In-class Programming Homework, Sept. 13

Due date: Thursday, Sept. 29 at midnight. You may work with up to one other person on the entirety of the assignment and turn in a joint program if you wish; in this case, each of you gets the same grade.

Introduction:

In this homework, you will write a C, C++ or a Fortran code with procedures defined in separate files. You will then use the code to analyze the dependence of the accuracy and error in your computation on the precision and approximation employed. These errors are caused by loss of significance when evaluating a function involving subtraction of nearly equal quantities. The files containing the procedures that you will use as a starting point are combined in "tar" files lab1_f.tar (Fortran), lab1_c.tar (C) and lab1_cc.tar. (C++). You might practice with the three programming languages if you wish, but please select only one for the submission of your assignment.

If you are using a computer with unix or linux shell, you can transfer these tar files to Agave by using the following commands:

```
sftp username@agave.asu.edu
sftp> cd APM512
(or the name of the working directory you created for the class)
sftp> put lab1_f.tar     (or lab1_c.tar or lab1_cc.tar)
sftp> quit
```

Login to Agave and extract the files, using the commands:

```
ssh username@agave.asu.edu
$ cd APM512 (or your working directory)
$ tar -xvf lab1_f.tar     (or lab1_c.tar or lab1_cc.tar)
```

The following folder should be created be created:
Fortran: LAB1_FOR: contains module_precisions.f, module_functions.f and main.f
C: LAB1_C: contains functions.c, head.h and main.c
C++: LAB1_CC: contains functions.cc, head.h and main.cc

To compile the code connect to a compute node:

```
$ interactive
```

Then and type the commands:

```
$ module load intel/2018x
$ ifort -FR -warn -o main module_precisions.f module_functions.f
              main.f (for Fortran)
$ icc -std=c99 -o main functions.c head.h main.c (for C)
$ icpc -o main functions.cc head.h main.cc (for C++)
$ ls
```

An executable file called "**main**" should be created. To run the file type the command

```
$ ./main
```

Assignment:

Consider the function

$$f(x,\omega) = \frac{\cos(\omega x) - \cos(\omega_o x)}{\omega^2 - \omega_o^2},$$

where $\omega_o = 0.16$ and $\omega = \omega_0 + \epsilon$ are parameters.

1) Write procedures and calling instructions to evaluate the function $f(x,\omega)$ with single and double precisions. The places to be modified are indicated in the program files.

2) Use your code to evaluate $f(\pi/8,\omega)$ with single, double and quadruple precisions for $\epsilon = 0.1$. Use scientific format and round the evaluations to 9 decimal digits (example: -7.70139683E-02).

3) Write instructions to evaluate the relative error in the computed values of $f(\pi/8,\omega)$ for single and double precisions with $\epsilon = 0.1$. Use the value evaluated at quadruple precision as your "true" value. Present you result in a scientific format and round the relative errors to 9 decimal digits.

4) Repeat steps **2)** and **3)** for $\epsilon = 0.001$.

5) Repeat steps **2)** and **3)** for $\epsilon = 0.000001$.

6) Your results should indicate that the accuracy of the computed values of $f(\pi/8,\omega)$ with double precision is much higher than that computed with single precision, which is expected. They should also show that for a given precision, the accuracy deteriorates as ϵ decreases. Explain the reason.

7) The function $f(x, \omega)$ can be approximated by using the Taylor series expansion of $\cos(\omega x)$ about $\omega_0 x$:

$$\cos(\omega x) = \sum_{n=0}^{\infty} \frac{(\omega x - \omega_o x)^n \cos(\omega_0 x + n\pi/2)}{n!}$$

Using this expansion gives the following approximation for $f(x, \omega)$:

$$\hat{f}(x, \omega) = \sum_{n=1}^{\infty} \frac{x^n (\omega - \omega_o)^{n-1} \cos(\omega_0 x + n\pi/2)}{n!(\omega + \omega_o)}$$

Add lines of code to evaluate the values of $\hat{f}(\pi/8, \omega)$ and their relative errors at single precision for $\epsilon = 0.1$ and $\epsilon = 0.000001$. Use the quadruple precision value computed in **2)** and **5)** as the "true" value for each ϵ. The function $\hat{f}(x, \omega)$ employing the first three terms of the expansion is coded in module_function.f, function.c and function.cc.

- How do the accuracies of the computed values $\hat{f}(\pi/8, \omega)$ and $f(\pi/8, \omega)$ at single precision compare for $\epsilon = 0.1$? Explain why?
- How do the accuracies of the computed values $\hat{f}(\pi/8, \omega)$ and $f(\pi/8, \omega)$ at single precision compare for $\epsilon = 0.000001$? Explain why?

Grading rubric:

- Upload the following files to Canvas for grading:
  - A tar file hw2.tar containing your program files. The tar file hw2.tar can be created by typing (from the command line):
    $ tar -cvf hw2.tar functions.c head.h main.c
    or analogously for Fortran or C++ filenames.
  - A pdf or word file containing your answers to questions **2)-7)** you don't need to include your lines of codes in this document.
- This assignment is worth 20 points, as follows:
  - (2 points) The program structure contains the files and the functions mentioned above with the appropriate naming
  - (3 points) Your program compiles without warnings or errors when -Wall (C,C++) or -warn (Fortran) options are used.
  - (10 points) When compiled and executed, your program produces accurate results for the evaluations of $f(\pi/8, \omega)$ and their relative errors.
  - (5 points) The program is adequately (but not excessively) commented.