APC/AST 523 Problem Set 2
Due Monday Oct. 31

Problem 1

a & b) Do problem 1 of Ch. 4 from *Iterative Methods for Sparse Linear Systems.* An online version of the first edition can be found at the website: www-users.cs.umn.edu/∼saad/books.html You may find some of the theorems in chapter 4 quite useful in solving the problem.

c) Letting $\alpha = 2$ as before, calculate how many times the number of iterations of the Jacobi algorithm are needed to converge to the same same error tolerance as the Gauss-Seidel algorithm?

d) Let $\alpha = 2$ again and consider the system

$$T_2 x = b. \tag{1}$$

Let $n = 100$, and let x and b be defined by

$$x = q_j = \left[\sin\left(\frac{j\pi}{n+1}\right),\ \sin\left(\frac{2j\pi}{n+1}\right), \ldots,\ \sin\left(\frac{nj\pi}{n+1}\right)\right] \tag{2}$$

$$b = \lambda_j q_j, \qquad \lambda_j = 2\left(1 - \cos\left(\frac{j\pi}{n+1}\right)\right) \tag{3}$$

i) Implement Jacobi's algorithm numerically for $j = 1$ using an initial starting guess of $x^0 = 0$. Iterate for 10000 iterations. Let the error for the k-th iteration be defined by

$$\text{Error} \equiv \max |x_i - x_i^k|, \qquad i = 1, \ldots, n. \tag{4}$$

Plot the error as a function of the number of iterations on a log-linear plot (y-axis is log, x-axis is linear). Is the convergence factor $\rho(G_J)$ in this case? If not, why not?
ii) Same as part (i) but now for $j = 2$. Is the convergence factor $\rho(G_J)$ in this case? What is the analytical expression for the convergence factor for the j-th eigenvector?
iii) Based on the findings of parts (i) and (ii), how would you expect the error to behave as a function of the number of iterations for an arbitrary x that is a linear combination of the eigenvectors, q_j? Is $\rho(G_J)$ an accurate estimator of the convergence rate when k, the number of iterations, is small? What about in the limit $k \to \infty$? Explain your reasoning.

You can use whatever programming language you like to implement the Jacobi algorithm. Submit your code on blackboard, including all relevant files and a Makefile or instructions on how to run the code.

2. Finding complex roots of a function is equivalent to two-dimensional root finding. If we have a function $f(x+iy)$, then

$$f(x+iy) = 0 \tag{5}$$

implies that

$$\|f(x+iy)\|^2 = 0, \tag{6}$$

where the vertical bars denote the norm. Define the function

$$f(\omega) \;=\; \epsilon^2(1-\epsilon^{-1}\omega^2)(\omega-1)^4 - (1-(\omega-1)^2)\omega^4, \qquad \epsilon = .5 \tag{7}$$

where $\omega = x + iy$ is in general complex. Write a program which uses Powell's method to find a root of $f(\omega)$ near $\omega_0 = .4 + .4i$, where ω_0 is the initial guess for the root. Your code should output the value of the root it finds as the answer.

If you are using C, complex numbers are implemented through the complex.h library. You can check your answer by e.g. using the Solve function in Mathematica. Submit your code on blackboard, including all relevant files and a Makefile or instructions on how to run the code.