Programming Assignment 3:
Krylov Methods

Mech 511

Due date: March 25

In this assignment, you will explore the use of Krylov methods for solving a large system of equations arising from a CFD simulation. The physical problem is the 2D energy equation, as in the programming assignment for Mech 510; feel free to re-use as much of your code from that assignment as you like. You will compute both the flux integral and the (full, unfactored) LHS matrix for the same conditions as the final part of that assignment:

$$\begin{aligned} u(x,y) &= 6\bar{u}y(1-y) \\ v(x,y) &= 0 \end{aligned}$$

with $\bar{u} = 3\frac{\mathrm{m}}{\mathrm{sec}}$. The temperature will be fixed at the inflow ($T = y$) and the walls (lower wall: $T = 0$, upper wall: $T = 1$), and the temperature profile will be considered fully-developed at the outflow. Finally, the flow conditions are such that $\mathrm{Re} = 25$, $\mathrm{Pr} = 0.7$, and $\mathrm{Ec} = 0.1$. Use a domain size of 40×1 with meshes as described below and a time step of $\Delta t = 0.25$.

1. As a baseline case, set up and solve the linear system numerically by direct inversion; you want to consider some numerically stable algorithm like LU rather than Gauss elimination, which is not numerically stable. Use mesh sizes starting with 20×10 (matrix size of 200×200), and double the size in each direction until run time is on the order of minutes. Tabulate run time as a function of mesh size.[1] How did you expect the run time to behave with increases in mesh size for this approach, and were your expectations met?

[1]If your operating system won't provide CPU time for program execution, you may need for small meshes to have an outer loop so that you repeat the whole calculation enough times to time it by hand.

2. Create the approximate-factored version of the LHS, and compute the update to the solution using approximate factorization. Again, tabulate run times. What behavior did you expect for this case, and how closely did the scheme match this? What is the error in the solution of the linear system for a 80×40 mesh, compared with the (exact) solution from part 1?

3. Create a GMRES subroutine to solve the full linear system from part 1. For each mesh, use 20, 30, and 40 vectors in the GMRES iteration, and compare the accuracy of the solution of the linear system obtained with each choice. As for the previous cases, tabulate run times and compare actual to expected performance. Is the error in the linear system solution larger or smaller than for the approximate factored scheme for a 80×40 mesh.

4. The GMRES scheme of part 3 has no preconditioning. Modify your subroutine to use the tri-diagonal factors as a quick and easy preconditioner. Compare performance for a moderately large mesh [2] of the pre-conditioned and non-pre-conditioned versions, and the approximate-factored version, in terms of both CPU time and accuracy for 20, 30, and 40 vectors.

Were you surprised by any of the results you obtained? What conclusions can you draw from your results?

[2] One for which the run time is a minute or more for the unpreconditioned case.