EECS 739: Final Exam
Handed out: Thursday, May 5, 2016
Due: Thursday, May 12, 2016

Academic Integrity Statement
“I have neither given nor received assistance on this exam.”

Print Name________________________________________
Signature________________________________________

The rules for this exam are as follows:

Exam rules: This exam has four questions and one extra-credit question and is four pages long. The questions are weighted equally. The exam counts for 35% of your final course grade. The exam is due between 10:30am and 1pm on Thursday, May 12, 2016, i.e., the final exam time period. You MUST turn-in your exam solutions in person. You may either turn them in by stopping by my office (3016 Eaton Hall) during this time period on Thursday, May 12. Alternatively, you can make an appointment with me to submit your exam in person prior to the final exam period. Exams found under my door, in my mailbox, etc., will not be accepted. The exam is open-book and open-notes. You may also consult outside published sources. If you use material from sources other than the textbooks, you must cite them.

Academic integrity: You are not allowed to collaborate at all on this exam. You should refrain from bringing up the questions in any kind of discussion until 1pm on Thursday, May 12. Please write and sign your name and date the following statement on a cover page to be submitted with your exam solutions: “I have neither given nor received unpermitted assistance on this exam.” You are not allowed to send any e-mail or otherwise make any on-line posting concerning the questions on this exam until after it is over. But you are allowed to consult publicly-available websites and search engines.

Help from the instructor: The only help available will be clarification of the questions. If you have questions, I can be reached via e-mail at shontz@ku.edu. No help will be given towards finding a solution.

Late policy: No late exam solutions will be accepted. The solutions must be turned in by 1pm on Thursday, May 12.

Partial credit: You must show ALL of—your work for partial/full credit. This includes any definitions, mathematics, figures, plots, pseudocode, etc. Unclear solutions will not receive partial credit.
1. (20 points; 5 points each)
   (a) Suppose the runtime of a serial program is given by \( T_1 = n^2 \), where the units of the runtime are in microseconds. Suppose that a parallelization of this program has runtime \( T_p = n^2/p + \log_2(p) \). Write a simple program that finds the speedups and efficiencies of this program for various values of \( n \) and \( p \). Run your program with \( n = 10, 20, 40, 80, 160, 320 \) and \( p = 1, 2, 4, 8, 16, 32, 64, 128 \). Show a plot of your results. What happens to the speedups and efficiencies as \( p \) is increased and \( n \) is held fixed? What happens when \( p \) is fixed and \( n \) is increased?
   
   (b) Suppose that \( T_p = T_1/p + T_o \), where \( T_o \) stands for the parallel overhead. Also suppose that we fix \( p \) and increase the problem size. Show that if \( T_0 \) grows more slowly than \( T_1 \), the parallel efficiency will increase as we increase the problem size. Show that if, on the other hand, \( T_0 \) grows faster than \( T_1 \), the parallel efficiency will decrease as we increase the problem size.
   
   (c) Give an example of an algorithm with poor strong scaling. Explain your answer.
   
   (d) Give an example of an algorithm with poor weak scaling. Explain your answer.

2. (20 points) Consider using a numerical method to approximately solve the following partial differential equation:
   \[
   \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 4, \quad 0 < x < 1, \quad 0 < y < 2; \\
   u(x, 0) = x^2, \quad u(x, 2) = (x - 2)^2, \quad 0 \leq x \leq 1; \\
   u(0, y) = y^2, \quad u(1, y) = (y - 1)^2, \quad 0 \leq y \leq 2.
   \]
   
   (a) (8 points) Using centered finite differences and \( h = k = \frac{1}{2} \), discretize the above partial differential equation and boundary conditions. Write down the resulting linear system of equations in matrix form.
   
   (b) (6 points) Solve the linear system to obtain an approximate solution to the above partial differential equation and corresponding boundary conditions. Compare your results to the exact solution given by \( u(x, y) = (x - y)^2 \).
   
   (c) (6 points) What parallel method should be used to solve the linear systems resulting from the discretization of Poisson's equation on an \( m \times n \) mesh? Assume that the mesh nodes are ordered in the natural ordering.

3. (20 points) Consider solving a linear system of the form \( Ax = b \), where \( A \) is an \( n \times n \), invertible sparse matrix of the form as shown at the top of Page 3.
   
   Here \( D_1, D_2, \ldots, D_6 \) are invertible diagonal matrices of various sizes positioned along the main diagonal, and \( E \) and \( F \) are sparse portions of \( A \) which collectively represent the remainder of the matrix.
(a) (8 points) Design an iterative method which can be used to solve the above linear system. Specify the iterative method focusing on the mathematical computations that are performed at each step. (In other words, you should not give a pseudocode or actual code.)

(b) (12 points) Write pseudocode for a parallel version of the above iterative solver for distributed memory machines. Be sure to specify the MPI functions which are used throughout your pseudocode.

4. (20 points) In class, we learned Newton’s method for use in finding a local minimum of an objective function. Recall that, on each iteration, the standard Newton method takes a full step in the computed direction. In this question, we consider the addition of a line search (on each iteration) in order to determine an appropriate step length to take in the computed direction. An “appropriate step length” here is one in which Newton’s method behaves in the desired manner, i.e., it makes progress towards a minimum.

(a) (5 points) Write pseudocode for a serial version of the modified Newton method. For this question, you can add any type of meaningful linesearch. (Assume the objective function is such that the relevant matrices are dense.)

(b) (5 points) Specify the advantages and disadvantages associated with the addition of the line search (in comparison to the standard Newton method).

(c) (10 points) Write pseudocode for a parallel version of your modified Newton method which employs both blocks and threads and uses CUDA to program a single GPU.

(10 points; 2 points each) Extra-Credit Question (Optional):

1. For what does the acronym CFD stand? (I am only interested in the acronym as it was used in Prof. Wang’s guest lecture.)

2. Name two distinct CFD applications which Prof. Wang discussed in his talk.
3. What were the two major classes of parallel numerical methods which Prof. Wang described in his talk?

4. What is the largest CFD problem that has been run to date? How large was the machine on which the code was run?

5. What is your favorite topic that we covered in class during the second half of the semester? Why?