Notes of a Numerical Analyst

Multivariate polynomials

NICK TREFETHEN FRS

My past twenty years have been spent working with polynomials. Polynomials are the starting point of numerical algorithms for integration, differentiation, root-finding, optimisation, and approximation, and in the software system Chebfun, every function is converted to this universal currency before you do anything with it. A typical f defined on $[-1,1]$ might be approximated to 16 digits by $p(x)=\sum_{k=0}^{n} a_k T_k(x)$ with $n = 500$, say, where T_k is the degree k Chebyshev polynomial.

But all this is univariate. What does one do in three dimensions, the base case of science and engineering? And how about dimensions $n > 3$, with applications from the many-particle systems of quantum physics to the high-dimensional search spaces of data science?

We all have our lacunae, and for years, one of mine was multivariate polynomials. You couldn't ask for a more respectable citizen of pure mathematics, as attested by ten Fields medals related to algebraic geometry, and I knew that one day, I would have to get serious and learn something about this subject. An excuse to put my house in order came recently in teaching a course at NYU. I decided to show the students case-by-case how, for each numerical problem, the basic 1D method you already know starts from univariate polynomials, and then in nD, there's a powerful analogue based on multivariate polynomials.

But as I tried to prepare my lecture I discovered, it wasn't so! When it comes to numerical computation, *multivariate polynomials are not used much.* Tensor products of univariate polynomials are used all the time (a special case), but not the multivariate version as normally understood, where we start from P_k, the set of polynomials of total degree $\le k$.

In numerical integration in a square or a cube, for example, there's an elegant cubature idea introduced by James Clerk Maxwell: interpolate function samples by an element of P_k, then integrate the interpolant. But implementation is difficult (challenges of unisolvency), and although there's plenty of theory, these formulae are rarely employed. Or in numerical PDE, the dominant method is finite elements, which in principle can be based on multivariate polynomials of arbitrary degree. But in practice, most applications stick to degrees 1–4. Or in approximation of functions, you *could* use multivariate polynomials, but few do.

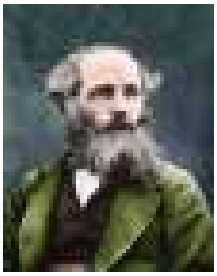

Figure 1. Maxwell proposed integrating a function via a multivariate polynomial interpolant. Why is this method so rarely used?

This got me thinking about another lacuna. Complex variables are my best-loved tool—why had I never mastered the multivariate case, several complex variables (SCV)? Given how much we gain from convergent series of polynomials, surely there's all the more to be gained from convergent series of multivariate polynomials? Well, with apologies to the experts, I now believe it isn't so. SCV is a fascinating field, full of challenges, but when it comes to developing numerical algorithms, it is the one-variable case we leverage.

Maybe the bedrock example is the Laplacian operator, the starting point of mathematical physics. In 3D we write it like this:

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}.$$

The Laplacian is the very archetype of an isotropic process—rotation-invariant—yet to work with it, we break it into univariate pieces.

Nick Trefethen
Trefethen is Professor of Numerical Analysis and head of the Numerical Analysis Group at the University of Oxford.