有限差分法与伪谱法探索

Exploring Finite Difference and Pseudo-Spectral Methods

Abstract

In this work, we present a unified and pedagogical introduction to several fundamental numerical methods for solving partial differential equations arising in computational physics. We begin by reviewing the finite difference method and providing a general framework for analyzing the stability and convergence of schemes for parabolic equations. Both classical von Neumann (Fourier) analysis and the energy/entropy method are discussed, highlighting their respective roles in understanding numerical stability.

We then introduce pseudo-spectral methods and develop the associated Gaussian quadrature and orthogonal polynomial machinery through a streamlined and conceptually transparent derivation, improving upon traditional proofs. Finally, we demonstrate how the time-independent Schrödinger equation can be discretized naturally within the pseudo-spectral framework, yielding accurate and efficient numerical approximations.

Our exposition aims to bridge foundational theory with computational practice, offering a coherent route from basic discretization techniques to modern spectral approaches.

Keywords Finite Difference Method; Stability and Convergence; Pseudo-spectral Methods; Gaussian Quadrature; Computational Physics

1 引言

The numerical solution of partial differential equations (PDEs) plays a central role in computational physics, where accurate and efficient discretization schemes are essential for modeling diffusion, wave propagation, quantum systems, and a variety of multiscale phenomena. Although a wide range of numerical techniques has been developed, a coherent understanding of their stability, convergence, and underlying mathematical structure remains fundamental for both theoretical analysis and practical computation.

This work is organized with the goal of building a conceptual path from classical discretization techniques to modern spectral methods. We begin with the finite difference method, whose simplicity makes it an ideal setting for introducing stability and convergence theory. For parabolic equations, we discuss two complementary tools: the von Neumann Fourier analysis, which provides a frequency-space perspective on amplification factors, and the energy/entropy method, which yields robust stability estimates even when Fourier analysis is not directly applicable.

Moving beyond local discretization, we introduce pseudo-spectral methods, where global basis functions lead to high-order—or even spectral—accuracy. To support this framework, we provide a streamlined derivation of Gaussian quadrature and orthogonal polynomials, emphasizing the structural properties that make spectral methods both powerful and elegant. As a concrete application, we formulate the time-independent Schrödinger equation within the pseudo-spectral setting and examine its resulting discrete eigenvalue problem.

Through this progression, the paper seeks to offer an integrated view of numerical PDE methods, illustrating how classical ideas naturally evolve into spectral techniques that are central to modern computational physics.

2 Finite Difference Methods

The fundamental role of *Schrödinger′ s equation* is no doubt, which is considered as a heat equation by certain variables transformation.

$$i\hbar\frac{\partial}{\partial t}\psi = \hat{H}\psi \tag{2.1}$$

Generally, it′ s demonstrated as below form.

$$u_t = u_{xx} + V(x)u \tag{2.2}$$

To discretize the equation, suppose $x_j = j\Delta x = jh_x$, $t_j = j\Delta t = jh_t$ as finite elements. While to distinguish the discretized form:

$$u(x,t) \to u[i](t) \to u[i,j] \tag{2.3}$$

A second order differentiation thus is represented:

$$u_{xx} = \frac{u[i+1] - 2u[i] - u[i-1]}{h_x^2} = \frac{1}{h^2}D_{x,+}D_{x,-}u[i] \tag{2.4}$$

$$u_t = \frac{1}{h^2}D_{x,+}D_{x,-}u[i] + V[i]u[i] = Hu[i] \tag{2.5}$$

We introduce few common methods in general:[1]

- Forward Euler method:

$$u[i,j] - u[i,j-1] = h_t Hu[i,j-1] \tag{2.6}$$

- Backward Euler method:

$$u[i,j] - u[i,j-1] = h_t Hu[i,j] \tag{2.7}$$

- 2nd order Runge-Kutta (RK2):

$$\begin{aligned} u[i,j+1] - u[i,j-1] &= h_t Hu\left[i, j+\frac{1}{2}\right] \\ u\left[i, j+\frac{1}{2}\right] &= u[i,j] + \frac{h_t}{2}Hu[i,j] \end{aligned} \tag{2.8}$$

The error comes from two sources, the first is the local truncation error due to finite differentiation, the second is the deduction error due to iterations[1]:

$$u_t = D_{t,+}u(x_i,t_j) + \left(u_t - D_{t,+}u(x_i,t_j)\right) = D_{t,+}u(x_i,t_j) + \tau_t[i,j] \tag{2.9}$$

$$\begin{aligned} u_{xx} &= D_{x,+}D_{x,-}u(x_i,t_j) + \left(u_{xx} - D_{x,+}D_{x,-}u(x_i,t_j)\right) \\ &= D_{x,+}D_{x,-}u(x_i,t_j) + \tau_{xx}[i,j] \end{aligned} \tag{2.10}$$

$$\tau[i,j] = \tau_{xx}[i,j] + \tau_t[i,j] = O(h_t) + O(h_x^2) \tag{2.11}$$

However, it only involves the difference error or local truncation error. We should adjust the deduction error:

$$e[i,j] = u(x_i, t_j) - u[i,j] \tag{2.12}$$

$$\begin{aligned} u(x_i, t_{j+1}) - u(x_i, t_j) = \frac{h_t}{h_x^2}(u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j)) \\ + h_t\tau[i,j] \end{aligned} \tag{2.13}$$

Subtract the exact differentiation terms by the real deduction terms, resulting error terms:

$$e[i,j+1] - e[i,j] = \frac{h_t}{h_x^2}(e[i+1,j] - 2e[i,j] + e[i-1,j]) + h_t\tau[i,j] \tag{2.14}$$

Absorbs all terms about $e[i,j]$ as G, a linear operator to be investigated.

$$e[i,j+1] = Ge[i,j] + h_t\tau[i,j] \tag{2.15}$$

Expanding all possible deductions:

$$\begin{aligned} \|e[i,j+1]\| &= \|Ge[i,j]\| + h_t\|\tau[i,j]\| \\ &\leq \left\|G^2e[i,j-1]\right\| + h_t(\|G\tau[i,j-1]\| + \|\tau[i,j]\|) \\ &\leq \left\|G^{j+1}e[i,0]\right\| + h_t\sum_{m=0}^{j}(\|G^m\tau[i,j-m]\|) \end{aligned} \tag{2.16}$$

Suppose the matrix G satisfies *stability* condition or *bounded* in *any* order:

$$\|G^n u[j]\| \leq C\|u[j]\| \tag{2.17}$$

For some scalar C independent of n.

$$\|e[i,j+1]\| \leq C_1\|e[i,0]\| + C_2\max_m|\tau[i,j-m]| \tag{2.18}$$

For possible scalar C_1 and C_2. It implies that we only need to restriction certain condition on τ. If the local truncation error is *consistent*:

$$\max_m\|\tau[i,j-m]\| = O(h_x^2) + O(h_t) \tag{2.19}$$

And the initial deduction error satisfies:

$$\|e[i,0]\| = O(h^2) \tag{2.20}$$

In such, the global true error:

$$\|e[i,j]\| = O(h_x^2) + O(h_t) \tag{2.21}$$

To systematically exemplify this, one commonly say that error is *convergent* if:

$$\|e[i,j]\| \to 0 \text{ if } h_{(\cdot)} \to 0 \tag{2.22}$$

Thus indeed, if operator G is locally *stable* for our estimated region and truncation error is *consistent*:

$$\textit{stability} + \textit{consistency} \Rightarrow \textit{convergence} \tag{2.23}$$

21 von Neumann Analysis

Given $u[i]$ as a discretization of a function, or vector-form with each elements $\{u[i]\}_{j\in\mathbb{Z}}$. One can define the norm:

$$\|u[i]\|_{L^2} = \sum_{i\in\mathbb{Z}} |u[i]|^2 \tag{2.24}$$

A usual way to analyze the stability is to use Fourier analysis[1]. To avoid confusion of i as imaginary unit and as indices, we use j as indices:

$$\hat{u}(\omega) = \frac{1}{2\pi}\sum_{j\in\mathbb{Z}} u[j]|_j \; e^{-i\omega j}, \omega \in [-\pi, \pi] \tag{2.25}$$

Notice $\hat{u}[\omega + 2\pi] = \hat{u}(\omega)$ due to integer indices.

- The shift operator is transformed to a multiplier function:

$$T_1 u[j] = u[j+1] \Rightarrow \widehat{T_1 u}(\omega) = e^{i\omega}\hat{u}(\omega) \tag{2.26}$$

It′ s shown that the differentiation becomes:

$$D_+ u[j] = \frac{1}{h_x}(T_1 u[j] - u[j]) \Rightarrow \widehat{D_+}\hat{u}(\omega) = \frac{1}{h_x}(e^{i\omega} - 1)\hat{u}(\omega) \tag{2.27}$$

- The Parseval equality:

$$\begin{aligned} \|\hat{u}(\omega)\|^2 &= \int_{-\pi}^{\pi} \frac{1}{(2\pi)^2} \sum_{j\in\mathbb{Z}} u^*[j]e^{-i\omega j} \sum_{k\in\mathbb{Z}} u[k]e^{-i\omega k} d\omega \\ &= \frac{1}{2\pi}\sum_{j\in\mathbb{Z}} u^*[j]u[j] = \frac{1}{2\pi}\|u[j]\|^2 \end{aligned} \tag{2.28}$$

To reduce notation, we omit 2π factor. Therefore the linear operator G can be expressed as:

$$u[k, j+1] = Gu[k,j] = \sum_{m=-l}^{n} a_m u[k+m, j] \tag{2.29}$$

$$(\hat{u}[k, j+1]) = \hat{G}(\hat{u}[k,j]) = \sum_{m=-l}^{n} a_k e^{im\omega}(\hat{u}[k,j]) \tag{2.30}$$

By Parseval equality, we only need to care about the operator in transformed domain:

$$\begin{aligned}\|u[k,j+1]\|^2 &= \|\hat{u}[k,j+1]\|^2 \\ &= \int_{-\pi}^{\pi} \left|\hat{G}(\omega)\right|^2 |\hat{u}[k,j]|^2(\omega)d\omega \\ &\leq \max_{\omega} \left|\hat{G}(\omega)\right|^2 \|u[k,j]\|\end{aligned} \tag{2.31}$$

Thus a sufficient condition for stability of a operator G must be:

$$\left\|\hat{G}\right\|_{\infty} < 1 \tag{2.32}$$

Suppose a backward Euler method for heat equation without potential:

$$\frac{1}{h_t}(u[k,j+1]-u[k,j]) = \frac{1}{h_x^2}(u[k+1,j+1]-2u[k,j+1]+u[k-1,j+1]) \tag{2.33}$$

$$\alpha := \frac{h_t}{h_x^2} \tag{2.34}$$

Every differential operator reduce to a linear combination:

$$\begin{aligned}u[k,j+1]-\alpha(u[k+1,j+1]-2u[k,j+1]+u[k-1,j+1])) &= u[k,j] \\ \hat{u}[k,j+1]-\alpha(e^{i\omega}-2+e^{-i\omega})\hat{u}[k,j+1] &= \hat{u}[k,j] \\ \hat{u}[k,j+1]\Big(1+4\alpha\sin^2\Big(\frac{\omega}{2}\Big)\Big) &= \hat{u}[k,j]\end{aligned} \tag{2.35}$$

It' s demonstrated that backward Euler method have no sufficient condition for stability, but it doesn' t mean it has no **such condition**.

$$\hat{G}_{BE}(\omega) = \frac{1}{1+4\alpha\sin^2(\frac{\omega}{2})} \tag{2.36}$$

Thus backward operator has no conditional stability. Readily, one can prove that for forward operator:

$$\hat{G}_{FE}(\omega) = 1-4\alpha\sin^2\Big(\frac{\omega}{2}\Big) \leq 1 \Rightarrow \frac{h_t}{h_x^2} \leq \frac{1}{2} \tag{2.37}$$

22 Energy/Entropy method

We notice that the action of G is a convex combination of $u[k-1,j], u[k,j], u[k+1,j]$, provided:

$$0 < \frac{h_t}{h_x^2} \leq \frac{1}{2} \tag{2.38}$$

$$\min\{u[k-1,j], u[k,j], u[k+1,j]\} \leq u[k,j+1] \leq \max\{...\} \tag{2.39}$$

Which leads to:

$$\begin{aligned}\min_k u[k,j+1] &\geq \min_k u[k,j] \\ \max_k u[k,j+1] &\leq \max_k u[k,j]\end{aligned} \tag{2.40}$$

$$\max_k |u[k,j+1]| \leq \max_k u[k,j] \tag{2.41}$$

For this reason, G is stable in $\|\cdot\|_\infty$. Given any convex combination of $u[k,j]$, with convex function $\eta(\cdot)$, we can extend:

$$\begin{aligned}\eta(u[k,j+1]) \leq \alpha\eta(u[k-1,j]) + \beta\eta(u[k,j]) + \gamma\eta(u[k+1,j]) \\ \alpha+\beta+\gamma = 1\end{aligned} \tag{2.42}$$

Summing over k, we deduce:

$$\sum_k \eta(u[k,j+1]) \leq \sum_k \eta(u[k,j]) \tag{2.43}$$

The convex function is called *entropy* in this setting. The above inequality means the *entropy* decreasing in time. Various convex functions can apply:

- $\eta(u) = u^2 \Rightarrow \|u[k,j+1]\|_2^2 \leq \|u[k,j]\|_2^2$
- $\eta(u) = u^p, 1 \leq p \leq \infty$ results:

$$\|u[k,j+1]\|_p^2 \leq \|u[k,j]\|_p^2 \tag{2.44}$$

Is the general L^p stability indicating $\|G\|_p \leq 1$.

3 PesudoSpectral Methods

To integrate a function, it$'$ s impossible to continuously achieve such while we have to discretize all data set. This force one to interpolate a certain data set as $\{(x_i, f(x_i)\}_{i=1}^N$. To decompose problems, one can regard data set as a combination of delta functions.

$$\{(x_i, f(x_i))\} \Rightarrow \{(x_i, f(x)\delta(x-x_i))\} \Rightarrow f(x)\{(x_i, \delta(x-x_i))\} \tag{3.45}$$

It allows us to analyze the function which must be a basis combination sharing the same property with delta function.

$$\delta(x-x_i) \rightsquigarrow l_i(x) \text{ s.t. } l_i(x_j) = \delta_{ij} \tag{3.46}$$

A function therefore can be approximated:

$$\begin{aligned}f(x) &\approx \sum_{i=1}^N f(x_i) l_i(x) \\ l_i(x_j) &= \delta_{ij}\end{aligned} \tag{3.47}$$

As for the integral of a function, it also must be the integral of each basis.

$$
\begin{aligned}
I[f] \coloneqq \int_a^b f(x)dx \approx \int_a^b p(x)dx = \int \sum_{i=0}^{N} f(x_i) l_i(x)dx = \sum_{i=0}^{N} w_i f(x_i) \\
w_i = \int_a^b l_i(x)dx
\end{aligned}
\tag{3.48}
$$

We call w_i as weight of basis. Suppose one want to interpolate a certain function by n-degree polynomials as basis, then the basis itself must be exactly match our interpolation.

$$
\int_a^b \sum_{i=0}^{n} a_i x^i dx = \sum_{i=0}^{n} a_i \frac{b^{i+1} - a^{i+1}}{(i+1)!} \tag{3.49}
$$

Interpolating itself suggests that weights must be a linear combination of the integral of itself.

$$
I_n \left[\sum_{i=0}^{n} a_i x^i \right] = \sum_{j=0}^{n} w_j \left(\sum_{i=0}^{n} a_i x_j^i \right) = \sum_{j=0}^{n} \left(w_j x_j^i \right) \sum_{i=0}^{n} a_i \tag{3.50}
$$

$$
\sum_{j=0}^{n} w_j x_j^i = \frac{b^{i+1} - a^{i+1}}{(i+1)!} \quad \forall i \in \{1, ..., n\} \tag{3.51}
$$

As a results, one should solve such *Vandermonde* matrix.

$$
\mathbf{V}^T \mathbf{w} = \mathbf{b},\, b_i = \int_a^b x^i dx \tag{3.52}
$$

In essence, a weight must be a linear combination of x^i.

$$
w_i = \int_a^b l_i(x)dx \overset{\text{polynomial}}{\Rightarrow} w_i = \sum_{j=0}^{n} a_{ij} \int_a^b x^j dx \tag{3.53}
$$

$$
\mathbf{w} = \mathbf{A}\mathbf{b} \tag{3.54}
$$

If $\mathbf{A} = \mathbf{V}^{(-1)^T}$, one finds the same weights produced by interpolating polynomial exactly. However, polynomials in evenly spaced grid isn$'$ t well-behaved in many conditions. Recall that Runge$'$ s function $f(x) = \frac{1}{1+x^2}$ in uniformly spaced points interpolation fails to uniformly converge to itself in the defect of boundary divergence. A expedience is *Clenshaw-Curtis* quadrature:

$$
x_j = \cos\left(\frac{j\pi}{n} \right) \in [-\pi, \pi], \quad j = 0, ..., n \tag{3.55}
$$

Such spaced points distributes tensely on the boundaries while sparely on the middle. Nonetheless, such interpolation is not satisfied which leading to *Gaussian Quadrature*.

31 Gaussian Quadrature

A approximation of function is a linear combination of basis functions.

$$
\begin{aligned}
f_n(x) &= \sum_{j=1}^{n} f(x_j) l_j(x) \\
f(x) &= \sum_{j\geq 1} f(x_j) l_j(x)
\end{aligned}
\tag{3.56}
$$

A insight is that we have maximally $2n$ unknown variables containing n grid x_j and n weights w_j. Hence we can approximate a $2n$ polynomial **exactly**.

$$
I_n[f] = \int_I f(x)dx = \sum_{j=0}^{n} f(x_j) \int_I l_j(x)dx = \sum_{j=0}^{n} f(x_j) w_j \tag{3.57}
$$

To adapt general space, we introduce a weight function for various possible basis.

$$
I_n[f] = \int_a^b f(x)w(x)dx = \sum_{j=0}^{n} f(x_j) \int_a^b l_j(x)w(x)dx = \sum_{j=0}^{n} f(x_j) w_j \tag{3.58}
$$

$$
w_j \coloneqq \int_a^b l_j(x)w(x)dx \tag{3.59}
$$

First, we adapt basis itself, which shows the common property of basis.

$$
\begin{aligned}
I_n\left[l_j\right] = \int_a^b l_j(x)w(x) = \sum_{k=0}^{n} l_j(x_k) \int_a^b l_k(x)w(x)dx \\
\Rightarrow l_k(x_j) = \delta_{kj}
\end{aligned}
\tag{3.60}
$$

A intriguing phenomena is that we plugin two basis into the integration, where $j,k < n \Rightarrow j+k < 2n$ must be exactly integrated:

$$
\begin{aligned}
I_n\left[l_j l_k\right] = \int_a^b l_j(x)l_k(x)w(x)dx = \sum_{l=0}^{n} l_j(x_l)l_k(x_l) \int_a^b l_l(x)w(x)dx \\
\int_a^b l_j(x)l_k(x)w(x)dx = \sum_{l=0}^{n} l_j(x_l)l_k(x_l)w_l = \delta_{jk}w_j
\end{aligned}
\tag{3.61}
$$

It follows that basis is *orthogonal* to each other. Consequently, a basis must be a linear combination of *orthogonal polynomials*. One plugin the orthogonal polynomials in with suitable degree less than n.

$$
I_n[p_{n-1}p_{m-1}] = \int_I p_{n-1}p_{m-1}wdx = \sum_{l=0}^{n} p_{n-1}(x_l)p_{m-1}(x_l)w_j = \delta_{nm} \tag{3.62}
$$

Now, here the crucial point. The orthogonal polynomials can be necessarily expanded into a delta function which is also a **linear combination** of itself.

$$
\begin{aligned}
p_{m-1}(x) = \sum_{k=1}^{n} p_{k-1}(x)\delta_{kn} &= \sum_{k=1}^{n} p_{k-1}(x) \sum_{l=1}^{n} p_{k-1}(x_l) p_{m-1}(x_l) w_l \\
&= \sum_{l=1}^{n} p_{m-1}(x_l) \sum_{k=1}^{n} p_{k-1}(x) p_{k-1}(x_l) w_l \\
&= \sum_{l=1}^{n} p_{m-1}(x_l) I_l(x)
\end{aligned} \tag{3.63}
$$

Consequently, a basis is represented below:

$$
l_i(x) = w_i \sum_{k=1}^{n} p_{k-1}(x) p_{k-1}(x_i) \tag{3.64}
$$

Although we still don′ t know the property of x_j and w_j. It pushes us to evaluate orthogonal polynomials. A essential property is orthogonal polynomial must be orthogonal to any polynomials with degree smaller than itself.

$$
\langle x p_n(x), p_k(x) \rangle = \langle p_n(x), x p_k(x) \rangle = 0 \Rightarrow k + 1 < n \tag{3.65}
$$

It means, $xp_n(x)$ should be a linear combination of only $n+1, n, n-1$-degree.

$$
\begin{aligned}
x p_n(x) &= A_n p_{n+1} + B_n p_n + C_n p_{n-1} \\
x p_{n+1}(x) &= A_{n+1} p_{n+2} + B_{n+1} p_{n+1} + C_{n+1} p_n
\end{aligned} \tag{3.66}
$$

It can be verified that $C_{n+1} = A_n$:

$$
C_{n+1} = \langle x p_{n+1}, p_n \rangle = \langle p_{n+1}, x p_n \rangle = A_n \tag{3.67}
$$

$$
x p_n(x) = C_{n+1} p_{n+1} + B_n p_n + C_n p_{n-1} \tag{3.68}
$$

For basis l_{n+1}, the terms $\sum_n p_n(x) p_n(x_i)$ is the key, motivating us to collects terms resemble to it.

$$
\begin{aligned}
x p_n(x) p_n(y) &= C_{n+1} p_{n+1}(x) p_n(y) + B_n p_n(x) p_n(y) + C_n p_{n-1}(x) p_n(y) \\
y p_n(x) p_n(y) &= C_{n+1} p_n(x) p_{n+1}(y) + B_n p_n(x) p_n(y) + C_n p_n(x) p_{n-1}(y)
\end{aligned} \tag{3.69}
$$

Subtract above two terms:

$$
\begin{aligned}
(x-y) p_n(x) p_n(y) = C_{n+1} &(p_{n+1}(x) p_n(y) - p_n(x) p_{n+1}(y)) \\
&- C_n (p_n(x) p_{n-1}(y) - p_{n-1}(x) p_n(y))
\end{aligned} \tag{3.70}
$$

We define $K_n(x, y)$ to simplify the notation:

$$
\begin{aligned}
K_n(x, y) &= p_{n+1}(x) p_n(y) - p_n(x) p_{n+1}(y) \\
K_{-1}(x, y) &= 0 - 0 = 0
\end{aligned} \tag{3.71}
$$

Summation of terms yields another expression of basis:

$$\begin{aligned}\sum_{i=0}^{n}(x-y)p_i(x)p_i(y) &= \sum_{i=0}^{n}(C_{n+1}K_n - C_nK_{n-1}) \\ &= \sum_{i=0}^{n}C_{n+1}K_n - \sum_{i=0}^{n-1}C_{n+1}K_n - K_{-1} \\ &= C_{n+1}K_n\end{aligned} \tag{3.72}$$

$$\sum_{i=0}^{n}p_i(x)p_i(y) = \frac{C_{n+1}K_n}{x-y} \tag{3.73}$$

To acquire the meaning of x_j, we use the fresh expression:

$$\begin{aligned}\delta_{jk} = l_k(x_j) &= w_j\sum_{i=0}^{n-1}p_i(x_j)p_i(x_k) \\ &= w_jC_{n+1}\frac{K_{n-1}(x_j,x_k)}{x_j-x_k} \\ &= w_jC_{n+1}\frac{p_n(x_j)p_{n-1}(x_k)-p_{n-1}(x_j)p_n(x_k)}{x_j-x_k}\end{aligned} \tag{3.74}$$

To force zero if $j \neq k$, we must have:

$$\begin{aligned}p_n(x_j)p_{n-1}(x_k) &= p_{n-1}(x_j)p_n(x_k) \\ \frac{p_n(x_j)}{p_{n-1}(x_j)} &= \frac{p_n(x_k)}{p_{n-1}(x_k)} \stackrel{?}{=} 0\end{aligned} \tag{3.75}$$

Indeed, it' s impossible unless we choose x_j, x_k both the roots of p_n or p_{n-1} orthogonal polynomials. To investigate the roots coincidence, we treating x_k as constant while manipulating x_j as variables:

$$\begin{aligned}\lim_{x_j\to x_k}\text{LHS} &= \frac{\frac{\mathrm{d}}{\mathrm{d}x_j}(p_n(x_j)p_{n-1}(x_k)-p_{n-1}(x_j)p_n(x_k))}{\frac{\mathrm{d}}{\mathrm{d}x_j}(x_j-x_k)} \\ &= p_n'(x_j)p_{n-1}(x_j) - p_{n-1}'(x_j)p_n(x_j) = \frac{1}{w_jC_{n+1}}\end{aligned} \tag{3.76}$$

For choice of roots, we follow the sign matter, choosing $x_j = x_k$ is the roots of p_n orthogonal polynomial. Derives the final expression of weight:

$$w_j = \frac{1}{C_{n+1}p_n'(x_j)p_{n-1}(x_j)} \tag{3.77}$$

$$l_k(x) = w_jC_{n+1}\frac{p_n(x)p_{n-1}(x_k)}{x-x_k} \tag{3.78}$$

A immediate consequence is any x_j will be the zeros of $p_n(x)$ for a certain basis set $\{l_i\}_{i=1}^n$. Take the weight into the basis expression:

$$l_k(x_k) = \frac{p_n(x)p_{n-1}(x_k)}{p_n'(x_k)p_{n-1}(x_k)(x-x_k)} = \frac{p_n(x)}{p_n'(x_k)(x-x_k)} \tag{3.79}$$

We recover l_k as the Lagrange basis in polynomial p_n, by removing the x_k zero. It shown that, given any orthogonal polynomials set, one only needs to compute the set of zeros and differentiations of polynomials at zeros which is pretty easy for nowadays computer.

Another choice of basis comes from *Shannon* and *Whittaker*[2]. We only needs two property to define the basis by zeros and orthogonality. It suggests that the orthogonality can be derived from Fourier analysis.

$$\operatorname{sinc}(x) = \frac{\sin(\pi x)}{\pi x} \tag{3.80}$$

The integral form of the sinc function is depicted by characteristic function:

$$\operatorname{sinc}(x) = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-i\omega x}dx = \hat{\chi}_{[-\pi,\pi]} \tag{3.81}$$

$$\chi_{[a,b]}(\omega) = \begin{cases} 1 & \text{if} -\pi \le \omega \le \pi \\ 0 & \text{otherwise} \end{cases} \tag{3.82}$$

It shown that orthogonality inherits from characteristic function by $\chi_{[-\pi+n\pi,\pi+n\pi]} \cdot \chi_{[-\pi+m\pi,\pi+m\pi]} = 0$:

$$\int_{-\infty}^{\infty} \operatorname{sinc}(x-m)\operatorname{sinc}(x-n)dx = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{i(n-m)\omega}d\omega = \delta_{nm} \tag{3.83}$$

So the only left part is zeros property, which must be $x = j, j \in \mathbb{Z}$.

$$l_j(x) = \operatorname{sinc}\left(\frac{x-x_j}{h}\right) \quad x_j = x_0 + j\delta x = x_0 + jh_x \tag{3.84}$$

$$l_j(x_i) = \begin{cases} 1 & \text{if } x_i = x_j \\ \sin(\pi(i-j)) = 0 & \text{if } x_i \neq x_j \end{cases} \tag{3.85}$$

$$\int_{-\infty}^{\infty} l_i(x)l_j(x)dx = h\delta_{ij} \tag{3.86}$$

In which, we constructs the basis entirely.

32 Application as Spectral Decomposition

Based on previous achievements, we can finally turn into the construction of *pesudospectral* methods. The key is not to find the spectral basis of certain linear operator, but to approximate the answer by predefined basis in Gaussian quadrature.

$$\frac{1}{\sqrt{w_i}}\int_I \sqrt{w(x)}l_i(x)\sqrt{w(x)}l_j(x)dx = \delta_{ij} \tag{3.87}$$

Define the basis function as:

$$\sqrt{\frac{w(x)}{w_i}} l_i(x) = \phi_i(x) \tag{3.88}$$

For the same reason, we omit w_i for simplicity.

Any function defined on the same domain of basis, can be decomposed by basis function with the merits of integration approximation:

$$\psi(x) = \sqrt{w(x)} \sum_{j \geq 1} \psi(x_j) l_j(x) = \sum_{j \geq 1} \psi(x_j) \phi_j(x) \tag{3.89}$$

$$\int_\Omega \psi(x)^* \psi(x) dx = \sum_{j \geq 1} \psi^*(x_j) \psi(x_j) \int_\Omega \phi^*(x) \phi(x) dx = \sum_{j \geq 1} \psi^2(x_j) \tag{3.90}$$

Based on the approximation of basis, the error should be $O(2n)$ order in integration.

A differential equation is usually a form of:

$$L\psi(x) = g(x) \tag{3.91}$$

Take the product of basis function and integrates in the space:

$$\begin{aligned} \int_\Omega \phi_j(x) L\psi(x) dx &= \int_\Omega \phi_j(x) g(x) dx \\ \sum_{i \geq 1} \psi(x_j) \int_\Omega \phi_j(x) L\, \phi_i(x) dx &= \int_\Omega \phi_j(x) g(x) dx \end{aligned} \tag{3.92}$$

One define the discretization of the equation, simplify the integration problem into a matrix problem.

$$\mathbf{L} = \int_\Omega \phi_j L\, \phi_i dx,\, \mathbf{g} = \int_\Omega \phi_j\, g\, dx, \mathbf{c} = \psi(x_j) \tag{3.93}$$

The eigenvalue problem follows the same routine:

$$L\psi^m(x) = \lambda_m \psi^m(x) \Rightarrow \mathbf{L}\mathbf{c}_m = \lambda_m \mathbf{c}_m \tag{3.94}$$

We present the reduction of Schrödinger′ s equation:

$$\hat{H}\psi = -\frac{\hbar^2}{2\mu} \frac{\partial^2}{\partial x^2} \psi + V\psi = E\psi \tag{3.95}$$

Next, we can use bracket symbol to formulate the basis that every integration is a summation of basis:

$$\langle i | n \rangle = \sqrt{w_i} l_n(x_i) \tag{3.96}$$

$$|i\rangle = |n\rangle\langle n|i\rangle = \sum_{n \geq 1} \sqrt{w_i} l_n(x_i) |n\rangle \tag{3.97}$$

The orthogonality property is illustrated below:

$$\begin{aligned}\langle j|i\rangle = \langle j|n\rangle\langle n|i\rangle = \sum_{n\geq 1}\sqrt{w_i w_j}l_n(x_i)l_n(x_j) = \delta_{ij}\\ \langle n|m\rangle = \langle n|i\rangle\langle i|m\rangle = \sum_{i\geq 1} w_i l_n(x_i)l_m(x_i) = \delta_{mn}\end{aligned} \tag{3.98}$$

Potential is a diagonal matrix:

$$\begin{aligned}\hat{V}|j\rangle = V(x_j)|j\rangle\\ V_{mn} = \left\langle i\middle|\hat{V}\middle|j\right\rangle = V(x_i)\delta_{ij}\end{aligned} \tag{3.99}$$

$$\hat{K} = \frac{\hbar^2}{2\mu}\hat{D}^\dagger\hat{D} \tag{3.100}$$

While kinetic is more complex involving the differential of basis[2]:

$$K_{ij} = \frac{\hbar^2}{2\mu}\left\langle i\middle|\hat{D}^\dagger\hat{D}\middle|j\right\rangle = \frac{\hbar^2}{2\mu}\left\langle i\middle|\hat{D}^\dagger\middle|k\right\rangle\left\langle k\middle|\hat{D}\middle|j\right\rangle = \frac{\hbar^2}{2\mu}\sum_{k\geq 1} D_{ki}D_{kj} \tag{3.101}$$

$$D_{ki} = \left\langle k\middle|\hat{D}\middle|n\right\rangle\langle n|i\rangle = \sqrt{w_k w_i}\sum_{n\geq 1} l_n'(x_k)l_n(x_i) \tag{3.102}$$

The Hamiltonian is:

$$H_{ij} = \frac{\hbar^2}{2\mu}\sum_{k\geq 1} D_{ki}D_{kj} + V(x_i)\delta_{ij} \tag{3.103}$$

4 结论

In this work, we have presented a coherent introduction to several core numerical methods used in computational physics, emphasizing the stability, convergence, and structural properties underlying their effectiveness. Beginning with finite difference schemes, we examined general stability theory for parabolic equations through both Fourier (von Neumann) analysis and the energy/entropy method, highlighting the complementary insights offered by these approaches.

We then turned to pseudo-spectral methods and provided a streamlined development of the Gaussian quadrature and orthogonal polynomial framework that supports them. This derivation not only simplifies classical proofs but also clarifies the mathematical mechanisms responsible for the high-order accuracy characteristic of spectral techniques. As an application, we demonstrated how the time-independent Schrödinger equation can be discretized within the pseudo-spectral setting, resulting in an efficient and accurate numerical representation.

Taken together, these elements form a continuous progression from local, low-order discretizations to global, high-accuracy spectral methods. We hope that the exposition offered here serves as a useful reference for readers seeking both conceptual understanding and practical tools for solving PDEs in computational physics.

参考文献

1. Chern, I.-L.: Finite Difference Methods for Solving Differential Equations. (2018)

2. Lo, J.Q.W.: Pseudospectral Methods in Quantum and Statistical Mechanics, https://open.library.ubc.ca/media/stream/pdf/24/1.0066507/1, (2008)