Parallel Computing for Partial Differential Equations — Matthias K. Gobbert
Sommersemester 2012 — Universität Kassel
Homework 2 — due on Wednesday, April 18, 2012

Each programming problem needs to start out with text explaining what you did; computer code, tables, or plots should *never* be the first page of any problem! This text should verbally introduce, explain, and interpret all other material including tables, plots, and computer code, in this order. Include relevant portions of code as part of your discussion for maximum clarity.

1. [4 or 6 points.] Please read an introduction to Matlab and work through its examples. I recommend the *Getting Started* guide provided by the software itself. Please see the Matlab area of my homepage under the header *How to Get Started with Matlab* for information how to get there. Read at least the first two sections "Introduction," and "Matrices and Arrays" immediately; make sure to try out the commands as you go along. Plan on reading the following sections "Graphics," "Programming," and possibly more (e.g., certain subsections under "Desktop Tools") also in the near future.

   Try out Matlab on several platforms available to you and report if everything behaved as desired. Platforms could include the IT Servicezentrum at the Uni Kassel and the computers in the Pool-Raum 2421 under Windows as well as under Linux. (Do not submit output here, just a report that says what you did and whether everything worked. Also try out relevant commands that we might have discussed in class like `spdiags`, `mesh`, etc.)

2. [8 or 14 points.] Use Matlab to solve the problem from the tech. rep. HPCF–2010–2. In particular, produce results such as Figure 3.1 and a table organized like Table 3.1 in that report for value of N that are as large as possible. You should have these results for versions of your code that use Gaussian elimination as well as that use the conjugate gradient method.

   (Note: For our overall goals, it does not matter if N is a little bit smaller or larger. But for best mathematical justification, use N such that the mesh spacing h is a power of 2. This might force the exact N values to be numbers like $2^\nu - 1$. To understand this fully, your report should state and rederive all relevant formulas.)

3. [Optional problem — 8 or 0 points.] Matlab is a commercial software package. You might be interested in an alternative that you can use on your own laptops, also in view of internet connectivity issues. I recommend the package Octave (`www.octave.org`). Repeat the steps from the previous problems in Octave, that is: Find and read some introductory documentation provided by the software, visit its webpage, try out the basic commands that you tried in Problem 1 in order to test if it behaves like Matlab or slightly differently; then solve the test problem from tech. rep. HPCF–2010–2 and provide tables for versions of your code with Gaussian elimination and the conjugate gradient method.

   (This problem is optional, since I do not want to force anyone to download and install software on his/her computer, but I wanted to provide it as a standalone alternative to Matlab. The background for my knowledge is tech. rep. HPCF–2011-10 available on the Publications page at `www.umbc.edu/hpcf`.)

4. [Optional problem — 0 points.] When I asked you to replace `u = A \ b` in Matlab by a call to the `pcg` function, you would use the same matrix `A` at first. But in preparation for our programming in C, can you develop an alternative that never assembles the matrix `A` at all? This is referred to a matrix-free implementation. Matlab's `pcg` function is prepared for this, sicne it can accept a function handle in the place of the matrix `A` in the call to `pcg`; see the documentation `doc pcg`. Concretely here, you would write a function `Ax.m` that takes a vector `u` as input and returns a vector `v` as output that is mathematically `v = A*u`. Notice that you can test this function `Ax` by itself (called 'unit testing') by comparing `Ax(u)` to `A*u` computed directly using `A = setupA(N)`.

5. [Optional problem — 0 points.] This is an exercise in effective Matlab programming. Notice that only the solution at the interior points is determined by the linear system $A\mathbf{u} = \mathbf{b}$. So, if you use `mesh(X,Y,U)` to produce the mesh plot, it is *not* exactly like the one in Figure 3.1 (a) of the tech. rep. HPCF–2010–2, because that plot includes the boundary points. How can you produce the plot with boundary points effectively in Matlab?