Algorithm for computing a Jordan basis (from Lecture 27).

Definition: Let V be a vector space, U a subspace of V and γ a subset of V. We will say that

(a) γ is *linearly independent mod* U if γ is linearly independent and $\mathrm{Span}(\gamma) \cap U = \{0\}$.

(b) γ is a basis for V/U if γ is linearly independent mod U and $V = \mathrm{Span}(\gamma) \oplus U$.

The following results are straightforward:

Proposition:

(i) If γ is a basis of V/U, it is linearly independent mod U

(ii) If γ is linearly independent mod U, then γ can be extended to a basis of V/U

(iii) Assume that $\dim(V) < \infty$ and γ is linearly independent mod U. Then γ is a basis of $V/U \iff |\gamma| = \dim(V) - \dim(U)$

(iv) If α is a basis of U and β is a basis of V which contains α, then $\beta \setminus \alpha$ is a basis of V/U.

Setup: Let V be a finite-dimensional vector space, $T \in \mathcal{L}(V)$, and assume that $\chi_T(x)$ splits. For each $\lambda \in Spec(T)$ let K_λ be the generalized eigenspace corresponding to λ. Below we describe an algorithm for computing a basis β_λ of K_λ s.t. $[T_{|K_\lambda}]_{\beta_\lambda}$ is in JCF. Then the ordered union $\sqcup_{\lambda \in Spec(T)} \beta_\lambda$ is a Jordan basis for T.

From now on we fix $\lambda \in Spec(T)$ and introduce the following notations:

- $D_{\lambda(T)}$ is the dot diagram of T corresponding to λ
- l is the largest size of a Jordan block in $JCF(T)$ corresponding to λ (which is equal to the number of rows in $D_{\lambda(T)}$)

- $K_\lambda(i) = \text{Ker}((T - \lambda I)^i)$. Thus we have

$$\{0\} = K_\lambda(0) \subset K_\lambda(1) \subset \ldots \subset K_\lambda(l) = K_\lambda.$$

- For $1 \leq i \leq k$ we put

$$n_\lambda(i) = \dim K_\lambda(i) - K_\lambda(i-1) =$$
$$\text{null}((T-\lambda I)^i) - \text{null}((T-\lambda I)^{i-1}) = \text{rk}((T-\lambda I)^{i-1}) - \text{rk}((T-\lambda I)^i).$$

  Thus, $n_\lambda(i)$ is the number of dots in the i^{th} row of $D_\lambda(T)$ and $\sum_{i=1}^{l} n_\lambda(i) = \dim K_\lambda$.

Algorithm:
Step 1: Choose a basis $w_{11}, \ldots, w_{1,m_1}$ for $K_\lambda(l)/K_\lambda(l-1)$. Note that $m_1 = n_\lambda(l) - n_\lambda(l-1)$. If $l = 1$, stop here.

Step 2: If $l > 1$, let $w_{2i} = (T - \lambda I)w_{1i}$ for $1 \leq i \leq m_1$. As proved in class, the set $\{w_{2i}\}_{i=1}^{m_1}$ lies in $K_\lambda(l-1)$ and is linearly independent mod $K_\lambda(l-2)$, so it can be extended to a basis $w_{21}, \ldots, w_{2m_2}$ (with $m_2 \geq m_1$) of $K_\lambda(l-1)/K_\lambda(l-2)$. Again note that $m_2 = n_\lambda(l-1) - n_\lambda(l-2)$. If $l = 2$, stop here.

Step 3: If $l > 2$, let $w_{3i} = (T - \lambda I)w_{2i}$ for $1 \leq i \leq m_2$ etc.
The algorithm will end at Step l.

We can arrange the obtained vectors $\{w_{ij}\}$ in a diagram as follows. Place the vectors $w_{11}, \ldots, w_{1,m_1}$ in the bottom row (from right to left). Then $w_{21}, \ldots, w_{2,m_2}$ in the next row, with $w_{2,i}$ directly above $w_{1,i}$ for $1 \leq i \leq m_1$ etc. Note that the diagram will have the same shape as the dot diagram $D_\lambda(T)$.

By construction, vectors in each column (to be read from top to bottom) form a nilpotent $(T-\lambda I)$-cycle. Also, by construction vectors in each row are linearly independent. In particular, this is true for vectors in the first row, the initial vectors of those nilpotent $(T-\lambda I)$-cycles, so by Lemma 18.3 all vectors in these cycles $\{w_{ij}\}$ are linearly independent. Finally, the total size of the set $\{w_{ij}\}$ is $\sum_{i=1}^{l}(n_\lambda(l+1-i)-n_\lambda(l-i)) = n_\lambda(l)-n_\lambda(0) = \dim(K_\lambda)$. Therefore, $\{w_{ij}\}$ is a Jordan basis for $T_{|K_\lambda}$ and we can set $\beta_\lambda = \{w_{ij}\}$.

Note that the order in which w_{ij} appear in β_λ matters – we need to list elements in each column consecutively from top to bottom (in the order opposite to how they were constructed). The order of columns is not essential (changing such order corresponds to permuting blocks in $JCF(T)$).

Below we see how the description of the above general algorithm can be simplified in some cases of small dimensions. We treat all three possible cases with $\dim(K_\lambda) = 3$ and one case (just as a sample) with $\dim(K_\lambda) = 4$. We represent dot diagrams by listing the lengths of columns, e.g. $D_\lambda(T) = 2+1+1$ will mean the diagram with two dots in the first column, one dot in the second and one dot in the third.

Example 1: $D_\lambda(T) = 1+1+1$ (one row of length 3). In this case $l = 1$, $n_\lambda(1) = 3$ and $K_1(\lambda) = K_\lambda = \mathrm{Ker}(T-\lambda I)$. Algorithm tells us to find a basis w_{11}, w_{12}, w_{13} of $K_1(\lambda)/K_0(\lambda)$ (which is the same as basis of K_λ since $K_0(\lambda) = \{0\}$) and stop there. This makes sense since $l = 1$ means that $K_\lambda = E_\lambda$ (the eigenspace), so any basis of E_λ will work as β_λ.

Example 2: $D_\lambda(T) = 3$ (one column of length 3). In this case $l = 3$ (so the algorithm will have three steps), $n_\lambda(1) = 1$, $n_\lambda(2) = 2$ and $n_\lambda(3) = 3$.

The first step of the algorithm tells us to find a basis w_{11} for $K_\lambda(3)/K_\lambda(2) = K_\lambda/K_\lambda(2)$. Since $\dim(K_\lambda(3)) - \dim(K_\lambda(2)) = n_\lambda(3) - n_\lambda(2) = 1$, any vector $w_{11} \in K_\lambda \setminus K_\lambda(2)$ will serve as such a basis.

Since $D_\lambda(T)$ has only one column, steps 2 and 3 amount to letting $w_{21} = (T-\lambda I)w_{11}$ and $w_{31} = (T-\lambda I)w_{21}$ (no bases extensions are needed).

Example 3: $D_\lambda(T) = 2+1$ (first column has length 2, second column has length 1). In this case $l = 2$, $n_\lambda(1) = 2$, $n_\lambda(2) = 3$.

The first step of the algorithm tells us to find a basis w_{11} for $K_\lambda(2)/K_\lambda(1) = K_\lambda/K_\lambda(1)$. As in Example 2, w_{11} can be any vector in $K_\lambda \setminus K_\lambda(1)$.

At the second step we let $w_{21} = (T-\lambda I)w_{11}$. Since $\dim(K_\lambda(1)) = 2$, we need one more vector w_{22} s.t. $\{w_{21}, w_{22}\}$ is a basis of $K_\lambda(1) = \mathrm{Ker}(T-\lambda I)$. Any $w_{22} \in \mathrm{Ker}(T-\lambda I) \setminus \mathrm{Span}(w_{21})$ will do the job.

Example 4: $D_\lambda(T) = 2 + 2$ (two columns of length 2). In this case $l = 2$, $n_\lambda(1) = 2$, $n_\lambda(2) = 4$.

The first step tells us to find a basis $\{w_{11}, w_{12}\}$ for $K_\lambda(2)/K_\lambda(1) = K_\lambda/K_\lambda(1)$. This time we need to work a little harder than in Examples 2 and 3 – according to Proposition (iv), what we can do is to find a basis $\{v_1, v_2\}$ of $K_\lambda(1) = \mathrm{Ker}(T-\lambda I)$ and then extend it to a basis $\{v_1, v_2, w_{11}, w_{12}\}$ of K_λ.

Since both columns hit the bottom row, at the second step we simply let $w_{21} = (T-\lambda I)w_{11}$ and $w_{22} = (T-\lambda I)w_{12}$. Note that $\{w_{21}, w_{22}\}$ is also a basis for $\mathrm{Ker}(T-\lambda I)$, but it may be completely different from the one we started with ($\{v_1, v_2\}$); the basis $\{v_1, v_2\}$ served an auxiliary role and has no significance for the final answer.

Note that in Example 4 there is an 'ad hoc' algorithm which in some cases will yield the answer faster: choose any basis $\{v_1, v_2\}$ of $\mathrm{Ker}(T-\lambda I)$, then find z_1, z_2 s.t. $(T-\lambda I)z_1 = v_1$ and $(T-\lambda I)z_2 = v_2$. Then $\{v_1, z_1, v_2, z_2\}$ is a Jordan basis for $T_{|K_\lambda}$. To justify this algorithm we need to explain two things:

(i) why $\{v_1, z_1, v_2, z_2\}$ is a basis and

(ii) why z_1 and z_2 with required properties can be found.

Condition (i) holds again by Lemma 18.3. To prove (ii) let $T_\lambda = (T-\lambda I)_{|K_\lambda}$. Then $T_\lambda^2 = 0$, so $\mathrm{Im}(T_\lambda) \subseteq \mathrm{Ker}(T_\lambda)$. We also know that $\dim \mathrm{Ker}(T_\lambda) = \dim \mathrm{Ker}(T-\lambda I) = 2$ and $\dim \mathrm{Im}(T_\lambda) = \dim(K_\lambda) - \dim \mathrm{Ker}(T_\lambda) = 4 - 2 = 2$. Combining the two facts, we get $\mathrm{Im}(T_\lambda) = \mathrm{Ker}(T_\lambda)$, which means that for any $v_1, v_2 \in \mathrm{Ker}(T_\lambda)$ the equations $T_\lambda(z_1) = v_1$ and $T_\lambda(z_2) = v_2$ can be solved for z_1 and z_2, which is precisely what we need for (ii).