1. Prove that if A is a symmetric positive-definite matrix with eigenvalues $\rho_1, \ldots, \rho_n$, and p is a polynomial, then $\|p(A)\|_A = \max_{1\le j\le n} |p(\rho_j)|$.

2. Prove that the for the conjugate gradient method the search directions s_i and the errors $e_i := x_* - x_i$ satisfy $s_i^T e_{i+1} \ge 0$ (in fact $s_i^T e_j \ge 0$ for all i, j). Use this to show that the l_2-norm of the error $\|e_i\|$ is a non-increasing function of i.

3. We analyzed preconditioned conjugate gradients, with a symmetric positive definite preconditioner M, as ordinary conjugate gradients applied to the problem $M^{-1}Ax = M^{-1}b$ but with the M-inner product rather than the l_2-inner product in $\mathbb{R}^n$. An alternative approach which doesn't require switching inner products in $\mathbb{R}^n$ is to consider the ordinary conjugate gradient method applied to the symmetric positive definite problem $(M^{-1/2}AM^{-1/2})z = M^{-1/2}b$ for which the solution is $z = M^{1/2}x$. Show that this approach leads to exactly the same preconditioned conjugate gradient algorithm.

4. The Matlab command `A=delsq(numgrid('L',n))` is a quick way to generate a symmetric positive definite sparse test matrix: it is the matrix arising from the 5-point finite difference approximation to the Laplacian on an L-shaped domain using an $n \times n$ grid (e.g., if $n = 40$, A will be $1,083 \times 1,083$ sparse matrix with $5,263$ nonzero elements and a condition number of about 325. Implement the conjugate gradient algorithm for the system $Ax = b$ for this A (and an arbitrary vector b, e.g., all 1's). Diagonal preconditioning does no good for this problem. (Why?) Try two other possibilities: tridiagonal preconditioning and incomplete Cholesky preconditioning (Matlab comes equipped with an incomplete Cholesky routine, `cholinc`, so you don't have to write your own). Study and report on the convergence in each case.