Temple 3043 Numerical Analysis I Fall 2011

Problem Set 8

(Out Mon 10/24/2011, Due Tue 11/01/2011)

Problem C

Download the Matlab code `temple3043_heateqn.m` from the course website, and run it. You will see a numerical approximation of the solution of the time-dependent homogeneous heat equation, as it approaches a constant temperature distribution. The numerical scheme is a second order accurate central-in-space, Crank-Nicolson-in-time finite volume scheme. Consult for more details:
`http://en.wikipedia.org/wiki/Heat_equation`
`http://en.wikipedia.org/wiki/Finite_volume_method`
`http://en.wikipedia.org/wiki/Crank-Nicolson`

The approach requires the solution of a linear system in each time step (plus one matrix-vector multiplication), and the program measures the time spent on these operations. In the given version, the Backslash operator is used. Since the system matrix is the same for each solve, and since the right hand side vectors may not change too much from time step to time step, one can probably invoke smarter solvers. Specifically, replace the Backslash solve by the following linear solvers (if possible, always use your own routines), and perform a rigorous run-time comparison (create a table, and then draw conclusions from it) with the following approaches

(a) Pre-compute the LU factorization, and use Backslash for the solves with L and U in each time step

(b) Same as (a), but compute the LU factorization in each time step (which is of course not smart)

(c) Pre-compute the Cholesky factorization, and use Backslash for the solves with R' and R

(d) Same as (c), but compute the Cholesky factorization in each time step

(e) Jacobi iteration with starting vector of zero

(f) Jacobi iteration with starting vector being the solution of the previous time step

(g) Gauß-Seidel iteration with starting vector of zero

(h) Gauß-Seidel iteration with starting vector being the solution of the previous time step

(i) SOR with optimal ω (found either by theory, or by numerical experiments) with zero starting vector

(j) The same SOR with starting vector being the solution of the previous time step

(k) Conjugate Gradients with starting vector of zero

(l) Conjugate Gradients with starting vector being the solution of the previous time step

for the following four tests

(1) $n = 500$ and $dt = 10^{-1}h$ (the default)

(2) $n = 500$ and $dt = 10^{-2}h$

(3) $n = 250$ and $dt = 10^{-1}h$

(4) $n = 250$ and $dt = 10^{-2}h$.

Note that with the given settings, the numerical approximation is within 10^{-5} of the true solution. Hence, for all approximate solvers, you should require an accuracy of at least 10^{-6}.