MATH 373 Introduction to Numerical Analysis

This document provides a template for both written reports and m-files required for each homework assignment. Written homework will be submitted in the following format, beginning on the next page. Feel free to copy and paste the following LaTeX code into your own file to use as a template for your homework.

Use the proper formatting available in LaTeX. Math should be in the math environment, e.g. $x - 3 = 5$, or

$$I = \int_a^b f(x)\,dx,$$

and computer variables and command line text should be in either the `verbatim` environment or typeset using `texttt`.

Formatting for m-files can be found after the formatting for the homework reports.

MATH 373-01 (or -02, or -03)
Brent Deschamp
Homework #1
January 1, 2017

---

1 Problem Statement

Problem statement. If the problem is written out in the LaTeX file, then feel free to copy the problem statement from the LaTeX file for the homework. If the problem is out of the book, then typeset the problem in your report.

2 Solution

This may include numerical values, functions, figures, derivations, etc. Be sure to use complete sentences and units when reporting answers. If the problem includes multiple parts, use the `enumerate` environment. Figures and code references should be linked using the `label` and `ref` system. Derivations should use the `align*` environment. Any math should use the math environment, including the names of variables. Check your spelling and grammar.

Also, make sure that you provide some justification for why you think the answers you are reporting are correct.

3 Command-line Usage

This section describes how you solved the problem. This will reference to any any m-files you wrote, as well as what was executed on the command line. Use either the `texttt` or `verbatim` environments to report command-line usage.

4 Code

Include any code you used to obtain your solution. Use the `listings` package and the `lstlisting` environment to report your code. Include the line

```
\lstset{language=Matlab}
```

in the preamble of your LaTeX file so that the code is formatted as MATLAB code.

Provide a description of what programs are included in this section using the `label` and `ref` commands. Each program should be placed in its own `listings` environment, which should usethe `caption` and `frame` options. The easiest way to include each program is with the `lstinputlisting` command.

The following provides guidance on formatting m-files. Note the included example using the `lstlisting` environment, which can be found in Listing ??.

M-files should be formatted using the template shown below.

- Each m-file should have a header, and all parts of the code should be properly documented.
- Be sure to include any input checks and any default values used for inputs. Always include `isempty` along with checking the number of inputs.
- Test cases should include inputs, as well as the expected output.
- Make sure all error messages state not only the problem but the required solution.
- Always check for division by 0.
- No foul language should appear in any part of an m-file. If it is found, a 0 will be assigned for the entire assignment.

```
function [outputs] = function_name(inputs)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name
% Date
% Function description

% Description of inputs, including units

% Description of outputs, including units
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Any test cases you've used to test your code.
% These are here mainly for convenience so that you don't need to remember what you
% were testing from session to session.  Be sure to include not only the inputs but
% the expected outputs.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Input testing and default values

%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Main body code, properly documented

end

% While the final end is not required by MATLAB,
% I have found that including it does help with debugging.
```

Listing 1: MATLAB code for plotting Kepler's equation.

```
function Kepler_plot()
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Brent Deschamp
% August 23, 2016
% Plot Kepler's Equation: E-e*sin(E) = M

% No inputs

% No outputs.  Plots are automatically generated
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Constants
e = 0.048;  % eccentricity for Jupiter
M = pi/4;   % mean anomaly

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Create values for E
E_vals = linspace(0,2*pi);
E_vals_finer = linspace(0,1);
E_vals_finer_still = linspace(0.7,0.9);

% Define function
Kepler = E_vals - e*sin(E_vals) - M;
Kepler_finer = E_vals_finer - e*sin(E_vals_finer) - M;
Kepler_finer_still = E_vals_finer_still - ...
        e*sin(E_vals_finer_still) - M;

% Plot the various functions
subplot(3,1,1);
plot(E_vals,Kepler);
grid
title(['Plotting Kepler''s Equation with e=' num2str(e)]);

subplot(3,1,2)
plot(E_vals_finer,Kepler_finer);
grid

subplot(3,1,3)
plot(E_vals_finer_still,Kepler_finer_still);
grid
```