A Geometric View of the Quotient Algebra
How Eigenvalues Solve Polynomial Equation Systems

Lars Hellström
Lars.Hellstrom@mdh.se

MAA155, 2020-05-19
The problem

Given a system of polynomial equations, with a finite number of solutions, find these solutions!

Variation: find just those solutions you’re interested in!
The traditional method

Mathematical folklore says that one should use Gröbner bases for this. The folklore often omits to say exactly how.

The traditional method is an elimination method: compute a Gröbner basis which eliminates variables, thereby reducing to a single variable problem.

This is a direct generalisation of Gaussian elimination for solving linear equation systems.
The traditional method

Mathematical folklore says that one should use Gröbner bases for this. The folklore often omits to say exactly how.

The traditional method is an elimination method: compute a Gröbner basis which eliminates variables, thereby reducing to a single variable problem.

This is a direct generalisation of Gaussian elimination for solving linear equation systems.
Problems with elimination

There are (at least) two practical problems with the elimination method:

1. Back-substitution is trickier for nonlinear systems: there is a much greater risk of numerical instability and one must test for false roots.

2. Elimination order Gröbner bases tend to be more (sometimes much more) expensive to compute than total degree order Gröbner bases.
Elimination cost example (slightly silly)

Consider the set of polynomials

\[ \{ x_i^2 - x_i \}_{i=1}^n \cup \left\{ y - \sum_{i=1}^n 2^{i-1}x_i \right\} \]

With respect to the monomial order

\[ x_1^2 > \cdots > x_n^2 > y > x_1 > \cdots > x_n \]

this is already a Gröbner basis.

But with respect to a monomial order

\[ x_1^2 > x_1 > x_2^2 > x_2 > \cdots > x_n^2 > x_n > y \]

for eliminating the \( x_i \) variables, the Gröbner basis must contain the polynomial

\[ \prod_{k=0}^{2^n-1} (y - k) \]

of degree \( 2^n! \).

Experience indicates that things like this tend to happen even in real life examples, when you pick an elimination order.
Buchberger in context

In algebra textbooks, the *Buchberger algorithm* for computing Gröbner bases is often presented as something very special, or possibly as the grand unification of several elementary algorithms (Gaussian elimination and Euclid’s algorithm, to mention two), but it is more illuminating to present it as a special case of Critical Pairs Completion, which is one of the fundamental algorithm schemata in *rewriting*. 
Rewriting is the formalisation of the mathematician’s activity of rewriting formulae. It needs two things:

1. A set of objects to rewrite. These are often general expressions relevant to the problem at hand. In the case treated in classical Gröbner bases theory, the basic objects can be taken to be polynomials over some field $\mathcal{K}$.

2. A set of rewrite rules that may transform one object into another object. That the rule $s$ transforms $l$ into $r$ is written $l \overset{s}{\rightarrow} r$.

In the equational reasoning branch of rewriting, the rationale for a rule $l \rightarrow r$ is that $l$ and $r$ are supposed to be equivalent, but $r$ is in some sense simpler than $l$. 
Gröbner bases and rewrite systems

The way a polynomial is interpreted as a rewrite rule depends on the ordering. A polynomial

\[ a\mu - \sum_{i=0}^{m} b_i\nu_i \quad \text{where} \quad a, b_i \in \mathcal{K}, \ a \neq 0, \ \text{and} \ \mu > \nu_i \ \text{are monomials} \]

gives rise to a rule

\[ \mu\lambda \rightarrow \frac{1}{a} \sum_{i=0}^{m} b_i\nu_i\lambda \]

for every monomial \( \lambda \); monomials not on the form \( \mu\lambda \) are left unchanged by this rule. The rules are then extended by linearity to act on arbitrary polynomials.

This rewrite system can be viewed as executing (generalised) division with respect to a set of polynomials.
Normal forms and confluence

An object which no rewrite rule can change is said to be on normal form.

A rewrite system is said to be terminating if every sequence of rewrite steps eventually stops at a normal form.

A rewrite system is said to be confluent if every object has exactly one normal form to which it can be mapped by a finite sequence of rewrite steps.

If a rewrite system is terminating and confluent—if the corresponding set of polynomials constitute a Gröbner basis—then one can effectively test whether two objects are equivalent by checking whether their normal forms are equal. This solves the ideal membership problem.
Critical Pairs

To prove that a terminating rewrite system is confluent, one only has to check that every expression that can be acted upon by rewrite rules in two different ways

\[ r_1 \xleftrightarrow{s_1} l \xrightarrow{s_2} r_2 \]

is such that the two preliminary results rewrite to a common normal form

\[ r_1 \rightarrow \cdots \rightarrow n \leftarrow \cdots \leftarrow r_2 \]

(the diamond condition).

Much detail work in specific theories goes into shrinking the sets of \( r_1 \xleftarrow{s_1} l \xrightarrow{s_2} r_2 \) instances that need to be checked; these are varyingly known as critical pairs, overlaps, ambiguities, and other things still. Critical pairs is the most common, but some define the critical pair to be \((s_1, s_2)\) and others to be \((r_1, r_2)\).
Completion

The completion procedure consists of doing the confluence check, and whenever something \textit{fails} you \textit{add a new rule} (new polynomial) that causes one of the purported normal forms to rewrite to the other one.

This, of course, can create new cases that need to be checked, so completion need not terminate in general. But in the commutative algebra case, Buchberger proved that it eventually terminates, so in that case this \textit{is} an algorithm.

The computational complexity can however be rather bad; the worst-case bounds are \textit{doubly exponential} in the number of variables, and there are examples which attain that bound. The poor complexity is not unexpected. It is trivial to encode an NP problem as an ideal membership problem, so computing a Gröbner basis has to be \textit{at least} NP-hard in general. (In fact is is PSPACE-hard.)
Completion

The completion procedure consists of doing the confluence check, and whenever something fails you add a new rule (new polynomial) that causes one of the purported normal forms to rewrite to the other one.

This, of course, can create new cases that need to be checked, so completion need not terminate in general. But in the commutative algebra case, Buchberger proved that it eventually terminates, so in that case this is an algorithm.

The computational complexity can however be rather bad; the worst-case bounds are doubly exponential in the number of variables, and there are examples which attain that bound. The poor complexity is not unexpected. It is trivial to encode an NP problem as an ideal membership problem, so computing a Gröbner basis has to be at least NP-hard in general. (In fact it is PSPACE-hard.)
Completion

The completion procedure consists of doing the confluence check, and whenever something fails you add a new rule (new polynomial) that causes one of the purported normal forms to rewrite to the other one.

This, of course, can create new cases that need to be checked, so completion need not terminate in general. But in the commutative algebra case, Buchberger proved that it eventually terminates, so in that case this is an algorithm.

The computational complexity can however be rather bad; the worst-case bounds are doubly exponential in the number of variables, and there are examples which attain that bound. The poor complexity is not unexpected. It is trivial to encode an NP problem as an ideal membership problem, so computing a Gröbner basis has to be at least NP-hard in general.

(In fact is is PSPACE-hard.)
Completion

The completion procedure consists of doing the confluence check, and whenever something fails you add a new rule (new polynomial) that causes one of the purported normal forms to rewrite to the other one.

This, of course, can create new cases that need to be checked, so completion need not terminate in general. But in the commutative algebra case, Buchberger proved that it eventually terminates, so in that case this is an algorithm.

The computational complexity can however be rather bad; the worst-case bounds are doubly exponential in the number of variables, and there are examples which attain that bound. The poor complexity is not unexpected. It is trivial to encode an NP problem as an ideal membership problem, so computing a Gröbner basis has to be at least NP-hard in general. (In fact is is PSPACE-hard.)
A shift of focus

The Gröbner basis and general rewriting machineries are just two different formalisms for the same thing, but they do place the focus slightly differently:

- Gröbner bases suggest focusing on the equations.
- Rewrite rules suggest focusing on the objects being rewritten. What do these represent?

In general, identifying equivalent objects amounts to forming a quotient. The normal forms are distinguished representatives of the equivalence classes in this quotient.
The quotient

The quotient $\mathcal{K}[X]/\mathcal{I}$ of a polynomial ring $\mathcal{K}[X]$ by an ideal $\mathcal{I}$ is a $\mathcal{K}$-algebra.

As a vector space, $\mathcal{K}[X]/\mathcal{I}$ is isomorphic to the vector space of normal forms. In particular, it has a basis consisting of all monomials that are normal forms (not divided by the leading monomials of any element in the Gröbner basis of $\mathcal{I}$).

The rewrite rules tell us how to calculate in the quotient: first do the operations as in $\mathcal{K}[X]$, then apply rewrite rules to find the normal form of the result.

We can easily compute multiplication tables for $\mathcal{K}[X]/\mathcal{I}$ with respect to this normal form basis.

And this works equally well for any Gröbner basis of $\mathcal{I}$. 
The quotient

The quotient $\mathcal{K}[X]/\mathcal{I}$ of a polynomial ring $\mathcal{K}[X]$ by an ideal $\mathcal{I}$ is a $\mathcal{K}$-algebra.

As a vector space, $\mathcal{K}[X]/\mathcal{I}$ is isomorphic to the vector space of normal forms. In particular, it has a basis consisting of all monomials that are normal forms (not divided by the leading monomials of any element in the Gröbner basis of $\mathcal{I}$).

The rewrite rules tell us how to calculate in the quotient: first do the operations as in $\mathcal{K}[X]$, then apply rewrite rules to find the normal form of the result.

We can easily compute multiplication tables for $\mathcal{K}[X]/\mathcal{I}$ with respect to this normal form basis.

And this works *equally well* for any Gröbner basis of $\mathcal{I}$. 
The quotient

The quotient $\mathcal{K}[X]/\mathcal{I}$ of a polynomial ring $\mathcal{K}[X]$ by an ideal $\mathcal{I}$ is a $\mathcal{K}$-algebra.

As a vector space, $\mathcal{K}[X]/\mathcal{I}$ is isomorphic to the vector space of normal forms. In particular, it has a basis consisting of all monomials that are normal forms (not divided by the leading monomials of any element in the Gröbner basis of $\mathcal{I}$).

The rewrite rules tell us how to calculate in the quotient: first do the operations as in $\mathcal{K}[X]$, then apply rewrite rules to find the normal form of the result.

We can easily compute multiplication tables for $\mathcal{K}[X]/\mathcal{I}$ with respect to this normal form basis.

And this works equally well for any Gröbner basis of $\mathcal{I}$. 
The quotient

The quotient $\mathcal{K}[X]/\mathcal{I}$ of a polynomial ring $\mathcal{K}[X]$ by an ideal $\mathcal{I}$ is a $\mathcal{K}$-algebra.

As a vector space, $\mathcal{K}[X]/\mathcal{I}$ is isomorphic to the vector space of normal forms. In particular, it has a basis consisting of all monomials that are normal forms (not divided by the leading monomials of any element in the Gröbner basis of $\mathcal{I}$).

The rewrite rules tell us how to calculate in the quotient: first do the operations as in $\mathcal{K}[X]$, then apply rewrite rules to find the normal form of the result.

We can easily compute multiplication tables for $\mathcal{K}[X]/\mathcal{I}$ with respect to this normal form basis.

And this works equally well for any Gröbner basis of $\mathcal{I}$. 
The quotient, geometrically

Geometrically, the quotient $\mathcal{K}[X]/\mathcal{I}$ is the coordinate ring of the variety $V(\mathcal{I})$.

The right intuition is that the quotient $\mathcal{K}[X]/\mathcal{I}$ is the set of functions $V(\mathcal{I}) \rightarrow \mathcal{K}$, but one must make some reservations:

1. If $\mathcal{K}$ is not algebraically closed, then the quotient may pick up points outside $\mathcal{K}^{|X|}$. (We’ll see how shortly.)

2. It can also happen that some degrees of freedom are used to keep track of the value of (partial) derivatives at some points. (This will also become apparent.)

3. If the quotient is not finite-dimensional then the variety can be infinite, and then one needs to restrict the kind of functions that may appear: the coordinate ring is generated by the coordinate functions. (But I’m not considering that case here; it’s what textbooks tend to be concerned about.)
The quotient, geometrically

Geometrically, the quotient $\mathcal{K}[X]/\mathcal{I}$ is the coordinate ring of the variety $V(\mathcal{I})$.

The right intuition is that the quotient $\mathcal{K}[X]/\mathcal{I}$ is the set of functions $V(\mathcal{I}) \rightarrow \mathcal{K}$, but one must make some reservations:

1. If $\mathcal{K}$ is not algebraically closed, then the quotient may pick up points outside $\mathcal{K}|_{\{X\}}$. (We’ll see how shortly.)

2. It can also happen that some degrees of freedom are used to keep track of the value of (partial) derivatives at some points. (This will also become apparent.)

3. If the quotient is not finite-dimensional then the variety can be infinite, and then one needs to restrict the kind of functions that may appear: the coordinate ring is generated by the coordinate functions. (But I’m not considering that case here; it’s what textbooks tend to be concerned about.)
The quotient, geometrically

Geometrically, the quotient $\mathcal{K}[X]/\mathcal{I}$ is the coordinate ring of the variety $V(\mathcal{I})$.

The right intuition is that the quotient $\mathcal{K}[X]/\mathcal{I}$ is the set of functions $V(\mathcal{I}) \rightarrow \mathcal{K}$, but one must make some reservations:

1. If $\mathcal{K}$ is not algebraically closed, then the quotient may pick up points outside $\mathcal{K}|^X$. (We’ll see how shortly.)

2. It can also happen that some degrees of freedom are used to keep track of the value of (partial) derivatives at some points. (This will also become apparent.)

3. If the quotient is not finite-dimensional then the variety can be infinite, and then one needs to restrict the kind of functions that may appear: the coordinate ring is generated by the coordinate functions. (But I’m not considering that case here; it’s what textbooks tend to be concerned about.)
The quotient, geometrically

Geometrically, the quotient $\mathcal{K}[X]/\mathcal{I}$ is the coordinate ring of the variety $V(\mathcal{I})$.

The right intuition is that the quotient $\mathcal{K}[X]/\mathcal{I}$ is the set of functions $V(\mathcal{I}) \rightarrow \mathcal{K}$, but one must make some reservations:

1. If $\mathcal{K}$ is not algebraically closed, then the quotient may pick up points outside $\mathcal{K}|^X$. (We’ll see how shortly.)

2. It can also happen that some degrees of freedom are used to keep track of the value of (partial) derivatives at some points. (This will also become apparent.)

3. If the quotient is not finite-dimensional then the variety can be infinite, and then one needs to restrict the kind of functions that may appear: the coordinate ring is generated by the coordinate functions. (But I’m not considering that case here; it’s what textbooks tend to be concerned about.)
Bounding the cardinality of the variety

An immediate corollary of the above is that

$$|V(I)| \leq \dim_K(K[X]/I).$$

**Example**

Q: How many points of intersection can there be between two cubic splines

$$P(t) = (1 - t)^3P_0 + 3(1 - t)^2tP_1 + 3(1 - t)t^2P_2 + t^3P_3,$$

$$Q(s) = (1 - s)^3Q_0 + 3(1 - s)^2sQ_1 + 3(1 - s)s^2Q_2 + s^3Q_3$$

where $P_0, P_1, P_2, P_3, Q_0, Q_1, Q_2, Q_3 \in \mathbb{R}^2$?
A: 9, like so:

To see that this is the maximum, consider the corresponding coordinate ring. $P(t) = Q(s)$ is two equations of the form

$$a_3t^3 + a_2t^2 + a_1t + a_0 = b_3s^3 + b_2s^2 + b_1s + b_0$$

so typically the Gröbner basis (with respect to e.g. degree-lexicographic order) will have one element with leading monomial $t^3$ and another with leading monomial $s^3$. Hence the normal form monomials are

$$1, s, s^2, t, st, s^2t, t^2, st^2, s^2t^2$$

and thus the coordinate ring is 9-dimensional; there can be at most 9 points in the variety.
A: 9, like so:

To see that this is the maximum, consider the corresponding coordinate ring. \( P(t) = Q(s) \) is two equations of the form

\[
a_3 t^3 + a_2 t^2 + a_1 t + a_0 = b_3 s^3 + b_2 s^2 + b_1 s + b_0
\]

so typically the Gröbner basis (with respect to e.g. degree-lexicographic order) will have one element with leading monomial \( t^3 \) and another with leading monomial \( s^3 \). Hence the normal form monomials are

\[
1, s, s^2, t, st, s^2 t, t^2, st^2, s^2 t^2
\]

and thus the coordinate ring is 9-dimensional; there can be at most 9 points in the variety.
Concretely, this example has

\[ P(t) = \left( 12t - 33t^2 + 22t^3, \frac{1}{10} - \frac{3}{10}t + \frac{33}{10}t^2 - \frac{11}{5}t^3 \right), \]

\[ Q(s) = \left( \frac{1}{10} - \frac{3}{10}s + \frac{33}{10}s^2 - \frac{11}{5}s^3, \frac{123}{10}s - \frac{168}{5}s^2 + \frac{223}{10}s^3 \right) \]

(These two S-shaped curves are almost mirror images of each other, but not quite, since if they were mirror images it would be possible to use ad hoc symmetry arguments to find the intersections.)
The choice of vector space basis

The awkward thing about the monomial basis description of the coordinate ring is however that it *completely mixes information about different points*; in general no element of the vector space basis is providing local information about just one point.

But we do have a method of discovering vector space bases that are better for analysing a given problem, namely that of finding the *eigenvectors*.
Eigenvectors

Let $x \in X$ be an arbitrary variable. We can define a multiply-by-$x$ operator $M_x : \mathcal{K}[X]/\mathcal{I} \rightarrow \mathcal{K}[X]/\mathcal{I}$ by $M_x f = (x + \mathcal{I})f$.

Now what does it mean for $f$ to be an eigenvector of $M_x$?

Viewing $f$ as a function $V(\mathcal{I}) \rightarrow \mathcal{K}$, the answer is obvious: $M_x f = \lambda f$ iff $f(p) = 0$ for all $p \in V(\mathcal{I})$ where the $x$-coordinate is $\neq \lambda$.

Or shorter, $x(p) \neq \lambda$ implies $f(p) = 0$.

In particular, the eigenvalues of $M_x$ are the possible values of $x$ at points of the variety!

This is nice, because it allows us to take advantage of the major advances that have been made in eigenstructure computations during the last couple of decades (especially concerning stability and parallelism).
What about the edge cases?

Another nice thing about this view is that the “edge cases” textbooks like to exclude are analysed with the rest.

\( \mathcal{K} \) not algebraically closed Variety points not in \( \mathcal{K}|^X^| \) show up as “complex” (living in some field extension) eigenvalues.

degrees of freedom used for derivatives These show up as generalised eigenvectors, for eigenvalues whose algebraic multiplicity is higher than the geometric multiplicity.
It’s not restricted to the variables

Note that there is nothing about the above that really is specific to $x$ being a variable. One can just as easily compute the possible values of some function.

In for example the spline intersection problem, the variables $s$ and $t$ are “times” in the parametrisation of these curves, but it is often more interesting to find the $(x, y)$-coordinates of the intersection points. Since $x$ and $y$ in that case are functions of $s$ and $t$, they are just as much elements of the quotient $\mathcal{K}[X]/\mathcal{I}$ as $s$ and $t$ are, and thus the multiply-by operators $M_x$ and $M_y$ have the intersection point coordinates as eigenvalues.
Detailed instructions

If I was to compute the coordinates of the middle right intersection point, I might seek to identify it as the point in the variety whose y-coordinate is closest to 1/3. This means its eigenvector will be the eigenvector of the largest eigenvalue of

\[ A = (M_y - \frac{1}{3}I)^{-1}. \]

This matrix \( A \in \mathbb{Q}^{9 \times 9} \) is straightforward to compute in exact arithmetic.

Then numerically iterate (standard power iteration)

\[ f_0 = 1, \quad f_{n+1} = \frac{1}{\|Af_n\|}Af_n \]

where \( \| \cdot \| \) is just some norm (e.g. standard Euclidean norm) until the direction of \( f_n \) stabilises. Then the sought intersection point will be \((\lambda_x, \lambda_y)\), where \( M_x f_n = \lambda_x f_n \) and \( M_y f_n = \lambda_y f_n \) (up to rounding errors).
The explicit calculation

In the quotient, \( y - \frac{1}{3} = \frac{11}{7360} s^2 + \frac{669}{736} t - \frac{679}{736} s - \frac{5353}{22080} \). The multiply-by-(\( y - \frac{1}{3} \)) operator has the matrix

\[
\begin{pmatrix}
-5353 & 0 & 44823 & 33 & 0 & -903 & -292527 & 0 & 880505181 \\
22080 & -5353 & 2976381 & 541696 & 33 & 1197312 & 79376512 & 11691 & 6455560241152 \\
669 & 22080 & 959856 & 541696 & 33 & 1547 & 19934128 & 33 & 3227780120576 \\
736 & 669 & 0 & 541696 & 33 & 0 & -19934128 & 0 & -2292527 \\
0 & 736 & -790661 & 2976381 & 33 & 18010983 & 199430193 & 199430193 & -134410152519 \\
-679 & 0 & 3250176 & 3250176 & 33 & 39868570816 & 39868570816 & 0 & 6455560241152 \\
7360 & -679 & 0 & 669 & 33 & 0 & 39868570816 & 0 & 11691 \\
0 & 7360 & -790661 & 4385570816 & 33 & 199430193 & 39868570816 & 33 & 199444128 \\
11 & 0 & 97509 & 1354718919427359417716468159317950 & 540177191082414403890227259873750 & 199199894925044910290333110447725 & 39868570816 & 388247727339 & 6455560241152 \\
7360 & 11 & -1083392 & -1083392 & 0 & 163083621 & 2392129536 & 0 & 2976381 \\
0 & -7360 & 0 & 1083392 & 0 & 1354718919427359417716468159317950 & 669 & 6455560241152 & 2355212677 \\
0 & 7360 & 0 & 0 & 669 & 736 & 0 & 163083621 & 6455560241152 \\
0 & 0 & 11 & 7360 & 0 & 0 & 1083392 & 2392129536 & 2392129536
\end{pmatrix}

The inverse \( A \) is

\[
\begin{pmatrix}
5353 & 0 & -44823 & -33 & 0 & -903 & -292527 & 0 & 880505181 \\
-22080 & 5353 & -2976381 & -541696 & -33 & -1547 & -19934128 & 0 & -6455560241152 \\
669 & -22080 & 959856 & -541696 & -33 & 0 & -19934128 & 0 & -3227780120576 \\
736 & 669 & 0 & -541696 & -33 & 0 & -19934128 & 0 & -2292527 \\
0 & 736 & -790661 & -2976381 & -33 & 18010983 & 199430193 & 199430193 & -134410152519 \\
-679 & 0 & 3250176 & -3250176 & -33 & 39868570816 & 39868570816 & 0 & 6455560241152 \\
7360 & -679 & 0 & 669 & -33 & 0 & 39868570816 & 0 & 11691 \\
0 & 7360 & -790661 & -4385570816 & -33 & 199430193 & 39868570816 & 33 & 199444128 \\
11 & 0 & -97509 & -3250176 & -540177191082414403890227259873750 & 199199894925044910290333110447725 & 39868570816 & 388247727339 & 6455560241152 \\
7360 & 11 & -1083392 & -1083392 & 0 & 163083621 & 2392129536 & 0 & 2976381 \\
0 & -7360 & 0 & 1083392 & 0 & 1354718919427359417716468159317950 & 669 & 6455560241152 & 2355212677 \\
0 & 7360 & 0 & 0 & 669 & 736 & 0 & 163083621 & 6455560241152 \\
0 & 0 & 11 & 7360 & 0 & 0 & 1083392 & 2392129536 & 2392129536
\end{pmatrix}
\]
Numerics

With elements truncated to two decimals, $A \approx$

$$
\begin{pmatrix}
-4.61 & -0.14 & -0.04 & -0.18 & -0.06 & -0.02 & -0.19 & -0.07 & -0.02 \\
30.29 & 8.66 & 3.62 & 5.86 & 2.18 & 0.81 & 6.26 & 2.33 & 0.87 \\
-24.98 & -7.18 & -2.10 & -5.20 & -1.94 & -0.72 & -5.55 & -2.06 & -0.76 \\
15.77 & 5.62 & 2.18 & 14.31 & 6.88 & 2.58 & 19.76 & 7.47 & 2.74 \\
119.03 & 44.18 & 16.63 & 125.00 & 48.76 & 18.77 & 154.65 & 57.80 & 21.49 \\
-34.88 & -12.97 & -4.84 & -36.78 & -13.91 & -5.11 & -41.10 & -14.07 & -5.12 \\
304.19 & 113.44 & 42.35 & 323.98 & 121.29 & 45.17 & 349.43 & 128.39 & 48.15 \\
\end{pmatrix}
$$

$f_0 = (1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)$, whereas $f_{50} \approx (-0.000, 0.014, -0.013, 0.035, -0.364, 0.333, -0.062, 0.640, -0.585)$ (rounded to 3 decimals). This is the eigenvector corresponding to the sought point, so its coordinates can be computed to be

$$
\lambda_x \approx \frac{f_{50}^T M_x f_{50}}{\|f_{50}\|^2} \approx 0.90639,
$$

$$
\lambda_y \approx \frac{f_{50}^T M_y f_{50}}{\|f_{50}\|^2} \approx 0.37438
$$
Graphics

Plotting confirms that this is indeed the coordinates of the sought point.

\begin{tikzpicture}[scale=3]
\path[use as bounding box] (-0.3,-0.3) rectangle (1.3,1.3);
\fill[cyan] (0.90639, 0.37438) circle (1pt);
\draw (0.1,0) .. controls (0,4.1) and (1,-3) .. (0.9,1);
\draw[red] (0,0.1) .. controls (4,0) and (-3,1) .. (1,0.9);
\end{tikzpicture}
A Physics analogy

What was done here is kind of similar to what one would do in quantum physics.

No direct means of observing the points?
First set up a particle with equal probability for all points of the space ($f_0$ state).
Then iterate a process ($A$) that will concentrate the probability towards the point you’re interested in.
When you’re satisfied with the concentration, measure the current position of the particle. ($M_x$ and $M_y$ are position operators.)

Since everything commutes, there are no Heisenberg uncertainty problems to worry about.
Exact root location

What if I’m not satisfied using numerical calculations for locating points, but require exact information?

The classical approach would be to compute the characteristic polynomial of the $M_x$ matrix—this has the eigenvalues of $M_x$ as its roots, and will thus provide an exact representation of the possible $x$ values.

But computing the characteristic polynomial using minor expansions is an $O(2^n)$ (plus change) operation, where $n = \dim(\mathcal{K}[X]/\mathcal{I})$. 
Exact root location

What if I’m not satisfied using numerical calculations for locating points, but require *exact* information?

The classical approach would be to compute the *characteristic polynomial* of the $M_x$ matrix—this has the eigenvalues of $M_x$ as its roots, and will thus provide an exact representation of the possible $x$ values.

But computing the characteristic polynomial using minor expansions is an $O(2^n)$ (plus change) operation, where $n = \dim(K[X]/I)$. 
Exact root location (2)

An alternative is to compute the \textit{minimal polynomial} by solving

\[
\sum_{i=0}^{m-1} \alpha_i M_x^i = M_x^m \quad \text{for} \quad \{\alpha_i\}_{i=0}^{m-1} \in \mathcal{K} \quad \text{and} \quad m \leq n \quad (1)
\]

—an $n^2 \times n$ linear equation system, but known to have a solution (if working in exact arithmetic).

Computing one matrix product is $O(n^3)$ operations in $\mathcal{K}$, so doing $n$ of them is $O(n^4)$. Solving (1) is $O(n^4)$ operations in $\mathcal{K}$ as well, so in total computing the minimal polynomial is $O(n^4)$ operations in $\mathcal{K}$.