Jordan Form Overview PhL 12.22.04

How I got to this subject. I was reading Jim's papers and saw mention of Fortran routines to get eigenvalues of a tridiagonal matrix by some strange QL method. This led me to do a broad review of "matrices", a piece of equipment long idle on my shop floor. When reading M&M, I saw their discussion of getting an arbitrary matrix to "triangular block diagonal form" which I called Pivot and Twist, and it was a torturous discussion. I later learned that something called the Schur Decomposition gets you by similarity to a completely triangular form, so at least this result was consistent with the M&M discussion. But things were still very hazy. I read the Fortran book about how you actually compute eigenvalues and eigenvectors numerically using the QR iteration method, a wonder of the world. But in the theory world, there is more going on here. Jordan did Jordan Form around 1870, see his bio.

Camille Jordan, by the way, was French **1838-1922**, an engineer and mathematician.
(See http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Jordan.html)

An arbitrary square matrix can be brought by similarity to Jordan Canonical Form. This form is triangular block diagonal as M&M said, but a lot more is known. It is really a double direct sum. The outer direct sum puts the eigenmanifolds into direct sum (block diagonal), and this is now proved from the Primary Decomposition Theorem which KR Matthews derives in his notes after LOTS of preparation.

Then within each eigenmanifold, there is further direct sum action. You end up with a set of small blocks with λ on diagonal, and only 1's on first off diagonal. Within each eigenmanifold, we have these facts:

(1) the total number of diagonal elements is of course the algebraic multiplicity (sum of block dims)

(2) the number of Jordan blocks is the geometric multiplicity

(3) the size of the largest block is the minimal polynomial exponent b for this eigenmanifold.

In my basic study of matrices, the notion of a "minimal polynomial" never came up. I now know what it is. It has exactly the same form as the characteristic polynomial, but the exponents can all be lower. I know exactly how to compute this polynomial (in principle) for any matrix A.

In order to "fill out" the Jordan structure of a particular manifold, you need three numbers: algemult, geomult and the min poly exponent. In some cases, you don't need all three, but in general you do need all three. You can see this by thinking about the case where alge mult = 4. I think beyond 6x6 you need even more information to get the form figured out.

When the geomult is full, you get all 1x1 blocks and you are fully diagonal, no surprise. When geomult is only 1, then you get a single full square block. In general, each square Jordan block does have a set of "generalized eigenvectors" that I know how to calculate from my EE263 notes. For each block, only one of these (the "first column") is a true eigenvector, the others are sort of fillers.

Just as with a fully diagonalizable matrix, the similarity matrix which takes you to Jordan form consists of the eigenvectors along with these filler generalized eigenvectors. Thus, you can in fact build the similarity

you need to get any matrix A into Jordan Canonical Form. I had wondered earlier how this worked when you did not have enough eigenvectors, I thought maybe some repeated. Now I know the answer.

The Jordan form is "as far as you can go". Those blocks are the primitives. Your distance from non-diagonalness is reflected by how many ones you have on that first off diagonal.

Just as a reminder, if the matrix A has no "defects", meaning geo = alg in each manifold, then the Jordan form will be fully diagonal. We see now why this has to be true if all eigenvalues are different!

The authors like to make a parallel between these Jordan blocks and prime numbers:

A = sum of irreducible Jordan blocks
N = product of powers of prime numbers

In each case, you have reduced something to the most basic primitive elements.

Conclusion: I think I know how to take any matrix to Jordan form now, I could do it manually. But Maple has a simple call jordan(A,'P') which computes the form *and* the similarity which gets you there. I suspect this program computes the basic eigenvectors, then computes the generalized eigenvectors, then has the similarity, and then just uses it to get J.