Lab 13—Advanced Diagonalization

Objective: To be able to diagonalize matrices with repeated and complex eigenvalues.

MATLAB Commands:

`angle(z)`	Given a complex number $z = x+iy$, this routine returns the polar form angle $\theta = \arctan(y/x)$. The angle is returned in radians.
`abs(z)`	Given a complex number $z = x+iy$, this routine returns the magnitude $R = \sqrt{x^2+y^2}$.
`rr(M)`	Step-by-step row reduction of matrix `M`.
`rref(M)`	Reduces matrix `M` (with real entries) to reduced row echelon form.
`eig(M)`	Finds eigenvalues of matrix `M`, including complex eigenvalues when appropriate.
`[P,D]=eig(M)`	Special syntax used in 3(d) below.

1. a) For each of the following matrices M, determine the eigenvalues.

$$M_1 = \begin{bmatrix} 4 & -2 & 1 \\ 2 & 0 & 1 \\ 2 & -2 & 3 \end{bmatrix} , \quad M_2 = \begin{bmatrix} -1 & 6 & 2 \\ 0 & 5 & -6 \\ 1 & 0 & -2 \end{bmatrix}$$

   b) Decide whether each matrix can be diagonalized over the real numbers and, if so, give the matrix P which diagonalizes it. Confirm that $P^{-1}MP$ is diagonal by using MATLAB to multiply the matrices together.

2. Here we will practise some simple complex arithmetic which may be useful in completing the next exercise. Consider the complex numbers

$$Z = 1 + 3i, \quad W = 2 - i$$

   Find $1/Z$ and WZ by hand calculations. Check using MATLAB. Use MATLAB's `abs()` and `angle()` functions to write Z in the polar form $Z = Re^{i\theta}$.

3. a) Find the eigenvalues of $M = \begin{bmatrix} 1 & 2 & -5 \\ -2 & 1 & 2 \\ 5 & -2 & 1 \end{bmatrix}$.

b) Now find the eigenvectors by row reduction. Use `rr()`, not `rref()` (see the Note at the bottom for an explanation). You may find it surprisingly hard to choose the correct row operations to carry out. [Hint: If you find the eigenvector of a complex eigenvalue by row reduction, you should then be able to find the eigenvector of the complex conjugate eigenvalue without further work.]

c) Write the matrix P that diagonalizes M and check that $P^{-1}MP$ is diagonal.

d) MATLAB provides a syntax for the `eig()` command which returns *both* the diagonal form $D = P^{-1}AP$ of a matrix A *and* the matrix P that diagonalizes it. The eigenvectors are of course the columns of P. Type the command

```
[P,D]=eig(M)
```

Check that $P^{-1}MP = D$. Do the columns of P look like the matrix you found in part (c)? If not, can you give an explanation (in words, supported by calculations if you can) for the apparent discrepency?

Note on **`rref()`**: If you try to use `rref(M-x*eye(3))` in Q3 to find the eigenvector belonging to the complex eigenvalue `x`, you will probably be surprised that MATLAB returns a matrix with three leading 1s. You should already know that `M-x*eye(3)` cannot be rank 3 since if **v** is an eigenvector, then so is $k\mathbf{v}$ for any (complex) number k, so you expect a free parameter in the solution for the eigenvector; thus `M-x*eye(3)` should be rank 2.

The problem here is round-off error. Most likely you entered a five-digit number for the eigenvalue `x`, which is only an approximation, not the exact eigenvalue. Since it's not exactly the eigenvalue, `M-x*eye(3)` should not be quite **0**. When row-reducing, this will have the effect that a row that should be full of zeroes will instead have an entry that isn't quite zero. A human eye using the `rr()` routine can detect this and compensate for the error, but MATLAB will instead divide the whole row by that entry, producing a leading 1 where there should really be a zero.