APC/AST 523 Problem Set 1
Due Wed. Oct. 12, 2011

Problem 1

You are provided with a program to perform matrix multiplication called AST523_mmulti. As its input, the program takes three integers, which are the number of rows in matrix A, the number of columns in matrix A, and the number of columns in matrix B. The matrix multiplication is represented as A*B=C, and the number of rows in B is by definition equal to the number of columns in A. The matrices A and B are initialized according to the routine sin_init in AST523_matrix_funcs.c, and the program returns the maximum value in matrix C along with the row and column number.

Use "make" to compile the program. Your first order of business is to check the program for memory leaks using valgrind (or whichever tool you prefer). In order for valgrind to work, the -g flag should be set in the makefile. If valgrind finds memory leaks, fix them. Note the -pg flag should be turned off when running valgrind, or valgrind may give strange results.

You are now asked to profile the program using gprof (or the tool of your choice). For gprof, the compiler and linker flags -pg should be set. Run the program with the input: "./AST523_mmulti 1000 1000 1000". The program may take a few seconds to run. Record the output of gprof; in particular take note of how long the program took to run and which routine took the most time. Now turn on the compiler optimization to -O1, then to -O2, and -O3. How much of a speedup do you get between -O0 and -O2?

Keep the flag set to -O2. The routine matrix_multiply in is not written in an optimal way. Optimize this routine, aiming for a factor of at least several in speedup. How were you able to achieve the speedup?

Finally, create a new file called AST523_matrix_funcs_omp.c in which you parallelize the routines in AST523_matrix_funcs.c using OpenMP. Compile with -fopenmp in the Makefile and run your program. How much of a speedup are you able to get using the maximum number of processors available to you? Note: valgrind may give strange results, so for timing you can either use a profiler that works with OpenMP or the functions in the time.h library.

Submit the modified AST523_matrix_funcs.c routine, the AST523_matrix_funcs_omp.c, and a text file called AST523_mmulti.txt describing what you did along each step of the way. *Do not modify the signature of any of the functions in AST523_matrix_funcs.c.* This means all of your modifications must conform to the definitions in AST523_mmulti.h.

Problem 2

In this problem you are asked to numerically evaluate the harmonic series

$$H_n = \sum_{k=0}^{n} k^{-1}. \tag{1}$$

The makefile generates a program called AST523_hseries, which evaluates the harmonic series shown above. As its input, AST523_hseries takes two arguments, which are an integer n, the upper limit of the summation, and a floating point number, $H_{n,e}$, which is the exact value of the harmonic series. As its output, AST523_hseries prints

$$\mathrm{abs}(H_{n,e} - H_{n,c}), \tag{2}$$

which is the error between the exact value of the harmonic series and the computed value, $H_{n,c}$. An example call to AST523_hseries with $n = 16$ and $H_{n,e} = 3.380728993$ looks like "./AST523_hseries 16 3.380728993".

The table below provides some values of $H_{n,e}$ to ten digits for various values of n along with the error from the program. Try it for yourself! Don't worry too much if you don't get exactly the same values of the error.

n	$H_{n,e}$	Error
$2^0 = 1$	1	0.
$2^4 = 16$	3.380728993	3.30×10^{-8}
$2^8 = 256$	6.124344963	8.17×10^{-7}
$2^{12} = 4096$	8.895103897	4.33×10^{-6}
$2^{16} = 65536$	11.66757818	1.50×10^{-4}
$2^{20} = 1048576$	14.44015975	3.65×10^{-2}
$2^{24} = 16777216$	17.21274803	1.81

Your first task is to diagnose the source of the error and to explain why it becomes of order unity around $n = 2^{24}$; a rough explanation is fine. Your second task is to reduce the error to $< 10^{-5}$ for $n = 2^{24}$. You are allowed to define your own variables and to modify the program as you wish, but *use only floats* in the evaluation of the harmonic series. Don't worry about optimization, but your program should not take more than a few seconds to run. Also, don't modify the input or output of AST523_hseries.

Submit the modified AST523_hseries.c file together with a file called AST523_hseries.txt in which you explain the cause of the error, why it becomes of order unity around $n = 2^{24}$, and how you were able to reduce it.