

Introduction to Numerical PDEs

Varun Shankar

February 16, 2016

1 Introduction

In this chapter, we will introduce a general classification scheme for linear second-order PDEs, and discuss when they have solutions. Then, we will introduce a very general method for solving PDEs called the Method of Weighted Residuals (MWR). Using the MWR as our foundation, we will discuss the following methods:

1. Global Collocation Methods.
2. Local Collocation Methods.
3. Global Galerkin Methods.
4. Local Galerkin Methods.

The remainder of this course will focus on global and local collocation methods. We will carefully use the classification of PDEs to derive appropriate global and local collocation methods.

2 Background on PDEs

2.1 Classifying second-order linear PDEs

Consider the following general form of a linear second-order PDE in two independent variables x, y and the dependent variable $u(x, y)$:

$$A \frac{\partial^2 u}{\partial x^2} + B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + D \frac{\partial u}{\partial x} + E \frac{\partial u}{\partial y} + Fu + G = 0, \quad (1)$$

where A through G are constants. Now, we classify PDEs based on conditions on these constants.

- If $B^2 - 4AC < 0$, the PDE is elliptic.
- If $B^2 - 4AC = 0$, the PDE is parabolic.
- If $B^2 - 4AC > 0$, the PDE is hyperbolic.

This classification allows us to develop numerical methods to solve these PDEs. However, in practice, the PDEs we encounter will have combinations of the above properties. We will discuss such scenarios also. Examples of the above PDEs are:

1. Elliptic- Laplace's equation, Poisson's equation.
2. Parabolic- the Heat equation
3. Hyperbolic- the wave equation
4. Mixed features- the incompressible Navier-Stokes equations.

This classification is typically retained even if we add more variables to the PDE. For example, the 2D heat equation is still viewed as parabolic.

2.2 Well-posedness

Practically speaking, simply considering a PDE and its solutions is useless. We typically solve PDEs on specific *domains* (of space-time), and impose conditions on the solutions of the PDE at the boundary of the domain. Through this process, we typically get useful unique solutions to PDEs.

(HADAMARD) A PDE problem is said to be well-posed if:

1. a solution to the problem exists;
2. the solution is unique; and
3. the solution depends continuously on the problem data.

“Problem data” here refers to: the coefficients in the PDE, the specific boundary and/or initial conditions on the PDE, and the domain on which the PDE is required to hold. If these conditions are not satisfied, the PDE is said to be ill-posed. An example of an ill-posed PDE is the Poisson Equation with an arbitrary Neumann boundary condition:

$$\frac{\partial^2 u}{\partial x^2} = f(x), x \in \Omega, \quad (2)$$

$$\frac{\partial u}{\partial x} = h(x), x \in \partial\Omega. \quad (3)$$

This is ill-posed unless $\int_{\partial\Omega} h(x)dS = \int_{\Omega} f(x)dx$. This can be shown via the divergence-theorem.

3 The Method of Weighted Residuals

We now turn our attention to the numerical solution of PDEs. There are many methods used to solve PDEs, and it is easy to become confused: how does one pick a method? When should one select finite differences over finite elements? Are these the only choices?

We will attempt to answer some of these questions. We now present an overarching method for solving boundary-value PDEs called the method of weighted residuals (MWR).

The basic concept of the MWR is to attempt to drive some *residual* to zero using orthogonality (or other) conditions. Consider the following general PDE problem:

$$\mathcal{L}u(x) = f(x) \in \Omega, Su(x) = g(x) \in \Gamma, \quad (4)$$

where $\mathcal{L}$ is some linear differential operator, and S is some other operator responsible for enforcing boundary conditions. Let the approximate solution be $U(x)$. We express this solution as

$$U(x) = \sum_{k=1}^N c_k \phi_k(x), \quad (5)$$

where $\{\phi_k\}_{k=1}^N$ span the so-called *trial space*. In other words, $U(x)$ is expressed as some linear combination of some *basis* functions, much like in interpolation. These basis functions are often called *trial functions*; sometimes, the $U(x)$ functions are themselves called *trial functions*. The goal now is to determine the set of unknown coefficients $\{c_k\}_{k=1}^N$. As we do so, we will also discuss how to select the trial functions (or basis functions).

We first define the *residual* $r(x)$ as

$$r(x) = \mathcal{L}U(x) - f(x). \quad (6)$$

Clearly, if $r(x) = 0$, we have solved the PDE exactly (assuming we somehow satisfied boundary conditions). We can safely generalize this condition and require

$$\int_{\Omega} r(x) w_j(x) dx = 0, \forall j, \quad (7)$$

where $\{w_j\}_{j=1}^M$ are *weight functions* or *test functions*. Thus, in the MWR, we seek c_k such that the above weighted average (in the function sense) of residuals is zero. We obtain different classes of methods by selecting the weight functions.

3.1 Collocation Methods

Collocation methods simply attempt to make the residual vanish at a specific set of points called the *collocation nodes* or *collocation points*. This is accomplished by selecting the weight functions (or test functions) to be the Dirac Delta function(al) centered at a collocation point. More precisely,

$$w_j(x) = \delta(x - x_j), x_j \in \Omega, \quad (8)$$

where $x_j, j = 1, \dots, M$ is the set of collocation points. Then, we have

$$\int_{\Omega} r(x)w_j(x)dx = \int_{\Omega} r(x)\delta(x - x_j)dx = r(x_j) = 0. \quad (9)$$

How to pick M , the number of collocation points? This will depend closely on the choice of ϕ , the trial functions (or basis functions). Note that the collocation method, by construction, only gives $U(x)$ at the collocation points x_j . Conventionally, the number of collocation points M equals the number N of unknown coefficients.

If there are more collocation points than unknowns, we have least-squares collocation methods.

3.2 Bubnov-Galerkin Methods

Bubnov-Galerkin methods take a different approach from collocation methods. Here, the approach is to simply ask that the weight functions (or test functions) be the same as the basis functions (or trial functions). More precisely,

$$\int_{\Omega} r(x)w_j(x)dx = \int_{\Omega} r(x)\phi_j(x)dx = 0. \quad (10)$$

Obviously, this is a requirement that $\phi_j(x)$ be orthogonal to residual. This condition is often called *Galerkin orthogonality*. Note that unlike the collocation method, the Galerkin method gives a $U(x)$ that is defined *everywhere*, though we only find it at a finite set of points in practice. On the other hand, this means that the residual does not necessarily exactly vanish at any point in the domain; it only vanishes in the average sense.

Henceforth, we will refer to Bubnov-Galerkin methods as simply Galerkin methods.

3.3 Petrov-Galerkin Methods

Petrov-Galerkin methods are a generalization of the Bubnov-Galerkin methods. Here, while the weight functions are orthogonal to the residual, they are not the same as the trial functions! Petrov-Galerkin methods can thus be more complicated to implement in practice, while offering greater flexibility than Galerkin methods.

4 Selecting the basis/trial functions for Galerkin Methods

We must now select our basis functions. These basis functions must both satisfy the boundary conditions and *span* approximately the same function space as $u(x)$ if we want $U(x)$ to be a good approximation to $u(x)$. Further, the choice of the basis functions will often determine how hard/easy it is to find the unknown coefficients c_k . We will first discuss the selection of basis functions for Galerkin methods, since these also end up being the test functions in this context.

There are two ways one could approach this. The first is straightforward: pick a very smooth, globally-supported basis function (that will also serve as the test function), akin to high-degree polynomial interpolation. The second is akin to piecewise polynomial interpolation: break the domain into subdomains, and use a piecewise-polynomial basis on each subdomain.

4.1 Global Galerkin Methods

The idea is the same as in interpolation: using a very smooth globally-supported basis can result in rapid convergence if the problem has a smooth solution. Such a method is called a *Spectral Galerkin Method*, and the name derives from the fact that such methods give spectral (exponential) convergence on smooth problems.

As for the basis functions, this will depend typically on the boundary conditions. If the boundary conditions are periodic, it is natural to use a Fourier basis, resulting in the spectral Fourier-Galerkin method. In other words, $\phi_k(x) = e^{ikx}$.

However, for problems involving other BCs, it is more common to use Legendre or Chebyshev polynomials as the basis functions. To make the basis functions satisfy the BCs, it is common to take *linear combinations* of these orthogonal polynomials as the trial functions (and test functions). By using orthogonal polynomials, we ensure that most of the inner products are zero. A quadrature rule will be required to evaluate inner products of the right hand side $f(x)$ with the basis functions.

4.2 Local Galerkin Methods

Local Galerkin methods adhere to the following philosophy: break up the domain Ω into a set of non-overlapping subdomains (or *elements*), then use piecewise polynomials supported on each subdomain as the trial functions.

Thus, instead of quadrature rules over the entire domain, we only need a quadrature rule on each subdomain. We lose spectral convergence, but gain computational efficiency and potential parallelism. To reduce the error in the ap-

proximate solution, we can either refinement the elements (referred to as h -refinement) or increase the order of the piecewise polynomials (p -refinement).

This method is called the Finite Element Method. There are many variants of this method, based on the type of polynomial basis used, the kinds of subdomains used etc. This is beyond the scope of our class.

5 Selecting trial/basis functions for Collocation methods

A similar philosophy applies to collocation methods: one could either select a very high-degree polynomial basis and obtain a *spectral collocation method* (sometimes called a *pseudospectral* method); or one could select low-degree piecewise polynomials, giving us the *finite-difference method*. However, unlike in Galerkin methods, *no quadrature rule is required!* For this reason, collocation methods are far simpler to implement than Galerkin methods. We will focus on collocation methods for the remainder of this class. We will develop the following methods:

1. (polynomial) finite-difference methods
2. (polynomial) spectral collocation methods
3. Radial Basis Function (RBF) spectral collocation methods; and
4. RBF Finite-Difference methods