October 20, 2015

Tools and Techniques of Computational Science - Fall 2015

Assignment 1

This homework is due by 11:59 PM on 10/26/2015. You are to submit a gzipped tarball of all your bash scripts and plots to Canvas. For this exercise, all plotting must be done using Gnuplot.

Exercise 1.0: Bash Scripting

1. Write a shell script that will add two integers supplied as command line arguments. If two arguments are not supplied, print the script's usage and exit with an error.

2. Write a shell script to determine the largest of three integers provided as command line arguments. Print error and usage information if sufficient arguments are not supplied.

3. Write a shell script that uses a while loop and expr to calculate the factorial of a supplied number.

4. Write a script that uses a for loop to generate the following output:

```
    1
   2 2
  3 3 3
 4 4 4 4
5 5 5 5 5
    .
   . .
  . . .
 . . . .
. . . . .
```

5. Write a script to convert a number from decimal to hexadecimal.

6. Write a shell script to convert Temperature from Celsius to Fahrenheit or vice versa. This script should be able to parse command line arguments or be used interactively.

7. Write a script to determine how many files and directories (including hidden files and directories) are in your home directory. How many of the files are executable? How large is each directory? Use a for loop and file tests to construct your solution.

Exercise 1.1: Numerical Bash Script

Write a bash shell script which uses numerical integration to evaluate an analytic function. For the numerical integration, use a simple composite mid-point rule to perform the integration. For a fixed interval, the midpoint approximation is as follows:

$$\int_{x_{min}}^{x_{max}} f(x)\,dx \approx h\,f(x_{min} + \frac{h}{2}) \tag{1}$$

where $h = x_{max} - x_{min}$. Note that h will be a constant in this assignment. The analytic function to integrate is:

$$f(x) = cos(x) \tag{2}$$

Your bash script should accept at least 3 command-line arguments that are used to control the limits of integration and the discretization interval. It should also have a "-h" option to show how to run the script. If no command-line arguments are provided, your script should show the help message and exit.

In addition to computing the numerical integration, your script should compute the error between your numerical solution and the exact analytic solution to the definite integral which is computed as:

$$\int_{x_{min}}^{x_{max}} cos(x)\,dx = sin(x_{max}) - sin(x_{min}) \tag{3}$$

Your bash script should approximate the definite integral and output the absolute error for a given NP, where NP is the number of discretization points between $[x_{min}, x_{max}]$ as:

$$Error = |Numerical - Exact| \tag{4}$$

Once your script is complete, run the following cases on $[x_{min}, x_{max}] = [0.0, 5.0]$ and time the runtime performance for each case:

- NP=10
- NP=50
- NP=100
- NP=200

Create two plots based on the results. The first plot should compare the wallclock runtime for the 4 cases. The second plot should show the error as a function of NP (on a logscale please).