Introduction to Gröbner Bases

Robert M. Corless

Fall, 1996

Contents

1 Key ideas from the last few weeks

1. A linear system of equations $Ax = b$ can be reduced to *row echelon form*, whence the solutions can just be read off.

2. Univariate polynomials have GCD's (Greatest Common Divisors) that can be computed by the Euclidean algorithm (though modular algorithms are significantly more efficient, and are what is actually implemented in most computer algebra systems).

3. An *ideal* I is a subset of a ring which is closed under addition and by multiplication from elements of the ring. We can think of the GCD of a pair of univariate polynomials as a useful generator (or basis) for the ideal generated by the original polynomials.

2 References for today's lecture

Much of the material for today has been taken from "Some examples of problem solving by using the symbolic viewpoint when dealing with polynomial systems of equations" by Laureano González-Vega, which I have in technical report form (from the University of Cantabria). The full reference is

Laureano Gonzalez-Vega, "Some Examples of Problem Solving by using the Symbolic Viewpoint when dealing with Polynomial Systems of Equations", in *Computer Algebra in Science and Engineering*, J. Fleischer, J. Grabmeier, F. W. Hehl, W. Küchlin, eds World Scientific, 1995, *pp.* 102–116.

Additional references include the wonderful book *Ideals, Varieties, and Algorithms*, by Cox, Little and O'Shea. It is in the library here at UWO, and has become one of my favourite texts. It is clear, coherent, deep and refreshing, as well as containing a lot of useful information. In particular, it has all the detailed proofs that I will be leaving out of these lectures.

Another useful reference is the book *Algorithms for Computer Algebra* by Geddes, Czapor, and Labahn. Chapter 10 is on Gröbner bases. Stephen Czapor wrote the `grobner` package in Maple.

3 Gröbner bases and systems of polynomial equations

I take it for granted that you are interested in solving systems of multivariate polynomial equations. Such systems arise in neural networks, chemical reactions, computer-aided design, singular point analysis of algebraic curves, and many others. Much computer algebraic manipulation is in fact simple polynomial system manipulation, which is why it is of intrinsic interest to computer algebra designers. Mathematica incorporates Gröbner bases directly into its simplification routines, for example.

We begin with a vague definition, and after some examples which should convince you that it is worthwhile to learn about Gröbner bases, I'll give a real definition.

3.1 Definitions

Temporary Working Definition: A *Gröbner basis* of a set of multivariate polynomials

$$F = \{f_i\}_{i=1}^m$$

where $f_i = f_i(x_1, x_2, \ldots, x_n)$ (note $m \neq n$ necessarily), is an equivalent but 'more convenient' set of multivariate polynomials

$$G = \{g_i\}_{i=1}^t$$

where t may not be the same as n or m, necessarily. We usually find Gröbner bases by use of computer algebra.

More details will be given on what we mean by 'equivalent' and 'more convenient', later.

Remark. Gröbner bases are computed with respect to a *monomial ordering*. That is, we need to be able to compare x^2y to xy or xy^2 and decide which is 'first'. There are several standard monomial orderings in the theory of Gröbner bases. The two maple uses are

1. `plex` or 'Pure Lexicographic' ordering. The standard name for this is simply 'lexicographic' order. It is familiar to everyone who can use a dictionary or phone book. Suppose we use the variables $x_,x_2, \ldots, x_n$ with the implicit ordering $x_1 > x_2 > \ldots > x_n$. This means x_1 is the 'most important' variable, say, down to x_n, the 'least important'. (These emotionally-charged words don't really mean anything, it's just a way of distinguishing). Of course if we start with x, y, z we can always rename so this ordering reflects our priorities (though often the choice of ordering among variables is arbitrary).

   Now consider the monomial term x^α, which we use as shorthand for $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$, and the monomial term x^β (which is a similar shorthand, of course). We say that $x^\alpha > x^\beta$ in the lexicographic order if the LEFTMOST nonzero entry in the vector $\alpha - \beta$ is POSITIVE. Thus $x^2yz^2 > xy^5z^3$ since $(2,1,2) - (1,5,3) = (1,-4,-1)$. So a lexicographic ordering counts powers of x as being more important than powers of other terms (making the identification $x_1 = x, x_2 = y, x_3 = z$, of course).

2. `tdeg` or 'total degree'. The standard name for this is 'graded reverse lexicographic order', and the emphasis is on the first two words. The 'graded' means that we consider total degree first: $x^\alpha > x^\beta$ if $\sum_{i=1}^n \alpha_i > \sum_{i=1}^n \beta_i$—that is, if the sum of the powers is larger, then that's the larger term. If we have a tie in powers (e.g. xy^2 vs x^2y) then we have to resort to a tie-breaking rule: it turns out to be useful to consider REVERSE lexicographic order, which says that $x^\alpha > x^\beta$ if the RIGHTMOST nonzero entry of $\alpha - \beta$ is NEGATIVE. Note the double-negative from pure lexicographic order, but it is the grading from the sums of powers that turns out to be most significant. Then $x^2yz^2 < xy^5z^3$ since the total degree of the second term is higher. Also, $xy^2z < x^2yz$ since $(1,2,1) - (2,1,1) = (-1,1,0)$.

We use only well-orderings: exactly one of $s > t$, $s = t$, or $s < t$ can hold. **Remark.** Lexicographic order gives the simplest answers, but it turns out to be (naturally) the hardest one to compute. There are other orderings than these ones, some of which are implemented in other computer algebra systems.

4 Recipes

We now look at some recipes for using Gröbner bases.

4.1 Recipe 0—Main property

The polynomial system F given by

$$f_1(x_1, \ldots, x_n) = 0, f_2(x_1, \ldots, x_n) = 0, \ldots, f_m(x_1, \ldots, x_n) = 0$$

has the same solutions as

$$g_1(x_1, \ldots, x_n) = 0, g_2(x_1, \ldots, x_n) = 0, \ldots, g_t(x_1, \ldots, x_n) = 0$$

where $G = \{g_i\}$ is any Gröbner basis of F with respect to any monomial ordering.

Example. If

$$\begin{aligned} x^3 + yx^2 - y^2 &= 0 \\ 2xy^2 + x^2 - y + 1 &= 0 \end{aligned}$$

then a Gröbner basis system for this polynomial system is (by Maple)

$$\begin{aligned} x + 4y^5 + 2y^4 + y^2 + y - 1 &= 0 \\ 8y^8 - 4y^7 + 4y^5 - 2y^4 - 3y^3 + 4y^2 - 3y + 1 &= 0. \end{aligned}$$

The reader can verify by Maple that every solution of the Gröbner basis is also a solution of the original system. It is a little harder to do an independent check that there are no solutions missed by this method, but one can use the theory of resultants to find a set of candidate solutions, for example, and if this is done (sure enough) we find that the solutions are indeed the same.

4.2 Recipe 1—existence of solutions

The polynomial system F has a solution if and only if the Gröbner basis with respect to any ordering is not the single polynomial $\{1\}$.

Example. Suppose $F = [x^3y - 2xy - 1, 3x + y^4x - 2xy, x^2 - y^2 + 1]$. Issuing the commands

```
with(grobner);
gbasis(F, [x, y], tdeg);
```

yields [1]. Hence the theory says there are no solutions to this system (well, you wouldn't expect any since there are three equations in 2 unknowns, and in fact this is easy to verify by hand in this example).

4.3 Recipe 2—existence of a finite number of solutions

We will see in the next sections that with a lex order basis it is easy to count the solutions explicitly (if there are a finite number). But how do we tell there are a finite number?

There are a finite number of solutions if and only if for each variable x_i there is a term t in the leading monomials of the Gröbner basis such that $t = c \cdot x_i^s$, that is, that at least one of the members of the Gröbner basis has its leading term as a pure power of x_i.

This gives us a simple and rational test for finite cardinality of solutions, given some Gröbner basis. Any monomial ordering may be used.

Example. If $F = [x^3y - 2xy + x, 3x + y^4x - 2y]$, then the total degree (graded reverse lexicographic or grevlex) Gröbner basis contains the polynomials $3x^4 - 2y^3 + 2x - 6x^2 + y^2$ and $2y^4 + 3x^2 - 2xy - y^3$. The reader should verify that the terms $3x^4$ and $2y^4$ are the highest-multidegree terms in these polynomials. Since they are pure powers, and the condition is satisfied for each variable, there are only a finite number of solutions.

4.4 Recipe 3—reduction to triangular form

This recipe uses the pure lexicographic order. It is a consequence of the fact that any power of x_1 outweighs any combination of the other variables, and so on recursively down to x_n, that *if the number of solutions is finite* then the Gröbner basis with respect to the lexicographic ordering has a special form:

$$
\begin{array}{rcl}
\omega_1^*(x_2,\ldots,x_n)x_1^{s_1} & + & \omega_1(x_1,x_2,\ldots,x_n) \\
\omega_2^*(x_3,\ldots,x_n)x_2^{s_2} & + & \omega_1(x_2,\ldots,x_n) \\
 & \vdots & \\
x_n^{s_n} & + & \omega_1(x_n)
\end{array}
$$

These equations are in *triangular* form. The last equation contains only x_n. The next-to-the last contains only x_{n-1} and x_n. Each equation as we move up contains only one more variable. This lets us count the number of solutions of the system, and in principle allows us to compute them. We solve the univariate polynomial for x_n, and then the next one for x_{n-1} in terms of each value of x_n we found, and so on up the list to x_1.

In practice there are numerical difficulties. This approach is a generalization of the characteristic polynomial approach to finding eigenvalues, which we know to be unstable. It is a generalization of the Row Echelon Form, which we know can be sensitive to errors (if the matrix is ill-conditioned). It is a generalization of the Euclidean algorithm, which can also be unstable. We will see later (time permitting) a linear algebraic approach that offers some slightly better hope of numerical stability.

4.5 Recipe 4—the Shape Lemma

In practice, the lex order Gröbner bases for many problems have the following shape. (I don't give the technical conditions that ensure this because (a) I don't know them, and (b) it doesn't matter anyway, just compute the Gröbner basis and see if it has the shape. The theorem says that it will usually happen).

$$
\begin{array}{rcl}
g(x_n) & = & 0 \\
x_{n-1} & = & g_{n-1}(x_n) \\
x_{n-2} & = & g_{n-2}(x_n) \\
 & \vdots & \\
x_1 & = & g_1(x_n)
\end{array}
$$

That is, once x_n is determined, all the other variables are determined by evaluation of polynomials at x_n. This simple shape is quite convenient to work with, on occasion.

Example. Eigenvalue problems. Eigenvalue problems with simple eigenvalues (i.e. no repeated roots) lead to Gröbner bases which have the specified shape. That is, we start with the polynomial system $Ax = \lambda x$ in the variables $x_1, x_2, \ldots, x_n, \lambda$. Then our ordinary analysis gives the equivalent system $p(\lambda) = 0$ from the characteristic polynomial, and once λ is identified we solve the linear system $(A - \lambda I)x = 0$ for the x's. This gives (on first blush) a solution x_i = rational function of λ (e.g. by Cramer's rule on a nonsingular submatrix), but remember that if $p(\lambda) = 0$ we can transform any rational function of λ into a polynomial in λ by simplification (e.g. $1/\lambda = -1/p_0(p_1 + p_2\lambda + \cdots p_n\lambda^{n-1})$). (I'm not certain simplicity of the eigenvalues is required).

This is clearly a Gröbner basis, and has the required shape.

5 Recipe 5—Linear Algebra Methods

To discuss these I must first discuss the normal form reduction (`normalf` in the `grobner` package in Maple). This is essentially the multivariate version of taking remainders (except that we take the remainder with respect to an ideal, instead of with respect to a single polynomial).

5.1 Some more definitions

1. The **multidegree** of $f = \sum a_\alpha x^\alpha$ is

$$\text{multideg}(f) = \max(\alpha \in \mathbb{Z}^n_{\geq 0} : a_\alpha \neq 0)$$

where the max is taken with respect to the monomial ordering chosen.

2. The **leading coefficient** of f is $\text{LC}(f) = a_{\text{multideg}(f)}$, which is in the base field K of coefficients.

3. The **leading monomial** of f is $\text{LM}(f) = x^{\text{multideg}(f)}$.

4. The **leading term** of f is $\text{LT}(f) = \text{LC}(f) \cdot \text{LM}(f)$.

5.2 Multivariate division by an ideal

We now discuss multivariate division and remainders. For a more complete discussion see Cox, Little, and O'Shea. We consider only the case when we are dividing by an ideal for which we have a Gröbner basis (otherwise the remainder is not unique).

Theorem 3: Division Algorithm in $K = k[x_1, \ldots, x_n]$. (Cox, Little, O'Shea p. 63). Fix a monomial order $>$ on $\mathbb{Z}^n_{\geq 0}$, and let $F = (f_1, \ldots, f_s)$ be an ordered s-tuple of polynomials in K. Then every f in K can be written as

$$f = a_1 f_1 + \cdots + a_s f_s + r$$

where $a_i, r \in K$, and either $r = 0$ or r is a k-linear combination of monomials, none of which is divisible by any of $\text{LT}(f_1)$, …, $\text{LT}(f_s)$. We will call r a **remainder** of f on division by F. Furthermore, if $a_i f_i \neq 0$, then we have

$$\text{multideg}(f) \geq \text{multideg}(a_i f_i)\ .$$

For the algorithm which constructs the a_i and r, and proof of its correctness, see Cox, Little, and O'Shea. Note that unless F is a Gröbner basis, the remainder r and quotients will *not be unique*. If, on the other hand, F is a Gröbner basis, then the remainder (but not the quotients) will be unique, and not depend on the order of polynomials in the basis. (This is proposition 1 in Cox, Little, O'Shea, p. 81.)

Remark. The fact that if r is not zero then it is a k-linear combination of 'small' monomials is crucial to the linear algebraic aspect of Gröbner bases. We give an example here.

Example Suppose $f_1 = x_1^2 x_2 - 2x_1^2 + x_2^2 + x_1 x_2$ and $f_2 = 2x_1^2 - x_2^2 + x_1 x_2$. This is not a Gröbner basis, but we can find using Maple that $G = [g_1, g_2, g_3]$ is a (grevlex or 'tdeg') Gröbner basis for F, where $g_1 = 2x_1^2 - x_2^2 + x_1 x_2$, $g_2 = x_1 x_2^2 - x_2^3 - 4x_1 x_2$, and $g_3 = x_2^4 + 6x_2^3 + 16 x_1 x_2$. Using Maple again, with the `normalf` routine, we find that the remainder of dividing (say) $f = 2x_1^4 x_2 - x_2^2 + x_1^3 + x_1 + 1$ by this ideal is

```
normalf(2*x1**4*x2 - x2**2 + x1**3 + x1 + 1,
        GB, [x1,x2], tdeg);
```

which gives $(1/2)x_2^3 - 13x_1 x_2 - x_2^2 + x_1 + 1$. One important point is that none of the terms in this remainder are divisible by the leading terms of the Gröbner basis (namely x_1^2, $x_1 x_2^2$, or x_2^4). Another is that the algorithm produces a unique remainder, no matter what ordering is chosen for the basis.

Finally, notice that using this remainder we can define an equivalence class in K, namely $f \equiv g$ if the remainder on dividing f by G is the same as the remainder on dividing g by G. This gives us the quotient ring K/I (see Cox, Little, and O'Shea for the details verifying that this all makes sense). We observe, then, that the set of monomials which are not divisible by the leading terms of the Gröbner basis form a basis for this quotient ring. This is what we explore, in section (5.4). But first, we take advantage of all this notation to give a real definition of a Gröbner basis, and describe Buchberger's algorithm for computing one.

5.3 S-polynomials and Buchberger's algorithm

We need two more definitions (Definition 4, p. 82 of Cox, Little, and O'Shea).

1. If $\text{multideg}(f) = \alpha$ and $\text{multideg}(g) = \beta$, then let $\gamma = (\gamma_1, \ldots, \gamma_n)$ where $\gamma_i = \max(\alpha_i, \beta_i)$ for each i. We call x^γ the **least common multiple** of $\text{LM}(f)$ and $\text{LM}(g)$. We write $x^\gamma = \text{LCM}(\text{LM}(f), \text{LM}(g))$.

2. The **S-polynomial** of f and g is the combination

$$S(f, g) = \frac{x^\gamma}{\text{LT}(f)} \cdot f - \frac{x^\gamma}{\text{LT}(g)} \cdot g\ .$$

(Note that we are inverting the leading coefficients here as well.)

For example, let $f = x^3 y^2 - x^2 y^3 + x$ and $g = 3x^4 y + y^2$ with the `tdeg` (graded reverse lexicographic) order. Then

$\gamma = (4, 2)$ and

$$\begin{aligned} S(f,g) &= \frac{x^\gamma}{\mathrm{LT}(f)} \cdot f - \frac{x^\gamma}{\mathrm{LT}(g)} \cdot g \\ &= \frac{x^4y^2}{x^3y^2} \cdot f - \frac{x^4y^2}{3x^4y} \cdot g \\ &= x \cdot f - \frac{1}{3} y \cdot g \\ &= -x^3y^3 + x^2 - \frac{1}{3} y^3 . \end{aligned}$$

The idea of an S-polynomial is that *leading terms are cancelled*. It turns out that every kind of cancellation can be expressed as a sequence of these S-polynomial combinations.

Definition of a Gröbner Basis (Theorem 6 in Cox, Little, and O'Shea, p. 84.)

Let I be a polynomial ideal. Then a basis $G = \{g_1, \ldots, g_t\}$ for I is a Gröbner basis for I if and only if for all pairs $i \neq j$, the remainder on division of $S(g_i, g_j)$ by G (listed in some order) is zero.

This is a computationally verifiable definition—to test if a given basis is a Gröbner basis, form all $\binom{t}{2}$ S-polynomials, and compute the remainders modulo G. If any of these remainders are nonzero, it isn't a Gröbner basis.

5.3.1 Buchberger's Algorithm

The closure of Gröbner bases with respect to S-polynomials suggests the simple algorithm of computing $S(f_1, f_2)$, taking its remainder, and if it is not zero, add it to the basis (as it is a combination of ideal elements, it is certainly in the ideal). Amazingly, this works—since we might be adding a polynomial every time, we have to check that eventually the process terminates, and indeed it does. This is Buchberger's algorithm, in its simplest and most expensive form.

Remark. When I attended ISSAC '89 in Portland, Oregon, which was my first ISSAC, I spent some time socializing. (This is only to be expected by those who know me). Several notables, including David Stoutemyer (the principal author of Derive), Fritz Schwarz, and James Davenport (who is the author of a fundamental monograph on the Risch integration algorithm), and I went to sample the fine products of the local microbreweries. We found one quite close to the hotel that had a very nice selection of inexpensive but tasty beverages, and had a genteel discussion. Several jugs later, Fritz Schwarz somehow had asked James Davenport about the proof of finiteness of Buchberger's algorithm. Davenport claimed that he could prove it in two lines, given the statement of the theorem. General disbelief followed—but he did it, then and there, on a napkin. General hilarity on noticing not two, but *three* lines. Chagrin, on Davenport's side—until he noticed and pointed out to all of us that the first line merely stated what was to be proved! I still have the napkin, by the way.

Example. We prove nothing here (the proof in Cox, Little and O'Shea is essentially the same as the one on my napkin), but give instead a simple example. We use the grevlex order (`tdeg`) for the ideal $I =< x^3 - 2xy, x^2y - 2y^2 + x >$.

$S(f_1, f_2) = -x^2$, and the remainder on division by I is again just $-x^2$. So we add this to the list, and call it f_3.

Now $S(f1, f2) = f_3$, and so the remainder on division by I is 0. But $S(f1, f3) = (x^3 - 2xy) - (-x)(-x^2) = -2xy$, which on division by I gives $-2xy$, not zero. So we add $-2xy$ to the basis, and call it f_4. Now $F = (f_1, f_2, f_3, f_4)$. We find as before that $S(f_1, f_2)$ and $S(f_1, f_3)$ reduce to zero. Also, $S(f_1, f_4) = y(x^3 - 2xy) - (-1/2)x^2(-2xy) = -2xy^2 = yf_4$ so on division by F we get 0.

Now $S(f_2, f_3) = (x^2y - 2y^2 + x) - (-y)(-x^2) = -2y^2 + x$, which has remainder $-2y^2 + x$, not zero. So we add this to the basis, and call it f_5. Now when we compute all $\binom{5}{2} = 10$ S-polynomials and reduce them modulo F we get zero in all cases. Thus we have found that

$$F = \{x^3 - 2xy, x^2y - 2y^2 + x, -x^2, -2xy, -2y^2 + x\}$$

is a Gröbner basis for I.

We now continue with the final recipe.

5.4 The vector space of reduced polynomials

For simplicity let us assume that the set of monomials not divisible by the leading term of any element of G is finite, as in our first example (where the monomials were $A = [1, x_2, x_2^2, x_2^3, x_1, x_1x_2]$). In this case we say the quotient ring is finite-dimensional, and this corresponds to a 'zero-dimensional' variety (it just means that there are only a finite set of solutions to the polynomial system).

Then we can express any given f and g, modulo the ideal, as say r_f and r_g respectively. It follows immediately that the representation of $f + g$ is $r_f + r_g$ and somewhat less immediately that the representation of fg is the remainder on division by I of $r_f * r_g$.

This produces a *linear vector space* as is easily verified. Let us consider the effects of multiplying by x_1 in this vector space.

Suppose $r = a_1\omega_1 + a_2\omega_2 + \cdots a_n\omega_s$, where the ω_i are the monomials in A and the a_i are coefficients. Suppose also that $x_1\omega_i$ reduces (on division by I) to $\alpha_{i1}\omega_1 + \alpha_{i2} +$

$\cdots \alpha_{is}\omega_s$, for $i = 1$ to s. Then it is easily seen that

$$x_1 \cdot r \bmod I = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1s} \\ \alpha_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \alpha_{s1} & \cdots & \cdots & \alpha_{ss} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_s \end{bmatrix}$$

That is, multiplication by x_1 acts on the space spanned by these monomials in a linear way, which we can represent by a matrix.

Obviously there's nothing special about x_1, and each variable is represented by such a matrix. Note that since multiplication by x_i and x_j in the polynomial ring *commutes*, these matrices P_{x_i} and P_{x_j} must also commute.

It should also be clear that knowledge of the action of these multiplication matrices should tell us everything possible about the ideal, in some sense. So the following miracle is actually explainable.

5.5 Roots of the system are eigenvalues of a commuting family

Theorem (Horn and Johnson, Matrix Algebra); If P is a commuting family of matrices, then there exists a unitary matrix U which simultaneously upper- triangularizes every member of P: $U^* P_{x_i} U = \operatorname{diag}(\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{is})$.

Observation. The roots of the original system are these associated eigenvalues: $r_1 = (\lambda_{11}, \lambda_{21}, \ldots, \lambda_{n1})$, etc. This observation is astonishing, to me; it is essentially a generalization of the *companion matrix* method for finding roots of univariate polynomials (which is how Matlab finds roots of polynomials).

To see that the eigenvalues give the roots of the polynomial system, let us work in the basis given by the matrix U (for simplicity let's assume all the roots are distinct). Then multiplication of a basis vector u_j by x_i gives us $\lambda_{ij}u_j$; multiplication of this by x_k gives $\lambda_{kj}\lambda_{ij}u_j$; addition of a constant times u_j gives the value $(\lambda_{kj}\lambda_{ij} - c)u_j$. Now u_j is, in the original basis, the vector of linear coefficients of a polynomial (call it $u_j(x)$). We now see that $f(x_1, x_2, \ldots, x_n)u_j(x)$ reduces to $f(\lambda_{1j}, \lambda_{2j}, \ldots, \lambda_{nj})u_j$. If we take $f \in I$, for example one of the Gröbner basis elements, then we know that $fu_j(x) \in I$ so it reduces to zero. Therefore $f(\lambda_{1j}, \lambda_{2j}, \ldots, \lambda_{nj})u_j = 0$, and by the linear independence of the basis we must have that the eigenvalues are roots of f.

(I don't see an easy way to show that all roots are eigenvalues, at this moment).

I don't wish to describe this algebra here, but I refer you to my paper in the proceedings of the ISSAC '95 conference in Montréal for a similar analysis, and to L. González-Vega's paper, and to the references contained therein. We content ourselves here with an example.

(Note that there is no package to do this automatically in Maple. I think that this would make an excellent project, now that I come to think of it).

We end with one last observation, and that is if the roots are distinct, then we can find this matrix U in a reasonably efficient way by finding the eigenvectors of *one* of these multiplication matrices; the matrix of eigenvectors then gives us our U.

For the example just given, the matrices are

$$P_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1 & -2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 0 & -1/2 & 2 \end{bmatrix}$$

and

$$P_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -6 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -16 & 1 & 4 \end{bmatrix}.$$

Then the eigenvalues are $[0,0,0,0,-2,-2]$ and $[0,0,0,0,2,-4]$, and the desired roots of the system are $(0,0)$, $(-2,2)$, and $(-2,-4)$. The multiple root at zero makes finding the unitary matrix U explicitly quite difficult, but in this case we can sort out which eigenvalue goes with which by simple elimination (because the problem is so small).