

23-Functions Part 2

text: Chapter 7.3, 7.11

ECEGR 101

Engineering Problem Solving with Matlab

Professor Henry Louie



Overview

- Example
- Variable Passing
- Local and Global Variables
- Nested Functions



Exercise

Write a function, called `parallelR`, that computes the equivalent resistance as follows:

$$r_{eq} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_4} \dots \frac{1}{R_n}}$$

The input argument is a vector `R`.

Run the function for:

$$R = [10 \ 10 \ 10 \ 10]$$

$$R = [3 \ 6]$$

$$R = [1 \ 10 \ 100]$$



Exercise

```
function [req] = parallelR(r)
% parallelR computes the equivalent parallel resistance
% Function parallelR calculates the equivalent
% resistance for a set of n parallelly connected resistors.
% Input:
% r - vector of n resistances
% Output:
% req - equivalent resistance

req = 1./sum(1./r);
```



Exercise

```
>> parallelR([10 10 10 10])
ans =
    2.5000
>>
>> parallelR([3 6])
ans =
    2
>>
>> parallelR([1 10 100])
ans =
    0.9009
>>
```



Variable Passing in MATLAB

- MATLAB programs communicate with their function using a **pass-by-value** scheme.
 - When a function call occurs, MATLAB makes a copy of the actual arguments and passes them to the function.
 - Even if the function modifies the input arguments, it won't affect the original data in the calling program.



Exercise

- A function is defined as shown on the right.

What are the values of x and z after the following commands are executed in the Command Window?

```
Editor - C:\Documents and Settings\amigu...
File Edit Text Cell Tools Debug Desktop Win...
[New] [Open] [Save] | [Cut] [Copy] [Paste] [Undo] [Redo] | [Print] [Help] [f]
1 function y = exampleFunction(x)
2 - x = abs(x);
3 - x = sqrt(x) + 10;
4 - y = x;
```

Command Window

```
>> x = 4;
>> z = exampleFunction(x)
```



Exercise

- A function is defined as shown on the right.

What are the values of x and z after the following commands are executed in the Command Window?

```
Editor - C:\Documents and Settings\amigu
File Edit Text Cell Tools Debug Desktop Win
[Icons]
1 function y = exampleFunction(x)
2 - x = abs(x);
3 - x = sqrt(x) + 10;
4 - y = x;
```

Command Window

```
>> x = 4;
>> z = exampleFunction(x)
```

```
z =
    12
>>
>> x
x =
     4
```




Local Variables

- All variables in a function file are **local variables**.
- All variables in a Command Window are local variables with respect to functions.
 - The function body commands cannot use them unless they are passed as input arguments.



Example

```
Editor - C:\Documents and Settings\amigue
File Edit Text Cell Tools Debug Desktop Wind
[Icons: New, Open, Save, Cut, Copy, Paste, Undo, Redo, Print, Find]
1 function x = randomInt()
2 % Output a random 8-bit integer.
3
4 - d = int8(100*rand);
5 - x = int8(10*rand);
```

```
Command Window
>> a = randomInt
a =
     9
>>
>> d
??? Undefined function or variable 'd'.
```



Local Variables

- How to make variable `d` available outside of the function?

```
Editor - C:\Documents and Settings\amigu...
File Edit Text Cell Tools Debug Desktop Wind
[Icons: File Explorer, Save, Cut, Copy, Paste, Undo, Redo, Print, Find]
1 function [x, d] = randomInt()
2 % Output a random 8-bit integer.
3
4 - d = int8(100*rand);
5 - x = int8(10*rand);
6
```

```
Command Window
>> [a,b] = randomInt
a =
    7
b =
   97
>>
```



Local Variables

```
Editor - C:\Documents and Settings\amiguelM
File Edit Text Cell Tools Debug Desktop Window
[Icons: New, Open, Save, Cut, Copy, Paste, Undo, Redo, Print, Find, Help]
1 function y = localVariableExample(x)
2
3 - y = x^2 + d;
4
```

```
Command Window
>> d = 10;
>>
>> y = localVariableExample(4)
??? Undefined function or variable 'd'.

Error in ==> localVariableExample at 3
y = x^2 + d;
```



Local Variables

- How to make variable `d` available to the function?

```
Editor - C:\Documents and Settings\amiguel\My
File Edit Text Cell Tools Debug Desktop Window
[Icons: New, Open, Save, Cut, Copy, Paste, Undo, Redo, Print, Find, Help]
1 function y = localVariableExample(x,d)
2
3 - y = x^2 + d;
4
```

```
Command Window
>> d = 10;
>>
>> y = localVariableExample(4, d)
y =
    26
>>
```



Exercise

Write a function, named `circuit1`, that takes as input arguments the following:

- a vector R (R has the elements $R_1 R_2 R_3 \dots R_n$)
- a voltage V

and has as output arguments current I and power P to be computed as follows:

$$I = \frac{V}{R_{EQ}}, \quad P = VI \quad \text{where} \quad R_{EQ} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \dots + \frac{1}{R_n}}$$

Use the function you created in previous exercise to compute R_{EQ} .



Exercise

```
function [I, P] = computeCurrentPower(R, V)
% Calculates I and P for a parallel circuit.
% Function computeCurrentPower calculates the current
% and power in a parallel circuit.
% Inputs:
% R - vector of resistor values
% V - voltage
% Output:
% I - current
% P - power

Req = parallelR(R);
I = V/Req;
P = V*I;
```



Exercise

```
>> [I, P] = computeCurrentPower([10 10 10], 100)
```

```
I =  
    30.0000
```

```
P =  
    3.0000e+003
```

```
>>
```

```
>> [I, P] = computeCurrentPower([20 30 10], 5)
```

```
I =  
    0.9167
```

```
P =  
    4.5833
```

```
>>
```




Global Variables

- Variables can be declared as global variables so they can be recognized by different functions and in the workspace.
- Syntax: `global variable_name`
- The variable has to be declared global in every function file that wants to access it.



Example

```
Editor - C:\Documents and Settings\amiguelM
File Edit Text Cell Tools Debug Desktop Window
[Icons: New, Open, Save, Cut, Copy, Paste, Undo, Redo, Print, Find, Help]
1 function y = localVariableExample(x)
2
3 - global d;
4
5 - y = x^2 + d;
6
```

```
Command Window
>> global d;
>> d = 10;
>>
>> y = localVariableExample(10)
y =
    110
>>
```



Global Variables

- Global variables are in general considered as **bad programming!**
- **Global variables**
 - Make program difficult to read:
 - It is often impossible to track where the variable was originally defined.
 - They interfere with the modular structure of the program.
 - Create naming conflicts.
 - Waste memory.
 - Can create difficult-to-trace bugs.



Global Variables

- Global variables are useful for sharing very large volumes of data among many functions, because the entire data set does not have to be copied each time a function is called.



Example

The distance between two points (x_1, y_1) and (x_2, y_2) on a Cartesian coordinate plane is given by the equation

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Write a function to calculate the distance between any two points (x_1, y_1) and (x_2, y_2) specified by the user. Use `fprintf` to display the coordinates entered and the distance between them on the screen.



Example

```
Editor - C:\Documents and Settings\amiguel\My Documents\_ECEGR 101 FALL 2006\LECTURES\calcDistance.m
File Edit Text Cell Tools Debug Desktop Window Help
[Icons] Stack: Base
1 function d = calcDistance()
2 % Calculates the distance between two points
3
4 % Ask the user for input
5 - clc
6 - x1 = input('Enter x1 coordinate value:');
7 - y1 = input('Enter y1 coordinate value:');
8 - fprintf('\n');
9 - x2 = input('Enter x2 coordinate value:');
10 - y2 = input('Enter y2 coordinate value:');
11
12 % Calculate the distance
13 - d = sqrt( (x1-x2)^2 + (y1-y2)^2 );
14
15 % Display the result
16 - fprintf('\nThe distance between points (%5.2f, %5.2f) and (%5.2f, %5.2f) is %5.2f\n\n', x1, y1, x2, y2, d);
17
```



Example

Command Window

```
>> d1 = calcDistance
```

Command Window

```
Enter x1 coordinate value:2
```

```
Enter y1 coordinate value:4
```

```
Enter x2 coordinate value:6
```

```
Enter y2 coordinate value:7
```

```
The distance between points ( 2.00, 4.00) and ( 6.00, 7.00) is 5.00
```

```
d1 =
```

```
5
```



Example

Write a function that calculates both the *harmonic mean* and *geometric mean* for a set of numbers.

$$\text{harmonic mean} = \frac{N}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_N}}$$

$$\text{geometric mean} = \sqrt[N]{x_1 x_2 x_3 \dots x_N}$$



Example

- We will solve this problem using **nested functions**:
 - Function 1: calculate the harmonic mean.
 - Function 2: calculate the geometric mean.
 - Main function: calls Function 1 and Function 2 and outputs the geometric and harmonic means.



Function 1

```
Editor - C:\Documents and Settings\amiguel\My Documents\_ECEGR
File Edit Text Cell Tools Debug Desktop Window Help
[Icons: New, Open, Save, Cut, Copy, Paste, Undo, Redo, Print, Run, Find, Help, Close, Save All, Print All, Close All, Save All and Close]
1 function [harmonicMean] = computeHarmonicMean(x)
2 % Calculates the harmonic mean for a set of numbers
3 %
4 % Input:
5 % x - row or column vector of numbers
6 %
7 % Output:
8 % harmonicMean - the harmonic mean of all values in x
9
10 - s = sum(1./x);
11 - harmonicMean = length(x)/s;
12
```

$$\text{harmonimean} = \frac{N}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_N}}$$



Function 2

```
Editor - C:\Documents and Settings\amiguel\My Documents\_ECEGR
File Edit Text Cell Tools Debug Desktop Window Help
[Icons]
1 function [geometricMean] = computeGeometricMean(x)
2 % Calculates the geometric mean for a set of numbers
3 %
4 % Input:
5 % x - row or column vector of numbers
6 %
7 % Output:
8 % geometricMean - the geometric mean of all values in x
9
10 - p = prod(x);
11 - N = length(x);
12 - geometricMean = p^(1/N);
13
```

$$\text{geometricmean} = \sqrt[N]{x_1 x_2 x_3 \dots x_N}$$



Main Function

```
Editor - C:\Documents and Settings\amiguel\My Documents\_ECEGR 101 FALL 20
File Edit Text Cell Tools Debug Desktop Window Help
[Icons: New, Open, Save, Cut, Copy, Paste, Undo, Redo, Print, Find, Run, Stop, Refresh, Save All, Print All, Stack: B]
1 function [hM, gM] = computeGHMean(x)
2 % Calculates the geometric and harmonic mean for a set of numbers
3 %
4 % Input:
5 % x - row or column vector of numbers
6 %
7 % Output:
8 % gM - the geometric mean of all values in x
9 % hM - the harmonic mean of all values in x
10
11 - hM = computeGeometricMean(x);
12 - gM = computeHarmonicMean(x);
13
```

Functions can call other functions



Example

Command Window

```
>> a = 1:10;  
>> [h, g] = computeGHMean(a)  
h =  
    4.5287  
g =  
    3.4142  
>>  
>> [h, g] = computeGHMean([1 4 5 6 2 9 10])  
h =  
    4.1611  
g =  
    3.0072  
>>
```



Exercise

- Write a function that increments by one the elements of an array that have an odd index (the first, third, fifth, etc. elements).

Example:

Input: $x = [1 \ 3 \ 5 \ 8 \ 10 \ -1 \ 8]$

Output: $y = [2 \ 3 \ 6 \ 8 \ 11 \ -1 \ 9]$



Exercise

```
Editor - C:\Documents and Settings\amiguel\My Documents\_ECEGR 101 FALL 2006\LE
File Edit Text Cell Tools Debug Desktop Window Help
[Icons: New, Open, Save, Cut, Copy, Paste, Undo, Redo, Print, Find, Run, Stop, Run and Debug, Run and Profile, Run and Coverage, Run and Memory, Run and Performance, Run and Profiling, Run and Coverage and Memory, Run and Coverage and Performance, Run and Coverage and Profiling, Run and Coverage and Memory and Performance, Run and Coverage and Memory and Profiling, Run and Coverage and Memory and Performance and Profiling] Stack: Base
1 function y = incrementOdd(x)
2 % incrementOdd adds one to the elements of an array with an odd index
3 %
4 % Input: x - vector or row array
5 % Output: y - array with odd-indexed elements incremented by one
6
7 - x(1:2:length(x)) = x(1:2:length(x)) + 1;
8 - y = x;
9
```



Exercise

Command Window

```
>> x = [1 3 5 8 10 -1 8]
x =
     1     3     5     8    10    -1     8
>>
>> y = incrementOdd(x)
y =
     2     3     6     8    11    -1     9
>>
>> incrementOdd([1 1 1 1 1 1])
ans =
     2     1     2     1     2     1
>>
```