Math 6008 Numerical PDEs–Lecture 2

Basic concepts in numerical PDE

Instructor: Lei Li, INS, Shanghai Jiao Tong University;
Email: leili2010@sjtu.edu.cn

---

1 Local Truncation error for finite difference approximation to derivatives

Suppose that $\mathcal{D}^m$ is some finite difference operator for $u^{(m)}(x)$ at point x, with step sizes $h_1, h_2, \dots, h_p$. Let $h = \max_i h_i$. The so-called truncation error for a smooth function $u(x)$ is

$$TE = \mathcal{D}^m u - u^{(m)}(x).$$

If for any smooth function u, there exists C_u which is not zero for some u such that

$$TE = C_u h^p + o(h^p),$$

we say the *order of accuracy* is p and $\mathcal{D}^m$ is called a p-th order approximation.

To determine the order, we do Taylor expansion:

$$u(x+h) = e^{h\frac{d}{dx}} u(x) = \sum_{n=0}^{\infty} \frac{h^n u^{(n)}(x)}{n!}.$$

By Taylor expansion, it's clear to see that the one-sided difference is first order accuracy:

$$TE = D_+ u(x) = \frac{1}{h}[u(x+h) - u(x)] = \frac{1}{h}(hu'(x) + \frac{1}{2}h^2 u''(x) + \dots) = u'(x) + O(h)$$

The centered difference D^2 is of second order accuracy ($TE = O(h^2)$).

Example: Show that

$$D_3 u(x) = \frac{1}{6h}[2u(x+h) + 3u(x) - 6u(x-h) + u(x-2h)]$$

gives a third order accuracy approximation.

Sample Code Presentation: Comparison of the three different finite differences.

2 Establishing finite difference schemes for PDEs

Currently, to approximate $u'(x)$, we have by Taylor expansion

$$D_+u(x) = \frac{u(x+h)-u(x)}{h} = u'(x)+O(h), \quad D_-u(x) = \frac{u(x)-u(x-h)}{h} = O(h)$$

and

$$D_0u(x) = \frac{u(x+h)-u(x-h)}{2h} = O(h^2).$$

To approximate $u''(x)$, we may do

$$D^2u(x) = \frac{u(x+h)-2u(x)+u(x-h)}{h^2} = D_+D_-u(x) = D_-D_+u(x) = u''(x)+O(h^2).$$

Consider approximating PDEs. Here, we take the advection equation (对流方程) (or transport equation, 输运方程)

$$\partial_t u + a\partial_x u = 0, \quad u(x,0) = g(x)$$

the diffusion equation (扩散方程)

$$\partial_t u = a\partial_x^2 u, \quad u(x,0) = g(x)$$

and the simple 1D elliptic equation (ODE with boundary conditions)

$$-u''(x) = f(x), \quad u(0) = u(1) = 0$$

as the examples.

Consider using **time step** τ and **spatial step** h. Set

$$t_n = n\tau, \quad x_j = jh.$$

2.1 The advection equation and the diffusion equations

Consider the advection equation. Using the forward one-sided difference for time and space, we may obtain the following equation for u_j^n, which is the numerical solution at (x_j, t_n):

$$\frac{u_j^{n+1}-u_j^n}{\tau} + a\frac{u_{j+1}^n - u_j^n}{h} = 0, \quad n = 0, 1, 2, \cdots, \quad , j = 0, \pm 1, \pm 2, \cdots.$$

Such equation is called the **finite difference equation** for the advection equation. This equation may be rewritten as

$$u_j^{n+1} = u_j^n - a\lambda(u_{j+1}^n - u_j^n), \tag{1}$$

where

$$\lambda = \frac{\tau}{h}.$$

The solutions for the same t_n is considered to be in the same "time level". Since the method only involves the data for two time levels in the formula, so it is a **two-level scheme** (两层格式). With the initial conditions, we may have

$$u_j^0 = \varphi_j$$

so that the solutions can be solved level by level. This thus forms a **finite difference method** for the advection equation. In this method, there is no dependence on the data at t_{n+1} for the spatial discretization so one can directly compute u_j^{n+1} using the formula from u_j^n. Such a scheme is called an **explicit scheme** (显式格式).

Of course, if one uses the centered difference for the spatial discretization, one may obtain another method,

$$\frac{u_j^{n+1} - u_j^n}{\tau} + a\frac{u_{j+1}^n - u_{j-1}^n}{2h} = 0$$

This is also a two-level explicit scheme.

In the *diffusion equation*, we have second order derivatives. We may use the D^2 operator above. Hence, one possible finite difference equation for the diffusion equation is

$$\frac{u_j^{n+1} - u_j^n}{\tau} - a\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2} = 0$$

or

$$u_j^{n+1} = u_j^n + a\mu(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$$

where $\mu = \tau/h^2$. Clearly, this scheme is also explicit and two-level. To solve this equation, we need to know the initial conditions, or data at level 0

$$u_j^0 = g_j.$$

Of course, for the time derivative, we may use the backward fintie difference to approximate the time derivative so that we have

$$\frac{u_j^n - u_j^{n-1}}{\tau} - a\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2} = 0$$

or equivalently,

$$\frac{u_j^{n+1} - u_j^n}{\tau} - a\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{h^2} = 0.$$

This can be rewriten as

$$-a\mu u_{j+1}^{n+1} + (1+2a\mu)u_j^n - a\mu u_{j-1}^n = u_j^{n-1}.$$

Again, we need the initial data

$$u_j^0 = g_j$$

to solve and again we can solve them level by level. However, the feature of such schemes is that the approximation of the spatial derivatives involves the solutions in the **unknonw level**. We cannot compute u_j^{n+1} directly using the data u_j^n. These are the **implicite schemes** (隐式格式). To solve one level, we have to formula an equation and solve it

$$AU^n = U^{n-1}$$

where $U^n = (u_j^n)_{j=0,\pm1,\cdots}$.

2.2 The elliptic equation

Let us consider the 1D elliptic equation $-u''(x) = f$ with boundary condition. We can approximate the second order derivatives using D^2 and obtain

$$-\frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} = f(x_j), \quad u_0 = u_N = 0.$$

For such equations, clearly, we cannot solve u_j from level to level. If we know u_0, we cannot use u_0 to find u_1 and use u_1 to find u_2. Instead, we must formulate a system of equations for **all values** u_j and solve them all at once!

Hence, there is no concept of time and thus no time level.

3 Other methods for deriving the finite difference methods

3.1 Integration oriented methods

The integration is often used for the finite volume methods which are about the cell averages (the unknowns are the average of solutions over a cell while in FDM, the unknowns are the point values at the grids). However, the integration method can also be used to derive the FDM using numerical integration.

If you are interested in this, you may read the book. Here, we skip this.

3.2 Method of lines discretizations

(Sec. 9.2 and 10.2 in the book of Leveque)

Idea: Approximate the spatial differential operators with finite difference and then we get a system of ODEs. Applying suitable ODE solvers, we then get the discretization of the PDEs.

- For the heat equation $u_t = u_{xx}$, we can approximate u_{xx} by the centered difference and have

$$u_j'(t) = \frac{1}{h^2}(u_{j+1}(t) - 2u_j(t) + u_{j-1}(t)).$$

  If we then apply the forward Euler method, we obtain the scheme:

$$\frac{u_j^{n+1} - u_j^n}{k} = \frac{1}{h^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n),$$

  where u_j^n means the the numerical value at $x_j = jh, t^n = nk$.

- For the advection equation $u_t + au_x = 0$ on $[0,1]$ with periodic boundary condition $u(0,t) = u(1,t)$ (if it's not periodic, then, the boundary condition must be imposed at the boundary where the characteristics come out), we may again use centered difference:

$$u_j'(t) = -\frac{a}{2h}(u_{j+1} - u_{j-1}).$$

With the forward Euler, we have

$$\frac{u_j^{n+1} - u_j^n}{k} = -\frac{a}{2h}(u_{j+1}^n - u_{j-1}^n).$$

In the methods generated by MOL, the same ODE solver is used for all aspects of the spatial discretization, which is sometimes not efficient and not appropriate. MOL, however, provides a useful tool and it is also helpful for understanding the stability.

4 The local truncation error and consistency

Given a scheme for a PDE, we use the local truncation error to measure the consistency.

Consider for example the heat equation

$$u_t = au_{xx}$$

and its corresponding scheme

$$\frac{u_j^{n+1} - u_j^n}{\tau} - a\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2} = 0. \tag{2}$$

Let $u(x,t)$ be the **exact solution** of the heat equation. We insert the exact solution $u(x,t)$ into the numerical scheme and determine how well it satisfies the PDE.

Define the **local truncation error** to be

$$T(x_j, t_n) = \frac{u(x_j, t_{n+1}) - u(x_j, t_n)}{\tau} - a\frac{u(x_{j+1}, t_n) - 2u(x_j, t_n) + u(x_{j-1}, t_n)}{h^2}.$$

Hence, by Taylor expansion, the local truncation error is given by

$$T(x,t) = \frac{u(x,t+\tau)-u(x,t)}{\tau} - \frac{1}{h^2}(u(x+h,t)-2u(x,t)+u(x-h,t))$$
$$= (u_t - u_{xx}) + \frac{1}{2}u_{tt}\tau + \frac{1}{12}u_{xxxx}h^2 + \dots$$

The error is $O(\tau + h^2)$. As we mentioned, the local truncation error on one hand measures how the exact solution satisfies the numerical scheme. Another way to understand it is: provided that the data at t_n is accurate $u^n = u(t_n)$, the error introduced at t_{n+1} is given by $\tau T(x_j, t_n)$.

For the implicit scheme,

$$\frac{u_j^{n+1} - u_j^n}{\tau} - a\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{h^2} = 0,$$

the local truncation error is defined as

$$T(x_j, t_n) = \frac{u(x_j,t_n) - u(x_j, t_{n-1})}{\tau} - a\frac{u(x_{j+1},t_n) - 2u(x_j,t_n) + u(x_{j-1},t_n)}{h^2}.$$

Using Taylor expansion, we find

$$T(x_j, t_n) = O(\tau) + O(h^2).$$

Exercise Derive the local truncation error of the following scheme for the advectin equation.

$$\frac{u_j^{n+1} - u_j^n}{\tau} + a\frac{u_{j+1}^n - u_{j-1}^n}{2h} = 0$$

Definition 1. *If the local truncation error of a scheme for a PDE is like $T = O(\tau^p) + O(h^q)$, then we say the scheme is of order p for temporal discretization and of order q for spatial discretization. If $p = q$, we say the scheme is of order p.*

Clearly, for the diffusion equation, the schemes we considered are both of first order in time and of second order in space.

Definition 2. *We say a finite difference scheme is consistent if $T(x,t) \to 0$ as $h \to 0, \tau \to 0$.*

In other words, as the step sizes become small, the solution of the PDE satisfies the numerical method very well and thus the numerical method is "close" to the PDE.

Similarly, for numerical schemes to *elliptic* equations, we can define the local truncation error and consistency as well.

Definition 3. *Consider an FDM $L_h(u_j) = f(x_j)$ for an elliptic equation $Lu = f(x)$. The local truncation error is*

$$T(x_j) = L_h(u_j) - f(x_j).$$

If $T(x_j) \to 0$ as $h \to 0$, we say the method is consistent.

5 Convergence

One natural questions is: if a scheme is consistent, will the numerical solution be always good?

Clearly, we care about whether the **error** $u(x_j, t_n) - u_j^n$ goes to zero or not.

Definition 4. *If $h, \tau \to 0$, we have the error $\|e_j^n\| = \|u(x_j, t_n) - u_j^n\| \to 0$, then we say the numerical method is convergent.*

Here, $\|e_j^n\|$ is some way to gauge the bigness of the errors $(e_j^n), n = 1, \cdots, N, j = 0, \pm 1, \cdots$. Such a quantity is called a "norm" (范数) in math. A typical norm we can use is $\|e_j^n\|_\infty = \max |e_j^n|$.

Consider the scheme (1) for the advection equation $u_t + au_x = 0$:

$$u_j^{n+1} = u_j^n + a\lambda(u_j^n - u_{j+1}^n),$$

with the initial value $u_j^0 = g_j$. From this scheme, we find that the solution u_j^n only depends on g_k for $j \le k \le j + n$ or the initial values at $x_j, x_{j+1}, \cdots, x_{j+n}$. However, the exact solution is given by

$$u(x, t) = g(x - at).$$

Hence, it depends on the initial value at some point less than x_j. Therefore, for such a method, the numerical solution cannot converge!

Exercise Derive the expression of u_j^n.

This example tells us that

A consistent scheme is not necessarily convergent.

Consider the scheme (2) for the diffusion equation. By the definition of the local truncation error, we have

$$u(x_j, t_{n+1}) = (1 - 2a\mu)u(x_j, t_n) + a\mu[u(x_{j+1}, t_n) + u(x_{j-1}, t_n)] + \tau T(x_j, t_n).$$

The corresponding scheme is

$$u_j^{n+1} = (1 - 2a\mu)u_j^n + a\mu[u_{j+1}^n + u_{j-1}^n].$$

Taking the difference, one has

$$e_j^{n+1} = (1 - 2a\mu)e_j^n + a\mu[e_{j+1}^n + e_{j-1}^n] - \tau T(x_j, t_n),$$

so that

$$|e_j^{n+1}| \leq (1 - 2a\mu)|e_j^n| + a\mu(|e_{j+1}^n| + |e_{j-1}^n|) + M\tau(\tau + h^2).$$

Define

$$E^n := \sup_j |e_j^n| = \|e_\cdot^n\|_\infty.$$

We thus have

$$|e_j^{n+1}| \leq (1 - 2a\mu)E^n + a\mu(E^n + E^n) + M\tau(\tau + h^2).$$

Hence,

$$E^{n+1} \leq E^n + M\tau(\tau + h^2)$$

if $2a\mu \leq 1$.

Hence, we find that if $2a\mu \leq 1$, the scheme for the diffusion scheme is convergence and the error is of the same order as the local truncation error.

6 Stability

A natural question is: from consistency to convergence, what do we need? The answer is **stability** (稳定性). Now try to investigate this issue.

In the numerical simulation, there could be some errors introduced. One type is the initial error. Another one is the one step error introduced $\tau T(x_j, t_n)$. These errors will be propagated along the numerical computation and then eventually accumulate in the final global error. Hence, one very important requirement is that these errors will not be amplified very much to conceal the true soluion. This is the notion of stability .

Consider again the following scheme for the advection equation

$$u_j^{n+1} = u_j^n + a\lambda(u_j^n - u_{j+1}^n) = u_j^n - a\lambda(u_{j+1}^n - u_j^n).$$

The error e_j^n satisfies

$$e_j^{n+1} = e_j^n - a\lambda(e_{j+1}^n - e_j^n) - \tau T(x_j, t_n).$$

Suppose that there is error at $t_n = 0$: $-\tau T(x_j, 0) = \epsilon(-1)^j$ where ϵ is very small and $\tau T(x_j, t_n) = 0$ for $n \geq 1$.

Hence, equivalently, we have

$$e_j^1 = \epsilon(-1)^j$$

and

$$e_j^{n+1} = e_j^n - a\lambda(e_{j+1}^n - e_j^n), \quad n \geq 1.$$

For this relation, we can find that $e_j^n = (-1)^j v_n$ (this is can be obtained by the discrete Fourier analysis) and thus

$$v_{n+1} = v_n(1 + 2a\lambda).$$

Hence,

$$v_n = (1 + 2a\lambda)^{n-1}\epsilon \Rightarrow e_j^n = (-1)^j(1 + 2a\lambda)^{n-1}\epsilon.$$

Clearly, the local error truncation error will be amplified very quickly if we fix $\lambda = \tau/h$ with $\tau, h \to 0$. This reflects the fact that the scheme is unstable.

Exercise Implement this method and run numerical simulation. Also run the numerical simulation for the following method

$$\frac{u_j^{n+1} - u_j^n}{\tau} + a\frac{u_j^n - u_{j-1}^n}{h} = 0$$

for $\lambda = 0.9$ and 2.

6.1 The concept of stability

The stability considers the numerical method itself without comparing to the exact solution. The notions for evolutionary equations and elliptic equations can be different.

Consider the time evolutionary equations. We need a quantity to gauge the bigness of the data at time level t_n. We need some "norm" (范数) to achieve this. The frequently used norms include

$$\|v^n\|_\infty = \sup_j |v_j^n|$$

and

$$\|v^n\|_h := \sqrt{\sum_j h(v_j^n)^2}.$$

This is the ℓ^2 norm for discrete data. (The difference is that we have extra factor h to be consistent with the continuous L^2 integral.)

Definition 5. *Suppose we have a numerical scheme*

$$u^n = Y^n(u^{n-1}, \cdots, u^0, h, \tau)$$

with initial data u^0. Let the numerical error for u^0 be ϵ^0 (or we have initial data $v^0 = u^0 + \epsilon^0$) and the numerical error at t_n be $\epsilon^n = v^n - u^n$ where v^n is obtained using the scheme with initial data v^0. We say the scheme is stable (稳定的) if there is $K > 0$ such that for all $\tau \le \tau_0$ and $n\tau \le T$ one has

$$\|\epsilon^n\|_h \le K\|\epsilon^0\|_h.$$

Consider the scheme

$$u_j^{n+1} = u_j^n - a\lambda(u_{j+1}^n - u_j^n).$$

Clearly, if u^0 has error ϵ^0, then the error (difference between two solutions to the scheme) is evolved by

$$e_j^{n+1} = e_j^n - a\lambda(e_{j+1}^n - e_j^n).$$

The stability means that e^n should be controlled. As we have seen, the local truncation error at t_0 can be viewed as the initial error for t_1. Hence, the stability can also be used to gauge how the local truncation error can be amplified. This is why it is useful.

For linear equations, the scheme is often given by

$$u_j^{n+1} = L_h u_j^n,$$

where L_h is a linear operator. Then, the stability can be equivalently be given by

$$\sup_{n\tau \leq T} \|L_h^n\| \leq K$$

or

$$\|u^n\|_h \leq K\|u^0\|_h.$$

For nonlinear equations, we have to use the definition to define the stability.

For elliptic equations, the stability is similar.

Definition 6. *Consider the scheme $L(U) = F$ for some elliptic equation. If F has some error δF so that $\tilde{F} = F + \delta F$ and the corresponding numerical solution $\tilde{U}$ has error*

$$e = \tilde{U} - U.$$

The method is said to be stable if

$$\|e\| \leq C\|\delta F\|$$

for some constant C independent of the step size $h \leq h_0$.