Numerical Computing :: Project Five

Using the matrices from last assignment (or any others you find interesting), perform the following study:

1. Generate a right-hand-side b of all ones of appropriate size.

2. Solve $Ax = b$ with a generic linear solver (eg, `numpy.linalg.solve` or Matlab's `backslash`). Call the resulting vector *truth*. This is the vector against which you will compute the error. Run a timing study with the generic linear solver.

3. Write a function that solves $Ax = b$ using either the LU decomposition or the Cholesky factorization, depending on whether the matrix is symmetric or not.

4. Write a function that solves $Ax = b$ using the Jacobi method. Run a timing study with your function.

5. Write a function that solves $Ax = b$ using the Gauss-Seidel method. Run a timing study with your function.

For parts 3-5, report the relative error compared to the *truth* that you computed in part 2.

Here are some questions that might help you think about this problem.

- Look up "diagonally dominant." It relates to the convergence of the iterative methods.

- Remember your paramedic training. Which methods work better for which matrices? Why do you think that is? And what do you mean by "work better"?

- Try interpreting the results from the iterative methods in light of the theory we know about fixed point methods.