Topics: User-defined function

Reading (ML): Sec 5.1, 5.2, 5.3 (exclude warning, inputname), 5.7 subfunctions
Optional Reading: other parts in Sec 5.3, Sec 5.4–5.6, Sec 5.7 private functions

General form of user-defined function

function [outarg1, outarg2, ...] = fname(inarg1, inarg2, ...)
% H1 comment line
% Other comment lines
executable code

Example: find prime numbers (again!)

Script file savePrime.m:

% Save prime numbers in [2,n] to vector prime

n= input('Enter number: ');
prime= 2; % vector to store prime #s
k= 3; % next number to be checked
while (k<=n)
    % check number k, save if prime

    % go to next number
    k= k+1;
end
prime

Function file isPrime.m. How to write the function header?

% Determine if n is prime, n>=2
% out <-- n if n is prime
% out <-- [] if n is composite

divisor= 2;
while ( mod(n,divisor) ~= 0 )
    divisor= divisor + 1;
end
if ( divisor==n )

else

end

Be sure you understand the example on p. 194 in Chapman.
User-Defined Function

- Can easily “reuse” code
- Functions can be independently tested
- Upon invocation, each function has its own memory space inaccessible by other functions or the command window space—variables in a function can be “seen” only inside the function
- Values stored in variables are not preserved between function calls.
- Arguments are “passed by value”

Subfunctions

- more than one function in an M-file
- top function is normal
- remaining functions are subfunctions, accessible only by top function

Global Memory

- Global memory can be accessed from any workspace
- Global variable must be declared to be global before it is used for the first time in a function.

  \[
  \text{global \ variable1 variable2 ...}
  \]

Persistent Memory

Persistent memory can be accessed from within the function only and is preserved unchanged between calls to the function.

  \[
  \text{persistent \ variable1 variable2 ...}
  \]