EECS 700: Homework 2

Due: Thursday, November 13, 2014 (In Lecture)

Each problem is worth 20 points.

1. Consider the following chemical reaction: $\mathrm{A} + 2\mathrm{B} \rightarrow \mathrm{R} + \mathrm{S}$, for which the mechanism proceeds through an intermediate species, X.

   The mechanism is given by the following reactions:

   $$\mathrm{A} + \mathrm{B} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} \mathrm{X}$$

   $$\mathrm{X} + \mathrm{B} \xrightarrow{k_3} \mathrm{R} + \mathrm{S}.$$

   The rate equations which correspond to these reactions are given by the following system of ODEs:

   $$\begin{aligned} \frac{d[A]}{dt} &= -k_1[A][B] + k_2[X] \\ \frac{d[B]}{dt} &= -k_1[A][B] + k_2[X] - k_3[X][B] \\ \frac{d[X]}{dt} &= k_1[A][B] - k_2[X] - k_3[X][B]. \end{aligned}$$

   A realistic set of initial conditions and rate constants are as follows: $[A]_0 = 1, [B]_0 = 1, [X]_0 = 0, k_1 = 1/10, k_2 = 1/10, k_3 = 2/10.$

   Integrate the system of ODEs from $t = 0$ to $t = 5$ using the 4^{th}-order Runge-Kutta method with stepsize $h = 0.5$. (I suggest you adapt my ForwardEulerTest.cpp code so that it integrates the ODE based on the IVP solver identified here. You will also need to modify the IVP being solved, the step size, etc.)

   For this question, please hand in your C++ code (which you will implement yourself), as well as your output and a plot (using your favorite plotting program) illustrating the behavior of the reactants as a function of time. Summarize the behavior of the reactants in a few sentences (or less).

2. Use the Triangle mesh generation software package to generate a series of 2D unstructured triangular meshes of the state of Kansas. (It's OK

to use a rough outline of the state.) The series of meshes you generate should demonstrate your ability to generate meshes of low, medium, and high quality as measured according to the minimum angle in each mesh.

For this question, hand in the *.poly files which you used as input to triangle, as well as a screenshot of your triangle session (illustrating which commands you used to call triangle and what its output was). Please also hand in figures of the resulting meshes drawn using ShowMe. (ShowMe is a software package that is used to visualize meshes and is a companion software package to Triangle. It should only take a few minutes to learn how to use ShowMe.)

3. Consider the following PDE (which specifies linear diffusion):

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2},$$

where $a > 0$ is a constant.

First, derive the equations for the Crank-Nicolson finite-difference approximation method (i.e., a semi-implicit method) for this particular PDE.

Second, convert your resulting linear system of equations into a matrix problem to be solved for $u(t, x)$.

Third, set $a = 2$, and solve the above PDE on a finite-difference grid on a unit square (i.e., $x \in [0, 1]$ and $t \in [0, 1]$) with $\Delta x = 0.2, \Delta t = 0.1$, and $u(t, 0) = 2at, u(t, 1) = 1 + 2at$ for the boundary conditions. Use $u(0, x) = x^2$ as the initial condition. (I suggest that you adapt my LU factorization function so that it performs the solution of the relevant tridiagonal system using the Thomas algorithm (i.e., bidiagonal LU algorithm). This will also require modifications in the main program as well.

For this question, please hand in your C++ code, as well as your output which generates an approximate solution to $u(t, x)$. Explain how you addressed computational efficiency in your implementation, as well. A few sentences will suffice.

4. Using MPI, implement a parallel version of the block matrix multiplication algorithm given in the Algorithm 8.3 pseudocode on p. 346 of our parallel computing handout. Experiment with running your code on various numbers of processors and with various block sizes. Use

matrices with 1 million rows and 1 millions columns for your experimental study.

Use your experimental data to determine if there is an optimal block size for a given number of processors.

Hand-in your code, tables of your experimental output, and a brief summary of your results in regards to the optimal block summary.

Important note: Please turn in a hard copy of your solutions for each problem. Please also send me an e-mail (to shontz@ku.edu) with your C++ cod and output with a single zip archive of your code for this assignment by 11am on the due date.