EECE5640
High Performance Computing
Homework 3
*Submit your work on Canvas in a single zip file.

1. (30) In this problem, you will utilize the IEEE 754 format and evaluate the performance implications of using floats versus doubles in a computation.

   a.) Compute f(x) = sin (x) using a Taylor/Maclaurin series expansion. To refresh your memory:

   $$\sin(x) = \sum_0^\infty \frac{(-1)^n}{(2n+1)!}\, x^{\,2n+1}$$

   $$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \ldots.$$

   You select the number of terms you want to compute (but at least 10 terms). Compute sin(x) for 4 different values, though be careful not to use too large a value. Generate two versions of your code, first defining x and sin(x) to use floats (SP), and second, defining them as doubles (DP). Discuss any differences you find in your results for f(x). You should provide an in-depth discussion on the results you get and the reasons for any differences.

   b.) Explore the benefits of compiling on the Discovery cluster with floating point vector extensions (e.g., AVX). Use the single-precision code from part (a). Additional information is provided on AVX support on Discovery. You should try to utilize AVX-512 in this part of the assignment. Generate an assembly listing (using the -S flag) and identify 2 different AVX-512 instructions that the compiler generated, explaining their operation. Note that the class reservation on Discovery has SandyBridge CPUs which only support AVX2.

   c.) For 10 points extra credit all, compute sin x using an alternative method, and evaluate for the single and double precision.

2. (30) In this problem, you will modify the matmul.c program provided, optimizing the execution of the matrix multiplication with first a dense matrix, and second with a sparse matrix. You are welcome to use pthreads, OpenMP or any of the optimizations that were presented in class to accelerate this code. There will be extra credit awarded for the fastest dense and the fastest sparse implementations (we will use our own test sets).

3. (30) In this problem, you will utilize the OpenBLAS library available on Discovery. To use OpenBLAS, you will need to issue load openblas/0.3.6. Using the matmul.c program, replace the math with a call to appropriate GEMM library function. Make

sure you specify the -lopenblas compiler switch. Compare the speed of your solution for problem 2 with the GEMM method you used.

4. (10 points) Linpack is a benchmark created by Jack Dongarra that is used to measure how fast a high-performance computer system can solve a dense n x n system of linear equations, solving the equation Ax = b. Linpack is used in producing the Top500 list. In this problem, you should compile and run Linpack on at least 3 different platforms (you can use Discovery for this, compiling and running on 3 different X86 nodes that have different CPU models. Describe the nodes you are running on and include the results of running the code provided. Source code for Linpack is provided.

5. (10 points for MS and 20 points for Undergraduate/Plus-One) (Extra quiz credit for everyone)
   Find a published paper from an ACM or IEEE conference that discusses a novel sparse matrix format that was not covered in class. Discuss why the proposed format is superior to the CSR or CSC format. Make sure cite your sources.

* Written answers to the questions should be included in your homework 3 write-up in pdf format. You should include your C/C++ programs and the README file in the zip file submitted.