

Mat1062: Introductory Numerical Methods for PDE

Mary Pugh

March 10, 2009

1 Ownership

These notes are the joint property of Rob Almgren and Mary Pugh.

2 Overview of Projection methods

Throughout this course, we have discussed Fourier modes in the context of stability. Generally, we assume that a PDE or a discrete scheme has a solution of the form $u(x, t) = U(t) \exp(i\xi x)$ where ξ is a real number. Then we look for the time dependence of $U(t)$, and if it grows, we conclude that the method is unstable. An implicit assumption in this approach is that *any* initial data $u_0(x)$ may be written as a combination of these modes, so these special solutions are stable.

Now we turn this point of view into a full numerical method.

Suppose we have a PDE which is either time-dependent, $u_t = \mathcal{L}u + f$ or $u_{tt} = \mathcal{L}u = f$, or is elliptic $\mathcal{L}u = -f$, where \mathcal{L} is a linear differential operator such as $\mathcal{L}u = u_{xx}$. The idea behind projection methods is to approximate the time-dependent solution $u(x, t)$ with

$$u_N(x, t) = \sum_{j=1}^N c_j(t) \phi_j(x)$$

and the elliptic solution $u(x)$ with

$$u_N(x) = \sum_{j=1}^N c_j \phi_j(x).$$

Here u_N is in the N -dimensional space spanned by the basis functions ϕ_1, \dots, ϕ_N .

How shall we choose the basis functions $\{\phi_j\}$? “Good” properties are

- *Localization in space.* This may give us a natural interpretation of the coefficients $c_j = u(x_j)$. It makes the linear system we’ll have to solve sparse so the linear algebra is doable though not necessarily easy. Also, if the basis functions are localised in space it makes it easier to refine the solution as needed — if something interesting is happening in a certain region of space we would add extra basis functions, choosing those that are localized where the action is.
- *Smoothness.* Our PDE depends on spatial derivatives, and it is nice if the approximating functions have at least a few. As we’ll see, if we seek a weak solution rather than a classical solution, we can use integration by parts to reduce the required number of derivatives.
- *Analytic simplicity.* They should be easy to work with.

Different methods focus on different combinations of virtues:

- **Finite element** methods take the ϕ_j to be piecewise low-order polynomials, based on a selection of node points. They are very well localized in space, but often have only barely enough smoothness. They work in strange-shaped regions in space. The errors are typically some power of the node point spacing h , or some negative power of the number of basis functions N .
- **Spectral methods** take the ϕ_j to be the eigenfunctions of the operator \mathcal{L} . For example, if $\mathcal{L} = \partial_{xx}$ with homogeneous Dirichlet, homogeneous Neumann, or periodic boundary conditions then ϕ_j would be functions of the form $\sin(jx)$, $\cos(jx)$, or $\exp(ijx)$ respectively. If the operator is self-adjoint then the eigenfunctions are orthogonal which is very helpful. Also they are smooth. One the downside, they aren’t localised in space and we have formulae for them only for certain domains. (There are theorems saying that they exist for more general domains but actually finding them is a different computational challenge.) For smooth problems, the error is typically *smaller than any power* of N .

- **Wavelet methods** use basis functions that are localized in space, but preserve some of the nice linear algebra properties of spectral methods. They are analytically rather complicated to work with.

The PDE acts on a space \mathcal{B} of functions. The approximation is in an N -dimensional subspace, \mathcal{B}_N , of \mathcal{B} . The main problems we have to address are 1) how to define the problem on \mathcal{B}_N given the original problem on \mathcal{B} and 2) how to show that as $N \rightarrow \infty$ our approximate solution u_N converges to the desired solution u .

- The **Galerkin** approximation is in terms of inner products (no node points). We rewrite the original problem as

$$\langle u_t, \phi_k \rangle = \langle \mathcal{L}u, \phi_k \rangle + \langle f, \phi_k \rangle \quad \text{for each } k = 1, \dots, N.$$

This gives the linear system

$$\sum_{j=1}^N c'_j(t) \langle \phi_j, \phi_k \rangle = \sum_{j=1}^N c_j \langle \mathcal{L}\phi_j, \phi_k \rangle + \langle f, \phi_k \rangle, \quad k = 1, \dots, N$$

which must be solved to give (c'_1, \dots, c'_N) in terms of (c_1, \dots, c_N) . If necessary (if the basis functions don't have enough smoothness), we can use integration by parts to rewrite the term $\langle \mathcal{L}\phi_j, \phi_k \rangle$. Clearly it will be very advantageous if $\langle \phi_j, \phi_k \rangle = \delta_{jk}$ so we don't have to do any linear algebra to find (c'_1, \dots, c'_N) in terms of (c_1, \dots, c_N) .

- The **collocation** method is based on a set of N node points x_1, \dots, x_N (no inner products). We keep track of the solution by its values u_1, \dots, u_N with $u_j = u(x_j)$, rather than directly by the coefficients c_j . Thus this is conceptually like the finite difference method. It is much easier to incorporate nonlinear terms than with Galerkin.