An Algorithm for Direct Numerical Extraction of the Characteristic Polynomial of a Matrix

Aaron M. Schinder

Abstract

A direct numerical method for the determination of the coefficients of the characteristic polynomial of a matrix was found while on a quixotic quest for a means of direct numerical solution of the eigenvalue problem. The geometric meaning of the determinant, and the linearity and behavior of the exterior algebra were used to derive a means of separating the unknown λ^k from the numerical part of the kth coefficient of the characteristic polynomial, expressed as a sum of ${}_N C_k$ determinants. The overall complexity class of the algorithm is $O(N^3 2^N)$, making it mostly useless from a computing perspective. My original goal may end up being torpedoed by an isomorphism between this problem and one whose solution is forbidden by the Abel-Ruffini theorem, which is discussed.

Disclaimer

This is not an academic paper. I haven't done any literature search for the originality of this method, or the history of the mathematics of linear algebra, beyond learning a lot of general skills in the subject in my math classes. In addition, I would be highly surprised if this method turned out to be original, (or even useful, given it's embarrassing complexity class), given that this is one of the most highly developed branches of mathematics, and is over 200 years old at this point.

This paper is organized like an academic paper mostly for the practice of quickly writing them. It is less formal in tone.

Introduction

Deriving the eigenvalues of a matrix and finding the associated eigenvectors is an important mathematical operation in the solution of linear algebra problems, linearizations of nonlinear systems of equations, and the solution of linear partial differential equations using approximations for the function space of the solution.

Eigenvectors have special properties in linear systems. They correspond to the stationary states of Schrodinger's equation, and other harmonic solutions to wave equations. They correspond to characteristic lines of the attractors of ordinary differential equation systems. Methods to efficiently find the eigenvalues and eigenvectors of a matrix are important.

This is not one of those methods.

Matlab, and other commercial solvers typically use an iterative method to transform the initial matrix into a triangular form. QR decomposition, and various refinements thereof for more specific input matrices, is used to solve not only the eigenvalue-eigenvector problem, but is used internally to iteratively solve for the complete set of roots of a polynomial by transforming the problem into an eigenvalue problem.

I was thinking about the solution to the eigenvalue problem, given that I have several linear partial differential equations that I am currently studying. When I have a problem like this, I tend to sometimes try to come up with an algorithm on my own prior to deep reading on the subject, as this is *occasionally* faster, and can give insight into the problems that cookbook application of someone else's process does not, when it doesn't lead me down all sorts of rabbit holes to no purpose.

Derivation

A matrix can be thought of, geometrically, as a series of differential one-forms, each pointing in an arbitrary direction in space. Each one form is related to one of the unit basis vectors. If your unit basis is Cartesian, and you aren't worrying about skew-transformations of coordinates, you don't have to pay as much attention to the transformation properties, covariance and contravariance, of each end of the basis tensor product. This would be specific to what sort of object you are representing anyway.

$$A = \boldsymbol{e_i} \otimes \boldsymbol{r_i} = \boldsymbol{e_i} \otimes a_{ij}\boldsymbol{e^j}$$

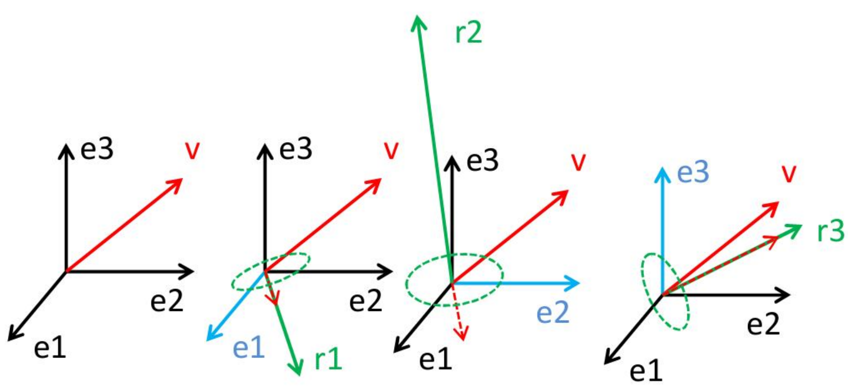
$$A\boldsymbol{v} = \boldsymbol{e_i} \otimes \boldsymbol{r_i}(\boldsymbol{v}) = \boldsymbol{e_i} \otimes a_{ij}v_k(\boldsymbol{e^j} \cdot \boldsymbol{e_k}) = (for\ cartesian\ basis)\ a_{ij}v_j\boldsymbol{e_i}$$

In this notation, capital letters are matrices, bold letters are vectors or dual-vectors (one-forms, etc). The subscripts are component indices in the Einstein summation convention.

The differential one-forms $\mathbf{r_i}$ correspond to the row vectors of the A matrix. ⊗ is the tensor product. Bolded quantities represent vectors or one-forms.

Geometrically speaking, the determinant of a matrix is the oriented magnitude of a differential N form, where N is the rank of the matrix. It is the inner product of the N-form, constructed from each of the component one-forms, with the unit N-form of your coordinate basis. In less abstract, general terms, it is the oriented N-volume of the parallelopiped formed by the set of row-vectors of the matrix.

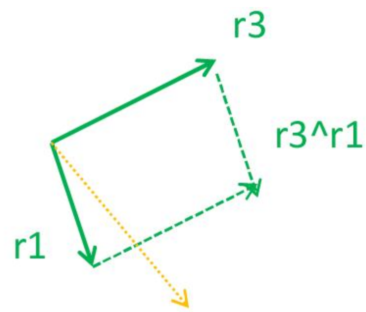
$$A = \begin{bmatrix} \boldsymbol{r_1} \\ \cdots \\ \boldsymbol{r_N} \end{bmatrix}$$



The N-form is constructed by a type of product defined in the branch of mathematics called exterior algebra. The exterior product takes two differential forms of a certain type, and constructs higher dimensional geometric objects from them. These geometric objects correspond to oriented areas, oriented volumes, oriented hypervolumes, and so on. They too have the properties of orientation and magnitude.

<u>Aside 1:</u>

You might recall that you've already seen a means of describing an oriented area in vector calculus: the cross product. The cross product and the wedge product are indeed intimately related. The exterior algebra is more general, and powerful, however, in that it behaves in a way that is systematically generalizable to any number of dimensions.

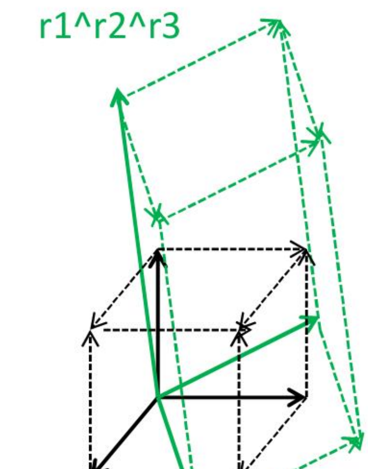


You might ask yourself how to describe the cross product of two vectors in four dimensions: There isn't one vector that is normal to each of the two cross-product vectors, but two! The differential two-form is well defined, however, with components in a basis {e1^e2, e3^e1, e2^e3, e2^e4, e4^e1, e4^e3}. There are six components in the basis for four dimensional differential areas, and only four components in the basis for four dimensional vectors, or four dimensional oriented 3-volumes for that matter. There are more ways to point an area in general, than there are to point a vector in higher dimensions! The number of bases of a differential area also corresponds to the number of degrees of freedom in a rotation group (or other metric preserving transform group, such as the Lorentz transform (well, provided what you are preserving is a 2-form, not a higher order form!))

In addition, this provides a natural way to extend the concept of the determinant of a matrix to include "sub-determinants", which take subsets of the set of component forms to form N-k forms.

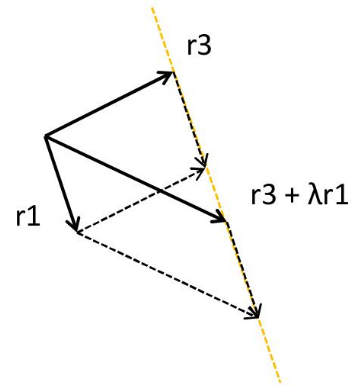
Determinants of Matrices, and Determinants of Sums of Matrices:

The determinant of a matrix is the oriented volume of the N-parralelopiped formed by the component row vectors:



$$det(A) = (\boldsymbol{r_1}\hat{}\boldsymbol{r_2}\hat{}\ldots\boldsymbol{r_N}) \cdot (\boldsymbol{e_1}\hat{}\boldsymbol{e_2}\hat{}\ldots\boldsymbol{e_N})$$

With this geometric picture in mind, it is easy to discern other properties of a matrix. If a matrix has a zero determinant, then it must have a null-space, and a dual-null space. The row addition operation does not alter the determinant (and from this, the process of bringing the matrix to upper triangular form via Gaussian elimination gives you the determinant by the product of the diagonal elements).



Or, taking advantage of the properties of exterior algebra, **a**^**a** = 0, and the linearity of the exterior product:

$$\boldsymbol{r_i}\hat{\ }\boldsymbol{r_{j\neq i}} = \boldsymbol{r_i}\hat{\ }(\boldsymbol{r_{j\neq i}} + \boldsymbol{\alpha r_i})$$

It is this property that can be used to calculate the coefficients of the characteristic equation. The eigenvalue is classically stated thus:

$$Av = \lambda v$$

$$(A - \lambda I)v = 0$$

The aforementioned means of finding the determinant via Gaussian elimination to bring A into upper triangular form does not allow you to produce a direct solution for the eigenvalues, as the row operations amount to the application of a transform matrix T, which transforms both A and the identity matrix. You no longer have a purely triangular matrix.

$$(TA - \lambda T)v = 0$$

T - upper (lower) triangular, TA lower (upper) triangular.

However, the matrix $(A - \lambda I)$, when decomposed into row operations, has the form:

$$\begin{bmatrix} \boldsymbol{r_1} - \lambda \boldsymbol{e_1} \\ \vdots \\ \boldsymbol{r_N} - \lambda \boldsymbol{e_N} \end{bmatrix}$$

As per our geometric picture before, the determinant of $(A - \lambda I)$ works out to:

$$(\boldsymbol{r_1} - \lambda \boldsymbol{e_1})\hat{\ }(\boldsymbol{r_2} - \lambda \boldsymbol{e_2})\hat{\ }\ldots(\boldsymbol{r_N} - \lambda \boldsymbol{e_N})$$

Which, when it is expanded, gives a polynomial:

$$(\boldsymbol{r_1}\hat{\ }\boldsymbol{r_2}\ldots\hat{\ }\boldsymbol{r_N}) + \{(-\boldsymbol{\lambda e_1}\hat{\ }\boldsymbol{r_2}\hat{\ }\ldots\boldsymbol{r_N}) + (\boldsymbol{r_1}\hat{\ } - \boldsymbol{\lambda e_2}\hat{\ }\ldots\boldsymbol{r_N}) + \ldots(\boldsymbol{r_1}\hat{\ }\boldsymbol{r_2}\hat{\ }\ldots -\boldsymbol{\lambda e_N})\} + \{\ldots\} + \ldots + (-\boldsymbol{\lambda e_1}\hat{\ } - \boldsymbol{\lambda e_2}\ldots\hat{\ } - \boldsymbol{\lambda e_N})$$

Using the linearity of the wedge product, the factors (-λ) can be removed from the polynomial, and you are left with a polynomial of order N in λ, otherwise known as the characteristic polynomial. The kth coefficeint, then, can be formed as a sum of the determinant of matrices formed with a combination of (N-k) chosen from {ri} and k chosen from {ei}.

$$kth\ term: (-1)^k \lambda^k \left(det\left(\begin{bmatrix} \boldsymbol{e_1} \\ \boldsymbol{e_k} \\ \vdots \\ \boldsymbol{r_{N-k}} \\ \boldsymbol{r_N} \end{bmatrix}\right) + det\left(\begin{bmatrix} \boldsymbol{e_1} \\ \boldsymbol{r_2} \\ \boldsymbol{e_{k+1}} \\ \vdots \\ \boldsymbol{r_N} \end{bmatrix}\right) + \ldots {}_N C_k\ combinations \right)$$

There are 2^N total combinations, so in total, the time complexity is 2^N times the time complexity of finding the determinants. The unknown factor λ is factored from the numerical coefficient. If a general method, such as Gaussian elimination is used to find the determinants, the overall time complexity for determining the coefficients of the characteristic polynomial is $O(N^3 * 2^N)$. ....Shut up.... At least it isn't as bad as the method of minors.

Aside 2:

While this algorithm is probably not numerically useful for the solution of the eigenvalue problem, it did bring to mind a few interesting isomorphisms between eigenvalue problems and the roots of polynomials:

If a direct method for the solution of the eigenvalues or eigenvectors were ever to be found, it would amount to a means of directly (exactly), as opposed to iteratively, solving for the roots of a polynomial of arbitrary order.

Even if a direct method for solving for one eigenvector of a matrix were found, this would lead to a direct method for both the solution of a polynomial of arbitrary order, and for finding all subsequent eigenvalues through the following process:

For polynomials P, matrices A
1. $Convert(P_N = 0) \rightarrow (A_N \boldsymbol{v} = \lambda \boldsymbol{v})$
2. $Direct\_Eigenvector\_Oracle(A_N \boldsymbol{v} = \lambda \boldsymbol{v}) \rightarrow \boldsymbol{v_i}$
3. $A_N \boldsymbol{v_i} \cdot \boldsymbol{v_i} / |\boldsymbol{v}| = \lambda_i$
4. $Polynomial\_Division(P_N \ / \ (x - \lambda_i)) \rightarrow P_{N-1}$
5. Iterate - Return to 1, feeding P_{N-1} into process.

The Abel-Ruffini theorem suggests that the latter of these processes, and by extension the former, may be impossible, at least in terms of radical expressions[1].

The λs correspond to the exact roots of the polynomial. If an oracle existed that returns even one eigenvector via an exact process, one λ may be found exactly. Polynomial division and iteration would return all the roots of P via an exact process. This would contradict the result of the Abel-Ruffini theorem.

Conclusion

An algorithm for the direct numerical production of the characteristic polynomial coefficients of a matrix has been developed. It doesn't allow direct, or even all that efficient, a solution to the eigenvalue problem. Commercial solvers proceed in the other direction, using iterative methods to solve the eigenvalue problem in order to find the roots of a polynomial. Interesting parallels between the direct solution of this problem and the direct solution of another unsolved problem (exact roots of an arbitrary order polynomial) have been developed.

Time has been wasted. Fun has been had. Trees have been mercilessly slaughtered in the pursuit of math.

References

[1] "Abel-Ruffini Theorem". *Wikipedia*, http://en.wikipedia.org/wiki/Abel-Ruffini_theorem. 31 May 2013.