in Celsius and Rankine.
- Generate a conversion table from 0°C to 100°C . Choose an appropriate spacing. Include a column for temperature in Celsius and Fahrenheit.

Recall that you will need to call your primary function from the command window or from a script M-file.