MAR 513-Lecture 2: The Finite-Difference Methods

Taylor Series Expansion

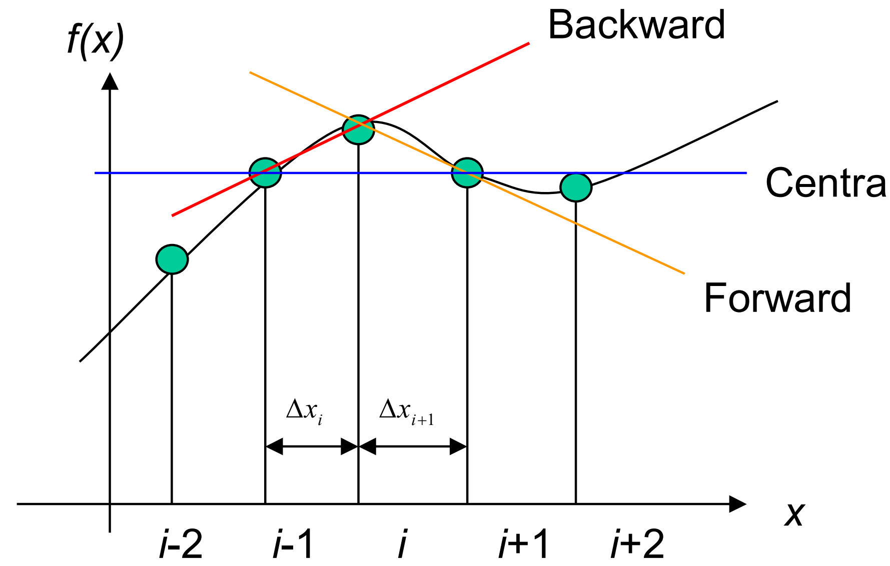
Suppose we have a continuous function $f(x)$. Its value in the vicinity of x_i can be approximately expressed using a Taylor series as

$$\boxed{f(x) = f(x_i) + (x - x_i)\left.\frac{\partial f}{\partial x}\right|_i + \frac{(x - x_i)^2}{2}\left.\frac{\partial f}{\partial x}\right|_i + \cdots + \frac{(x - x_i)^n}{n!}\left.\frac{\partial^n f}{\partial x^n}\right|_i + \cdots} \tag{2.1}$$

Using (2.1), we have derived the discrete expression for the first-order derivative as

$$\left.\frac{\partial f}{\partial x}\right|_i = \frac{f(x) - f(x_i)}{x - x_i} - \left[\frac{(x - x_i)}{2}\left.\frac{\partial f}{\partial x}\right|_i + \cdots + \frac{(x - x_i)^{n-1}}{n!}\left.\frac{\partial^n f}{\partial x^n}\right|_i + \cdots\right] \tag{2.2}$$

Replacing x *by* x_{i+1} or $x_{i-1,}$ in (2.2) or substracting Taylor expansion equation for x_{i-1} *from* x_{i+1}, we can get



$$\left.\frac{\partial f}{\partial x}\right|_i = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} - [\frac{(x_{i+1} - x_i)}{2}\left.\frac{\partial^2 f}{\partial x^2}\right|_i + \cdots + \frac{(x_{i+1} - x_i)^{n-1}}{n!}\left.\frac{\partial^n f}{\partial x^n}\right|_i + \cdots]$$

$$\left.\frac{\partial f}{\partial x}\right|_i = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} + \frac{(x_i - x_{i-1})}{2}\left.\frac{\partial^2 f}{\partial x^2}\right|_i - \cdots + (-1)^{n-1}\frac{(x_{i+1} - x_i)^{n-1}}{n!}\left.\frac{\partial^n f}{\partial x^n}\right|_i + \cdots$$

$$\left.\frac{\partial f}{\partial x}\right|_i = \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}} - \frac{(x_{i+1} - x_i)^2 - (x_i - x_{i-1})^2}{2(x_{i+1} - x_{i-1})}\left.\frac{\partial^2 f}{\partial x^2}\right|_i + \frac{(x_{i+1} - x_i)^3 + (x_i - x_{i-1})^3}{6(x_{i+1} - x_{i-1})}\left.\frac{\partial^3 f}{\partial x^3}\right|_i + \cdots$$

Expressing $x_{i+1} - x_i = \Delta x_{i+1}$ $\quad x_i - x_{i-1} = \Delta x_i$ $\quad \Delta x_i = \Delta x_{i+1} = \Delta x$ we obtain

$$\left.\frac{\partial f}{\partial x}\right|_i = \frac{f(x_{i+1}) - f(x_i)}{\Delta x} + O(\Delta x); \qquad \text{Forward Difference Scheme (FDS)}$$

$$\left.\frac{\partial f}{\partial x}\right|_i = \frac{f(x_i) - f(x_{i-1})}{\Delta x} + O(\Delta x); \qquad \text{Backward Difference Scheme (BDS)}$$

$$\left.\frac{\partial f}{\partial x}\right|_i = \frac{f(x_{i+1}) - f(x_{i-1})}{2\Delta x} + O[(\Delta x)^2]; \qquad \text{Central Difference Scheme (CDS)}$$

$$O(\Delta x) \qquad O[(\Delta x)^2]$$

The order of the higher-order terms that are deleted from the right-hand sides of the discrete equation. In numerical method, they are called "**truncation errors**".

General forms:

$$\boxed{O[(\Delta x)^m]}$$

$$\begin{cases} m=1 & \text{First order accurate} \\ m=2 & \text{Second order accurate} \end{cases}$$

FDS: The first order accurate

BDS: The first order accurate

CDS: The second order accurate

Exercise: Derive CDS and determine its truncation error

For the second order derivative, we also can use the same approach.

Example: $\dfrac{\partial^2 f}{\partial x^2}$ Use the uniform grid

$$f(x_{i+1}) = f(x_i) + \Delta x \frac{\partial f}{\partial x}\bigg|_i + \frac{(\Delta x)^2}{2}\frac{\partial^2 f}{\partial x^2}\bigg|_i + \frac{(\Delta x)^3}{6}\frac{\partial^3 f}{\partial x^3}\bigg|_i + \frac{(\Delta x)^4}{6}\frac{\partial^4 f}{\partial x^4}\bigg|_i + \cdots \quad (2.3)$$

$$f(x_{i-1}) = f(x_i) - \Delta x \frac{\partial f}{\partial x}\bigg|_i + \frac{(\Delta x)^2}{2}\frac{\partial^2 f}{\partial x^2}\bigg|_i - \frac{(\Delta x)^3}{6}\frac{\partial^3 f}{\partial x^3}\bigg|_i + \frac{(\Delta x)^4}{6}\frac{\partial^4 f}{\partial x^4}\bigg|_i + \cdots \quad (2.4)$$

Eq. (2.3) + Eq. (2.4)

$$f(x_{i+1}) + f(x_{i-1}) = 2f(x_i) + (\Delta x)^2 \frac{\partial^2 f}{\partial x^2}\bigg|_i + \frac{(\Delta x)^4}{3}\frac{\partial^4 f}{\partial x^4}\bigg|_i + \cdots$$

Then, we have

$$\frac{\partial^2 f}{\partial x^2}\bigg|_i = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{(\Delta x)^2} + O[(\Delta x)^2]$$

Centered Difference Scheme (CDS) with second order accuracy

Example of constructing the difference equation

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = g(x, y)$$

Select the CDS

$$\frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{(\Delta x)^2} + \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{(\Delta y)^2} = g_{i,j}$$

$$\boxed{f_{i+1,j} + f_{i-1,j} + \frac{(\Delta x)^2}{(\Delta y)^2}(f_{i,j+1} + f_{i,j-1}) - 2[1 + \frac{(\Delta x)^2}{(\Delta y)^2}]f_{i,j} = g_{i,j}}$$

X: i=1, 2, 3….N
Y: j=1,2,3…...M

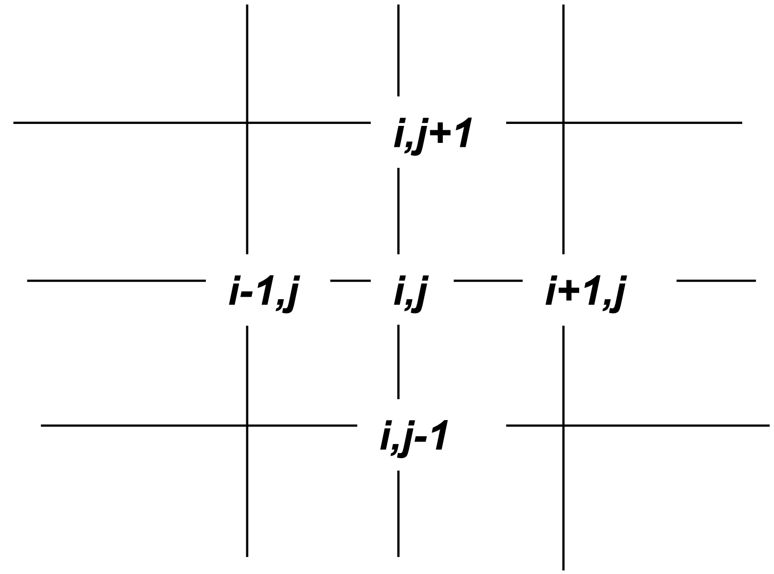


Fig. 2.2: Uniform rectangular grids.

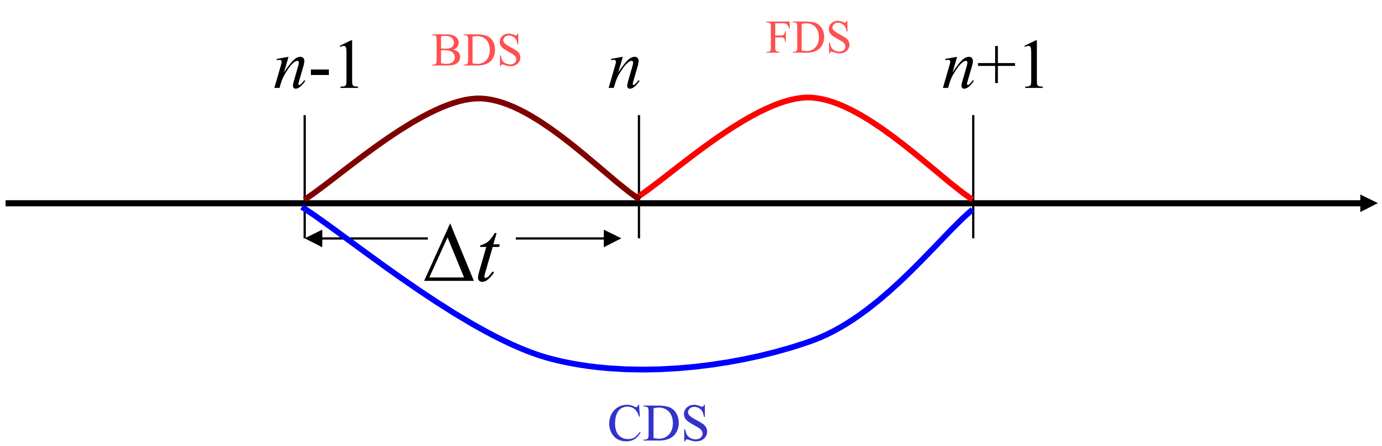
The basic idea for the finite-difference method is to replace the derivatives using the discrete approximation and convert the differential equation to a set of algebraic equations.

The time derivatives

$$\left.\frac{\partial f}{\partial t}\right|_n = \frac{f(t_{n+1}) - f(t_n)}{\Delta t} + O(\Delta t); \qquad \text{Forward Difference Scheme (FDS)}$$

$$\left.\frac{\partial f}{\partial t}\right|_n = \frac{f(t_n) - f(t_{n-1})}{\Delta t} + O(\Delta t); \qquad \text{Backward Difference Scheme (BDS)}$$

$$\left.\frac{\partial f}{\partial x}\right|_n = \frac{f(t_{n+1}) - f(t_{n-1})}{2\Delta t} + O[(\Delta t)^2]; \qquad \text{Central Difference Scheme (CDS)}$$



Numerical Schemes

Explicit scheme: A numerical scheme in which the numerical value at time step (n+1) is calculated directly from its previous value at the time step n. This means that once the values at the time step n are known, we can "predict" a new value at the time step (n+1) by a direct time integration.

Implicit scheme: A numerical scheme in which the numerical value at the time step (n+1) is not explicitly obtained from its previous value at the time step n. This value must be solved from an algebraic equation formed at the time step n+1.

Example:

$$\frac{\partial f}{\partial t} + C\frac{\partial f}{\partial x} = 0$$

It is solvable, with a general form

$$f = g(x - Ct)$$

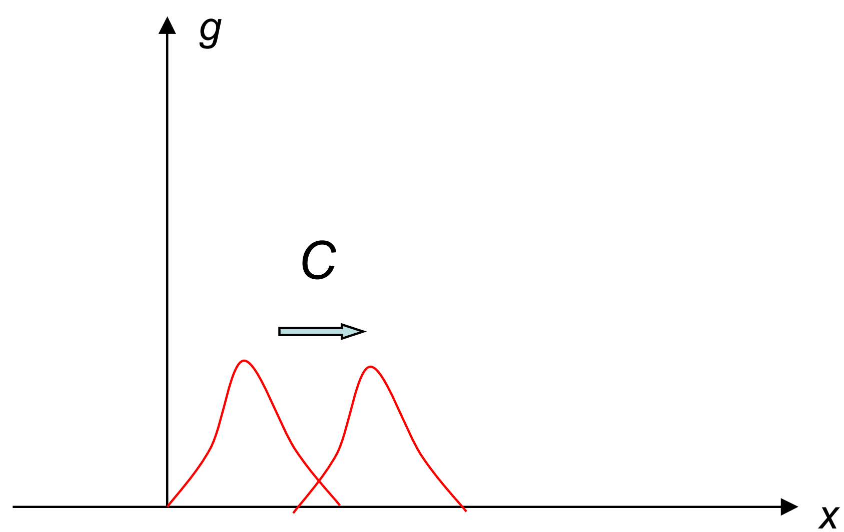


Fig. 2.3: Schematic of a propagation of a blob.

a) Leapfrog Scheme

$$\frac{f_i^{n+1} - f_i^{n-1}}{2\Delta t} + C\frac{f_{i+1}^n - f_{i-1}^n}{2\Delta x} = 0 \qquad (2.5)$$

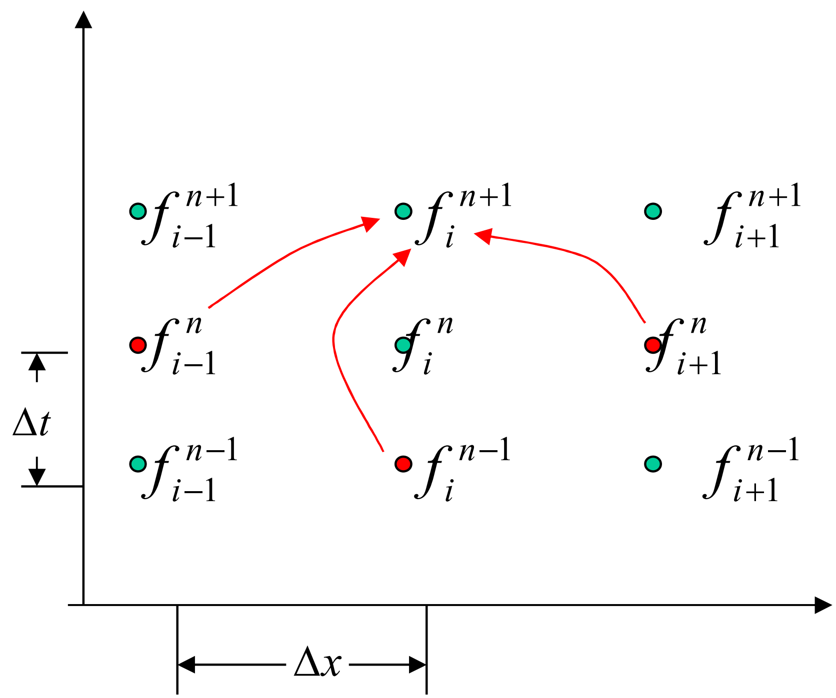
Truncation error:

$O[(\Delta t)^2]$ for time derivative

$O[(\Delta x)^2]$ for space derivative

Then, the difference equation is

$$\boxed{f_i^{n+1} = f_i^{n-1} - C\frac{\Delta t}{\Delta x}(f_{i+1}^n - f_{i-1}^n)}$$



b). Forward Time/Central Space Scheme (Euler Scheme)

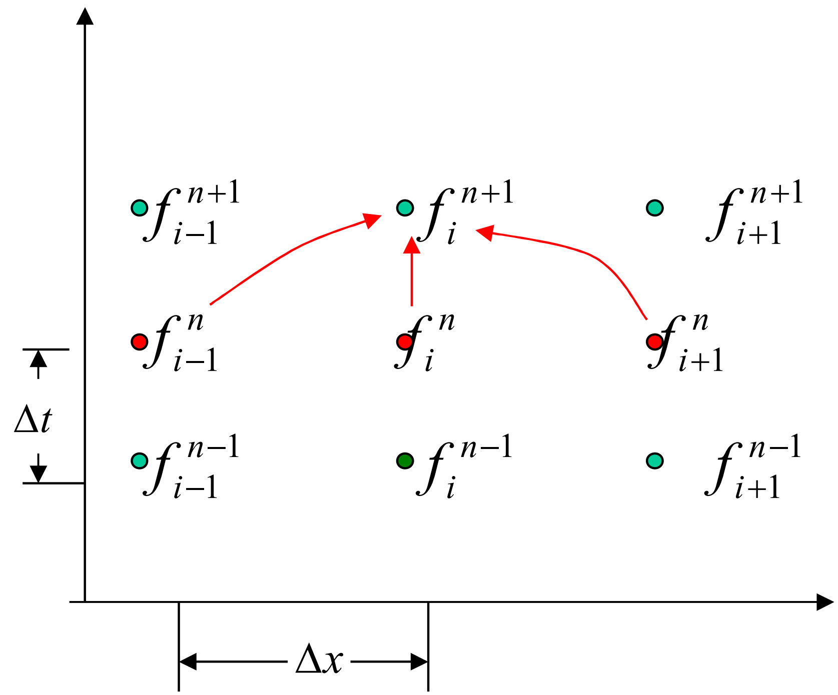
$$\frac{f_i^{n+1} - f_i^n}{\Delta t} + C\frac{f_{i+1}^n - f_{i-1}^n}{2\Delta x} = 0$$

Truncation error:

$O(\Delta t)$ First-order accurate

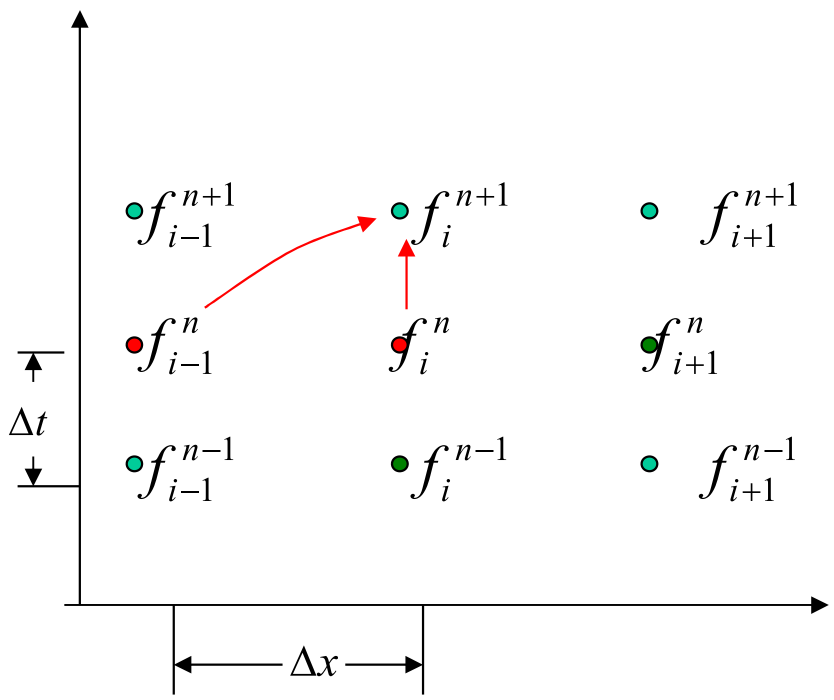
$O[(\Delta x)^2]$ Second-order accurate

$$\boxed{f_i^{n+1} = f_i^n - C\frac{\Delta t}{2\Delta x}(f_{i+1}^n - f_{i-1}^n)}$$



c). Forward Time/Backward Space Scheme

$$\frac{f_i^{n+1} - f_i^n}{\Delta t} + C\frac{f_i^n - f_{i-1}^n}{\Delta x} = 0$$



Truncation errors:

$O(\Delta t)$ First-order accurate

$O(\Delta x)$ First-order accurate

$$\boxed{f_i^{n+1} = f_i^n - C\frac{\Delta t}{\Delta x}(f_i^n - f_{i-1}^n)}$$

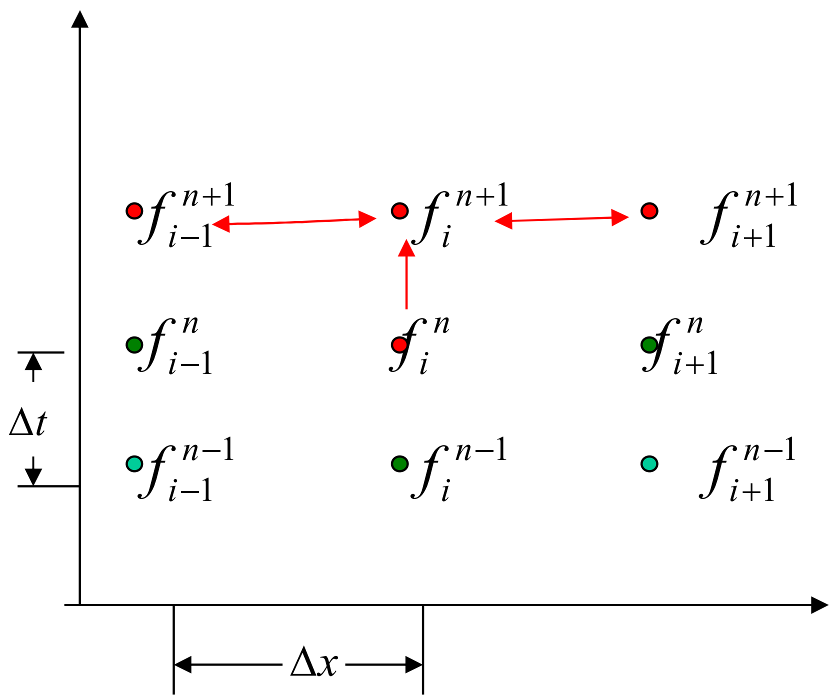
Sometime, it is also called the upwind scheme for the case C > 0.

d). Forward Time/Implicit Central Space Scheme

$$\frac{f_i^{n+1} - f_i^n}{\Delta t} + C\frac{f_{i+1}^{n+1} - f_{i-1}^{n+1}}{2\Delta x} = 0$$

$O(\Delta t)$ First-order accurate

$O[(\Delta x)^2]$ Second-order accurate

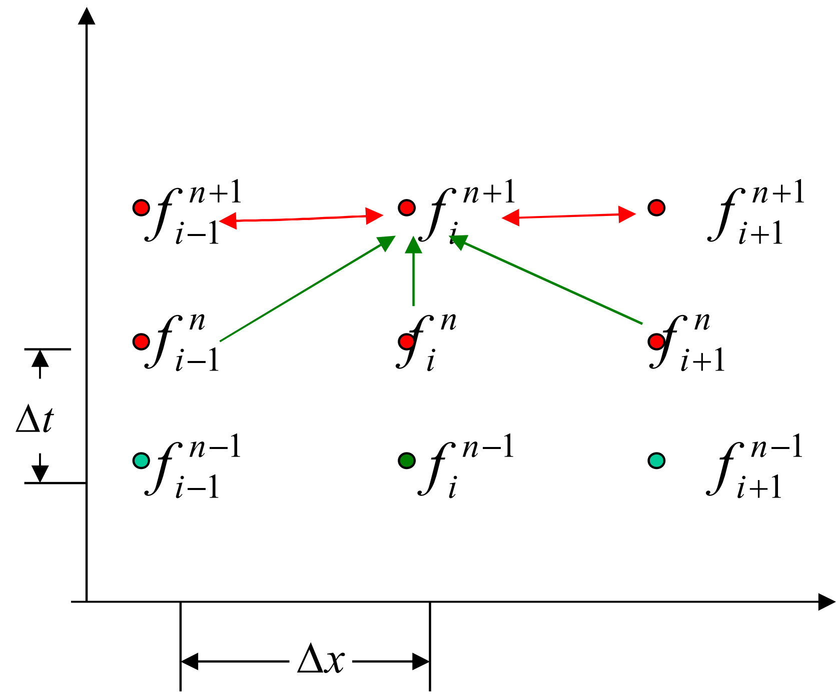


$$\boxed{C\frac{\Delta t}{2\Delta x}f_{i+1}^{n+1} + f_i^{n+1} - C\frac{\Delta t}{2\Delta x}f_{i-1}^{n+1} = f_i^n}$$

This is a fully-implicit scheme!

e). Crank-Nicolson Scheme—Semi-implicit Scheme

$$\frac{f_i^{n+1}-f_i^n}{\Delta t}+\frac{C}{2}\left(\frac{f_{i+1}^{n+1}-f_{i-1}^{n+1}}{2\Delta x}+\frac{f_{i+1}^{n}-f_{i-1}^{n}}{2\Delta x}\right)=0$$



Truncation errors:

$O(\Delta t)$ First-order accurate

$O[(\Delta x)^2]$ Second-order accurate

$$C\frac{\Delta t}{4\Delta x}f_{i+1}^{n+1}+f_i^{n+1}-C\frac{\Delta t}{4\Delta x}f_{i-1}^{n+1}=f_i^n-C\frac{\Delta t}{4\Delta x}(f_{i+1}^n-f_{i-1}^n)$$

f). Lax-Wendroff Scheme

$$\frac{f_{i-1/2}^{n+1/2} - \frac{1}{2}(f_i^n + f_{i-1}^n)}{\frac{1}{2}\Delta t} = -C\frac{f_i^n - f_{i-1}^n}{\Delta x}$$

$$\frac{f_{i+1/2}^{n+1/2} - \frac{1}{2}(f_{i+1}^n + f_i^n)}{\frac{1}{2}\Delta t} = -C\frac{f_{i+1}^n - f_i^n}{\Delta x}$$

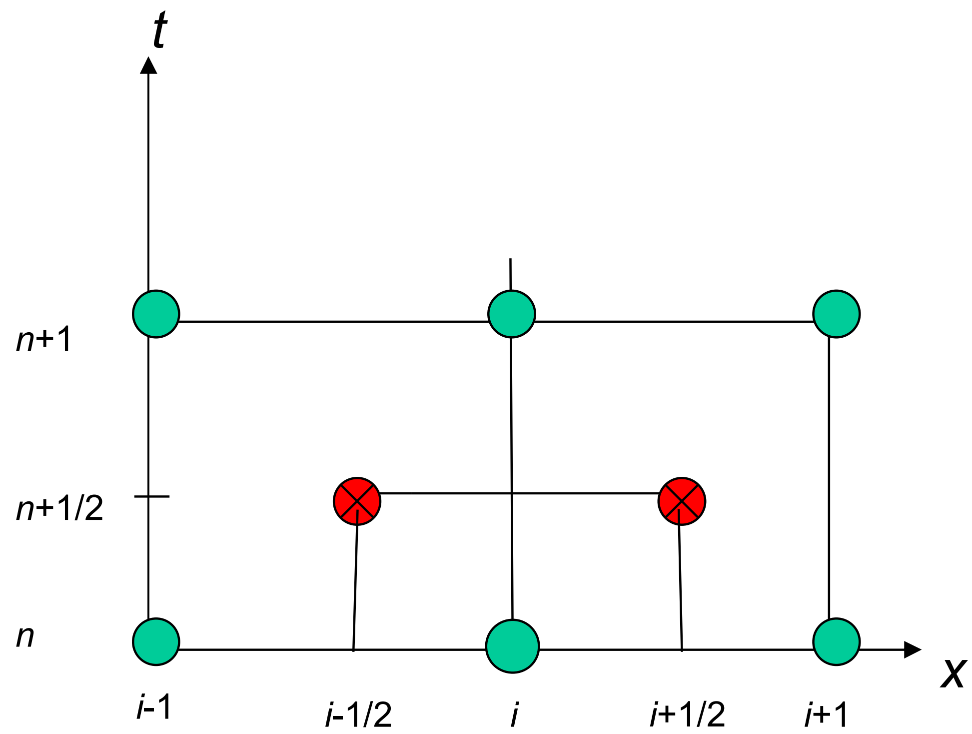


Fig. 2.5: The space-time stencil used to construct the Lax-Wendroff scheme

Then

$$f_{i+1/2}^{n+1/2} - f_{i-1/2}^{n+1/2} = \frac{1}{2}(f_{i+1}^n - f_{i-1}^n) - \frac{c\Delta t}{2\Delta x}(f_{i+1}^n - 2f_i^n + f_{i-1}^n)$$

$$f_{i+1/2}^{n+1/2} - f_{i-1/2}^{n-1/2} = -\frac{\Delta x}{C}\frac{f_i^{n+1} - f_i^n}{\Delta t}$$

$$\boxed{\frac{f_i^{n+1} - f_i^n}{\Delta t} = -C\frac{f_{i+1}^n - f_{i-1}^n}{2\Delta x} + \frac{1}{2}C^2\Delta t\frac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{(\Delta x)^2}}$$

Truncation errors: $O[(\Delta t)^2 + (\Delta x)^2]$

QS: How could we know which scheme is better?

Or How do we evaluate these schemes?

Next !

Numerical Stability

Stability: Will the numerical solution remain finite at all times and at all locations?

Example: The diffusion equation

$$\frac{\partial T}{\partial t} = A_h \frac{\partial^2 T}{\partial x^2} \tag{3.1}$$

Forward time scheme for time and central difference scheme for space, we have

$$\frac{T_k^{n+1} - T_k^n}{\Delta t} - A_h \frac{T_{k+1}^n - 2T_k^n + T_{k-1}^n}{(\Delta x)^2} = 0 \tag{3.2}$$

Let us rewrite

$$r = \frac{\Delta t}{(\Delta x)^2} \tag{3.3}$$

, then

$$\boxed{T_i^{n+1} = (1 - 2rA_h)T_i^n + rA_h(T_{i+1}^n + T_{i-1}^n)} \tag{3.4}$$

Assume that an error ε appears at a point i_o at initial, let us examine if this error would grow up with time. If does, the numerical scheme is unstable. If doesn't, the numerical scheme is stable or at least neutral stable.

Case 1: $r = \frac{1}{2A_h}$

$$\begin{cases} T_k^{n+1} = \frac{1}{2}(T_{k+1}^n + T_{k-1}^n) \\ T_k^0 = T(k,0) \end{cases} \tag{3.5}$$

When the error appears at the initial, the approximate solution is

$$\begin{cases} \widetilde{T}_k^{n+1} = \frac{1}{2}(\widetilde{T}_{k+1}^n + \widetilde{T}_{k-1}^n) \\ \widetilde{T}_k^0 = \begin{cases} \widetilde{T}(k,0) & k \neq k_o \\ \widetilde{T}(k_o,0) + \varepsilon & k = k_o \end{cases} \end{cases} \tag{3.6}$$

(3.6)-(3.5) producing an error equation

$$\begin{cases} \varepsilon_k^{n+1} = \frac{1}{2}(\varepsilon_{k+1}^n + \varepsilon_{k-1}^n) \\ \varepsilon_k^0 = \begin{cases} 0 & k \neq k_o \\ \varepsilon & k = k_o \end{cases} \end{cases} \tag{3.7}$$

Table of the errors

n=5	0	0.16525ε	0	0.3125ε	0	0.3125ε	0	0.15625ε	0
n=4	0.0625ε	0	0.25ε	0	0.375ε	0	0.25ε	0	0.0625ε
n=3		0.125ε	0	0.375ε	0	0.375ε	0	0.125ε	
n=2			0.25ε	0	0.5ε	0	0.25ε		
n=1				0.5ε	0	0.5ε			
n=0					ε				
n / ***i***	k_o-4	k_o-3	k_o-2	k_o-1	k_o	k_o+1	k_o+2	k_o+3	k_o+4

Errors decreases with time integration, so the method is stable!

Case 2: $r = \dfrac{1}{A_h}$

$$T_k^{n+1} = T_{k+1}^n + T_{k-1}^n - T_k^n \tag{3.8}$$

Error equation is

$$\begin{cases} \varepsilon_k^{n+1} = \varepsilon_{k+1}^n + \varepsilon_{k-1}^n - \varepsilon_k^n \\ \varepsilon_k^0 = \begin{cases} 0 & k \neq k_o \\ \varepsilon & k = k_o \end{cases} \end{cases}$$

n=4	ε	-4ε	10ε	-16ε	19ε	-16ε	10ε	-4ε	ε
n=3		ε	-3ε	6ε	-7ε	6ε	-3ε	ε	
n=2			ε	-2ε	3ε	-2ε	ε		
n=1				ε	-ε	ε			
n=0					ε				
	k_o-4	k_o-3	k_o-2	k_o-1	k_o	k_o+1	k_o+2	k_o+3	k_o+4

Error increases with time integration, so the method is unstable!

QS 1: Do we have to make a table every time to determine the stability of a numerical scheme?

QS 2: What is the definition of the numerical stability?

Numerical stability is generally built on an assumption that the boundary values are accurate and errors appear at the initial integration. Stability analysis is to determine if this error would grow up.

For a linear *t-x* equation, the error always could be expressed by a wave function such as

$$T_i^n = T^0 A^n e^{ik\Delta x} \qquad (3.9)$$

where A is the amplitude factor, T_o is the initial value, n is the number of the tine integration, k is the node point

$$\|A\| \begin{cases} > 1 & \text{Unstable} \\ \leq 1 & \text{Stable} \end{cases} \qquad (3.10)$$

Example:

$$T_i^{n+1} = (1-2rA_h)T_i^n + rA_h(T_{i+1}^n + T_{i-1}^n)$$

Substituting (3.9) into the above equation, we have

$$\cancel{T^0} A^{n+1} \cancel{e^{ik\Delta x}} = (1-2rA_h)\cancel{T^0} A^n \cancel{e^{ik\Delta x}} + rA_h[e^{i(\cancel{k}+1)\Delta x} + e^{i(\cancel{k}-1)\Delta x}]\cancel{T^0} A^n$$

After removing the common factors, we get

$$A^{n+1} = (1-2rA_h)A^n + rA_h(e^{i\Delta x} + e^{-i\Delta x})A^n$$

Then

$$\boxed{A = 1-2rA_h + 2rA_h \cos\Delta x}$$

To make the method stable, we must have

$$\|A\| = \|1-2rA_h(1-\cos\Delta x)\| \le 1$$

That is equivalent to

$$-1 \le 1 - 2rA_k(1 + |\cos \Delta x|) \le 1$$

So, the stability condition is:

$$\boxed{r = \frac{\Delta t}{(\Delta x)^2} \le \frac{1}{A_k(1 + |\cos \Delta x|)}} \qquad (3.11)$$

For case 1: $r = 1/(2A_\mathrm{h})$

$$\frac{1}{2A_h} \le \frac{1}{A_h(1 + |\cos \Delta x|)}$$ stable!

For case 2: $r = 1/\mathrm{A_h}$

$$r = \frac{1}{A_h} > \frac{1}{A_k(1 + |\cos \Delta x|)}$$ unstable!

Now, let us use this method to analyze all the advection schemes described in the last class:

1. Leapfrog scheme

$$f_k^{n+1} = f_k^{n-1} - C\frac{\Delta t}{\Delta x}(f_{k+1}^n - f_{k-1}^n)$$

Central difference for both time and space.

Assume that

$$f_k^n = f(0)A^n e^{ik\Delta x}$$

Then, we have

$$A^{n+1} = A^{n-1} - C\frac{\Delta t}{\Delta x}(e^{i\Delta x} - e^{-i\Delta x})A^n \longrightarrow A = A^{-1} - C\frac{\Delta t}{\Delta x}2i\sin(\Delta x)$$

$$A^2 + C\frac{\Delta t}{\Delta x}2i\sin(\Delta x) - 1 = 0$$

Solution is

$$\boxed{A_{\pm} = -iC\frac{\Delta t}{\Delta x}\sin(\Delta x) \pm [1 - (C\frac{\Delta t}{\Delta x}\sin(\Delta x))^2]^{\frac{1}{2}}}$$

Let's define that

$$\alpha = C\frac{\Delta t}{\Delta x}\sin(\Delta x)$$

, then the solution can be expressed as

$$A_{\pm} = -i\alpha \pm [1-\alpha^2]^{\frac{1}{2}}$$

For stability, $|A_{\pm}| \leq 1$

If $\alpha \leq 1$, $\quad |A_{\pm}| = (\alpha^2 + 1 - \alpha^2)^{1/2} = 1 \longrightarrow$ Stable !

If $\alpha > 1$, $\quad |A_{\pm}| = \left|-\alpha \pm (\alpha^2 - 1)^{1/2}\right| \not\leq 1$ because $|A_-| > 1$

Therefore, the leapfrog scheme is only stable if $\alpha \leq 1$, i. e.

$$\boxed{C\frac{\Delta t}{\Delta x} \leq 1}$$

2) Forward Time/Central Space Scheme (Euler Scheme)

$$f_k^{n+1} = f_k^n - C\frac{\Delta t}{2\Delta x}(f_{k+1}^n - f_{k-1}^n)$$

Assume that

$$f_k^n = f(0)A^n e^{ik\Delta x}$$

Then, we have

$$A^{n+1} = A^n - C\frac{\Delta t}{2\Delta x}(e^{i\Delta x} - e^{-i\Delta x})A^n \longrightarrow A = 1 - C\frac{\Delta t}{\Delta x} i \sin(\Delta x)$$

$$|A| = 1 + (C\frac{\Delta t}{\Delta x})^2 \sin^2(\Delta x) > 1 \text{ if } \Delta t \neq 0 \text{ and } \Delta x \neq 0$$

Therefore, no matter what the ratio of time step to space grid size is chosen, the Euler scheme is an absolutely unstable scheme!

3). Forward Time/Backward Space Scheme

$$f_k^{n+1} = f_k^n - C\frac{\Delta t}{\Delta x}(f_k^n - f_{k-1}^n)$$

Assume that $f_k^n = f(0)A^n e^{ik\Delta x}$

Then, we have $A^{n+1} = A^n - C\frac{\Delta t}{\Delta x}(1-e^{-i\Delta x})A^n \longrightarrow A = 1 - C\frac{\Delta t}{\Delta x}[1-\cos(\Delta x) - i\sin(\Delta x)]$

$$A = 1 - C\frac{\Delta t}{\Delta x} + +iC\frac{\Delta t}{\Delta x}\sin(\Delta x)$$

$$|A|^2 = [1 - C\frac{\Delta t}{\Delta x} + C\frac{\Delta t}{\Delta x}\cos(\Delta x)]^2 + \left(C\frac{\Delta t}{\Delta x}\right)^2 \sin^2(\Delta x)$$

$$= (1 - C\frac{\Delta t}{\Delta x})^2 + 2(1 - C\frac{\Delta t}{\Delta x})C\frac{\Delta t}{\Delta x}\cos(\Delta x) + \left(C\frac{\Delta t}{\Delta x}\right)^2$$

$$= 1 - 2C\frac{\Delta t}{\Delta x} + 2(1 - C\frac{\Delta t}{\Delta x})C\frac{\Delta t}{\Delta x}\cos(\Delta x) + 2\left(C\frac{\Delta t}{\Delta x}\right)^2$$

$$= 1 - 2C\frac{\Delta t}{\Delta x}[1 - C\frac{\Delta t}{\Delta x}][1-\cos(\Delta x)]$$

For $|A|^2 \leq 1$, then

$$2C\frac{\Delta t}{\Delta x}[1 - C\frac{\Delta t}{\Delta x}][1-\cos(\Delta x)] \geq 0$$

which can only be assured if

$$C\frac{\Delta t}{\Delta x} \leq 1$$

So, if we use the upstream information (backward) for space gradient and forward time stepping, the method is conditionally stable!

Since $|A| < 1$, the scheme is artificial numerical damping!

4). Forward Time/Implicit Central Space Scheme

$$C\frac{\Delta t}{2\Delta x}f_{i+1}^{n+1} + f_i^{n+1} - C\frac{\Delta t}{2\Delta x}f_{i-1}^{n+1} = f_i^n$$

Assume that $f_k^n = f(0)A^n e^{ik\Delta x}$

Then, we have $C\frac{\Delta t}{2\Delta x}A^{n+1}e^{i\Delta x} + A^{n+1} - C\frac{\Delta t}{2\Delta x}A^{n+1}e^{-i\Delta x} = A^n$

$$A^{n+1}(1 + C\frac{\Delta t}{\Delta x}i2\sin\Delta x) = A^n \longrightarrow A = \frac{1}{1 + C\frac{\Delta t}{\Delta x}i2\sin\Delta x} = \frac{1 - C\frac{\Delta t}{\Delta x}i2\sin\Delta x}{1 + 4(C\frac{\Delta t}{\Delta x})^2\sin^2\Delta x}$$

$$\boxed{|A| = \frac{1}{1 + 4(C\frac{\Delta t}{\Delta x})^2\sin^2\Delta x}\sqrt{1 + 4(C\frac{\Delta t}{\Delta x})^2\sin^2\Delta x} \le 1}$$

Unconditionally stable!

No matter what Δt and Δx are chosen, this scheme is always stable. However, the accuracy of the scheme still depend on Δt and Δx because it only O(Δt) and O(Δx^2) . So, you can get serious phase errors for both long and short waves if C(Δt/Δx) is too large. In fact, the implicit schemes essentially have C = ∞ in that the information can be spread throughout the grids over one time step.

damping scheme!

5. Crank-Nicolson Scheme—Semi-implicit Scheme

$$C\frac{\Delta t}{4\Delta x} f_{i+1}^{n+1} + f_i^{n+1} - C\frac{\Delta t}{4\Delta x} f_{i-1}^{n+1} = f_i^n - C\frac{\Delta t}{4\Delta x}(f_{i+1}^n - f_{i-1}^n)$$

Assume that $f_k^n = f(0)A^n e^{ik\Delta x}$

$$C\frac{\Delta t}{4\Delta x} A^{n+1}e^{i\Delta x} + A^{n+1} - C\frac{\Delta t}{4\Delta x} A^{n+1}e^{-i\Delta x} = A^n[1 - C\frac{\Delta t}{4\Delta x}(e^{i\Delta x} - e^{-i\Delta x})]$$

$$\downarrow$$

$$A^{n+1}(1 + C\frac{\Delta t}{2\Delta x} i\sin\Delta x) = A^n(1 - C\frac{\Delta t}{2\Delta x} i\sin\Delta x)$$

$$\downarrow$$

$$A = \frac{1 - C\frac{\Delta t}{2\Delta x} i\sin\Delta x}{1 + C\frac{\Delta t}{2\Delta x} i\sin\Delta x} = \frac{(1 - C\frac{\Delta t}{2\Delta x} i\sin\Delta x)^2}{1 + (C\frac{\Delta t}{2\Delta x})^2 \sin^2\Delta x}$$

$$\boxed{|A| = 1}$$

Crank-Nicolson scheme is unconditionally stable!

This scheme helps reducing the artificial numerical damping.

Central difference scheme for the time integration produces two numerical solutions, even though it provides a second-order numerical accuracy. This can produces additional numerical errors!

Example: Leapfrog Scheme: there are two numerical solutions: $A_{\pm}$

$$f_k^n = (aA_+^n + bA_-^n)e^{ik\Delta x}$$

For advection equation, the analytical solution is

$$f(x,t) = Ce^{ik(x-ct)}$$

But now, the central difference scheme produces two solutions:

$$f(1) = ae^{-ic\hat{k}(n\Delta t)+i\hat{k}(k\Delta x)} \approx ae^{ik(x-ct)} \longrightarrow \text{Physical mode}$$

$$f(2) = ae^{ic\hat{k}(n\Delta t)+i\hat{k}(k\Delta x)} \approx ae^{ik(x+ct)} \longrightarrow \text{Computational mode}$$

Computational mode has also a finite amplitude bounded by $|A| \leq 1$, but its amplitude goes like $(-1)^n$. It changes sign with each time step and causes significantly artificial numerical error waves. Therefore, when a central difference scheme is used for the time integration, we must try to choose the initial conditions so that the computational mode is zero. However, it will grow due to numerical errors like truncation roundoff, so the computational mode can not be ignored.

Computational dispersion

$$\frac{\partial u}{\partial t} + C\frac{\partial u}{\partial x} = 0 \quad \mathrm{C} = \text{constant} \tag{3.12}$$

This advection equation has an analytical solution in the form of a single harmonic component

$$u(x,t) = \mathrm{Re}[U(t)e^{ikx}] \tag{3.13}$$

Substituting (3.13) into (3.12), we have

$$\frac{dU}{dt} + ikCU = 0 \tag{3.14}$$

This is an oscillation equation where kC is the frequency (called ω). Therefore,

$$C = \frac{\omega}{k} \tag{3.15}$$

Since C is constant, so that waves with different wave lengths propagate with the same phase speed. The function $u(x,t)$ is advected with no change in shape at a constant velocity C along the x axis:

There are no dispersion!

Now, let us consider the difference equation like

$$\frac{\partial u_j}{\partial t} + C\frac{u_{j+1} - u_{j-1}}{2\Delta x} = 0 \tag{3.16}$$

Assume that

$$u(x,t) = \mathrm{Re}[U(t)e^{ikj\Delta x}] \tag{3.17}$$

Substituting (3.17) into (3.16), we have

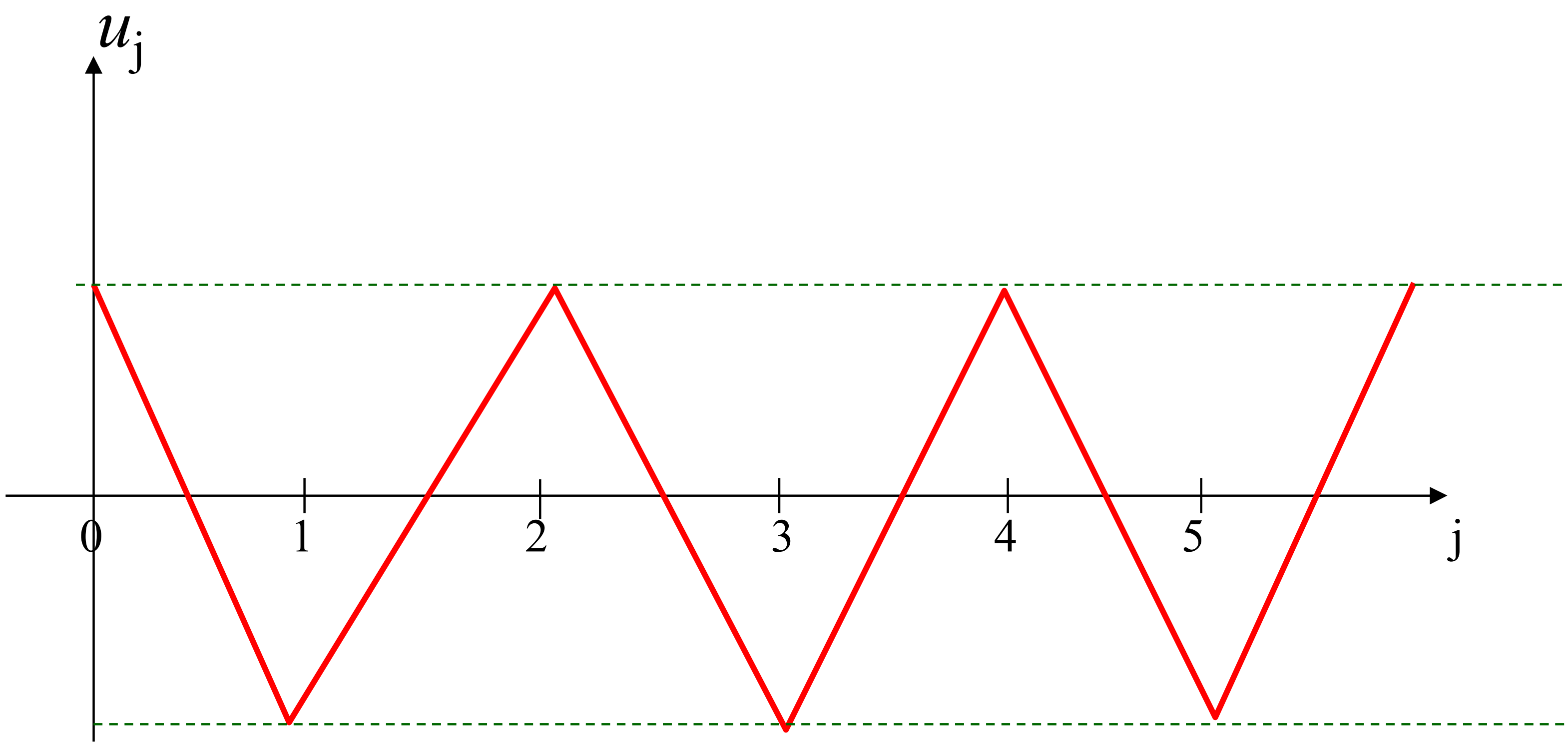
$$\frac{dU}{dt} + ik(C\frac{\sin k\Delta x}{k\Delta x})U = 0 \tag{3.18}$$

Therefore,

$$\boxed{\hat{c} = C\frac{\sin k\Delta x}{k\Delta x}}$$

Numerical phase speed is a function of the wave number k, so that the finite differencing in space cause a dispersion of the waves! It is also called " computational dispersion"!

As Δx increases from zero, the phase speed monotonically decreases relative to C. It becomes zero for the shortest resolvable wavelength $2\Delta x$ ($k\Delta x = \pi$)-stationary. Thus, all waves propagate at a speed less than the true phase speed C.



$$u_{j+1} = u_{j-1}$$

$$\frac{\partial u_j}{\partial t} + C\frac{u_{j+1} - u_{j-1}}{2\Delta x} = \frac{\partial u_j}{\partial t} = 0$$

Using the finite-difference method to solve the advection equation, we have encountered two difficulties:

1) The advection speed is less than the true advection speed

2) The advection speed changes with wave number: this false dispersion is particularly serious for the shortest waves

If the pattern that is being advected represents a superposition of more than one wave, this false dispersion will result in a deformation of that pattern.

Examples: 1) Fronts (both ocean and atmosphere): the initial fields could be deformed quickly!

If the patterns are advected with a wave-number dependent phase speed, the nature of the energy propagation will be affected! Therefore, the difference scheme might also destroy the true energy balance due to the difference approach.

The group velocity

$$c_g = \frac{d\omega}{dk} = \frac{d(kC)}{dk} = C \qquad (3.19)$$

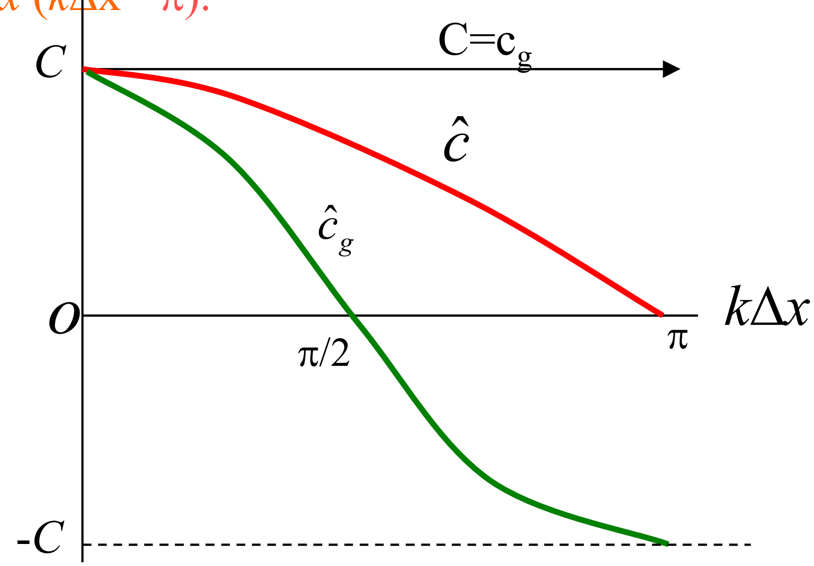
So, in the advection equation with a constant C, its group velocity is also constant and equals to its phase speed.

After the equation is discretized, its group velocity becomes

$$\hat{c}_g = \frac{d\omega}{dk} = \frac{d(k\hat{c})}{dk} = \frac{d}{dk}(kC\frac{\sin k\Delta x}{k\Delta x}) = C\cos k\Delta x \qquad (3.20)$$

As $k\Delta$x increase from zero, the group velocity decreases monotonically relative to c_g and becomes $-c_g$ for the shortest resolvable wavelength of $2\Delta x$ ($k\Delta$x= π).

This implies that the central difference scheme in space could cause the energy propagate in the opposite direction!



Nonlinear Instability

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{3.21}$$

This is usually called "advective equation for momentum and sometimes is also called "shock equation".

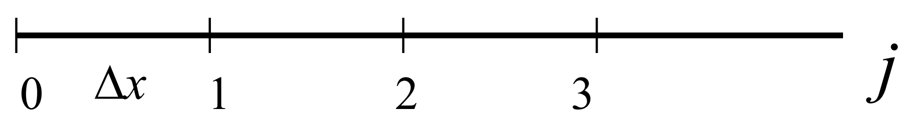
The general solution of (3.21) is

$$u = f(x - ut) \tag{3.22}$$

where *f* is an arbitrary function.

Minimum length of a wave is 2Δx, therefore, the maximum wave number

$$k_{\max}\Delta x = \pi \longrightarrow k_{\max} = \frac{\pi}{\Delta x}$$



When preformed in finite differences, nonlinear terms could result in an error related to the inability of the discrete grids to resolve the wave lengths shorter than 2Δx, For example,

$$u = \sin kx \qquad \text{and } k < k_{\max} \tag{3.23}$$

Then,

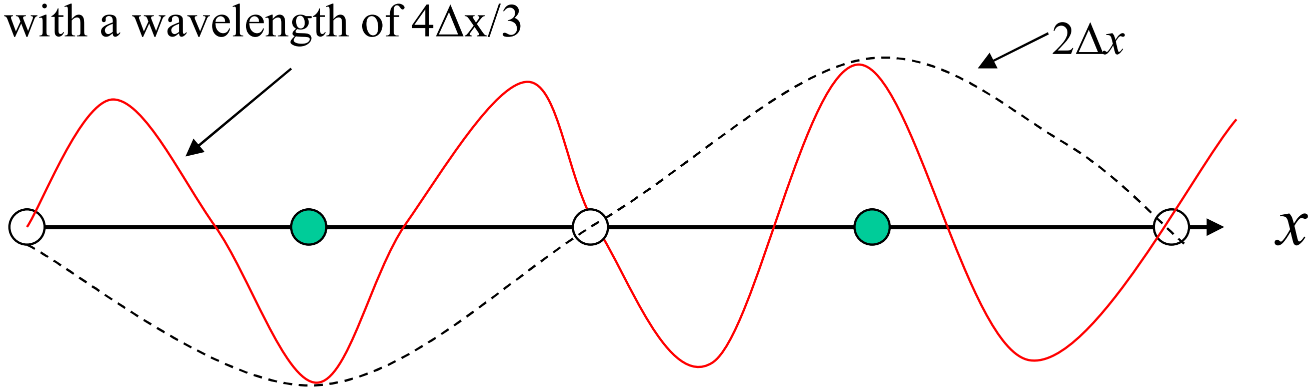
$$u\frac{\partial u}{\partial x} = k \sin kx \cos kx = \frac{1}{2} k \sin 2kx \tag{3.24}$$

If the wave number in (3.23)

$$\frac{1}{2}k_{\max} < k \le k_{\max}$$

Then, the nonlinear term will produce a wave number ($2k > k_{max}$) that can not resolved by the grid! This can not be properly reproduced in a finite-difference calculation!

A nonlinear wave with a wavelength of 4Δx/3



In a more general case, suppose that a function consists of a number of harmonic components

$$u = \sum_n u_n \tag{3.25}$$

Then, the nonlinear term will produce a products of harmonics of different wave lengths such as

$$\sin k_1 x \sin k_2 x = \frac{1}{2}[\cos(k_1 - k_2)x - \cos(k_1 + k_2)x]$$

Then, even if a finite difference calculation is started with waves which all have $k \le k_{max,}$ very soon through this process of nonlinear interaction waves will be formed with $k > k_{max}$, and a misrepresentation of waves will occur!

This misrepresentation error is called "aliasing error"

Energy spectrum

For example, the combination of two waves: $k_1+k_2 > k_{max.}$ Because the energy of this combined high wave number wave can not be resolved in the finite-difference scheme, it will be misplaced into the energy pool with $k <_{kmax.}$

Comments:

1) Aliasing errors: misrepresents the wave shapes and energy and it can cause the nonlinear instability of the numerical calculation. Such an error is not be able to control by the horizontal resolution, even it is related to it.

2) Nonlinear instability can be determined using the harmonic analysis method shown in the linear equation. This instability does not depend on the ratio of the time step to space resolution.

The best way to control the nonlinear instability:

Develop a numerical scheme to conserve the momentum and mass in the individual cells and entire computational domain.