ChE-101
H-1: Computer Programming Guidelines and Quality Matlab Code

(last updated 28 March 2005)

This handout contains important information that should be included in each Matlab file as well as an example of a quality Matlab Code.

1. Computer Programming Guidelines

Below are a number of generic rules that apply to writing computer programs. Some of them will apply to all programs or program files, and others are more specific.

- Every file you create or edit should contain your name, the date it was created or last edited and a brief description of what problem the file solves. Remember that many problems require several files. This guideline applies to all of them.

- Your output in the Command Window should contain your name, the date, course, and problem number. This rule and the previous rule can be accomplished with a single set of commands.

- Every file should also include a comment referring to any other files required by the file. This does not include built-in MATLAB files, but does include any function files or script files referred to or from the current file.

- Each program should include sufficient comments that the major steps can be followed by a user with a knowledge level equal to your own. This doesn't mean each line of code must be commented. Blocks of code that perform some operation can be commented together. This requirement does include the need for the units on numerical values in the code.

- Effective use of variable names can minimize the need for comments. Variable names that are descriptive should be used whenever possible.

- Use spaces and indents to improve the readability of your code.

- When input is requested from the user, the units need to be included in the request on the screen, and the format of the input must be specified.

- Output should be formatted as to be easily readable. Use tabular data when appropriate. Include units and report values to the proper number of significant figures.

- The code should be written such that it can be used for a modified problem with as few changes as possible. The primary method of doing this is to size loops by using the 'length' command.

- The code must be written as to prevent the possibility of getting stuck in an infinite loop. This is usually done by putting a maximum iteration number and an if-break structure in any "while" loop.

- Any input or output values should be checked whether they are outside the feasible range.

- Your code should always be checked with a test case, either given to you by the instructor or created by you.

- It should always be clear when viewing the program output why the program stop executing. Was the convergence criteria met, or was a maximum number of steps exceeded?

- When checking for program convergence, unless it is obvious what the magnitude of the answer is going to be, normalized convergence criteria should be used. Also be sure to use the absolute value when checking convergence.

- The rules of proper graphing (labels, symbol types, titla, legend, etc) should be followed when creating a graph in Matlab.

2. Example of a Quality Matlab Code and Output

```
ChE 101
Fall 2004

Matlab HW #5  _  Answer Key
___________________________________________________________

Here is the main file for Problem 5
___________________________________________________________

clear

%  Darin Ridgway
%  ChE 101
%  Fall 2004

%  Matlab HW 5

%  This problem calculates the material balances on a process
%  for the combustion of methanol with excess oxygen for feed
%  flowrates of methanol from 100 to 700 mol/h.

%  This file calls the function file mat_hw5f_F04.m.  It must be available.


%  Set the values of the methanol feed in mol/h
F = [100:50:700];

%  Set the initial guess and the options array
z0 = [ 1000  1000  1000  0.5 ];
options = optimset('Largescale','off');

for i = 1:length(F)

   %  Call fsolve
   y = fsolve(@mat_hw5f_F04, z0, options, F(i));

   %  Store the answers
   N1(i) = y(1);     %  N1 = the flowrate of oxygen in mol/h
   N2(i) = y(2);     %  N2 = the flowrate of dry gas out in mol/h
   N3(i) = y(3);     %  N3 = the flowrate of water out in mol/h
   x(i) = y(4);      %  x = the mole fraction of nitrogen in the outlet
```

```
end

%  Create the output
clc
disp('                    ')
disp('                    ')
disp('  Darin Ridgway     ')
disp('  ChE 101           ')
disp('  Fall 2004         ')
disp('                    ')
disp('  Matlab HW 5       ')
disp('                    ')
disp('   This problem calculates the material balances on a process  ')
disp('   for the combustion of methanol with excess oxygen for feed  ')
disp('   flowrates of methanol from 100 to 700 mol/h.                ')


fprintf('\n\n')
fprintf('   methanol fed  oxygen fed   dry gas out   water out   N2 out
\n')
fprintf('     (mol/h)      (mol/h)       (mol/h)      (mol/h)   (mol frac)
\n\n')

for i = 1:length(F)
    fprintf('     %7.1f     %7.1f       %7.1f       %7.1f     %4.2f  \n',...
        F(i), N1(i), N2(i), N3(i), x(i))
end

fprintf('\n\n\n')
```

Here is the function file for Problem 5

```
%  Darin Ridgway
%  ChE 101
%  Fall 2004, 2004

%  Matlab Homework 5

%  This file is called by fsolve to solve the probelm in main file
mat_hw5_F04.m

%  The equations represent mass balances on a methanol combustion process

%  F = the feed rate of methanol (mol/h)
%  z(1) = the feed flowrate of oxygen (mol/h)
%  z(2) = the product flowrate of dry gas (mol/h)
%  z(3) = the product flowrate of water (mol/h)
%  z(4) = the mole fraction of nitrogen in the product gas

function fz = mat_hw5f_F04(z, F)

fz(1) = 0.1034*z(2) _ F;
fz(2) = 0.0336*z(2) + 2*z(3) _ F*4;
fz(3) = z(4)*z(2) _ 3.76*z(1);
fz(4) = z(2)*(0.1804 + 2*(0.897 _ z(4))) + z(3) _ 2*z(1) _ F;
```

Here is the response in the Command Window for Problem 5

Darin Ridgway
ChE 101
Fall 2004

Matlab HW 5

This problem calculates the material balances on a process for the combustion of methanol with excess oxygen for feed flowrates of methanol from 100 to 700 mol/h.

methanol fed (mol/h)	oxygen fed (mol/h)	dry gas out (mol/h)	water out (mol/h)	N2 out (mol frac)
100.0	209.4	967.1	183.8	0.81
150.0	314.1	1450.7	275.6	0.81
200.0	418.7	1934.2	367.5	0.81
250.0	523.4	2417.8	459.4	0.81
300.0	628.1	2901.4	551.3	0.81
350.0	732.8	3384.9	643.1	0.81
400.0	837.5	3868.5	735.0	0.81
450.0	942.2	4352.0	826.9	0.81
500.0	1046.9	4835.6	918.8	0.81
550.0	1151.6	5319.1	1010.6	0.81
600.0	1256.2	5802.7	1102.5	0.81
650.0	1360.9	6286.3	1194.4	0.81
700.0	1465.6	6769.8	1286.3	0.81