Assignment 3. Due Friday, Feb. 11.

Reading: Generalizations of the Field of Values ...

1. Let $A$ be a normal matrix whose eigenvalues lie on a line in the complex plane. Show that the polynomial numerical hull of degree 2 of $A$ is equal to the spectrum of $A$: $H_2(A) = \sigma(A)$.

2. Let $J$ be an $N$ by $N$ Jordan block with eigenvalue 0. The polynomial numerical hull of degree $N - 1$ of $J$ is a disk about 0 of radius $r$, where $1 > r > 1 - \log(2N)/N$.

   (a) For a given $\zeta \neq 0$, write $(\zeta I - J)^{-1}$ as a polynomial of degree $N - 1$ in $J$.

   (b) Give a lower bound on $\| (\zeta I - J)^{-1} \|_2$ based on $H_{N-1}(J)$. [Note: In this way, one can use polynomial numerical hulls to estimate pseudospectra.]

3. Download SDPT3 version 2.3 from
   
   www.math.nus.edu.sg/~mattohkc/sdpt3.html
   
   and follow the directions to install it on your computer. [I have been able to make it work under Unix, but not under Windows. If you have trouble, let me know; I have a copy that works on zeno.] Write a code to compute the polynomial numerical hull of a given degree of the matrix with $-1$’s on the subdiagonal and 1’s on the main diagonal and the first three superdiagonals. These computations may take a long time, so use a fairly small matrix, say, 20 by 20, and try some fairly low degree polynomial numerical hulls, say, $k = 2$ and $k = 6$.

   There is a routine in the SDPT3 package that directly solves the problem
   
   $$ \min_{c_1, \ldots, c_k} \| I - \sum_{j=1}^{k} c_j B^j \|. $$
   
   This routine is called igmres. You will need to loop over different values of $\zeta$, set $B = A - \zeta I$, call igmres (sending it $B$ and the degree $k$ of the polynomial numerical hull you are computing), and then test if the resulting norm is equal to 1. (I test if it is greater than .999999, to allow for a little roundoff.) If the test passes, then $\zeta \in H_k(A)$; otherwise it is not.

   The following code will test if a given point zeta is in $H_k(A)$:
   
   ```matlab
   B = A - zeta*eye(size(A));
   [blk, AA, C, b, X0, y0, Z0, objval, p] = igmres(B, k, 0, 1); % Check the documentation
   if abs(objval(1)) > .999999,
   ```
\% zeta is inside
else
\% zeta is outside
end;

Turn in plots showing the points that are inside the polynomial numerical hulls that you compute.