Back to the roots:

Solving polynomial systems with numerical linear algebra tools

Prof. Dr. Bart De Moor

ESAT-STADIUS, Department of Electrical Engineering, KU Leuven, Belgium

E: bart.demoor@kuleuven.be W: www.bartdemoor.be

Finding the roots of a set of multivariate polynomials has numerous applications in geometry and optimization, system and control theory, modeling and identification, statistics and bioinformatics, and many other scientific disciplines. It is an old yet fascinating problem, that has intrigued scientists throughout the ages, starting with the Greeks, over Fermat and Descartes, Newton, Leibniz, Bezout and many many others.

It all started with trying to find the roots of a polynomial equation with real coefficients in one unknown. In the beginning of the 19-th century, formulas were known for the roots of polynomials up to degree 4, but it was Galois who showed that no general formulas exist for degree 5 and higher. This implies that roots of polynomials of degree higher than 4 in general can only be found using (iterative) numerical algorithms. Soon thereafter, Sylvester and other contemporains started research on how to find the roots of sets of multivariate polynomials. Sylvester derived an elimination algorithm, in which he eliminates variables one by one, ending up with a ‘characteristic equation’ in one variable only. If one then has obtained the roots of this last equation, one can then ‘back-substitute’ root-by-root into the other equations, and hence in principle find all roots. Sylvester’s algorithm is the equivalent of Gaussian elimination for linear equations, but more importantly, his results imply that finding the roots of a polynomial system is an eigenvalue problem!

Later on, in the 20th century, there was a booming mathematical development, which gave birth to a discipline that today is called algebraic geometry, with a fabulous rich history, to which famous mathematicians, such as Hilbert, but also many others, contributed. It also led to the machinery of Gröbner bases and the like, which today are ubiquitous in books and symbolic methods in algebraic geometry, with numerous applications in fields like geometric design, combinatorics and integer programming, coding theory, robotics, etc…

We will however not talk about these developments, but return to the very roots and early days of the problem.

In this talk, we will elaborate on a research program, the objective of which is to translate the many – symbolic - algorithms from algebraic geometry, into numerical linear algebra algebra algorithms. Our talk develops ideas on three complementary levels:

- Geometric linear algebra, which deals with column and row vector spaces, dimensions, orthogonality, kernels, eigenvalue problems and the like;

- Numerical linear algebra, where we conceptually deal with tools like Gram-Schmidt orthogonalization, the singular value decomposition, ranks, angles between subspaces, etc… ;
- Numerical algorithms, which implement the linear algebra tools into an efficient and numerically robust method. Here we can exploit matrix structures (e.g. Toeplitz or sparsity), investigate variations of iterative methods (e.g. power methods) or try to speeden up convergence (e.g. by FFT).

Starting from a set of multivariate polynomials, we will show how, from the matrix of coefficients, we can construct a so-called Macaulay matrix, the row space of which represents 'the ideal' of the polynomial system, and the kernel its 'variety', which also contains the roots of the polynomial system. Its co-rank, which is the dimension of the kernel, reveals the number of roots, which can be finite or infinite, and which can have multiplicities larger than 1. We will show how to find them, using insights from system theory (more specifically, we will use multi-dimensional realization theory).

This basically leads to 2 approaches to find all roots: a kernel based algorithm, in which one first has to calculate the kernel of the Macaulay matrix, and then by applying multi-dimensional realization theory one can construct an eigenvalue problem from which the roots are obtained. Secondly a 'data-driven' algorithm, in which we directly operate with the elements of the Macaulay matrix, solve a set of linear equations using the QR-decomposition, and then obtain an eigenvalue problem in terms of certain blocks of the triangular factor R.

Our claim is, that in due time, we will have a numerical linear algebra based tool set to efficiently and robustly find all roots of a set of multivariate polynomials. Notions from linear algebra we use are column and row spaces, ranks, kernels and the eigenvalue problem, but also Grassmann's dimension theorem and angles between subspaces. The tool set we use are algorithms such as the QR-, the CS- and the singular value decomposition (SVD).

We will illustrate our results with some motivating examples, such as:

- How to find the minimizing root of polynomial optimization problems, i.e. optimization problems in which the objective function and constraints are multivariate polynomials. In particular, we will show that optimization algorithms in system identification, such as Prediction Error Methods, in principle try to solve an eigenvalue problem, as is also the case in Structured Total Least Squares problems. We will also elaborate on the H2-model reduction problem.
- Algebraic statistics, in which the maximum likelihood estimation of the parameters of discrete statistical models (such as in Bayesian networks or Hidden Markov Models), leads to finding the roots of a set of multivariate polynomials.
- Multi-way arrays of data or numerical tensors, which arise in statistics, biomedical data and signal processing, system identification, data mining etc… In particular, the problem of approximating a given data tensor in a least squares sense by one of lower multi-linear rank, results in a set of multivariate polynomials, and hence is an eigenvalue problem.