Numerical Solutions for Partial Differential Equations

Lucy He
Harvard University

Max Guo
Harvard University

Jacqueline Wei
Harvard University

I. Introduction

Partial differential equations (PDEs) can be notoriously difficult to solve analytically, so various numerical methods have been developed to approximate solutions. In this paper, we survey three standard numerical methods: finite difference, finite element, and spectral methods for solving PDEs. We provide only a bare-bones exposition of these methods compared to the existing, expansive research. For the sake of brevity and accessibility, we demonstrate each of the methods for solving the Poisson Equation. Note that, in one dimension, this is actually an ordinary differential equation (ODE). Nevertheless, the one-dimensional case illustrates important concepts that generalize to multidimensional problems. The one-dimensional Poisson Equation can be written as follows:

$$-u_{xx}(x) = f(x) \tag{1}$$

where we impose boundary conditions $x \in (0,1)$, $u(0) = u(1) = 0$, and f is some continuous function.

Finally, we also provide a detailed application of the finite element method (FEM) applied to the standard heat equation in three dimensions. This extends the discussion in this Github repository by one additional dimension. Our code is located attached. The simplest form of the problem that we solve is given by:

$$u_t(t,x,y,z) = u_{xx} + u_{yy} + u_{zz} \tag{2}$$

on a rectangular box of dimensions $L \times W \times H$, with initial conditions

$$u(0,y,z) = 1 \tag{3}$$

$$u(L,y,z) = 0 \tag{4}$$

$$u_y(x,y=0,W,z) = 0 \tag{5}$$

$$u_z(x,y,z=0,H) = 0. \tag{6}$$

In other words, we heat one side to a constant temperature of 1 and the opposite side to a lower temperature of 0. Moreover, we enforce that the temperature should be the same across any cross-section parallel to the yz plane.

II. Finite Difference Method

In finite difference methods, we discretize the space and then approximate all derivatives with finite differences. The intuition is based on Taylor expansions and the definition of the derivative. For small h and continuous u:

$$u(x+h) = u(x) + \frac{u'(x)}{h} + \dots \tag{7}$$

$$\implies u'(x) \approx \frac{u(x+h) - u(x)}{h}. \tag{8}$$

Note that the definition of the derivative is exactly this as $h \to 0$:

$$u'(x) = \lim_{h \to 0} \frac{u(x+h) - u(x)}{h}. \tag{9}$$

For our Poisson problem, we need a second derivative. As a first step, we can use the approximation:

$$u''(x) \approx \frac{\frac{u(x+h)-u(x)}{h} - \frac{u(x)-u(x-h)}{h}}{h} \tag{10}$$
$$= \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}. \tag{11}$$

Now we follow the excellent exposition of Khoo et al. (2003) for discretizing and solving the problem. We divide the interval $(0, 1)$ into $n + 1$ equal length intervals such that $h = \frac{1}{n+1}$. For $0 \leq j \leq n + 1$, let $x_j = jh$, and $\hat{u}_j \approx u(x_j)$ be our numerical approximations for the final solution. Then our initial conditions and the approximations give us the following set of equations:

$$-\frac{\hat{u}_{j+1} - 2\hat{u}_j + \hat{u}_{j-1}}{h^2} = f(x_j), \qquad 1 \leq j \leq n \tag{12}$$
$$\hat{u}_0 = \hat{u}_{n+1} = 0. \tag{13}$$

Writing this in terms of matrices:

$$A = \frac{1}{h^2}\begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \vdots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{bmatrix}, \quad \hat{u} = \begin{bmatrix} \hat{u}_1, \\ \vdots, \\ \hat{u}_n \end{bmatrix}, \quad f = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \tag{14}$$
$$A\hat{u} = f \tag{15}$$

We can then use standard matrix solvers to solve this matrix equation. One can show that A is a symmetric positive definite matrix, which implies that $\hat{u}$ has a unique existence.

The analysis of finite difference methods may extend to their convergence and stability properties, especially by analyzing the truncation error. Moreover, there are many ways of utilizing surrounding points to approximate an mth derivative with finite differences. We note that our current approximation of $u''(x)$ in Equation (11) uses centered coefficients $[-1, 2, 1]$. In general, there are several approaches for obtaining accurate coefficients, including Lagrangian interpolation and methods of undetermined coefficients. The coefficients may also generalize to higher order differentials, in which case the coefficients are organized into stencils (Emmons (1944)). For example, a two-dimensional case of the Laplacian operator can be approximated as follows:

$$\Delta u(x, y) = u_{xx}(x, y) + u_{yy}(x, y) \tag{16}$$
$$\approx \frac{u(x+h, y) - 2u(x, y) + u(x-h, y)}{h^2} + \frac{u(x, y+h) - 2u(x, y) + u(x, y-h)}{h^2} \tag{17}$$

which lends itself to the following stencil:

$$\frac{1}{h^2}\begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix}. \tag{18}$$

Thus, finite differences represent an understandable and standard way of approximating PDE solutions. In the next section, we discuss another common numerical method, the finite element method.

III. Finite Element Method

Finite Element Method solves differential equations by subdividing the original system into smaller parts called finite elements. Finite elements are obtained by discretizing the space to create the numerical domain for the solution, which involves finite number of points.

In this section, we demonstrate solving a differential equation with the Finite Element Method following this high-level procedure:

1. Obtain the weak formulation of the problem.
2. Make function space finite-dimensional.
3. Convert the discrete problem into linear systems and solve.

III.I. Equivalent formulations

We denote the scalar product as $(v, w) := \int_\Omega v(x) \cdot w(x) dx$. Define function space

$$V := \{v | v \in \mathcal{C}[0, 1], v' \text{ is piecewise continuous and bounded on [0,1]}, v(0) = v(1) = 0\}.$$

Define energy function

$$F : V \to \mathbf{R}$$
$$v \mapsto \frac{1}{2}(v', v') - (f, v)$$

Our original problem is equivalent to finding $u \in \mathcal{C}^2$ that satisfies $-u_{xx} = f$ with $u(0) = u(1) = 0$.

Furthermore, we can show that the followings are equivalent formulations:

(1) Find $u \in \mathcal{C}^2$ that satisfies $-u_{xx} = f$ with $u(0) = u(1) = 0$.

(2) Find $u \in V$ such that $(u', \phi') = (f, \phi), \forall \phi \in V$.

(3) Find $u \in V$ such that $F(u) \leq F(\phi), \forall \phi \in V$.

Detailed proof is included in the lecture notes in Wick (2020).

III.II. Basis function and dimension-reduction

In our 1-D case, the mesh would be intervals on the axis. In particular, we can let nodal points be equidistant, hence $x_0 = 0 < x_1 < x_2 < \ldots < x_n < 1 = x_{n+1}$ where $x_j = jh$ and $h = \frac{1}{n+1}$ (equivalent to the finite difference method). Now we can formally define a finite element.

Definition: A *finite element* is a triplet (K, P_K, Σ) where K is an element (a cell of the mesh), P_K are polynomials on K, and Σ is a set of degrees of freedom.

Then we proceed to make the function space finite-dimensional by finding a set of basis functions:

$$V_h = V_h^{(k)} := \{v \in \mathcal{C}[0, 1] \mid v_{K_i} \in P^{(k)} \text{ on each element } K_i, v(0) = v(1) = 0\}$$

where $P^{(k)}$ denotes the space of kth order polynomials. Consider the case for linear basis functions. $P^{(1)} := \{a_0 + a_1 x | a_0, a_1 \in \mathbf{R}\}$. All functions within this V_h are hat functions satisfying

$$\phi_j(x) = \begin{cases} 0 & \text{if } x \notin [x_{j-1}, x_{j+1}] \\ \frac{x - x_{j-1}}{h} & \text{if } x \in [x_{j-1}, x_j] \\ \frac{x_{j+1} - x}{h} & \text{if } x \in [x_{j-1}, x_j] \end{cases} \tag{19}$$

with the property

$$\phi_j(x_i) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{20}$$

Then its derivative is

$$\phi_j(x)' = \begin{cases} 0 & \text{if } x \notin [x_{j-1}, x_{j+1}] \\ \frac{1}{h} & \text{if } x \in [x_{j-1}, x_j] \\ \frac{-1}{h} & \text{if } x \in [x_{j-1}, x_j] \end{cases} \tag{21}$$

Note that for each of the function $v_h \in V_h$, we have a unique representation

$$v_h(x) = \sum_{j=1}^{n} v_{h,j} \phi_j(x), \forall x \in [0, 1], v_{h,j} \in \mathbf{R}$$

In this way, we can simplify by working with a discrete function space. Concretely, solving (2) is equivalent to solving

$$\begin{aligned} &\text{Find } u_h \in V_h : (u_h', \phi_h') = (f, \phi_h), \forall \phi_h \in V_h \\ &\Leftrightarrow \left(\left(\sum_{j=1}^{n} u_j \phi_j \right)', \phi_h' \right) = (f, \phi_h) \forall \phi_h \in V_h \\ &\Leftrightarrow \sum_{j=1}^{n} u_j (\phi_j', \phi_h') = (f, \phi_h), \forall \phi_h \in V_h \\ &\Leftrightarrow \sum_{j=1}^{n} u_j (\phi_j', \phi_i') = (f, \phi_i) \text{ for } i \in \{1, 2, ...n\} \end{aligned}$$

III.III. Solve linear system

With the above steps, we now only need to find $(u_j)_{j=1}^n \in \mathbf{R}^n$ such that $\Leftrightarrow \sum_{j=1}^n u_j(\phi_j', \phi_i') = (f, \phi_i)$ for $i \in \{1, 2, ...n\}$. This can be written as a linear system $AU = F$ in which we aim to solve for U, and we have

$$A = \begin{bmatrix} (\phi_1', \phi_1') & (\phi_1', \phi_2') & \cdots & (\phi_1', \phi_n') \\ (\phi_2', \phi_1') & (\phi_2', \phi_2') & \cdots & (\phi_2', \phi_n') \\ \vdots & \vdots & \ddots & \vdots \\ (\phi_n', \phi_1') & (\phi_n', \phi_2') & \cdots & (\phi_n', \phi_n') \end{bmatrix}, U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}, F = \begin{bmatrix} (f, \phi_1) \\ (f, \phi_2) \\ \vdots \\ (f, \phi_n) \end{bmatrix}$$

Using the derivative $\phi_j(x)'$ we noted above, we can derive

$$A_{ij} = \int_\Omega \phi_j(x)' \phi_i(x)' dx = \begin{cases} -\frac{1}{h} & \text{if } j = i - 1 \\ \frac{2}{h} & \text{if } j = i \\ -\frac{1}{h} & \text{if } j = i + 1 \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

The F matrix depends explicitly on the value of f. In the simple case of $f = 1$, we will get $F = \begin{bmatrix} h \\ h \\ \vdots \\ h \end{bmatrix}$.

Finally, by the boundary condition, we know that the first row of matrix A should be $\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix}$ and the last row $\begin{bmatrix} 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$, and the first and last row of F should both be 0. Then we can solve this linear system and obtain numerical numerical approximation of the PDE.

IV. Spectral Methods

Spectral methods presume that the solution to a differential equation can be written as a sum of basis functions (i.e. Fourier series), and then choose coefficients to satisfy the differential equation as closely as possible. Below we show both a linear (Poisson) and nonlinear (heat equation) example.

IV.I. Poisson Equation

For simplicity, suppose that our function $f(x)$ is periodic, so $f(x + 2\pi) = f(x)$. We assume that u and f can each be approximately written as a sum of Fourier series to n terms (where n can be chosen based for the desired accuracy) such that

$$u(x) = \sum_{j=0}^{n} a_j e^{ijx} \tag{23}$$

$$f(x) = \sum_{j=0}^{n} b_j e^{ijx} \tag{24}$$

Then, by plugging this form of both equations into the original PDE, we obtain

$$\sum_{j=0}^{n} -a_j (j^2) e^{ijx} = \sum_{j=0}^{n} b_j e^{ijx} \tag{25}$$

From here it is obvious that if the coefficients of each j match, then the differential equation will match. By the uniqueness theorem of Fourier series, we can equate the series term by term to obtain

$$a_j = -\frac{b_j}{j^2} \tag{26}$$

Note that if $b_0 = 0$, we can freely choose a_0 and the Poisson equation has a solution, which is equivalent to choosing the integration constant. When numerically computing, we compute only a finite number of terms in the series. It can be shown then that the error of the approximation is $O(\frac{1}{n^n})$.

IV.II. Nonlinear Heat Equation

Here we consider a version of the heat diffusion equation where the diffusion coefficient scales linearly with heat itself. This transforms the linear heat diffusion equation into a nonlinear problem

$$u_t - c u u_{xx} = 0 \tag{27}$$

where the diffusion coefficient is now cu. We repeat the same process as above by estimating

$$u \approx \sum_{j=0}^{n} \hat{u}_j e^{ijx} \tag{28}$$

and taking derivatives. Doing so yields

$$\sum_{k=0}^{n} \partial_t(\hat{u}_k) e^{ikx} + \left(c \sum_{l=0}^{n} \hat{u}_l e^{ilx} \right) \sum_{k=0}^{N} k^2 \hat{u}_k e^{ikx} = 0 \tag{29}$$

Unlike in the linear case, there is no easy relationship between the coefficients. In particular, if we expand the multiplication of sums on the right-hand side, our coefficients become $k^2 \hat{u}_l \hat{u}_k$, which takes $O(n^2)$ to compute (as opposed to $O(n)$ for finite difference). See Costa (2004) for the complete derivation.

V. Solving the Heat Equation in Three Dimensions

As an application of the finite element method, we solve the heat equation in 3 dimensions on a cube (Equation (2))[1]. We generalize the approach in this Github repository but develop the relevant expressions for one dimension higher. Our code and work are in the attached notebook. We do not use any sophisticated packages for solving PDE's, other than using scipy for generating a Delaunay triangulation.

VI. Conclusion

This essay briefly explores key concepts in finite difference, finite element, and spectral methods for solving partial differential equations. We summarize the methods and demonstrate each of them applied to the simple one-dimensional Poisson equation. The finite difference method discretizes the space and then approximates all derivatives with finite differences. The finite element method solves differential equations by subdividing the original system into smaller parts called finite elements. Finite elements are obtained by discretizing the space to create the numerical domain for the solution, which involves a finite number of points. Spectral methods presume that the solution to the PDE can be written as a sum of basis functions, and then choose coefficients to satisfy the differential equation as closely as possible. We also show with code the finite element method applied to the 3D heat equation on a cube.

References

Boo Cheong Khoo, Jacob White, Jaime Peraire, and Anthony T. Patera. Lecture notes in numerical methods for partial differential equations, Spring 2003.

Howard W Emmons. The numerical solution of partial differential equations. *Quarterly of Applied Mathematics*, 2(3): 173–195, 1944.

Thomas Wick. Numerical methods for partial differential equations, October 2020.

Bruno Costa. Spectral methods for partial differential equations. *Cubo*, 4, 01 2004.

[1]This project task was graciously provided by Sumit during office hours in lieu of solving the Cahn Hilliard Equation from scratch, given that the latter is very difficult to do without, say, FEniCS. Quote: "Given the state of the situation, just do this."