ENGR570 HW 3

Linear Solvers

Dylan Lyon

October 27, 2022

1 Problem 2

1.1 Logic

So the first order FD analog of Laplace eqn for a 2D star stencil is:

$$-4 \cdot u_{ij}^{(k)} + u_{i+1,j}^{(k)} + u_{i-1,j}^{(k)} + u_{i,j+1}^{(k)} + u_{i,j-1}^{(k)} = 0 \tag{1}$$

To rearrange this for an update, use

$$u_{ij}^{(k+1)} = \frac{u_{i+1,j}^{(k)} + u_{i-1,j}^{(k)} + u_{i,j+1}^{(k)} + u_{i,j-1}^{(k)}}{4} \tag{2}$$

For treatment of BC's, assume $u = 0$ along edges. So first order interp between any border square and its ghost square outside the border should be zero in the middle. Ergo, for squares on the edges, use $-u_{ij}$ as the ghost node value.

1.2 Questions

(1) My formula for spectral radius was $\rho = \text{RelTol}^{1/\text{ItersToConvergence}}$ where RelTol=1e-8. I used this prediction to get ItersTo1e-6 = $\frac{ln(\text{RelTol})}{ln(\rho)}$ where RelTol=1e-6. There is a better formula; After the first few iterations, the update vector tends to be an exponential walk towards the solution. Take a few samples of r_l, then use their regressed exponent to get an approximation for $\rho(T)$ in $err^{(n)} = \rho(T)^n err^{(0)}$.

(2) Predictions:

- Fastest time to solution: RBBJ
- Fastest time per iteration: GS
- Smallest spectral radius: RBBJ

(3) Results:

- Fastest time to solution was GS. Probably due to each iteration averaging out to being nearly twice as valuable. Also, were it not for relative convergence, GS doesn't need to remember previous values and can simply overwrite its own vector.

- Fastest time per iteration was RBBJ, though it was similar to JI. GS was slightly slower, maybe because of reading/writing to same vector? I'm not sure.
- Smallest spectral radius: GS at .9935. Really solely a factor of the number of iterations here.

2 Problem 6

2.1 Logic

Really just followed the wikipedia algorithm to the T : `https://en.wikipedia.org/wiki/Biconjugate_gradient_stabilized_method#Algorithmic_steps`
My method had 139 iterations with 1.371884 ms per iteration. PETSC had 136 iterations with 1.242647 ms per iteration based on sys time. I can't believe it's close!!!!!!!!!
Here's the plot: Cool problem, was worth it.

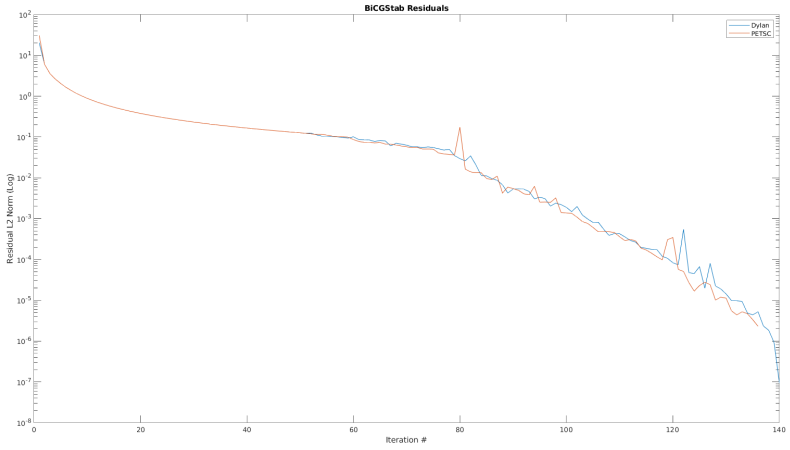

Figure 1: Residual comparison. Note: That looks oddly like a truncated gamma function, don't know how that gets there.