



AE1106: Programming I

Session 4: Scripts, Functions, and Simple Plotting

3-7-2018



Topics

- Scripts
- Functions

Scripts versus functions

Script file

- File name: *name_s.m*
- List of Matlab commands
- All variables in workspace can be used inside script file
- All variables used in script file are available in command window (global)

Function file

- File name: *name_f.m*
- First line: *function [o]=name_f(i)*
with o: all output variables
and i: all input variables
- List of Matlab commands
- Inputs only can be used inside script file
- Only outputs of function file are available in command window; all other variables not (local)

Function files

- Other names:
 - Function (C)
 - Subroutine (Fortran)
 - Procedure (Pascal)
 - Class (Java)
- Well defined part of a larger program
- Advantages
 - Modular approach (improved code development)
 - Can be checked separately (so, improved code debugging)
 - Future application is possible

Rudra Pratap
Getting Started with Matlab
page 108 (2010 version):

- Pseudo-code
- Readability
- Modularity
- Robustness
- Expandability

Function files; example

separate file

Communication with main program:

- Input and output variables (other names can be used as well)

All other variables in m-file are local (i.e. 'unknown' in main program)

Application (command window, other M-file)

```
>> p0=101.3e3
>> v=70
>> pd=dynpres(v)
>> ptot=p0+pd
>> rho
```

```
function pdyn=dynpres(u)
% syntax: function pdyn=dynpres(u)
%
% Calculation of dynamic pressure
% Input:
%   u: speed (m/s)
% Output:
%   pdyn: dynamic pressure (Pa)
% density of air (kg/m^3)
rho=1.225;
% dynamic pressure (Pa)
pdyn=1/2*rho*u^2;
```

Pitfalls

- name m-file ↔ name variable
name m-file ↔ name built-in routine
use *exist* or *which*
- `[o1,o2,o3]=example(i1)`, all outputs are specified
`o=example(i1)` returns first output *o1* only
- specific m-file should be located in 'current directory'
or
be located in one of the directories specified in *path*