An Algorithm for Finding the Jordan Canonical Form of a Matrix

Jim.Swift@nau.edu

Given a square $n \times n$ matrix A, this algorithm finds the matrix P such that $B := P^{-1}AP$ is in Jordan Canonical Form (JCF). It is assumed that A has integer entries, or possibly rational entries, so all calculations can be done exactly. This algorithm is not appropriate if the entries are floating point numbers.

See section 1.8 in Lawrence Perko's book "Ordinary Differential Equations and Dynamical Systems" for background. We follow his notation, but the algorithm given here is much more efficient than the one quoted in the book.

This algorithm is based on Chapter 9 in the Schaum's Outline on Matrix Operations, by Richard Bronson. The algorithm is for humans rather than computers, since it takes some higher-order thinking to implement in step 2.2.

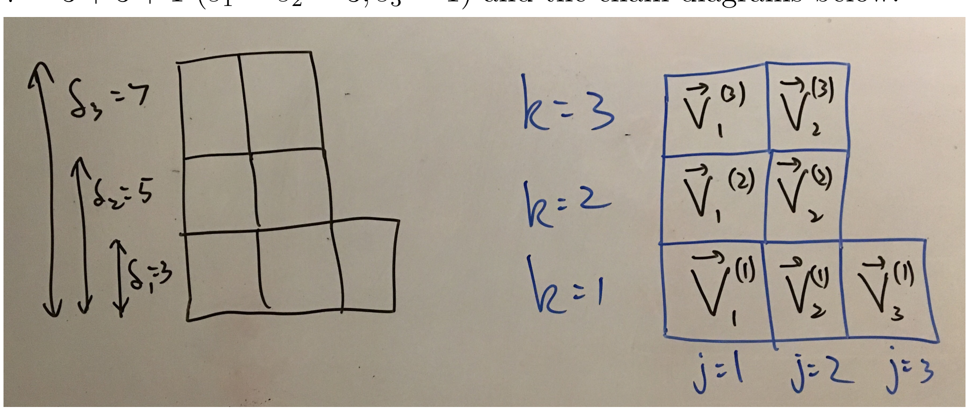
- **Step 1:** Find the eigenvalues of A. This is hard by hand. I use Mathematica's Eigenvalues[ ] command. We assume that the multiplicity of at least one eigenvalue is larger than 1. Otherwise we would just diagonalize the matrix!

- **Step 2:** For each eigenvalue λ of A, let m denote the multiplicity of the eigenvalue. We will find an $n \times m$ matrix P_λ that contains m columns which are generalized eigenvectors for λ. Initialize with $P_\lambda = 0$.

  - **step 2.1:** Compute $(A - \lambda I_n)^k$, rref$((A - \lambda I_n)^k)$ and the deficiency index $\delta_k = \dim(\mathrm{Ker}((A - \lambda I_n)^k))$ for $k = 1, 2, \ldots$ until $\delta_k = m$. Note that δ_k is the number of zero rows in the reduced row echelon form of $(A - \lambda I_n)^k$. I use Mathematica's RowReduce[ ] function.
  - **step 2.2:** Construct the chain diagram for this λ. The row at height (or rank) $k = 1$ has δ_1 boxes. The rows at height $\leq k$ have δ_k boxes. The boxes in the chain diagram should be left-justified. Each column represents a chain of generalized eigenvalues. The chain diagram defines the size of the chains s_j for $j = 1, 2, \ldots \delta_1$. This gives a partition of the integer $m = s_1 + s_2 + \cdots + s_{\delta_1}$ with $s_1 \geq s_2 \geq \cdots \geq s_{\delta_1} \geq 1$. Then put the generalized eigenvector $\mathbf{v}_j^{(k)}$ in chain j at height k. These generalized eigenvectors are all nonzero vectors that satisfy the "chain condition"

$$(A - \lambda I_n)\mathbf{v}_j^{(k)} = \mathbf{v}_j^{(k-1)}, \text{ for } k > 1, \text{ and } (A - \lambda I_n)\mathbf{v}_j^{(1)} = \mathbf{0}.$$

For example, if $m = 7$, $\delta_1 = 3$, $\delta_2 = 5$ and $\delta_3 = 7$, then we get the partition $7 = 3 + 3 + 1$ ($s_1 = s_2 = 3, s_3 = 1$) and the chain diagrams below.



- **step 2.3:** For $j = 1, 2, \ldots, \delta_1$, choose the top vector of the jth chain, $\mathbf{v}_j^{(s_j)}$, such that the bottom of the chain, $\mathbf{v}_j^{(1)} = (A - \lambda I_n)^{s_j - 1}\mathbf{v}_j^{(s_j)}$, is nonzero and linearly independent from the eigenvectors chosen earlier. (These are the eigenvectors $\mathbf{v}_1^{(1)}, \mathbf{v}_2^{(1)}, \ldots, \mathbf{v}_{j-1}^{(1)}$.) Compute all the vectors in the chain, and add these generalized eigenvectors to P with k increasing. That is, P_λ is updated from $P_\lambda = [\ldots]$ to $P_\lambda = [\ldots\ \mathbf{v}_j^{(1)}\ \mathbf{v}_j^{(1)}\ \mathbf{v}_j^{(2)}\ \ldots \mathbf{v}_j^{(s_j)}]$

In our example, the resulting matrix is $P_\lambda = [\mathbf{v}_1^{(1)}\ \mathbf{v}_1^{(2)}\ \mathbf{v}_1^{(3)}\ \mathbf{v}_2^{(1)}\ \mathbf{v}_2^{(2)}\ \mathbf{v}_2^{(3)}\ \mathbf{v}_3^{(1)}]$ after step 2.3 is finished.

- **Step 3:** Construct the full, $n \times n$ matrix P by concatenating the P_λ computed for each eigenvalue.

Then, $B := P^{-1}AP$ is in JCF, and thus $\exp(Bt)$ is easily computed. The general solution to $\dot{\mathbf{x}} = A\mathbf{x}$ is $\mathbf{x}(t) = P\exp(Bt)\mathbf{c}$, and the solution to the IVP with $\mathbf{x}(0) = \mathbf{x}_0$ is $\mathbf{x}(t) = P\exp(Bt)P^{-1}\mathbf{x}_0$.

If some of the eigenvalues of A are complex and non-real, and A has real entries, then P will have complex entries but $\exp(At) = P\exp(Bt)P^{-1}$ will be real. On the other hand, the more geometrically intuitive form $\mathbf{x}(t) = P\exp(Bt)\mathbf{c}$ may give complex valued solutions if the vector of constants $\mathbf{c}$ is not chosen correctly. There is a link on my web site to "The general solution for a 4x4 matrix with repeated complex conjugate eigenvalues." describing how to deal with this.

Some examples using this algorithm are available as "ExamplesFromPerko.pdf" on my web site. The worked examples are from Section 1.8 in "Differential Equations and Dynamical Systems" by Lawrence Perko (Third Edition).

Confession

I (JWS) don't know of a proof that this algorithm works! But "Hey, what can go wrong?" The problem is that perhaps the generalized eigenvectors you obtain are not linearly independent. I should mention that if the matrix P you obtain by the algorithm is nonsingular then it *will* put A into JCF.

Bronson's algorithm in the Schaum's outline (first edition) says that in my step 2.3 we should choose $\mathbf{v}_j^{(s_j)}$ to be "linearly independent from all previously determined generalized eigenvectors associated with λ." That is potentially a lot more checking than I do, and it's not correct! See the scanned "Example 4 Done Incorrectly", the last page of "ExamplesFromPerko.pdf" on my web site. It gives and example where following Bronson's algorithm gives a linearly dependent set of generalized eigenvectors. It is my conjecture (as stated in step 2.3) that we only need to check that the bottom row of true eigenvectors is linearly independent.

Another example of what can go wrong is described in the note in the box of Example 3 in "ExamplesFromPerko.pdf". If we choose the size 1 chain first, and then the size 2 chain, we might not be able to get three linearly independent generalized eigenvectors. That is why both my algorithm and Bronson's tell us to choose the chains in order of decreasing size.