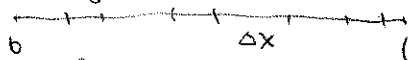


§8.1 Computation of solutions: An Introduction

Explicit formulas can only be found for a little portion of PDEs. Most problems in practice cannot be solved by formula. In order to get quantitative results, we have to use computers to get the numerical approximations.

The idea of the finite difference method is to replace derivatives by difference quotients. For instance, consider a function $u(x)$ of one variable. Choose a mesh size Δx : $x_j = j\Delta x$



Recall the definition of the derivative

$$u'(x) = \lim_{\Delta x \rightarrow 0} \frac{u(x+\Delta x) - u(x)}{\Delta x}$$

Then $u'(x) \approx \frac{u(x+\Delta x) - u(x)}{\Delta x}$ for a small Δx

Denoting $u_j = u(j\Delta x)$, we have the following three standard approximations for the first derivative:

$$\text{Forward difference: } u'(j\Delta x) \sim \frac{u_{j+1} - u_j}{\Delta x}$$

$$\text{Backward difference: } u'(j\Delta x) \sim \frac{u_j - u_{j-1}}{\Delta x}$$

$$\text{Centered difference: } u'(j\Delta x) \sim \frac{u_{j+1} - u_{j-1}}{2\Delta x}$$

To estimate the error in the above approximations, we consider the Taylor expansion of $u(x)$ ($u(x) \in C^4$)

$$u(x+\Delta x) = u(x) + u'(x)\Delta x + \frac{1}{2!}u''(x)\Delta x^2 + \frac{1}{3!}u'''(x)\Delta x^3 + O(\Delta x^4)$$

$$u(x-\Delta x) = u(x) - u'(x)\Delta x + \frac{1}{2!}u''(x)\Delta x^2 - \frac{1}{3!}u'''(x)\Delta x^3 + O(\Delta x^4)$$

From these two expansions, we deduce that

$$\frac{u(x+\Delta x) - u(x)}{\Delta x} = u'(x) + O(\Delta x)$$

$$\frac{u(x) - u(x-\Delta x)}{\Delta x} = u'(x) + O(\Delta x)$$

$$\frac{u(x+\Delta x) - u(x-\Delta x)}{2\Delta x} = u'(x) + O(\Delta x^2) \quad \text{second-order accuracy}$$

} first-order accuracy

For the second derivative, the simplest approximation centered second difference: $u''(j\Delta x) \sim \frac{u_{j+1} - 2u_j + u_{j-1}}{(\Delta x)^2}$

$$u''(x) = \frac{u(x+\Delta x) - 2u(x) + u(x-\Delta x)}{(\Delta x)^2} + O(\Delta x^2)$$

so the centered second difference is valid with an error of $O(\Delta x^2)$.

Sources of errors:

- ① Numerical approximation \rightarrow Truncation error
e.g. $O(\Delta x)$, $O(\Delta x^2)$
- ② Computer arithmetic \rightarrow Roundoff error
e.g. 16 digits

Usually ① is much larger than ②.

For function $u(x, t)$ with two variables, we choose mesh sizes $\Delta x, \Delta t$ for x, t variables separately.

$$u(j\Delta x, n\Delta t) \sim u_j^n$$

where the n is a superscript.

For instance, $\frac{\partial u}{\partial t}(j\Delta x, n\Delta t) \sim \frac{u_j^{n+1} - u_j^n}{\Delta t}$

$$\frac{\partial u}{\partial x}(j\Delta x, n\Delta t) \sim \frac{u_{j+1}^n - u_j^n}{\Delta x}$$

Example: Let us solve the heat equation via finite differences.

$$\begin{cases} u_t = u_{xx} \\ u(x, 0) = \phi(x) \end{cases}$$

Choosing a grid with mesh size Δx and Δt , we have

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}$$

difference equation

The truncation error on the left is $O(\Delta t)$, while $O(\Delta x^2)$ on the right. To balance that, we choose $\Delta t = 5(\Delta x)^2$, and

$$u_j^{n+1} = (1-2s)u_j^n + s(u_{j+1}^n + u_{j-1}^n)$$

This is a numerical scheme for computing the numerical solution $\{u_j^n\}$, called the FTCS scheme (forward time, centered space). This is an explicit scheme since values at the time level $n+1$ depends on the previous time level.

If $s=1$,
$$u_j^{n+1} = u_{j+1}^n - u_j^n + u_{j-1}^n$$

Assume the initial data $\phi(x)$ a step function. At the grid points $j\Delta x$,

$$\phi_j: \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0$$

A sample calculation with these simple initial data can be done by hand by simply "marching in time". That is, $\phi(x)$ provides u_j^0 , then the scheme give u_j^1 , u_j^2 , and so forth.

We can schematically use the diagram (template, or stencil),

$$\begin{array}{cccc} & \circ & 1 & \circ \\ & & * & -1 & * & & \circ & & 1 \\ & & & & & & & & & & \circ & & & & \circ \end{array}$$

which means that in order to get u_j^{n+1} (*) you just add or subtract its three lower neighbors as indicated (o)

$n=4$	1	-4	10	-16	19	-16	10	-4	1	1
$n=3$	0	1	-3	6	-7	6	-3	1	0	1
$n=2$	0	0	1	-2	3	-2	1	0	0	1
$n=1$	0	0	0	1	-1	1	0	0	0	1
$n=0$	0	0	0	0	1	0	0	0	0	1

The result is horrendous. The true solution of the PDE will always be between 0 and 1 by maximum principle.

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

$$\begin{array}{cc} * \\ \circ \frac{1}{2} \quad \circ \frac{1}{2} \end{array}$$

$n=4$	$\frac{1}{16}$	0	$\frac{1}{4}$	0	$\frac{3}{8}$	0	$\frac{1}{4}$	0	$\frac{1}{16}$	1
$n=3$	0	$\frac{1}{8}$	0	$\frac{3}{8}$	0	$\frac{3}{8}$	0	$\frac{1}{8}$	0	1
$n=2$	0	0	$\frac{1}{4}$	0	$\frac{1}{2}$	0	$\frac{1}{4}$	0	0	1
$n=1$	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	0	1
$n=0$	0	0	0	0	1	0	0	0	0	1

We can see the dissipative behavior, which is expected for the heat equation.

The difference between $s=1$ and $s=1/2$ is the former scheme is unstable while the latter is stable,

~~which~~ which is the stability of numerical methods.