MCS 471 - Computer Problem 3

Write a Matlab function, called *myeigen*, to implement the shifted inverse power method for the eigenvalue problem $Ax = \lambda x$. The value of the shifted parameter is $\alpha$ and is input by the user. Recall how one might guess the values of $\alpha$. For example, if $\alpha = 0$, then this iteration method yields the smallest (in absolute value) eigenvalue and its corresponding eigenvector.

The inputs into the function should be the matrix $A$, the value of $\alpha$, and the initial starting vector for the iteration. The outputs should be the approximation to the smallest eigenvalue and its corresponding eigenvector.

To invert the matrix $A$ in the iteration use Matlab’s $LU$ decomposition command and then invert the $L$ and $U$ separately. The stopping criterion is when the 2 norm of the difference between two consecutive eigenvector iterates is less than $10^{-4}$. Compare your results with the results obtained using the build-in Matlab command *eig*:

```matlab
[lambda, v] = eig(A)
```

1. Test your program on:

$$
A = \begin{pmatrix}
2.5 & -2.5 & 3.0 & 0.5 \\
0.0 & 5.0 & -2.0 & 2.0 \\
-0.5 & -0.5 & 4.0 & 2.5 \\
-2.5 & -2.5 & 5.0 & 3.5
\end{pmatrix}
$$

Use the shifts $\alpha = 0, 2.5, 4.5$ and an initial vector with all ones.

2. Consider

$$
A = \begin{pmatrix}
0 & 11 & -5 \\
-2 & 17 & -7 \\
-4 & 26 & -10
\end{pmatrix}
$$

Use the shifts $\alpha = 0, 4.2$ and the starting vector $(1, 2, 1)'$. Repeat using the same shifts but with the starting vector $(1, 1, 1)'$. Write a short explanation describing the outcome.

Here is a crude program for the power method

```matlab
a = [ 0 11 -5; -2 17 -7; -4 26 -10]
x = [1 1 1]'
err = 1;
while (err-0.001 >= 0)
y = a*x;
% locate the maximum element in |y| and its index k
[c,k] = max(abs(y));
c1 = y(k);
% normalize the vector y
y = y/c1;
err = norm(y-x);
x=y; end
x
c1
```