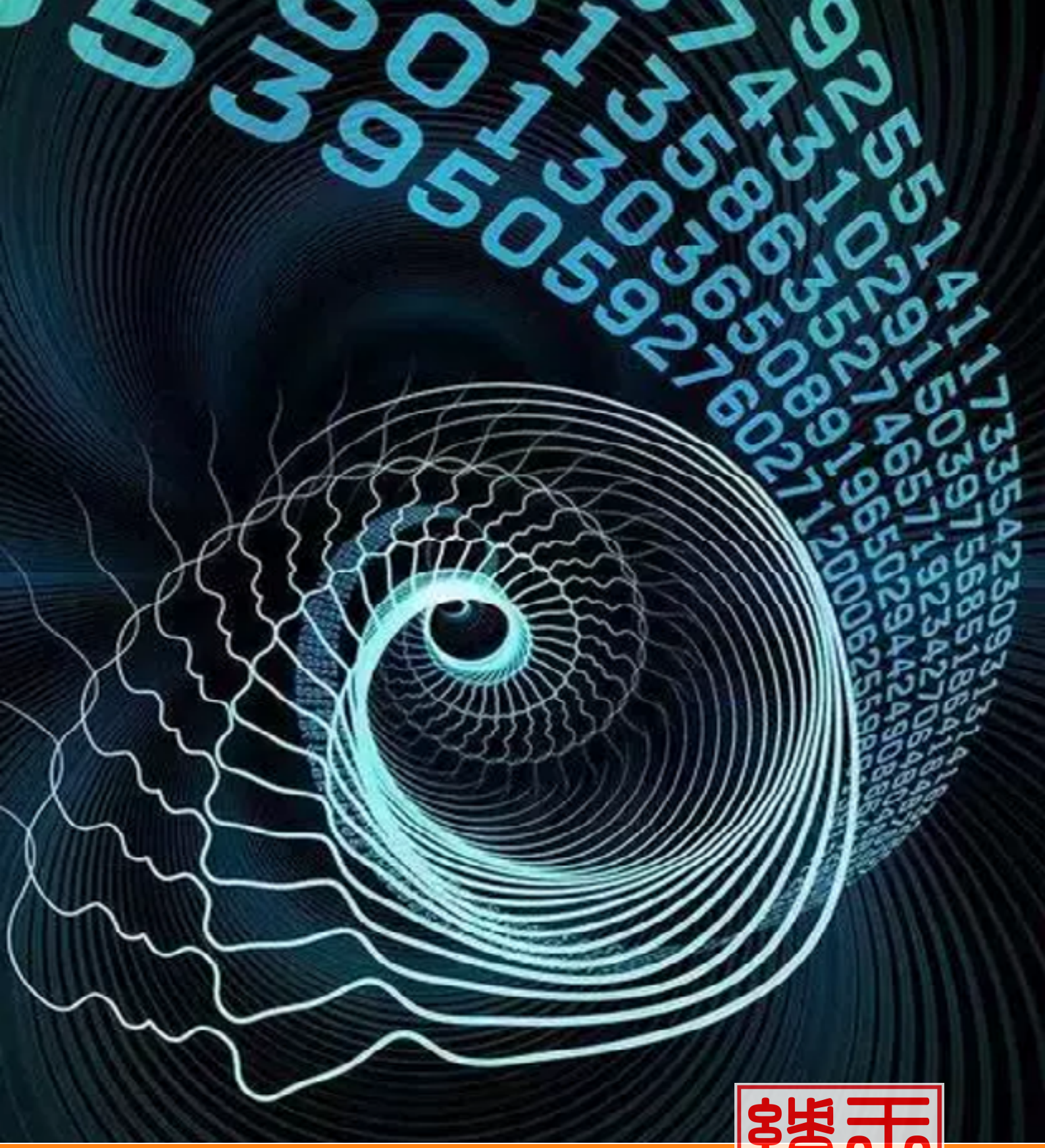

Finite Difference Method

Author: Zhuangji Wang

Talent is enduring patience. – Voltaire

Contents

Chapter 1 Parabolic Equations

1.1 Warming Up, Solving Linear Equations

After discretization, the original problem that solving a differential equation can be reduced to solving a system of linear equations. Usually, the coefficient matrix is ill-conditioned, but it may have good properties, e.g., the coefficient matrix can be sparse, tri-diagonal and diagonally dominant. We would like to first discuss solving linear equations because (a) it is a necessary step in numerical PDE with nearly all kinds of schemes and (b) it is a good programming exercise for beginners. Recall that the basic properties of the linear system derived from numerical PDEs are sparsity, diagonal dominance, ill-condition.

Remark *(Condition Number)*
Given $A \in \mathbb{R}^N$ is a non-singular matrix, $\lambda_1, \lambda_2, ..., \lambda_N$ are the eigenvalues listed in an increasing order w.r.t. the complex metric. Then $\|A\|\|A\|^{-1}$ is the condition number, and $|\lambda_N|/|\lambda_1|$ is the spectral condition number.

THOMAS ALGORITHM

Thomas algorithm is a direct method in solving linear systems. Consider an n-dimensional, tri-diagonal linear system

$$\begin{pmatrix} b_1 & c_1 & 0 & 0 & ... & 0 \\ a_2 & b_2 & c_2 & 0 & ... & 0 \\ 0 & a_3 & b_3 & c_3 & ... & 0 \\ ... & ... & ... & ... & ... & ... \\ 0 & 0 & ... & 0 & a_n & b_n \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ ... \\ u_n \end{pmatrix} = d$$

The MATLAB pseudo code is

```
% ------------ Thomas algorithm --------------------
r(1)=c(1)/b(1);y(1)=d(1)/b(1);
for i=2:n  r(i)=c(i)/(b(i)-r(i-1)*a(i));y(i)=(d(i)-y(i-1)*a(i))/(b(i)-r(i-1)*a(i)); end
u(n)=y(n);
for i=n-1:-1:1  u(i)=y(i)-r(i)*u(i+1); end
```

GENERAL ITERATION METHOD

Consider an n-dimensional linear system $Au = b$ and recast it into an iterative form

$$u^{(k+1)} = Tu^{(k)} + d,\ k = 0, 1, ... \tag{1.1}$$

If the spectral norm of the iteration matrix T, i.e., the square root of the maximum eigenvalue of $T^T T$, $\rho(T) < 1$, then the convergence of the iteration is ensured.

1. Jacobi iteration: Let $D = diag\{a_{11}, a_{22}, ..., a_{NN}\}$, and $B = D - A$, then $Au = Du - Bu = b$, i.e., $u = D^{-1}Bu + D^{-1}b = (I - D^{-1}A)u + D^{-1}b$. Recall that D is diagonal, s.t. obtaining D^{-1} is simple. The Jacobi iteration can be summarized as
$$u^{(k+1)} = (I - D^{-1}A)u^{(k)} + D^{-1}b \tag{1.2}$$
2. Gauss-Seidel iteration: This is a stochastic version of Jacobi iteration. Let $D = diag\{a_{11}, a_{22}, ..., a_{NN}\}$, $-C$ be the upper triangular part of A (not including the diagonal), and $-B$ be the lower triangular part

of A, i.e., $A = D - B - C$. The Jacobi iteration can be rewritten as the Gauss-Seidel iteration

$$u^{(k+1)} = D^{-1}Bu^{(k+1)} + D^{-1}Cu^{(k)} + D^{-1}b \tag{1.3}$$

3. Richardson iteration: This is a relaxed version of Jacobi iteration. Consider the iteration form $u^{(k+1)} = u^{(k)} - \omega(Au^{(k)} - b)$, i.e., the iteration matrix is $T = I - \omega A$. Suppose λ_i is the eigenvalue of A. If A is SPD and $0 < \lambda_1 \leq \lambda_i \leq \lambda_N$, the optimal value of ω is $\omega_{opt} = 2/(\lambda_1 + \lambda_N)$. Then the iteration form follows.

1.2 Simple Linear Parabolic Equations and Explicit Schemes

A model heat equation problem can be shown as follows

$$\begin{cases} u_t(x,t) = u_{xx}(x,t), & 0 \leq t \leq t_F, 0 \leq x \leq 1 \\ u(0,t) = u(1,t) = 0, & 0 \leq t \leq t_F \\ u(x,0) = u_0(x), & 0 \leq x \leq 1 \end{cases} \tag{1.4}$$

$t \in \mathbb{R}_+$ represents the time domain, and we restrict it to a finite time period $[0, t_F]$.$x \in [0,1] \subseteq \mathbb{R}$ represents the spatial domain. Numerically solving a PDE includes two steps, (a) to discretize the PDE by finite difference and (b) to solve the discretized PDE. Let us focus on the domain $[0, t_F] \times [0, 1]$.

Partition $[0,1]$ into J equal intervals with mesh nodes $\{x_0, x_1, ..., x_j, ..., x_{J-1}, x_J\}$, $x_j = j\Delta x, j = 0, 1, 2, ..., J$ or $\Delta x = J^{-1}$. Similarly, Partition $[0, t_F]$ into N equal intervals with mesh nodes $\{t_0, t_1, ..., t_n, ..., t_{N-1}, t_N\}$, $t_n = n\delta t, n = 0, 1, 2, ..., N$. The goal of FDM is to approximate the solution of $u(x,t)$ at all of the mesh nodes $(x_j, t_n) = (j\Delta x, n\Delta t)$, and we denote the numerical solution as $U_j^n = u(x_j, t_n)$.

FDM uses finite difference to approximate the derivatives based on the Taylor expansions, i.e.,

$$\begin{aligned} u(x, t+\Delta t) &= u(x,t) + u_t(x,t)\Delta t + \frac{1}{2}u_{tt}(x,t)\Delta t^2 + ... \\ u(x+\Delta x, t) &= u(x,t) + u_x(x,t)\Delta x + \frac{1}{2}u_{xx}(x,t)\Delta x^2 + \frac{1}{6}u_{xxx}(x,t)\Delta x^3 + ... \end{aligned}$$

There are three kinds of finite differences that will be discussed

1. Forward differences:

$$\begin{aligned} \Delta_{+t}u(x,t) &= u(x, t+\Delta t) - u(x,t) \\ \Delta_{+x}u(x,t) &= u(x+\Delta x, t) - u(x,t) \end{aligned} \tag{1.5}$$

2. Backward differences:

$$\begin{aligned} \Delta_{-t}u(x,t) &= u(x,t) - u(x, t-\Delta t) \\ \Delta_{-x}u(x,t) &= u(x,t) - u(x-\Delta x, t) \end{aligned} \tag{1.6}$$

3. Central differences (including the second order):

$$\begin{aligned} \delta_t u(x,t) &= u(x, t+0.5\Delta t) - u(x, t-0.5\Delta t) \\ \delta_x u(x,t) &= u(x+0.5\Delta x, t) - u(x-0.5\Delta x, t) \\ \Delta_{0x}u(x,t) &= \frac{1}{2}(\Delta_{+x} + \Delta_{-x})u(x,t) = \frac{1}{2}\left[u(x+\Delta x, t) - u(x-\Delta x, t)\right] \\ \delta_x^2 u(x,t) &= u(x+\Delta x, t) - 2u(x,t) + u(x-\Delta x, t) \end{aligned} \tag{1.7}$$

We can use the Taylor expansion to interpret the forward finite differences

$$u(x,t+\Delta t)-u(x,t)=u(x,t)+u_t(x,t)\Delta t+\frac{1}{2}u_{tt}(x,t)\Delta t^2-u(x,t)=u_t(x,t)\Delta t+O(\Delta t^2)$$
$$u_t(x,t)=\frac{u(x,t+\Delta t)-u(x,t)}{\Delta t}+O(\Delta t)=\frac{\Delta_{+t}u(x,t)}{\Delta t}+O(\Delta t)$$

Using forward difference in the time domain and central difference in the spatial domain, the original heat equation Eq.1.4 at $t-t_n, x=x_j$ can be discretized as

$$u_t(x_j,t_n)\approx\frac{\Delta_{+t}u(x_j,t_n)}{\Delta t}=\frac{U_j^{n+1}-U_j^n}{\Delta t}$$
$$u_{xx}(x_j,t_n)\approx\frac{\delta_x^2u(x_j,t_n)}{\Delta x^2}=\frac{U_{j+1}^n-2U_j^n+U_{j-1}^n}{\Delta x^2}$$

That induces the following **EXPLICIT SCHEME (FORWARD EULER METHOD)**

$$\begin{aligned}&U_j^{n+1}=U_j^n+\mu\left(U_{j+1}^n-2U_j^n+U_{j-1}^n\right), && \mu=\Delta t/(\Delta x)^2\\ &U_0^n=0,\ U_J^n=0, && n=0,1,2,...\\ &U_j^0=u_0(x_j), && j=0,1,2,...,J-1,J\end{aligned}\tag{1.8}$$

Remark *(Separation of Variables)*

Recall the exact solution of the heat equation

$$u_t(x,t)=u_{xx}(x,t),\ t\geq 0,\ x\in[0,1]$$

Using the separation of variables, let $u(x,t)=f(x)g(x)$ *and we obtain*

$$f\cdot g'=f''\cdot g\Rightarrow\frac{f''}{f}=\frac{g'}{g}=-k^2$$

Suppose k *is a position constant, and we can have* $f''+k^2f=0, g'+k^2g=0$. *Use the BCs* $u(0,t)=u(1,t)=0$ $u_k(x,t)=e^{-k^2t}\sin kx$. *Choose* $k=m\pi, m\in\mathbb{N}_+$, *then the linear combination w.r.t.* k.

$$u(x,t)=\sum_{m=1}^{\infty}a_me^{-(m\pi)^2t}\sin m\pi x$$

Align $u(x,t)$ *with the IC, we obtain* $u_0(x)=\sum_{m=1}^{\infty}a_m\sin m\pi x$ *with the Fourier expansion*

$$a_m=2\int_0^1u_0(x)\sin m\pi x dx$$

In summary, we suppose to use FDM to solve the heat equation in Eq.1.4 with $u_0(0)=u_0(1)=0$. First, the computational domain should be discretized, and the derivatives should be approximated with finite differences (the three types of finite difference schemes in Eqs. 1.5-1.7). Second, the discretized PDE should be solved as a difference equation. An explicit finite difference scheme is

$$\frac{U_j^{n+1}-U_j^n}{\Delta t}\approx u_t(x,t)=u_{xx}(x_j,t_n)\approx\frac{U_{j+1}^n-2U_j^n+U_{j-1}^n}{\Delta x^2}\tag{1.9}$$

or in a simplified form

$$U_j^{n+1}=U_j^n+\mu\left(U_{j+1}^n-2U_j^n+U_{j-1}^n\right),\ \mu=\Delta t/\Delta x^2\tag{1.10}$$

Then, we will shift to some theoretical issues of FDM.

1.2.1 Consistency

Consistency denotes whether the difference equation approximates the original PDE faithfully as $\Delta t\to 0, \Delta x\to 0$. Truncation error ($T(x,t)$) is a measure of consistency, where in the difference scheme, U_j^n is replaced with

$u(x_j, t_n)$, i.e.,

$$T(x,t) = \frac{u(x,t+\Delta t) - u(x,t)}{\Delta t} - \frac{u(x+\Delta x,t) - 2u(x,t) + u(x-\Delta x,t)}{\Delta x^2} = \frac{\Delta_{+t}u(x,t)}{\Delta t} - \frac{\delta_x^2 u(x,t)}{\Delta x^2}$$

We would like to show $T(x,t) \to u_t - u_{xx} = 0$ as $\Delta t \to 0, \Delta x \to 0$.

Proof Use Taylor expansion at (x,t)

1. For the time domain
$$\Delta_{+t}u(x,t) = \left(u(x,t) + u_t(x,t)\Delta t + \frac{1}{2}u_{tt}(x,t)\Delta t^2 + ...\right) - u(x,t) = u_t(x,t)\Delta t + \frac{1}{2}u_{tt}(x,t)\Delta t^2 + ...$$
2. For the spatial domain
$$\begin{aligned}\delta_x^2 u(x,t) = &\left(u(x,t) + u_x(x,t)\Delta x + \frac{1}{2}u_{xx}(x,t)\Delta x^2 + \frac{1}{6}u_{xxx}(x,t)\Delta x^3 + \frac{1}{24}u_{xxxx}(x,t)\Delta x^4 + ...\right)\\ &+ \left(u(x,t) - u_x(x,t)\Delta x + \frac{1}{2}u_{xx}(x,t)\Delta x^2 - \frac{1}{6}u_{xxx}(x,t)\Delta x^3 + \frac{1}{24}u_{xxxx}(x,t)\Delta x^4 + ...\right)\\ &-2u(x,t) = u_{xx}(x,t)\Delta x^2 + \frac{1}{12}u_{xxxx}(x,t)\Delta x^4 + ...\end{aligned}$$

Therefore,

$$\begin{aligned}T(x,t) =& \underbrace{u_t(x,t) - u_{xx}(x,t)}_{=0} + \frac{1}{2}u_{tt}(x,t)\Delta t - \frac{1}{12}u_{xxxx}(x,t)\Delta x^2 + O(\Delta t^2 + \Delta x^3)\\ =& \frac{1}{2}u_{tt}(x,t)\Delta t - \frac{1}{12}u_{xxxx}(x,t)\Delta x^2 + O(\Delta t^2 + \Delta x^3)\\ =& \frac{1}{2}u_{tt}(x,\eta)\Delta t - \frac{1}{12}u_{xxxx}(\zeta,t)\Delta x^2\end{aligned}$$

where η is between t and $t+\Delta t$, $\zeta \in (x-\Delta x, x+\Delta x)$, and we applied the original PDE $u_t(x,t) - u_{xx}(x,t) = 0$. Suppose the solution $u(x,t)$ is smooth enough, s.t. $M_{tt} = \max|u_{tt}| < \infty$, $M_{xxxx} = \max|u_{xxxx}| < \infty$. Then

$$|T(x,t)| \le \frac{1}{2}M_{tt}(x,\eta)\Delta t + \frac{1}{12}M_{xxxx}(\zeta,t)\Delta x^2 \to 0, \text{ as } \Delta t \to 0, \Delta x \to 0$$

for $\forall (x,t) \in [\tau, t_F) \times (0,1)$, $\tau > 0$. We say this scheme is unconditionally consistent. ■

Remark *(Order of Accuracy)*

1. *If $\mu = \Delta t/\Delta x^2 \le \infty$, then $T(x,t) = O(\Delta t)$, and we say that the scheme has the* **FIRST ORDER ACCURACY**.
2. *As a special case, for a PDE with constant coefficients, i.e., $u_t = u_{xx}$, we automatically have $u_{tt} = u_{xxt} = u_{xxxx}$. Then, it is possible to simplify the truncation error further.*
$$T(x,t) = \frac{1}{2}u_{tt}(x,\eta)\Delta t - \frac{1}{12}u_{xxxx}(\zeta,t)\Delta x^2 \approx \frac{1}{2}\left(1 - \frac{1}{6\mu}\right)u_{tt}(x,\eta)\Delta t + O(\Delta t^2)$$
Take $\mu = 1/6$, then $T(x,t) = O(\Delta t^2)$ and the scheme has the **SECOND ORDER ACCURACY**.

1.2.2 Convergence

Convergence focuses on that if U_j^n converges to $u(x_j, t_n)$ as $\Delta t \to 0, \Delta x \to 0$. Given a sequence of pairs $(\Delta x, \Delta t)$, we could obtain a set of numerical solution U_j^n. Therefore, $U_j^n \to u(x_j, t_n)$ as $\Delta t \to 0, \Delta x \to 0$ can be stated as

$$\left(U_j^n\right)_i \to u(x_j, t_n), \ \left(U_j^n\right)_i \in \left\{\left(U_j^n\right)_i\right\}_{i=0}^{\infty}$$

We say the numerical scheme is convergent if for any fixed point $(x^*, t^*) \in (\tau, t_F) \times (0,1)$, $U_j^n \to u(x_j, t_n)$ as $\Delta t \to 0, \Delta x \to 0$.

Proof Suppose the truncation error $T_j^n = T(x_j, t_n)$ is bounded at any mesh point, i.e., $|T_j^n| \leq \overline{T} < \infty$, with $\overline{T}$ as some constant. Define the approximation error $e = U - u$ or $e_j^n = U_j^n - u(x_j, t_n)$. For the numerical solution,

$$U_j^{n+1} = U_j^n + \mu \left(U_{j+1}^n - 2U_j^n + U_{j-1}^n\right)$$

For the exact solution,

$$T(x_j, t_n) = \frac{u(x_j, t_{n+1}) - u(x_j, t_n)}{\Delta t} - \frac{u(x_{j+1}, t_n) - 2u(x_j, t_n) + u(x_{j-1}, t_n)}{\Delta x^2}$$
$$\Rightarrow u(x_j, t_{n+1}) = u(x_j, t_n) + \mu \left[u(x_{j+1}, t_n) - 2u(x_j, t_n) + u(x_{j-1}, t_n))\right] + T(x_j, t_n)\Delta t$$

Using the definition of the approximation error, and take the difference between the two equations above

$$\begin{aligned} e_j^{n+1} &= e_j^n + \mu \left(e_{j+1}^n - 2e_j^n + e_{j-1}^n\right) - T_j^n \Delta t \\ &= \mu e_{j+1}^n + (1 - 2\mu) e_j^n + \mu e_{j-1}^n - T_j^n \Delta t \end{aligned}$$

If $0 \leq \mu \leq 1/2$, we have $\mu \geq 0, 1 - 2\mu \geq 0$. Let $E^n = \max \left|e_j^n\right|, j = 0, 1, 2, ..., J$ be the maximum error on n^{th} time step, then

$$\left|e_j^{n+1}\right| \leq \mu E^n + (1 - 2\mu) E^n + \mu E^n + T_j^n \Delta t \leq E^n + \overline{T}\Delta t, \; j = 0, 1, 2, ..., J$$

In another word, we have

$$E^{n+1} \leq E^n + \overline{T}\Delta t$$

We just need $E^n \to 0$, or say, we just need $E^0 \to 0$. Fortunately, based on the IC, we have $E^0 = 0$ as $n = 0$. Therefore, the recursive relation leads to $E^n \leq n\overline{T}\Delta t, n = 0, 1, 2, ..., N$. Since $n\Delta t \leq t_F$ and $\overline{T} \leq \frac{1}{2} M_{tt} \Delta t + \frac{1}{12} M_{xxxx} \Delta x^2$, we obtain,

$$E^n \leq t_F \left[\frac{1}{2} M_{tt} \Delta t + \frac{1}{12} M_{xxxx} \Delta x^2\right]$$

Then, $\Delta t \to 0, \Delta x \to 0$ implies $E^n \to 0$. We also call this scheme is of the **FIRST ORDER ACCURACY**. ■

Remark *(Another Definition of Truncation Error)*
There is another way to define the truncation error. We know that the truncation error is defined by replacing the numerical solution with the exact solution in the (discretized) numerical schemes. So traditionally, the truncation error is written as

$$T(x, t) = \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} - \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{\Delta x^2}$$
$$T(x_j, t_n) = \frac{u(x_j, t_{n+1}) - u(x_j, t_n)}{\Delta t} - \frac{u(x_{j+1}, t_n) - 2u(x_j, t_n) + u(x_{j-1}, t_n)}{\Delta x^2}$$

But there is another way to define it from

$$U_j^{n+1} = U_j^n + \mu \left(U_{j+1}^n - 2U_j^n + U_{j-1}^n\right)$$

i.e., define T^* as

$$T^*(x, t) = u(x, t + \Delta t) - u(x, t) - \mu \left[u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)\right]$$

Thus, $T^*(x, t) = T(x, t)\Delta t$. Although $T^*(x, t) = T(x, t)\Delta t = O(\Delta t^2)$, we still call this scheme has the first order accuracy, and in general, $T^*(x, t) = O(\Delta t^{n+1})$ has the n^{th} order accuracy.

In this example of the heat equation, since the exact solution is stable, and the numerical solution converges to the exact solution, then we cna show the stability of the numerical solution simultaneously. We summarize our results as follows.

Definition 1.1 (Refinement Path)

A refinement path is defined as the sequence of sets

$$\{(\Delta x)_i, (\Delta t)_i,\ i, i = 0, 1, 2, ..., N,\ (\Delta x)_i, (\Delta t)_i \to 0\}$$

and we define the grid ratio or the mesh ratio

$$\mu_i = (\Delta t)_i/(\Delta x)_i^2$$

An example refinement path can be shown in Fig. 1.1.

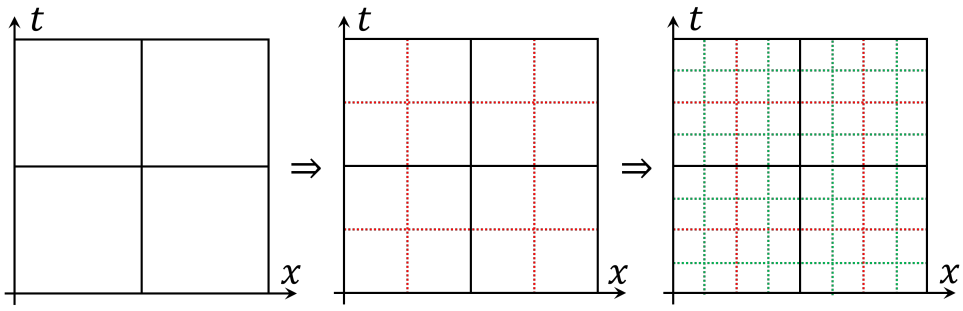

Figure 1.1: An example of the "refinement path"

Theorem 1.1 (Convergence of the Explicit Scheme)

If a refinement path satisfies $\mu_i \leq 1/2$ for all sufficient large values of i, $\exists$ positive numbers n_i, j_i s.t. $n_i(\Delta t)_i \to t$ and $j_i(\Delta x)_i \to x$, and $|u_{tt}| \leq M_{tt}$ and $|u_{xxxx}| \leq M_{xxxx}$ uniformly, then $\left\{U_j^n\right\}_{i=0}^{\infty}$ converges to $u(x,t)$ of the original PDE uniformly.

1.2.3 Stability with Fourier Analysis

We apply the **FOURIER STABILITY ANALYSIS** or the **VON NEUMANN TECHNIQUE** to demonstrate the stability and convergence. In practice, a more general case is that we use this proposed way to first prove the stability of the numerical scheme and then prove the convergence. We first check the exact solution. For the heat equation $u_t = u_{xx}$, using the separation of variables, we obtain

$$u(x,t) = f(x) \cdot g(t) \Rightarrow f \cdot g' = f'' \cdot g \Rightarrow \frac{f''}{f} = \frac{g'}{g} = -k^2$$

Hence, $g(t) = e^{-k^2 t}, f(x) = e^{ikx}$. Let $k = m\pi, m \in \mathbb{Z}$, the solution can be written as follows, and the coefficients a_m can be determined from the IC.

$$u(x,t) = \sum_{m=-\infty}^{\infty} a_m e^{-(m\pi)^2 t} e^{im\pi x},\ a_m = \int_0^1 u_0(x) e^{-im\pi x}$$

Recall the numerical scheme $U_j^{n+1} = U_j^n + \mu\left(U_{j+1}^n - 2U_j^n + U_{j-1}^n\right)$, and we can assume the numerical solution also follows a Fourier mode, i.e., $U_j^n = \lambda^n e^{ikj\Delta x}$ and $U_j^{n+1} = \lambda^{n+1} e^{ikj\Delta x} = \lambda U_j^n$, which is similar to the evolution of the exact solution. Plug the Fourier mode into the explicit scheme

$$\lambda^{n+1} e^{ikj\Delta x} = \lambda^n e^{ikj\Delta x} + \mu\left(\lambda^n e^{ik(j+1)\Delta x} - 2\lambda^n e^{ikj\Delta x} + \lambda^n e^{ik(j-1)\Delta x}\right)$$

We can write λ as a function of k, i.e., $\lambda = \lambda(k)$,

$$\lambda(k) = 1 + \mu\left(e^{ik\Delta x} - 2 + \lambda^n e^{-ik\Delta x}\right) = 1 - 2\mu(1 - \cos k\Delta x) = 1 - 4\mu \sin^2 \frac{1}{2} k\Delta x$$

λ is called the **AMPLIFICATION FACTOR** of the mode k. Then $\forall k = m\pi$, we can claim a numerical approximation as follows.

$$U_j^n = \sum_{m=-\infty}^{\infty} A_m e^{im\pi(j\Delta x)} |\lambda(m\pi)|^n$$

We note that when $t = 0$, i.e., $n = 0$, and if the Fourier expansion of the IC exists, then $A_m = a_m$.

To compare U_j^n and $u(x,t)$, we partition the solutions into two portions:

1. In the portion with relatively low frequency, i.e., $|k|, |m|$ are small, for the numerical solution, we have the expansion for $\lambda(m\pi)$. For the exponential decreasing part in the exact solution, $e^{-(m\pi)^2 t} = e^{-(m\pi)^2 n\Delta t} = \left[e^{-(m\pi)^2\Delta t}\right]^n$, we can also do the Taylor expansion. Therefore, compare the following equations

$$\lambda(m\pi) = 1 - 4\mu \sin^2 \frac{1}{2}(m\pi)\Delta x = 1 - (m\pi)^2\Delta t + \frac{1}{12}(m\pi)^4\Delta t(\Delta x)^2 + ..., \quad \mu = \frac{\Delta t}{\Delta x^2}$$

$$e^{-(m\pi)^2\Delta t} = 1 - (m\pi)^2\Delta t + \frac{1}{2}(m\pi)^4\Delta t^2 + ...$$

   The first two terms in $e^{-(m\pi)^2\Delta t}$ and $\lambda(m\pi)$ are of the same. Therefore, $\lambda(m\pi)$ provides the first order approximation of $e^{-(m\pi)^2\Delta t}$, or $\lambda(m\pi)$ approximates $e^{-(m\pi)^2\Delta t}$ at least up to $O(\Delta t)$.
2. In the portion with relatively high frequency, i.e., $|k|, |m|$ are large, we observe the fact that $u(x,t)$ in such modes will be bounded by $e^{-(m\pi)^2 t}$, which vanish rapidly w.r.t. m and t.

For stability, we hope that the numerical solution can be bounded by $\lambda, \forall n$. Thus, we need to estimate λ.

$$\lambda(m\pi) = 1 - 4\mu \sin^2 \frac{1}{2}(m\pi)\Delta x \underbrace{\leq 1 - 4\mu}_{\text{the worst case}} \tag{1.11}$$

To ensure $|\lambda(m\pi)| \leq 1$, it is sufficient to show $|1 - 4\mu| \leq 1$, i.e., $0 \leq \mu \leq 1/2$. Thus, when $0 \leq \mu \leq 1/2$, $|\lambda(m\pi)| \leq 1$ and $|\lambda(m\pi)|^n \leq 1$ are bounded. The requirement of μ matches the requirement in the convergence section.

Remark *When $\mu > 1/2$, in particular, when $m\Delta x = 1$, s.t. $\sin^2(m\pi)\Delta x/2 = 1$, $|\lambda(m\pi)| > 1$ and the associate modes will grow unbounded w.r.t. n.*

Definition 1.2 (Stability)

For the model problem in Eq.1.4, a numerical scheme is stable if $\exists K$, s.t. $\forall m$, K is independent to $k = m\pi$, and $|\lambda(m\pi)|^n \leq K, \forall n\Delta t \leq t_F$. ♣

The condition can be relaxed if $|\lambda(m\pi)| < 1 + K'\Delta t$ for all of the modes, i.e., k or m. That is necessary and sufficient for the convergence of a consistent finite difference scheme. If $|\lambda(m\pi)| < 1 + K'\Delta t$, then $|\lambda(m\pi)|^n < (1 + K'\Delta t)^n \leq 1 + O(\Delta t)$.

We redo the convergence analysis using von Neumann technique, where the smoothness of the exact solution is not necessary, i.e., the error may not be expressed based on the partial derivatives, such as u_{tt} and u_{xxxx}.

Suppose the Fourier expansion of the IC is absolutely convergent, then the error $e = U - u(x,y)$ or $e_j^n = U_j^n - u(x_j, t_n)$ can be expressed as

$$e_j^n = U_j^n - u(x_j, t_n) = \sum_{m=-\infty}^{\infty} A_m e^{im\pi j\Delta x} \left[|\lambda(m\pi)|^n - e^{-(m\pi)^2 n\Delta t}\right]$$

Equipped with the assumption of absolute convergence, $\forall \varepsilon > 0$, $\exists M_0 > 0$, s.t. $\sum_{|m|>M_0} |A_m| \leq \varepsilon/2$. That is associated with the portion of high frequency, where $e^{-(m\pi)^2 n\Delta t}$ also vanishes fast w.r.t. m, and $|\lambda(k)|^n \leq 1$ and at most of order $O(1)$ since Eq.1.11.

$$|e_j^n| \leq \frac{1}{2}\varepsilon + \underbrace{\sum_{|m|\leq M_0} |A_m| \times \left||\lambda(m\pi)|^n - e^{-(m\pi)^2 n\Delta t}\right|}_{\text{the low frequency portion}}$$

A fact shows that if $|a| \leq 1, |b| \leq 1$, then $|a^n - b^n| \leq n|a-b|$. Assume $\mu = \Delta t/(\Delta x)^2 \leq 1/2$, and since $\lambda(m\pi) \leq 1$, $e^{-(m\pi)^2 n\Delta t} \leq 1$, we can apply that fact.

$$\begin{aligned} |e_j^n| &\leq \frac{1}{2}\varepsilon + \sum_{|m|\leq M_0} n \times |A_m| \times \left||\lambda(m\pi)| - e^{-(m\pi)^2\Delta t}\right| \\ &\leq \frac{1}{2}\varepsilon + \sum_{|m|\leq M_0} |A_m|\, nC(\mu)(m\pi)^4(\Delta t)^2 \leq \frac{1}{2}\varepsilon + t_F C(\mu)\pi^4 \left[\sum_{|m|\leq M_0} |A_m| m^4\right] \Delta t \end{aligned}$$

In the second step, $C(\mu)$ is a constant that depends on $\mu = \Delta t/(\Delta x)^2$, which can be derived based on the Taylor expansions of $\lambda(m\pi)$ and $e^{-(m\pi)^2\Delta t}$. $|m| \leq M_0$ indicates the sum is finite. Therefore, it is always possible to choose a sufficiently small Δt, s.t.

$$t_F C(\mu)\pi^4 \left[\sum_{|m|\leq M_0} |A_m| m^4\right] \Delta t \leq \frac{1}{2}\varepsilon \Rightarrow |e_j^n| \leq \frac{1}{2}\varepsilon + \frac{1}{2}\varepsilon = \varepsilon$$

Remark *The assumption on the smoothness of the solution is stronger than the assumption on the absolute convergence of the Fourier expansion of the IC. Thus, the von Neumann technique relaxes the condition for the convergence and stability. The von Neumann technique is useful in handling linear problems. E.g., when the IC is a step function, the smoothness of the solution will fail, but the Fourier expansion still converges absolutely, although not uniformly.*

We provide a summary to the explicit scheme. Three theoretical issues are considered. Consistency is measured with the truncation error. Stability is processed with the Fourier stability analysis. Convergence can be proved either using the Taylor expansion of the error functions (need smoothness condition) or using the von Neumann technique (need absolute convergence of the Fourier expansion). The numerical computation of the explicit scheme is simple. However, the explicit scheme is conditionally stable, i.e. $\mu \leq 1/2$. Thus, if solving problems with large t_F, the computation could be expensive.

1.3 Fully Implicit Schemes

The **FULLY IMPLICIT SCHEME (BACKWARD EULER METHOD)** can be obtained by slightly revising the explicit scheme.

$$\begin{aligned} &\frac{U_j^{n+1} - U_j^n}{\Delta t} = \frac{U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}}{\Delta x^2} \\ &-\mu U_{j-1}^{n+1} + (1+2\mu)U_j^{n+1} - \mu U_{j+1}^{n+1} = U_j^n,\ \mu = \Delta t/\Delta x^2 \end{aligned} \tag{1.12}$$

We provide theoretical discussion of the fully implicit scheme in the way similar to the explicit scheme. However, we will do stability analysis first, which will naturally induce convergence.

1.3.1 Consistency

Define the truncation error as

$$T(x,t+\Delta t)=\frac{u(x,t+\delta t)-u(x,t)}{\Delta t}-\frac{u(x+\Delta x,t+\Delta t)-2u(x,t+\Delta t)+u(x-\Delta x,t+\Delta t)}{\Delta x^2}$$

Apply Taylor expansion at $(x,t+\Delta t)$,

$$T(x,t+\Delta t)=[u_t(x,t+\Delta t)-u_{xx}(x,t+\Delta t)]+\frac{1}{2}u_{tt}(x,t+\delta t)\Delta t-\frac{1}{12}u_{xxxx}(x,t+\Delta t)\Delta x^2+...$$

Assume the smoothness of the exact solution and suppose $\Delta t\sim\Delta x^2$, then $T(x,t+\Delta t)=O(\Delta t)$, corresponding to the first order accuracy.

1.3.2 Stability

Recall that stability implies that the numerical solution stays bounded at any time steps, which can be proved with the von Neumann stability analysis (Fourier stability analysis). Suppose the numerical solution includes k or m Fourier modes

$$\lambda^n e^{ikj\Delta x}=\lambda^n e^{i(m\pi)j\Delta x},\ k=m\pi,\ m\in\mathbb{Z}$$

Plug that specific mode into the fully implicit scheme

$$-\mu\lambda^{n+1}e^{i(m\pi)(j-1)\Delta x}+(1+2\mu)\lambda^{n+1}e^{i(m\pi)j\Delta x}-\mu\lambda^{n+1}e^{i(m\pi)(j+1)\Delta x}=\lambda^n e^{i(m\pi)j\Delta x}$$

Solve λ, we obtain

$$\lambda(k)=\lambda(m\pi)=\frac{1}{1+4\mu\sin^2\frac{1}{2}(m\pi)\Delta x} \tag{1.13}$$

Therefore, $\forall\mu>0$, $0<\lambda\leq 1$ automatically, $\forall k=m\pi$. Thus, the fully implicit scheme is unconditionally stable, i.e., μ can be arbitrarily large, although Δt should be small in terms of accuracy. Nevertheless, we can define Δt much larger than the time step in the explicit scheme, which implies the computing load for the fully implicit scheme can be ameliorated.

1.3.3 Convergence

As shown in the explicit scheme, there are two methods we can apply for convergence analysis.

VON NEUMANN ANALYSIS (FOURIER ANALYSIS)

Recall that we need one "absolute convergence" condition in this analysis. Assume that the Fourier expansion of the initial condition is absolutely convergent, i.e., $\sum_{m=-\infty}^{\infty}|A_m|<\infty$. Let $e_j^n=U_j^n-u(x_j,t_n)$, and we separate the series of e_j^n into a low frequency portion $m\leq M_0$ and a high frequency portion $m>M_0$.

$$\begin{aligned}\left|e_j^n\right|=\left|U_j^n-u(x_j,t_n)\right|&=\left|\sum_{m=-\infty}^{\infty}A_m e^{i(m\pi)j\Delta x}\left[\lambda(m\pi)^n-e^{-(m\pi)^2n\Delta t}\right]\right|\\&\leq\underbrace{\frac{1}{2}\varepsilon+\sum_{|m|<M_0}|A_m|\left|\lambda(m\pi)^n-e^{-(m\pi)^2n\Delta t}\right|}_{\text{the low frequency portion}}\leq\frac{1}{2}\varepsilon+\sum_{|m|<M_0}|A_m|\,n\left|\lambda(m\pi)-e^{-(m\pi)^2\Delta t}\right|\end{aligned}$$

Recall in the approximation of the high frequency portion, we not only uses the fact that $\sum_{|m|<M_0}|A_m|\leq\varepsilon/2$, but also use the facts that $e^{-(m\pi)^2n\Delta t}$ vanish fast for large m and $|\lambda(m\pi)|\leq 1$ since Eq.1.13.

For the low frequency portion, we have (for $k = m\pi, \mu = \Delta t/\Delta x^2$)

$$\begin{aligned}
&\lambda(k) - e^{-k^2\Delta t} = \frac{1}{1 + 4\mu \sin^2 \frac{1}{2}k\Delta x} - e^{-k^2\Delta t} \\
=&\left(1 - 4\mu \sin^2 \frac{1}{2}k\Delta x + 16\mu^2 \sin^4 \frac{1}{2}k\Delta x + ...\right) - \left(1 - k^2\Delta t + \frac{1}{2}k^4\Delta t^2 + ...\right) \\
=&\left(1 - k^2\Delta t + \frac{1}{12}k^4\Delta t\Delta x^2 + ...\right) + 16\mu^2 \sin^4 \frac{1}{2}k\Delta x + ... - \left(1 - k^2\Delta t + \frac{1}{2}k^4\Delta t^2 + ...\right) \\
\leq& C(\mu)k^4\Delta t^2
\end{aligned}$$

Therefore, $\lambda(k) - e^{-k^2\Delta t} \leq C(\mu)k^4\Delta t^2$ is bounded. Hence

$$\begin{aligned}
\left|e_j^n\right| = \left|U_j^n - u(x_j, t_n)\right| &\leq \frac{1}{2}\varepsilon + \sum_{|m|\leq M_0} |A_m| n C(\mu)(m\pi)^4\Delta t^2 \\
&\leq \frac{1}{2}\varepsilon + t_F C(\mu)\pi^4 \left[\sum_{|m|\leq M_0} |A_m| n m^4\right] \Delta t \leq \frac{1}{2}\varepsilon + \frac{1}{2}\varepsilon \leq \varepsilon, \text{ as } \Delta t \to 0
\end{aligned}$$

Hence we prove the convergence of the numerical solution.

Remark *This proof cannot tell us the order of convergence, except for an arbitrarily small ε. In order to show its order of convergence, we have to process an error estimation method.*

ERROR ESTIMATION

Recall that we need one "smoothness" condition in this analysis, where we can show that the maximum of the truncation error is bounded, i.e., $\overline{T} = \max |T_j^{n+1}| \sim O(\Delta t)$. The numerical scheme is

$$-\mu U_{j-1}^{n+1} + (1 + 2\mu)U_j^{n+1} - \mu U_{j+1}^{n+1} = U_j^n$$

The truncation error can be expressed as

$$-\mu u(x_j - \Delta x, t_n + \Delta t) + (1 + 2\mu)u(x_j, t_n + \Delta t) - \mu u(x_j + \Delta x, t_n + \Delta t) = u(x_j, t_n) + \Delta t T_j^{n+1}$$

Take the difference between the equations and let $e_j^n = U_j^n - u(x_j, t_n)$

$$-\mu e_{j-1}^{n+1} + (1 + 2\mu)e_j^{n+1} - \mu e_{j+1}^{n+1} = e_j^n - \Delta t T_j^{n+1}$$

Let $E^n = \max_{0\leq j\leq J} \left|e_j^n\right|$ be the maximum error in the nth time step, then

$$\begin{aligned}
(1 + 2\mu)\left|e_j^n\right| &= \left|\mu e_{j-1}^{n+1} + \mu e_{j+1}^{n+1} + e_j^n - \Delta t T_j^{n+1}\right| \\
&\leq \mu\left|e_{j-1}^{n+1}\right| + \mu\left|e_{j+1}^{n+1}\right| + \left|e_j^n\right| + \Delta t\left|T_j^{n+1}\right| \leq \mu E^{n+1} + \mu E^{n+1} + E^n + \Delta t\overline{T}
\end{aligned}$$

Take the maximum of the LHS, we obtain a recursive relation

$$(1 + 2\mu)E^{n+1} \leq 2\mu E^{n+1} + E^n + \Delta t\overline{T} \Rightarrow E^{n+1} \leq E^n + \Delta t\overline{T}$$

Since the IC is given, i.e., $E^0 = 0$, hence

$$E^n \leq n\Delta t\overline{T} < t_F\overline{T} \approx t_F O(\Delta t)$$

Thus, the fully implicit scheme is of the first order accuracy.

At the end of this section, we provide comments on the difference equations and the linear system obtained using the fully implicit method. Recall the numerical scheme is $-\mu U_{j-1}^{n+1} + (1 + 2\mu)U_j^{n+1} - \mu U_{j+1}^{n+1} = U_j^n$ with BCs $U_0^{n+1} = U_J^{n+1} = 0$ embedded. Then, the coefficient matrix of the implicit scheme has the following properties, tri-diagonal and diagonally dominant. Such a linear system can be solved directly with Gaussian or Thomas algorithms, or in iterative ways with Jacobi or Gauss-Seidel algorithms.

1.4 Weighted Average Scheme

We discuss one sophisticated design of the implicit method, which is referred to as the **WEIGHTED AVERAGE SCHEME** or the **θ-METHOD**. Somehow it is a combination of the explicit scheme and the fully implicit scheme. By doing that, we can include six nodes to approximate the 2^{nd} derivative of x.

$$\text{Fully Implicit Scheme:} \quad \frac{U_j^{n+1}-U_j^n}{\Delta t}=\frac{U_j^{n+1}-2U_j^{n+1}+U_{j-1}^{n+1}}{\Delta x^2}$$

$$\text{Explicit Scheme:} \quad \frac{U_j^{n+1}-U_j^n}{\Delta t}=\frac{U_j^{n}-2U_j^{n}+U_{j-1}^{n}}{\Delta x^2}$$

The linear combination of the explicit scheme and the fully implicit scheme can be expressed as

$$\begin{matrix} U_j^{n+1}-U_j^n=\mu\delta_x^2U_j^{n+1} & \times\theta \\ U_j^{n+1}-U_j^n=\mu\delta_x^2U_j^{n} & \times(1-\theta) \end{matrix}, \quad 0\le\theta\le 1$$

And the θ-method can be obtained

$$U_j^{n+1}-U_j^n=\mu\theta\delta_x^2U_j^{n+1}+\mu(1-\theta)\delta_x^2U_j^n \tag{1.14}$$

1.4.1 Consistency

In the explicit scheme, we use Taylor expansion at (x_j,t_n) to obtain the truncation error T_j^n. In the fully implicit scheme, we use Taylor expansion at (x_j,t_{n+1}) to obtain the truncation error T_j^{n+1}. However, for θ-method, we expand the Taylor series and obtain the truncation error at $(x_j,t_{n+1/2})$, i.e., $T_j^{n+1/2}$.

$$\begin{aligned} T_j^{n+1/2}=&\frac{1}{\Delta t}\left(u(x_j,t_{n+1})-u(x_j,t_n)\right)-\frac{1}{\Delta x^2}\left(\mu\theta\delta_x^2u(x_j,t_{n+1})+\mu(1-\theta)\delta_x^2u(x_j,t_n)\right)\\
=&\left[u_t(x_j,t_{n+1/2})+\frac{1}{24}u_{ttt}(x_j,t_{n+1/2})\Delta t^2+...\right]\\
&-\theta\left[u_{xx}(x_j,t_{n+1})+\frac{1}{12}u_{xxxx}(x_j,t_{n+1})\Delta x^2+\frac{1}{360}u_{xxxxxx}(x_j,t_{n+1})\Delta x^4\right]\\
&-(1-\theta)\left[u_{xx}(x_j,t_{n})+\frac{1}{12}u_{xxxx}(x_j,t_{n})\Delta x^2+\frac{1}{360}u_{xxxxxx}(x_j,t_{n})\Delta x^4\right]\\
=&\left[u_t(x_j,t_{n+1/2})-u_{xx}(x_j,t_{n+1/2})\right]\\
&+\left[\left(\frac{1}{2}-\theta\right)u_{xxt}(x_j,t_{n+1/2})\Delta t-\frac{1}{12}u_{xxxx}(x_j,t_{n+1/2})\Delta x^2\right]\\
&+\left[\frac{1}{24}u_{ttt}(x_j,t_{n+1/2})\Delta t^2-\frac{1}{8}u_{xxtt}(x_j,t_{n+1/2})\Delta t^2\right]\\
&+\left[\frac{1}{12}\left(\frac{1}{2}-\theta\right)u_{xxxxt}(x_j,t_{n+1/2})\Delta t\Delta x^2-\frac{1}{360}u_{xxxxxx}(x_j,t_{n+1/2})\Delta x^4\right]+...\end{aligned}$$

In general, $T_j^{n+1/2}=O(\Delta t+\Delta x^2)$. A special case occurs when $\theta=1/2$, and use the fact $u_{ttt}=u_{ttxx}=u_{xxtt}$ if exists,

$$T_j^{n+1/2}=-\frac{1}{12}\left[u_{ttt}(x_j,t_{n+1/2})\Delta t^2+u_{xxxx}\Delta x^2\right]\sim O(\Delta t^2+\Delta x^2)$$

That results in the **CRANK-NICOLSON (C-N) SCHEME**, and it is of the second order accuracy in the time domain. Thus, we can choose larger time steps with the same accuracy.

1.4.2 Stability

For linear problems, we can still apply the von Neumann (Fourier) stability analysis. Recall $k = m\pi$, and $\forall k$, we hope the corresponding mode in the numerical solution can be bounded at any time step n, i.e., based on the θ-method,

$$\begin{aligned}\lambda(k)^{n+1}e^{ikj\Delta x} - \lambda(k)^n e^{ikj\Delta x} = \mu\theta\left[\lambda(k)^{n+1}e^{ik(j+1)\Delta x} - 2\lambda(k)^{n+1}e^{ikj\Delta x} + \lambda(k)^{n+1}e^{ik(j-1)\Delta x}\right]\\ +\mu(1-\theta)\left[\lambda(k)^n e^{ik(j+1)\Delta x} - 2\lambda(k)^n e^{ikj\Delta x} + \lambda(k)^n e^{ik(j-1)\Delta x}\right]\end{aligned}$$

The amplification factor can be solved as

$$\lambda(k) = \lambda(m\pi) = \frac{1 - 4\mu(1-\theta)\sin^2\frac{1}{2}k\Delta x}{1 + 4\mu\theta\sin^2\frac{1}{2}k\Delta x} \tag{1.15}$$

For stability, we need $|\lambda| \leq 1$. If $\lambda > 0$, we observe that the denominator, hence the nominator in Eq.1.15, should be > 0, which implies $\lambda < 1$ automatically. If $\lambda < 0$, we need to show that $\lambda \geq -1$, the good thing is that the denominator is still positive, i.e.

$$\frac{1 - 4\mu(1-\theta)\sin^2\frac{1}{2}k\Delta x}{1 + 4\mu\theta\sin^2\frac{1}{2}k\Delta x} \geq -1 \Rightarrow 2(1-2\theta)\mu\sin^2\frac{1}{2}k\Delta x \leq 1$$

Suppose $\sin^2\frac{1}{2}k\Delta x$ always achieves its maximum value. If $1 - 2\theta \leq 0$, then the inequality can be satisfied. If $1 - 2\theta \geq 0$, a conservative condition will be $(1-2\theta)\mu \leq 1/2$, s.t. the stability criterion can be satisfied for $\forall k$. To summary, the necessary condition for the stability of θ-method is $(1-2\theta)\mu \leq 1/2$ (This is correct even for $1 - 2\theta \leq 0$).

Remark *(Special Cases of the θ-method)*

1. *$\theta = 0$, the θ-method reduces to explicit scheme.*
2. *$\theta = 1$, the θ-method reduces to fully explicit scheme.*
3. *$\theta \in [1/2, 1)$, the θ-method has more implicit component, hence it is unconditionally stable.*
4. *$\theta \in (0, 1/2]$, the θ-method has more explicit component. It is conditionally stable with $\mu \leq 1/[2(1-2\theta)]$.*

1.4.3 Convergence

Analogizing to the error estimations mentioned in the explicit and implicit schemes, define $e_j^n = U_j^n - u(x_j, t_n)$. Write the θ-method as

$$(1+2\theta\mu)U_j^{n+1} = \theta\mu(U_{j+1}^{n+1} + U_{j-1}^{n+1}) + (1-\theta)\mu(U_{j+1}^n + U_{j-1}^n) + [1 - 2\mu(1-\theta)]U_j^n$$

Suppose the exact solution is smooth enough, and the truncation error is bounded

$$\begin{aligned}(1+2\theta\mu)u(x_j, t_{n+1}) =&\theta\mu\left[u(x_{j+1}, t_{n+1}) + u(x_{j-1}, t_{n+1})\right]\\ &+ (1-\theta)\mu\left[u(x_{j+1}, t_n) + u(x_{j-1}, t_n)\right] + [1 - 2\mu(1-\theta)]u(x_j, t_n) + \Delta t T_j^{n+1/2}\end{aligned}$$

Take the difference between the two equations

$$(1+2\theta\mu)e_j^{n+1} = \theta\mu(e_{j+1}^{n+1} + e_{j-1}^{n+1}) + (1-\theta)\mu(e_{j+1}^n + e_{j-1}^n) + [1 - 2\mu(1-\theta)]e_j^n - \Delta t T_j^{n+1/2}$$

Since $0 \leq \theta \leq 1$, we have $1-\theta \geq 0$. Assume $1-2\mu(1-\theta) \geq 0$, i.e., $\mu(1-\theta) \leq 1/2$. Let $E^n = \max_{0\leq j\leq J}\left|e_j^n\right|$. Then

$$\begin{aligned}(1+2\theta\mu)|e_j^{n+1}| \leq& 2\mu\theta E^{n+1} + 2\mu(1-\theta)E^n + [1 - 2\mu(1-\theta)]E^n + \Delta t\left|T_j^{n+1/2}\right|\\ =& 2\mu\theta E^{n+1} + E^n + \Delta t\left|T_j^{n+1/2}\right|\end{aligned}$$

Thus, we obtain

$$(1+2\theta\mu)E_j^{n+1} = 2\mu\theta E^{n+1} + E^n + \Delta t\left|T_j^{n+1/2}\right|$$
$$E^{n+1} \le E^n + \Delta t\left|T_j^{n+1/2}\right|$$

From the IC, we naturally have $E^0 = 0$ and suppose the truncation error is also bounded $\overline{T} = \max\left|T_j^{n+1/2}\right| < \infty$. Thus

$$E^n \le n\Delta t\left|T_j^{n+1/2}\right| \le t_F\overline{T}$$

For general θ-methods, $\overline{T} \sim O(\Delta t + \Delta x^2)$. For the C-N scheme, $\overline{T} \sim O(\Delta t^2 + \Delta x^2)$. Note that the condition $\mu(1-\theta) \le 1/2$ used in this error estimation procedure is more restrict than the condition $(1-2\theta)\mu \le 1/2$ in the stability analysis, since $\theta \ge 0$ and $(1-2\theta)\mu \le \mu(1-\theta)$.

Besides consistency, convergence and stability, there is another theoretical issue that need to be considered. This is a characteristics come from the original PDE. E.g., in the model problem $u_t = u_{xx}$, the solution u should be bounded by the extreme values in ICs and Dirichlet BCs. In a simple sense, the solution u should be a smoother for the the extreme IC and BC values. That characteristics is referred to as the **MAXIMUM PRINCIPLE**.

Theorem 1.2 (Maximum Principle of θ-Method)

The θ-scheme for the model problem $u_t = u_{xx}$ with $0 \le \theta \le 1$ and $\mu(1-\theta) \le 1/2$ yields the numerical solution U_j^n satisfying

$$U_{\min} \le U_j^n \le U_{\max},\ 0 \le j \le J$$
$$\text{where } U_{\min} \overset{def}{=\!=} \min\left\{U_0^m, 0 \le m \le n, U_j^m, 0 \le m \le n, U_j^0, 0 \le j \le J\right\}$$
$$U_{\max} \overset{def}{=\!=} \max\left\{U_0^m, 0 \le m \le n, U_j^m, 0 \le m \le n, U_j^0, 0 \le j \le J\right\}$$

Thus, the maximum and the minimum values are accessed at the boundaries. ♡

Proof Start with the θ-scheme $U_j^{n+1} - U_j^n = \mu\left\{\theta\delta_x^2 U_j^{n+1} + (1-\theta)\delta_x^2 U_j^n\right\}$, and we rewrite this θ-scheme as

$$(1+2\theta\mu)U_j^{n+1} = \theta\mu(U_{j+1}^{n+1} + U_{j-1}^{n+1}) + (1-\theta)\mu(U_{j+1}^n + U_{j-1}^n) + [1-2\mu(1-\theta)]U_j^n$$

Suppose the maximum can be accessed at U_j^{n+1}, an internal point of the computational domain. We can obtain,

$$U_j^{n+1} \ge \max\left\{U_{j-1}^{n+1}, U_{j+1}^{n+1}, U_{j-1}^n, U_{j+1}^n, U_j^n\right\}$$

However, use the fact that U_j^{n+1} is an internal maximum in the θ-scheme, we would have

$$(1+2\theta\mu)U_j^{n+1} \ge \theta\mu(U_{j+1}^{n+1} + U_{j-1}^{n+1}) + (1-\theta)\mu(U_{j+1}^n + U_{j-1}^n) + [1-2\mu(1-\theta)]U_j^n$$

Note that "=" holds iff all of $U_{j-1}^{n+1}, U_{j+1}^{n+1}, U_{j-1}^n, U_{j+1}^n, U_j^n$ are all the maxima. That implies the solution will be constant. That indicates the maximum may still be transmitted to the boundaries. Similar procedure can be done for the minimum case. ■

Remark

1. *For Fourier analysis, $\mu(1-2\theta) \le 1/2$ is an iff condition for stability, which can be usually obtained when the problem has constant coefficients.*
2. *For the maximum principle, $\mu(1-\theta) \le 1/2$ is an sufficient condition, which can be accessed when the coefficients are not constant.*

1.5 Maximum Principle and Non-homogeneous Problems

The **MAXIMUM PRINCIPLE** is a requirements from the original PDE problem, rather than from the numerical schemes. E.g., for $|u_x| = 1$ with $u(0) = 0$, the solutions can be $u = x$ or $u = -x$, which are not unique. But in many real physical applications, there may be only one reasonable solution, such as the viscosity solution $u = x$. Although the maximum principle is not from the numerical schemes, it can be a useful tool for stability analysis. We will present two examples, one is for non-homogeneous BCs and the other one is for non-homogeneous PDEs.

NON-HOMOGENEOUS BOUNDARY CONDITION

Consider the following problem

$$\begin{cases} \text{PDE:} & u_t(x,t) = u_x x(x,t), & (x,t) \in [0,1] \times [0,t_F] \\ \text{BC:} & u_x(x,t) = \alpha(t)u(x,t) + g(t), & x = 0 \\ \text{BC:} & u_(1,t) = 0, & x = 1 \\ \text{IC:} & u(x,t) = u_0(x), & t = 0 \end{cases}$$

At $x = 0$, we have

$$\frac{U_1^n - U_0^n}{\Delta x} = \alpha^n U_0^n + g^n$$
$$\Rightarrow \quad U_0^n = \beta^n U_1^n - \beta^n g^n \Delta x,\ \beta^n = \frac{1}{1 + \alpha^n \Delta x}$$

For $j = 1, 2, ..., J-1$, use the θ-scheme, as $U_J^n = 0$. Combining it with the first equation, i.e., the non-homogeneous BC, we obtain,

$$\begin{aligned} \text{Eq:} \quad & -\theta\mu U_{j-1}^{n+1} + (1 + 2\theta\mu)U_j^{n+1} - \theta\mu U_{j+1}^{n+1} = [1 + (1-\theta)\mu\delta_x^2]U_j^n \\ \text{BC:} \quad & U_0^{n+1} - \beta^{n+1}U_1^{n+1} = -\beta^{n+1}g^{n+1}\Delta x \end{aligned}$$

The Fourier stability analysis cannot be processed in this case, since without homogeneous BCs, the solution cannot be simply written as a sum of harmonics. Thus, for this example with non-homogeneous and time-dependent BCs, we have to propose the analysis with the maximum principle.

We just need to analyze the difference equation when $j = 1$, since that is the node directly interacts with the non-homogeneous BC. For the boundary node at $j = 0$, we do not need to discuss its stability since it does not include time evolution, i.e., there is no time derivative in such a BC.

Use the θ-scheme at $j = 1$, the central difference scheme can be reformulated based on the BC, i.e.,

$$U_1^{n+1} - U_1^n = \mu\left\{\theta\delta_x^2 U_1^{n+1} + (1-\theta)\delta_x^2 U_1^n\right\}$$

where

$$\begin{aligned} \delta_x^2 U_1^n &= U_2^n - 2U_1^n + U_0^n = U_2^n - 2U_1^n + (\beta^n U_1^n - \beta^n g^n \Delta x) \\ &= U_2^n - (2 - \beta^n)U_1^n - \beta^n g^n \Delta x \\ \delta_x^2 U_1^{n+1} &= U_2^{n+1} - 2U_1^{n+1} + U_0^{n+1} = U_2^{n+1} - 2U_1^{n+1} + \left(\beta^{n+1}U_1^{n+1} - \beta^{n+1}g^{n+1}\Delta x\right) \\ &= U_2^{n+1} - (2 - \beta^{n+1})U_1^{n+1} - \beta^{n+1}g^{n+1}\Delta x \end{aligned}$$

$$\begin{aligned} \Rightarrow \quad U_1^{n+1} - U_1^n &= \mu(1-\theta)\left\{U_2^n - (2-\beta^n)U_1^n - \beta^n g^n \Delta x\right\} \\ &\quad + \mu\theta\left\{U_2^{n+1} - (2-\beta^{n+1})U_1^{n+1} - \beta^{n+1}g^{n+1}\Delta x\right\} \end{aligned}$$

Let $e_j^n = U_j^n - u(x_j, t_n)$ be the error between the numerical solution and the exact solution. We have

$$e_1^{n+1} - e_1^n = \mu\left\{\theta\left[e_2^{n+1} - (2-\beta^{n+1})e_1^{n+1}\right] + (1-\theta)\left[e_2^n - (2-\beta^n)e_1^n\right]\right\} - \Delta t T_1^{n+1/2}$$

Rearranging this equation yields

$$\left[1+\theta\mu\left(2-\beta^{n+1}\right)\right]e_1^{n+1} = \left[1-(1-\theta)\mu(2-\beta^n)\right]e_1^n + \theta\mu e_2^{n+1} + (1-\theta)\mu e_2^n - \Delta t T_1^{n+1/2} \tag{1.16}$$

From the error equation above, to use the maximum principle, we should make the following assumptions,

1. All the coefficient should be non-negative, s.t., we can use the triangular inequality.
2. The sum of the coefficients on the RHS should be $\leq$ the sum of the coefficients on the LHS, s.t., we can derive a contraction.

In Eq.1.16, for the first assumption, we make some rough estimation, which will be sufficient but not necessary to fulfill the first assumption, and doable in practical applications.

1. LHS: It is sufficient to have $2-\beta^{n+1} \geq 0$, i.e., $\alpha^{n+1}\Delta \geq -1/2$.
2. RHS: We need $1-(1-\theta)\mu(2-\beta^n) \geq 0$, and we can set $1-2\mu(1-\theta) \geq 0$, i.e., $\mu(1-\theta) \leq 1/2$.

Remark *$2-\beta^{n+1} \geq 0$ is a magic in this analysis. Suppose it is $1-\beta^{n+1} \geq 0$, then we have no choice but let $\alpha^{n+1}\Delta x \geq 0$, which implies $\alpha(t) \geq 0$. That may not be true for some PDE. However, $\alpha^{n+1}\Delta \geq -1/2$ can always be achieved under correctly choice of Δx.*

In Eq.1.16, for the second assumption,

1. LHS: the sum of coefficients is $1+\mu\theta(1-\beta^{n+1})+\mu\theta$.
2. RHS: the sum of coefficients is $1-\mu(1-\theta)(1-\beta^n)+\mu\theta$.

Therefore, what we need is $-\mu(1-\theta)(1-\beta^n) \leq \mu\theta(1-\beta^{n+1})$

Remark *Unfortunately, after some trials, we have to restrict $\beta^n \leq 1$, i.e., $\alpha^n \geq 0$ to make the numerical scheme valid based on the second requirement, which yields*

$$-\mu(1-\theta)(1-\beta^n) \leq 0 \leq \mu\theta(1-\beta^{n+1})$$

This observation should also remind us what the physics represented by $\alpha(t) \geq 0$ and $\alpha(t) \leq 0$. E.g., let $\alpha(t) = \alpha \geq 0$ be a constant number, as $u(0,t)$ increases, the derivative u_x will increase, while the flux is in the opposite direction (the negative direction) should also increase if exists. That could be considered as a "negative feedback", which is stable. Otherwise, if $\alpha(t) < 0$, the system will be of "positive feedback", where at the boundary $x = 0$, the larger the u, more weight will be added to u. By making this observation, we should say the maximum principle also has its limitations.

Nevertheless, based on the two assumptions, we have

$$\begin{aligned}
(1+\theta\mu)|e_1^{n+1}| &\leq [1+\theta\mu(2-\beta^{n+1})]|e_1^{n+1}| \\
&\leq [1-(1-\theta)\mu(2-\beta^n)]E^n + \theta\mu E^{n+1} + (1-\theta)\mu E^n + \Delta|T_1^{n+1/2}| \\
&= [1-(1-\theta)\mu(1-\beta^n)]E^n + \theta\mu E^{n+1} + \Delta|T_1^{n+1/2}| \\
&\leq E^n + \theta\mu E^{n+1} + \Delta|T_1^{n+1/2}|
\end{aligned}$$

In the derivation, implicitly define $E^n = \max_{1\leq j\leq J}|e_j^n|$, and we also make one more step reformulation, i.e.

$$(1+2\theta\mu)|e_1^{n+1}| \leq E^n + 2\theta\mu E^{n+1} + \Delta|T_1^{n+1/2}|$$

Recall that in the convergence analysis of the θ-scheme, for the interior nodes $j = 2, 3, ..., J-1$, we have

$$(1 + 2\theta\mu)E_j^{n+1} = 2\mu\theta E^{n+1} + E^n + \Delta t \left|T_j^{n+1/2}\right|$$

Thus, for $j = 1, 2, 3, ..., J-1$, we combine the two inequalities above and obtain

$$\begin{aligned}
(1 + 2\theta\mu)|e_j^{n+1}| \leq & E^n + 2\mu\theta E^{n+1} + \Delta t \left|T_j^{n+1/2}\right| \\
\leq & E^n + 2\mu\theta E^{n+1} + \Delta t \max_{1\leq j\leq J-1} \left|T_j^{n+1/2}\right| \\
\leq & E^n + 2\theta\mu E^{n+1} + \Delta t\overline{T} \\
\Rightarrow \quad (1 + 2\theta\mu)E^{n+1} \leq & E^n + 2\theta\mu E^{n+1} + \Delta t\overline{T}
\end{aligned}$$

Thus, the following recursive relationship can be obtained,

$$E^{n+1} \leq E^n + \Delta t\overline{T}$$

Remark *(Truncation Errors of General BCs)*

We should measure the truncation error of the discretized equation of general BCs. Suppose the BC can be expressed as

$$\frac{U_1^n - U_0^n}{\Delta x} = \alpha U_0^n + g^n$$

Use Taylor expansion at (x_0, t_n),

$$R_0^n = \frac{u(x_1, t_n) - u(x_0, t_n)}{\Delta x} - \alpha^n u(x_0, t_n) - g^n = \left[\frac{1}{2}u_{xx}(x_0, t_n)\Delta x + \frac{1}{6}u_{xxx}(x_0, t_n)\Delta x^2 + ...\right]$$

$$\Rightarrow \quad u(x_0, t_n) = \beta^n u(x_1, t_n) - \beta^n g^n \Delta x - \beta^n R_0^n \Delta x$$

In FDM, We use U_0^n *from the BC in the first difference equation, which yields,*

$$\begin{aligned}
\frac{1}{\Delta t}(U_1^{n+1} - U_1^n) = & \frac{1}{\Delta x^2}\theta\left[U_2^{n+1} - (2 - \beta^{n+1})U_1^{n+1} - \beta^{n+1}g^{n+1}\Delta x\right] \\
& + \frac{1}{\Delta x^2}(1 - \theta)\left[U_2^n - (2 - \beta^n)U_1^n - \beta^n g^n \Delta x\right]
\end{aligned}$$

WLOG, let $\theta = 0$*, i.e., let us focus on the explicit scheme*

$$\frac{1}{\Delta t}(U_1^{n+1} - U_1^n) = \frac{1}{\Delta x^2}\left[U_2^n - (2 - \beta^n)U_1^n - \beta^n g^n \Delta x\right]$$

The truncation error can be represented as

$$\begin{aligned}
T_1^{n+1/2} = & \frac{u(x_1, t_{n+1}) - u(x_1, t_n)}{\Delta t} - \frac{\delta_x^2 u(x_1, t_n)}{\Delta x^2} - \frac{\beta^n R_0^n \Delta x}{\Delta x^2} \\
= & \left[\frac{1}{2}u_{tt}(x_1, t_{n+1/2})\Delta t - \frac{1}{12}u_{xxxx}(x_1, t_n)\Delta x^2 + ...\right] - \beta^2\left[\frac{1}{2}u_{xx}(x_0, t_n) + ...\right]
\end{aligned}$$

We have a bad observation, that $T_1^{n+1/2} \approx -\frac{1}{2}\beta^n u_{xx}(x_0, t_n)$*. Therefore, the truncation is of the order* $O(1)$*. In order to solve this issue, we need* R^n *is of the second order accuracy, i.e.,* $R^n \sim O(\Delta x^2)$*. Let*

$$\frac{U_1^n - U_0^n}{\Delta x} = \frac{1}{2}\alpha(U_0^n + U_1^n) + g^n$$

Then, the truncation error will be

$$\begin{aligned}
R_{1/2}^n = & \frac{u(x_1, t_n) - u(x_0, t_n)}{\Delta x} - \frac{1}{2}\alpha^n\left[u(x_0, t_n) + u(x_1, t_n)\right] - g^n \\
= & \frac{1}{24}u_{xxx}(x_{1/2}, t_n)\Delta x^2 - \frac{1}{8}\alpha_{1/2}^n u_{xx}(x_{1/2}, t_n)\Delta x^2 + ...
\end{aligned}$$

Then, $T_1^{n+1/2} \sim O(\Delta x)$*. Similar procedure can be performed for implicit scheme and general* θ*-schemes.*

GENERAL PARTIAL DIFFERENTIAL EQUATION

Consider a model problem

$$u_t = b(x,t)u_{xx}$$

E.g., Using the explicit scheme where b_j^n is a constant,

$$\frac{U_j^{n+1}-U_j^n}{\Delta t} = b_j^n \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{\Delta x^2}$$
$$\Rightarrow \quad U_j^{n+1} = \mu b_j^n U_{j+1}^n + (1-2\mu b_j^n)U_j^n + \mu b_j^n U_{j-1}^n$$

Let T_j^n be the consistency error, and $e_j^n = U_j^n - u(x_j,t_n)$ be the convergence error (solution error), we have

$$e_j^{n+1} = \mu b_j^n e_{j+1}^n + (1-2\mu b_j^n)e_j^n + \mu b_j^n e_{j-1}^n - \Delta t T_j^n$$

To adopt the maximum principle, we need $1-2\mu b_j^n \geq 0$, i.e., $\mu b_j^n \leq 1/2$.

We can also use the θ-scheme. Usually, we will need $b^* = b_j^{n+1/2}$. Then the θ-scheme can be expressed as,

$$\frac{U_j^{n+1}-U_j^n}{\Delta t} = \frac{b^*}{\Delta x^2}\left\{\theta\delta_x^2 U_j^{n+1} + (1-\theta)\delta_x^2 U_j^n\right\}$$

The truncation error can be represented as

$$T_j^{n+1/2} = \left(\frac{1}{2}-\theta\right)u_{xxt}\Delta t - \frac{b^*}{12}u_{xxxx}\Delta x^2 + \frac{1}{24}u_{ttt}\Delta t^2 - \frac{b^*}{8}u_{xxtt}\Delta t^2 + ...$$

The convergence error (solution error) can be represented as

$$\frac{e_j^{n+1}-e_j^n}{\Delta t} = \frac{b^*}{\Delta x^2}\left\{\theta\delta_x^2 e_j^{n+1} + (1-\theta)\delta_x^2 e_j^n\right\} - T_j^{n+1/2}$$
$$\Rightarrow \quad (1+2\theta\mu b^*)e_j^{n+1} = \theta\mu b^* e_{j+1}^{n+1} + \theta\mu b^* e_{j-1}^{n+1} + (1-\theta)\mu b^* e_{j+1}^n$$
$$+ [1-2\mu(1-\theta)b^*]e_j^n + (1-\theta)\mu b^* e_{j-1}^n - \Delta t T_j^{n+1/2}$$

Following the two requirements when using the maximum principle, firstly, all of the coefficients should be positive, s.t. the triangular inequality can be used,

$$1-2\mu(1-\theta)b^* \geq 0 \Rightarrow \mu(1-\theta)b^* \leq 1/2$$

Secondly, the sum of coefficients in the RHS should be $\leq$ the sum of coefficients in the LHS, which is automatically satisfied, s.t., a contraction can be formulated. Usually, we assume

$$b^* = b_j^{n+1/2} = \frac{b_j^n + b_j^{n+1}}{2}$$

For the general case

$$u_t = b(x,t)u_{xx} + a(x,t)u_x + c(x,t)u + d(x,t)$$

Assume $b(x,t) > 0$ and use central difference in the convection term $a(x,t)u_x$,

$$\frac{U_j^{n+1}-U_j^n}{\Delta t} = b_j^n \frac{\delta_x^2 U_j^n}{\Delta x^2} - a_j^n \frac{U_{j+1}^n - U_{j-1}^n}{2\Delta x} + c_j^n U_j^n + d_j^n$$

In the case when the sign of $a(x,t)$ changes, we may revise the central difference and apply an **UPWIND DIFFERENCE**,

1. $a_j^n > 0$, the information comes from Node $j-1$, and we should use U_{j-1}^n and U_j^n to construct the difference scheme, i.e., $(U_j^n - U_{j-1}^n)/\Delta x$.
2. $a_j^n < 0$, the information comes from Node $j+1$, and we should use U_j^n and U_{j+1}^n to construct the difference scheme, i.e., $(U_{j+1}^n - U_j^n)/\Delta x$.

1.6 2D Parabolic Problems

Consider the following linear 2D problem with Dirichlet BCs and ICs, where b is a constant

$$\begin{aligned} u_t &= b(u_{xx}+u_{yy}), && G\times[0,t_F]=[0,X]\times[0,Y]\times[0,t_F] \\ u|_{\partial G} &= f(x,y), && \text{Dirichlet BC} \\ u|_{t=0} &= u_0(x,y), && \text{IC} \end{aligned} \tag{1.17}$$

To employ FDM, suppose the temporal and spatial domains can be discretized as

$$\{(x_i,y_j,t_n)=(i\Delta x, j\Delta y, n\Delta t),\ i=1,2,...,I,\ j=1,2,...,J,\ n=0,1,...,N\}$$

The explicit scheme can be written as

$$\frac{U_{i,j}^{n+1}-U_{i,j}^{n}}{\Delta t}=b\left[\frac{\delta_x^2 U_{i,j}^n}{\Delta x^2}+\frac{\delta_y^2 U_{i,j}^n}{\Delta y^2}\right]$$

To check the theoretical issues of the explicit scheme, we first define the truncation error and provide consistency analysis.

$$T_{i,j}^n=\frac{1}{2}u_{tt}(x_i,y_j,t_n)\Delta t-\frac{b}{12}\left[u_{xxxx}\Delta x^2+u_{yyyy}\Delta y^2\right]_{i,j}^n\sim O(\Delta t+\Delta x^2+\Delta y^2)$$

Thus, the explicit scheme is of the first order accuracy in time and second order accuracy in space.

For the stability analysis, since the basic solution to the PDE is $e^{-[k_x^2+k_y^2]bt}e^{\tau(k_x i\Delta x+k_y j\Delta y)}$ where τ is the imaginary unit $\sqrt{-1}$, and $k=(k_x,k_y)=(m_x\pi,m_y\pi)$. To adopt the Fourier stability analysis, suppose the Fourier mode contained in the numerical solution is $U_{i,j}^n=\lambda^n e^{\tau(k_x i\Delta x+k_y j\Delta y)}$, and use the Fourier mode in the explicit scheme

$$\begin{aligned} &\lambda^{n+1}e^{\tau(k_x i\Delta x+k_y j\Delta y)}-\lambda^n e^{\tau(k_x i\Delta x+k_y j\Delta y)} \\ =&\frac{b\lambda^n\Delta t}{\Delta x^2}\left[e^{\tau[k_x(i-1)\Delta x+k_y j\Delta y]}-2e^{\tau[k_x i\Delta x+k_y j\Delta y]}+e^{\tau[k_x(i+1)\Delta x+k_y j\Delta y]}\right] \\ +&\frac{b\lambda^n\Delta t}{\Delta y^2}\left[e^{\tau[k_x i\Delta x+k_y(j-1)\Delta y]}-2e^{\tau[k_x i\Delta x+k_y j\Delta y]}+e^{\tau[k_x i\Delta x+k_y(j+1)\Delta y]}\right] \end{aligned}$$

Let $\mu_x=b\Delta t/\Delta x^2, \mu_y=b\Delta t/\Delta y^2$, we can solve for

$$\lambda(k)=1-4\left[\mu_x\sin^2\frac{1}{2}k_x\Delta x+\mu_y\sin^2\frac{1}{2}k_y\Delta y\right]$$

To ensure $|\lambda(k)|\le 1$, we need $\mu_x+\mu_y\le 1/2$, i.e., the explicit scheme is conditionally stable,

For the convergence analysis, we employ the Fourier analysis and separate the low frequency bound and high frequency bound. For the Fourier analysis, recall we need the Fourier expansion of the IC should be absolutely convergent.

$$\begin{aligned} e_{i,j}^n &= U_{i,j}^n-u(x_i,y_j,t_n) \\ &= \sum_{m_x,m_y=-\infty}^{\infty}A_{m_x,m_y}e^{\tau(m_x\pi i\Delta x+m_y\pi j\Delta y)}\left[|\lambda(m_x\pi,m_y\pi)|^n-e^{-[(m_x\pi)^2+(m_y\pi)^2]bn\Delta t}\right] \end{aligned}$$

With the absolute convergence condition, $\forall\varepsilon>0$, $\exists M>0$, s.t., $\sum_{|m|>M}|A_m|\le\varepsilon$ Then, since λ^n and $e^{-[(m_x\pi)^2+(m_y\pi)^2]b\Delta t}$ are all decreasing functions w.r.t. n, m_x, m_y, s.t. the high frequency portion can be bounded by $\varepsilon/2$. Thus,

$$|e_{i,j}^n|\le\frac{1}{2}\varepsilon+\sum_{|m|\le M}|A_m|\left||\lambda(m_x\pi,m_y\pi)|^n-e^{-[(m_x\pi)^2+(m_y\pi)^2]b\Delta t}\right|$$

For the low frequency portion

$$\begin{aligned}\lambda(m_x\pi, m_y\pi) &= 1 - 4\left[\mu_x \sin^2\frac{1}{2}m_x\pi\Delta x + \mu_y \sin^2\frac{1}{2}m_y\pi\Delta y\right] \\ &= 1 - [(m_x\pi)^2 + (m_y\pi)^2]b\Delta t + \frac{1}{12}[(m_x\pi)^4\Delta x^2 + (m_y\pi)^4\Delta y^2]b\Delta t \\ e^{-[(m_x\pi)^2+(m_y\pi)^2]b\Delta t} &= 1 - [(m_x\pi)^2 + (m_y\pi)^2]b\Delta t + \frac{1}{2}[(m_x\pi)^2 + (m_y\pi)^2]^2b^2\Delta t^2\end{aligned}$$

Comparing the expansion form, we have

$$\begin{aligned}|e^n_{i,j}| &\le \frac{1}{2}\varepsilon + \sum_{|m|\le M}|A_m|\left||\lambda(m_x\pi, m_y\pi)|^n - e^{-[(m_x\pi)^2+(m_y\pi)^2]bn\Delta t}\right| \\ &\le \frac{1}{2}\varepsilon + n\sum_{|m|\le M}|A_m|\left||\lambda(m_x\pi, m_y\pi)| - e^{-[(m_x\pi)^2+(m_y\pi)^2]b\Delta t}\right| \\ &\le \frac{1}{2}\varepsilon + n\sum_{|m|\le M}|A_m|C(\mu_x,\mu_y)[(m_x\pi)^2 + (m_y\pi)^2]^2b^2\Delta t^2 \\ &\le \frac{1}{2}\varepsilon + t_F C(\mu_x,\mu_y)b^2\left[\sum_{|m|\le M}|A_m|[(m_x\pi)^2 + (m_y\pi)^2]^2\right]\Delta t\end{aligned}$$

Therefore, by choosing Δt small enough, we can obtain $|e^n_{i,j}| \le \varepsilon$.

We can apply error analysis to provide another version of the convergence analysis, which will require the (sufficient) smoothness of the exact solution. Let $e^n_{i,j} = U^n_{i,j} - u(x_i, y_j, t_n)$, and the difference equation for the error can be shown as

$$\begin{aligned}e^{n+1}_{i,j} &= e^n_{i,j} + \left[\mu_x\delta_x^2 e^n_{i,j} + \mu_y\delta_y^2 e^n_{i,j}\right] - \Delta t T^n_{i,j} \\ &= [1 - 2(\mu_x + \mu_y)]e^n_{i,j} + \mu_x e^n_{i-1,j} + \mu_x e^n_{i+1,j} + \mu_y e^n_{i,j-1} + \mu_y e^n_{i,j+1} - \Delta t T^n_{i,j}\end{aligned}$$

We require $1 - 2(\mu_x + \mu_y) \ge 0$, and let $|\overline{T}| = \max_{i,j}|T^n_{i,j}|$ and $E^n = \max_{i,j}|e^n_{i,j}|$. Thus, we can have

$$\begin{aligned}& E^{n+1} \le E^n + \Delta t|\overline{T}| \\ \Rightarrow\quad & E^n \le n\Delta t|\overline{T}| \le t_F|\overline{T}| \le t_F\left[\frac{1}{2}\Delta t M_t t - \frac{b}{12}[\Delta x^2 M_{xxxx} + \Delta y^2 M_{yyyy}]\right] + \ldots \\ & \sim O(\Delta t + \Delta x^2 + \Delta y^2)\end{aligned}$$

1.6.1 Alternative Direction Implicit Scheme

We first present the θ-scheme for the motivation of the **ALTERNATIVE DIRECTION IMPLICIT SCHEME (ADI)**. Suppose the general θ-scheme and the C-N scheme ($\theta = 1/2$) can be written as

$$\begin{aligned}&\theta\text{-scheme:} \quad \frac{U^{n+1}_{i,j} - U^n_{i,j}}{\Delta t} = b\left[\frac{(1-\theta)\delta_x^2U^n_{i,j} + \theta\delta_x^2U^{n+1}_{i,j}}{\Delta x^2} + \frac{(1-\theta)\delta_y^2U^n_{i,j} + \theta\delta_y^2U^{n+1}_{i,j}}{\Delta y^2}\right] \\ &\text{C-N scheme:} \quad \left[1 - \frac{1}{2}\mu_x\delta_x^2 - \frac{1}{2}\mu_y\delta_y^2\right]U^{n+1}_{i,j} = \left[1 + \frac{1}{2}\mu_x\delta_x^2 + \frac{1}{2}\mu_y\delta_y^2\right]U^n_{i,j}\end{aligned} \tag{1.18}$$

Therefore, the accuracy of the general θ-scheme is $T^{n+1/2}_{i,j} \sim O(\Delta t + \Delta x^2 + \Delta y^2)$, while the accuracy of the C-N scheme is $T^{n+1/2}_{i,j} \sim O(\Delta t^2 + \Delta x^2 + \Delta y^2)$ Thus, we need to solve the following equation system to obtain $U^{n+1} = \{U^{n+1}_{i,j}\}_{i=0,1,2,\ldots,I;j=0,1,2,\ldots J}$, i.e., $AU^{n+1} = b^n$.

In this equation, $b = \{\left[1 + \frac{1}{2}\mu_x\delta_x^2 + \frac{1}{2}\mu_y\delta_y^2\right]U^n_{i,j}\}_{i=0,1,2,\ldots,I;j=0,1,2,\ldots J}$. The size of A is $[(I+1)\times(J+1)]^2$. For each row of A, there are 5 non-zero entries. Thus, although A is diagonally dominant, A is not tri-diagonal.

Thus, the Thomas method cannot be applied.

In order to convert the problem into a tri-diagonal linear system, we propose the ADI scheme. Consider we split one time step into two half time steps. In the first half step, the implicit scheme only occurs in the x-axis, while in the second half step, the implicit scheme only occurs in the y-axis. I.e., we introduce an intermediate step $U_{i,j}^{n+1/2}$ and yield

$$\begin{cases} \dfrac{U_{i,j}^{n+1/2}-U_{i,j}^{n}}{\Delta t/2}=\dfrac{\delta_x^2 U_{i,j}^{n+1/2}}{\Delta x^2}+\dfrac{\delta_y^2 U_{i,j}^{n}}{\Delta y^2} & \Rightarrow \quad \left(1-\dfrac{1}{2}\mu_x\delta_x^2\right)U_{i,j}^{n+1/2}=\left(1+\dfrac{1}{2}\mu_y\delta_y^2\right)U_{i,j}^{n} \\ \dfrac{U_{i,j}^{n+1}-U_{i,j}^{n+1/2}}{\Delta t/2}=\dfrac{\delta_x^2 U_{i,j}^{n+1/2}}{\Delta x^2}+\dfrac{\delta_y^2 U_{i,j}^{n+1}}{\Delta y^2} & \Rightarrow \quad \left(1-\dfrac{1}{2}\mu_y\delta_y^2\right)U_{i,j}^{n+1}=\left(1+\dfrac{1}{2}\mu_x\delta_x^2\right)U_{i,j}^{n+1/2} \end{cases}$$

Motivated by the C-N scheme in Eq.1.18, we rewrite the ADI scheme as

$$\left(1-\frac{1}{2}\mu_x\delta_x^2\right)\left(1-\frac{1}{2}\mu_y\delta_y^2\right)U_{i,j}^{n+1}=\left(1+\frac{1}{2}\mu_x\delta_x^2\right)\left(1+\frac{1}{2}\mu_y\delta_y^2\right)U_{i,j}^{n} \tag{1.19}$$

That can be derived using the two-half-step form mention above, i.e., apply $\left(1+\frac{1}{2}\mu_x\delta_x^2\right)$ to the first equation, apply $\left(1-\frac{1}{2}\mu_x\delta_x^2\right)$, and add the two equations. Note that $\left(1+\frac{1}{2}\mu_x\delta_x^2\right)$ and $\left(1-\frac{1}{2}\mu_x\delta_x^2\right)$ are commutative since they only contains differences in x-axis. Hence, some terms can be cancelled. However, that may not apply when differences in x-axis and y-axis occur together.

Suppose $T_{i,j}^{n+1/2}$ be the truncation error of the C-N scheme, then the truncation error of the ADI scheme, $T_{i,j}^{*,n+1/2}$, can be written as follows, where the accuracy of the ADI scheme is not reduced.

$$\Delta t T_{i,j}^{*,n+1/2}=\frac{1}{4}\mu_x\mu_y\delta_x^2\delta_y^2[u(x_i,y_j,t_{n+1})-u(x_i,y_j,t_n)]+\Delta t T_{i,j}^{n+1/2}\sim\Delta t O(\Delta t^2+\Delta x^2+\Delta y^2)$$

To solve the linear system induced from the ADI scheme, use intermediate step $U_{i,j}^{n+1/2}$, where

$$\begin{cases} \left(1-\dfrac{1}{2}\mu_x\delta_x^2\right)U_{i,j}^{n+1/2}=\left(1+\dfrac{1}{2}\mu_y\delta_y^2\right)U_{i,j}^{n}, & i=0,1,...,I \\ \left(1-\dfrac{1}{2}\mu_y\delta_y^2\right)U_{i,j}^{n+1}=\left(1+\dfrac{1}{2}\mu_x\delta_x^2\right)U_{i,j}^{n+1/2}, & j=0,1,...,J \end{cases}$$

1. In the first equation, fix j and we can solve $U_{i,j}^{n+1/2}$ for all i , where the linear system $AU_{i,j}^{n+1/2}=b_j^n$ has tri-diagonal coefficient matrix A of order $(I+1)^2$ and
$$b_j^n=\left[\left(1+\frac{1}{2}\mu_y\delta_y^2\right)U_{0,j}^{n},\left(1+\frac{1}{2}\mu_y\delta_y^2\right)U_{1,j}^{n},...,\left(1+\frac{1}{2}\mu_y\delta_y^2\right)U_{I,j}^{n}\right]^T$$
2. In the second equation, fix i and we can solve $U_{i,j}^{n}$ for all j , where the linear system $AU_{i,j}^{n}=b_i^{n+1/2}$ has tri-diagonal coefficient matrix A of order $(J+1)^2$ and
$$b_i^{n+1/2}=\left[\left(1+\frac{1}{2}\mu_x\delta_x^2\right)U_{i,0}^{n+1/2},\left(1+\frac{1}{2}\mu_x\delta_x^2\right)U_{i,1}^{n+1/2},...,\left(1+\frac{1}{2}\mu_x\delta_x^2\right)U_{i,J}^{n+1/2}\right]^T$$

We finish this section by providing the Fourier stability analysis and the maximum principle analysis.

FOURIER STABILITY ANALYSIS/VON NEUMANN TECHNIQUE

Start with the ADI scheme in Eq.1.19, and assume the Fourier mode as $U_{i,j}^{n}=\lambda^n e^{\tau(m_x\pi i\Delta x+m_y\pi j\Delta y)}$. Using the Fourier mode into the ADI scheme yields

$$\lambda(m_x,m_y)=\frac{\left(1-\frac{1}{2}\mu_x\sin^2\frac{1}{2}m_x\pi\Delta x\right)\left(1-\frac{1}{2}\mu_y\sin^2\frac{1}{2}m_y\pi\Delta y\right)}{\left(1+\frac{1}{2}\mu_x\sin^2\frac{1}{2}m_x\pi\Delta x\right)\left(1+\frac{1}{2}\mu_y\sin^2\frac{1}{2}m_y\pi\Delta y\right)}$$

Thus, $|\lambda| \leq 1$, and the ADI scheme is unconditionally stable.

MAXIMUM PRINCIPLE

Start with the first equation

$$\left(1-\frac{1}{2}\mu_x\delta_x^2\right)U_{i,j}^{n+1/2}=\left(1+\frac{1}{2}\mu_y\delta_y^2\right)U_{i,j}^{n}$$

$$\Rightarrow \quad (1+\mu_x)U_{i,j}^{n+1/2}=(1-\mu_y)U_{i,j}^{n}+\frac{1}{2}\mu_y(U_{i,j-1}^{n}+U_{i,j+1}^{n})+\frac{1}{2}\mu_x(U_{i+1,j}^{n+1/2}+U_{i-1,j}^{n+1/2})$$

Based on the requirement of the maximum principle, we have $\mu_y \leq 1$.

Similarly, for the second equation

$$\left(1-\frac{1}{2}\mu_y\delta_y^2\right)U_{i,j}^{n+1}=\left(1+\frac{1}{2}\mu_x\delta_x^2\right)U_{i,j}^{n+1/2}$$

$$\Rightarrow \quad (1+\mu_y)U_{i,j}^{n+1}=(1-\mu_x)U_{i,j}^{n+1/2}+\frac{1}{2}\mu_x(U_{i-1,j}^{n+1/2}+U_{i+1,j}^{n+1/2})+\frac{1}{2}\mu_y(U_{i,j+1}^{n+1}+U_{i,j-1}^{n+1})$$

Based on the requirement of the maximum principle, we have $\mu_x \leq 1$.

The proof shows that the scheme should be stable unconditionally. However, if we require the solution satisfying the maximum principle, it is necessary to set an additional condition, i.e., $\max\{\mu_x, \mu_y\} \leq 1$.

CONVERGENCE ANALYSIS

First give the expression for the truncation error

$$\begin{aligned}
LHS=&\left(1-\frac{1}{2}\mu_x\delta_x^2\right)\left(1-\frac{1}{2}\mu_y\delta_y^2\right)u(x_i,y_j,t_{n+1})\\
=&u(x_i,y_j,t_{n+1})-\frac{1}{2}\mu_x\left[u(x_{i+1},y_j,t_{n+1})+u(x_{i-1},y_j,t_{n+1})-2u(x_i,y_j,t_{n+1})\right]\\
&-\frac{1}{2}\mu_y\left[u(x_i,y_{j+1},t_{n+1})+u(x_i,y_{j-1},t_{n+1})-2u(x_i,y_j,t_{n+1})\right]\\
&+\frac{1}{4}\mu_x\mu_y\left[u(x_{i+1},y_{j+1},t_{n+1})+u(x_{i-1},y_{j+1},t_{n+1})+u(x_{i+1},y_{j-1},t_{n+1})+u(x_{i-1},y_{j-1},t_{n+1})\right]\\
&+\frac{1}{4}\mu_x\mu_y\left[-2u(x_{i+1},y_j,t_{n+1})-2u(x_{i-1},y_j,t_{n+1})-2u(x_i,y_{j+1},t_{n+1})-2u(x_i,y_{j-1},t_{n+1})\right]\\
&+\frac{1}{4}\mu_x\mu_y\left[4u(x_i,y_j,t_{n+1})\right]\\
RHS=&\left(1+\frac{1}{2}\mu_x\delta_x^2\right)\left(1+\frac{1}{2}\mu_y\delta_y^2\right)u(x_i,y_j,t_n)\\
=&u(x_i,y_j,t_n)+\frac{1}{2}\mu_x\left[u(x_{i+1},y_j,t_n)+u(x_{i-1},y_j,t_n)-2u(x_i,y_j,t_n)\right]\\
&+\frac{1}{2}\mu_y\left[u(x_i,y_{j+1},t_n)+u(x_i,y_{j-1},t_n)-2u(x_i,y_j,t_n)\right]\\
&+\frac{1}{4}\mu_x\mu_y\left[u(x_{i+1},y_{j+1},t_n)+u(x_{i-1},y_{j+1},t_n)+u(x_{i+1},y_{j-1},t_n)+u(x_{i-1},y_{j-1},t_n)\right]\\
&+\frac{1}{4}\mu_x\mu_y\left[-2u(x_{i+1},y_j,t_n)-2u(x_{i-1},y_j,t_n)-2u(x_i,y_{j+1},t_n)-2u(x_i,y_{j-1},t_n)\right]\\
&+\frac{1}{4}\mu_x\mu_y\left[4u(x_i,y_j,t_n)\right]
\end{aligned}$$

The truncation error can be expressed as follows. We temporally use $u(\natural) = u(x_i, y_j, t_{n+1/2})$ to indicate that

Taylor expansion is performed at $\natural = (x_i, y_j, t_{n+1/2})$. Recall $\mu_x = b\Delta t/\Delta x^2$, $\mu_y = b\Delta t/\Delta y^2$,

$$
\begin{aligned}
&T_{i,j}^{n+1/2} = \frac{LHS - RHS}{\Delta t} = \frac{1}{24}u_{ttt}(\natural)\Delta t^2 - \frac{b}{8}u_{xxtt}(\natural)\Delta t^2 - \frac{b}{8}u_{yytt}(\natural)\Delta t^2 \\
&-\frac{b}{12}u_{xxxx}(\natural)\Delta x^2 - \frac{b}{96}u_{xxxxtt}(\natural)\Delta x^2\Delta t^2 - \frac{b}{12}u_{yyyy}(\natural)\Delta y^2 - \frac{b}{96}u_{yyyytt}(\natural)\Delta y^2\Delta t^2 \\
&+\frac{b^2}{4}u_{xxyyt}(\natural)\Delta t^2 + \frac{b^2}{48}u_{xxxxyyt}(\natural)\Delta x^2\Delta t^2 + \frac{b^2}{48}u_{xxyyyyt}(\natural)\Delta y^2\Delta t^2 + \frac{b^2}{576}u_{xxxxyyyyt}(\natural)\Delta x^2\Delta y^2\Delta t^2 + ... \\
&\sim O(\Delta t^2 + \Delta x^2 + \Delta y^2)
\end{aligned}
$$

Second let $e_{i,j}^n = U_{i,j}^n - u(x_i, y_j, t_n)$ be the error, and we have the following error expression using Eq.1.19

$$\left(1 - \frac{1}{2}\mu_x\delta_x^2\right)\left(1 - \frac{1}{2}\mu_y\delta_y^2\right)e_{i,j}^{n+1} = \left(1 + \frac{1}{2}\mu_x\delta_x^2\right)\left(1 + \frac{1}{2}\mu_y\delta_y^2\right)e_{i,j}^{n} - \Delta t T_{i,j}^{n+1/2}$$

To expand this expression, we have

$$
\begin{aligned}
&LHS = \left(1 - \frac{1}{2}\mu_x\delta_x^2\right)\left(1 - \frac{1}{2}\mu_y\delta_y^2\right)e_{i,j}^{n+1} \\
=&e_{i,j}^{n+1} - \frac{1}{2}\mu_x\left[e_{i+1,j}^{n+1} + e_{i-1,j}^{n+1} - 2e_{i,j}^{n+1}\right] - \frac{1}{2}\mu_y\left[e_{i,j+1}^{n+1} + e_{i,j-1}^{n+1} - 2e_{i,j}^{n+1}\right] \\
+&\frac{1}{4}\mu_x\mu_y\left[e_{i+1,j+1}^{n+1} + e_{i-1,j+1}^{n+1} + e_{i+1,j-1}^{n+1} + e_{i-1,j-1}^{n+1} - 2e_{i+1,j}^{n+1} - 2e_{i-1,j}^{n+1} - 2e_{i,j+1}^{n+1} - 2e_{i,j-1}^{n+1} + 4e_{i,j}^{n+1}\right] \\
&RHS = \left(1 + \frac{1}{2}\mu_x\delta_x^2\right)\left(1 + \frac{1}{2}\mu_y\delta_y^2\right)e_{i,j}^{n} - \Delta t T_{i,j}^{n+1/2} \\
=&-\Delta t T_{i,j}^{n+1/2} + e_{i,j}^{n} + \frac{1}{2}\mu_x\left[e_{i+1,j}^{n} + e_{i-1,j}^{n} - 2e_{i,j}^{n}\right] + \frac{1}{2}\mu_y\left[e_{i,j+1}^{n} + e_{i,j-1}^{n} - 2e_{i,j}^{n}\right] \\
+&\frac{1}{4}\mu_x\mu_y\left[e_{i+1,j+1}^{n} + e_{i-1,j+1}^{n} + e_{i+1,j-1}^{n} + e_{i-1,j-1}^{n} - 2e_{i+1,j}^{n} - 2e_{i-1,j}^{n} - 2e_{i,j+1}^{n} - 2e_{i,j-1}^{n} + 4e_{i,j}^{n}\right]
\end{aligned}
$$

We can isolate the terms $\frac{1}{4}\mu_x\mu_y\delta_x^2\delta_y^2(e_{i,j}^{n+1} - e_{i,j}^{n})$, since it is of the same order as the truncation error, and then the error equation is reduced to the one in the C-N scheme. I.e., letting $E^n = \max_{i,j}|e_{i,j}^n|$ and $\overline{T} = \max_{i,j}|T_{i,j}^{n+1/2}|$, we rearrange the expression and have

$$
\begin{aligned}
(1 + \mu_x + \mu_y)e_{i,j}^{n+1} = &(1 - \mu_x - \mu_y)e_{i,j}^{n} + \frac{1}{2}\mu_x\left[e_{i+1,j}^{n+1} + e_{i-1,j}^{n+1}\right] + \frac{1}{2}\mu_y\left[e_{i,j+1}^{n+1} + e_{i,j-1}^{n+1}\right] \\
&+ \frac{1}{2}\mu_x\left[e_{i+1,j}^{n} + e_{i-1,j}^{n}\right] + \frac{1}{2}\mu_y\left[e_{i,j+1}^{n} + e_{i,j-1}^{n}\right] - \Delta t T_{i,j}^{n+1/2} \\
\Rightarrow\quad &(1 + \mu_x + \mu_y)E^{n+1} \le (1 - \mu_x - \mu_y)E^n + \mu_x(E^{n+1} + E^n) + \mu_y(E^{n+1} + E^n) + \Delta\overline{T} \\
\Rightarrow\quad &E^{n+1} \le E^n + \Delta\overline{T}
\end{aligned}
$$

In the derivation, we implicitly require that $1 - \mu_x - \mu_y \ge 0$, s.t. the triangular inequality can be applied. The convergence analysis for the ADI scheme is heuristic, and a strict proof of the convergence of the ADI scheme is not trivial.

Remark *(3D version of the ADI scheme)*
The 3D parabolic equation can be written as

$$u_t = b(u_{xx} + u_{yy} + u_{zz})$$

A popular method in solving this problem is known as the **LOCALLY ONE-DIMENSION SCHEME (LOD)**.

In order to reduce the complexity, we rewrite the problem into three 1D problem, i.e.

$$\left(1-\frac{1}{2}\mu_x\delta_x^2\right)U_{i,j,k}^{n+*}=\left(1+\frac{1}{2}\mu_x\delta_x^2\right)U_{i,j,k}^{n}$$
$$\left(1-\frac{1}{2}\mu_y\delta_y^2\right)U_{i,j,k}^{n+**}=\left(1+\frac{1}{2}\mu_y\delta_y^2\right)U_{i,j,k}^{n+*}$$
$$\left(1-\frac{1}{2}\mu_z\delta_z^2\right)U_{i,j,k}^{n+1}=\left(1+\frac{1}{2}\mu_z\delta_z^2\right)U_{i,j,k}^{n+**}$$

1.7 Exercises

✍ **Exercise 1.1** Consider the BVP with periodic BCs

$$u_t(x,t)=u_{xx}(x,t),\ (x,t)\in[0,1]\times[0,2]$$
$$u(x,0)=\begin{cases}2x, & x\in[0,0.5]\\ 2-2x, & x\in[0.5,1]\end{cases}$$
$$u(0,t)=u(1,t)=0$$

1. Derive the analytical solution by the separation of variables.
2. Solve the PDE with the explicit scheme, prove the consistency, and derive the condition for stability. Compare the numerical results at $t=0,0.02,0.04,0.06,0.08,0.1$ with $\Delta=0.05,\Delta t=0.001$ and $\Delta=0.05,\Delta t=0.002$. Comment on the results.
3. Solve the PDE with the C-N scheme, prove the consistency, derive the condition for stability, and derive the condition for the maximum principle. Compare the numerical results at $t=0,0.02,0.04,0.06,0.08,0.1$ with $\Delta=0.05,\Delta t=0.001$ and $\Delta=0.05,\Delta t=0.01$. Comment on the results.

Solve

1. *Suppose $u(x,t)=\phi(x)\psi(t)$, then the PDE becomes $\phi(x)\psi'(t)=\phi''(x)\psi(t)$,*

$$\frac{\psi'(t)}{\psi(t)}=\frac{\phi''(x)}{\phi(x)}=-k^2,\ k\geq 0$$

For the spatial part,

$$\begin{aligned}&\phi''(x)+k^2\phi(x)=0\\ &u(0,t)=u(1,t)=0\end{aligned}\Rightarrow\phi(x)=C_1\sin kx=C_1\sin(m\pi x)$$

For the temporal part,

$$\psi'(t)+k^2\psi(t)=0\Rightarrow\psi(t)=C_2e^{-k^2t}=C_2e^{-(m\pi)^2t}$$

Therefore, the general solution to the original PDE can be written as the sum w.r.t. m,

$$u(x,t)=\sum_{m=0}^{\infty}C_me^{-(m\pi)^2t}\sin(m\pi x),\ C_m=C_1C_2,\forall m=0,1,2,...$$

Match the general solution to the sine expansion of the IC, we have

$$u(x,t)=\sum_{m=0}^{\infty}\frac{8}{(m\pi)^2}\sin\frac{m\pi}{2}e^{-(m\pi)^2t}\sin(m\pi x),\ \forall(x,t)\in[0,1]\times[0,2]$$

2. *In the consistency analysis, substituting the exact solution to the explicit scheme, and using Taylor expansion at (x_j,t_n),*

$$\begin{aligned}T_j^n&=\frac{1}{\Delta t}[u(x_j,t_{n+1})-u(x_j,t_n)]-\frac{1}{\Delta x^2}[u(x_{j+1},t_n)-2u(x_j,t_n)+u(x_{j-1},t_n)]\\ &=\frac{1}{2}u_{tt}(x_j,t_n)\Delta t-\frac{1}{12}u_{xxxx}(x_j,t_n)\Delta x^2+...\\ &\sim O(\Delta t+\Delta x^2)\to 0,\qquad \text{as } \Delta t,\Delta x\to 0\end{aligned}$$

In the stability analysis, suppose the Fourier mode of the numerical solution is $U_j^n=\lambda^n e^{im\pi j\Delta x}$, and $\mu=\Delta t/\Delta x^2$. Use the

Fourier mode into the explicit scheme

$$\lambda^{n+1}e^{im\pi j\Delta x}-\lambda^n e^{im\pi j\Delta x}=\mu\left[\lambda^n e^{im\pi(j-1)\Delta x}-2\lambda^n e^{im\pi j\Delta x}+\lambda^n e^{im\pi(j+1)\Delta x}\right]$$

Since $\lambda\neq 0$, we can solve

$$\lambda=1-4\mu\sin^2\frac{m\pi}{2}\Delta x$$

The stability criterion is $|\lambda|\leq 1$, which implies $0\leq\mu\sin^2\frac{m\pi}{2}\Delta x\leq 1/2$. Thus, $0\leq\mu\leq 1/2$ will be sufficient to ensure the stability of the explicit scheme.

For the numerical solution, we just provide a hint that when $\Delta=0.05,\Delta t=0.001$, the numerical solution is stable, when $\Delta=0.05,\Delta t=0.002$ the numerical solution is unstable.

3. *In the consistency analysis, substituting the exact solution to the C-N scheme, and using Taylor expansion at $(x_j,t_{n+1/2})$, we can obtain the following results after some tedious calculation*

$$\begin{aligned}T_j^{n+1/2}=&\frac{1}{\Delta t}[u(x_j,t_{n+1})-u(x_j,t_n)]-\frac{1}{2\Delta x^2}[u(x_{j+1},t_n)-2u(x_j,t_n)+u(x_{j-1},t_n)]\\&-\frac{1}{2\Delta x^2}[u(x_{j+1},t_{n+1})-2u(x_j,t_{n+1})+u(x_{j-1},t_{n+1})]\\=&-\frac{1}{12}u_{xxxx}(x_j,t_{n+1/2})\Delta x^2+\left[\frac{1}{24}u_{ttt}(x_j,t_{n+1/2})-\frac{1}{8}u_{xxtt}(x_j,t_{n+1/2})\right]\Delta t^2\\&-\frac{1}{96}u_{xxxxtt}(x_j,t_{n+1/2})\Delta x^2\Delta t^2-\frac{1}{360}u_{xxxxxx}(x_j,t_{n+1/2})\Delta x^4+\dots\\\sim&O(\Delta t^2+\Delta x^2)\to 0,\qquad \text{as } \Delta t,\Delta x\to 0\end{aligned}$$

In the stability analysis, use the Fourier mode into the C-N scheme

$$\begin{aligned}\lambda^{n+1}e^{im\pi j\Delta x}-\lambda^n e^{im\pi j\Delta x}=&\frac{1}{2}\mu\left[\lambda^{n+1}e^{im\pi(j-1)\Delta x}-2\lambda^{n+1}e^{im\pi j\Delta x}+\lambda^{n+1}e^{im\pi(j+1)\Delta x}\right]\\&+\frac{1}{2}\mu\left[\lambda^n e^{im\pi(j-1)\Delta x}-2\lambda^n e^{im\pi j\Delta x}+\lambda^n e^{im\pi(j+1)\Delta x}\right]\end{aligned}$$

Since $\mu\neq 0$, we can solve for λ and observe that the C-N scheme is unconditionally stable.

$$-1\leq\lambda=\frac{1-2\mu\sin^2(m\pi\Delta x/2)}{1+2\mu\sin^2(m\pi\Delta x/2)}\leq 1$$

In the maximum principle, let $e_j^n=U_j^n-u(x_j,t_n)$ be the error, and the numerical scheme becomes

$$\begin{aligned}&e_j^{n+1}-e_j^n=\frac{\mu}{2}\left[\left(e_{j+1}^{n+1}-2e_j^{n+1}+e_{j-1}^{n+1}\right)+\left(e_{j+1}^n-2e_j^n+e_{j-1}^n\right)\right]-\Delta tT_j^{n+1/2}\\\Rightarrow\quad&(1+\mu)e_j^{n+1}=\frac{\mu}{2}e_{j+1}^{n+1}+\frac{\mu}{2}e_{j-1}^{n+1}+\frac{\mu}{2}e_{j+1}^n+(1-\mu)e_j^n+\frac{\mu}{2}e_{j-1}^n-\Delta tT_j^{n+1/2}\end{aligned}$$

Recall the first requirement of using the maximum principle is that all of the coefficients should be ≥ 0. Hence $\mu\leq 1$. The second requirement is that the sum of coefficients in the LHS should be $\geq$ the sum of coefficients in the RHS, which is already satisfied. Thus, the requirement of using the maximum principle is $\mu\leq 1$.

For the numerical solution, we just provide a hint. Both time and spatial steps satisfy the stability criterion, s.t. a stable numerical solution can be provided.

However, we claim that $\Delta=0.05,\Delta t=0.01$ do not satisfy the requirement for the maximum principle. Thus, during the iteration, the numerical solution at interior points may access maximum or minimum values. Such effect may not be clearly shown in this example, but we can observe that the corresponding numerical example is not smooth, especially near the middle point. If we enlarge the time step to $\Delta t=0.01$, we can observe the interior points, rather than the boundary points, access the maximum or minimum values

▲

Appendix. The MATLAB code

```
% ------------ Math 517 Chap.1 Exercise 1 (Explicit Scheme) --------
clear all; close all; clc
```

```
u0=@(x) (x>=0&x<=0.5).*2.*x+(x>0.5&x<=1).*(2-2.*x);
% --------------- dx=0.05; dt=0.001 -------------------
dx=0.05; dt=0.001; X=0:dx:1; T=0:dt:2; LX=length(X); LT=length(T);
[Xaxis,Taxis]=meshgrid(X,T); U1=zeros(LT,LX); mu=dt./dx./dx; U1(1,:)=u0(X);
for t=2:LT
    U1(t,1)=0; U1(t,LX)=0;
    for j=2:LX-1
       U1(t,j)=U1(t-1,j)+mu*(U1(t-1,j-1)-2*U1(t-1,j)+U1(t-1,j+1)); end end
figure(1), mesh(Taxis,Xaxis,U1), xlabel('T'), ylabel('X'), zlabel('U')
% --------------- dx=0.05; dt=0.002 -------------------
dx=0.05; dt=0.002; X=0:dx:1; T=0:dt:2; LX=length(X); LT=length(T);
[Xaxis,Taxis]=meshgrid(X,T); U2=zeros(LT,LX); mu=dt./dx./dx; U2(1,:)=u0(X);
for t=2:LT
    U2(t,1)=0; U2(t,LX)=0;
    for j=2:LX-1
       U2(t,j)=U2(t-1,j)+mu*(U2(t-1,j-1)-2*U2(t-1,j)+U2(t-1,j+1)); end end
figure(2), mesh(Taxis,Xaxis,U2), xlabel('T'), ylabel('X'), zlabel('U')
% ------ for t=0, 0.02, 0.04, 0.06, 0.08, 0.10 ------
figure(3), plot(X,U1(1,:),X,U2(1,:)),xlabel('T'), ylabel('X'),legend('stable','unstable')
figure(4), plot(X,U1(21,:),X,U2(11,:)),xlabel('T'), ylabel('X'),legend('stable','unstable')
figure(5), plot(X,U1(41,:),X,U2(21,:)),xlabel('T'), ylabel('X'),legend('stable','unstable')
figure(6), plot(X,U1(61,:),X,U2(31,:)),xlabel('T'), ylabel('X'),legend('stable','unstable')
figure(7), plot(X,U1(81,:),X,U2(41,:)),xlabel('T'), ylabel('X'),legend('stable','unstable')
figure(8), plot(X,U1(101,:),X,U2(51,:)),xlabel('T'), ylabel('X'),legend('stable','unstable')

% ------------ Math 517 Chap.1 Exercise 1 (C-N Scheme) --------
clear all; close all; clc
u0=@(x) (x>=0&x<=0.5).*2.*x+(x>0.5&x<=1).*(2-2.*x);
% --------------- dx=0.05; dt=0.001 -------------------
dx=0.05; dt=0.001; X=0:dx:1; T=0:dt:2; LX=length(X); LT=length(T);
[Xaxis,Taxis]=meshgrid(X,T); U1=zeros(LT,LX); mu=dt./dx./dx; U1(1,:)=u0(X);
A=-0.5.*mu.*ones(LX-2,1); B=(1+mu).*ones(LX-2,1); C=-0.5.*mu.*ones(LX-2,1);
A(1)=0; C(LX-2)=0;
for t=2:LT
    D=zeros(LX-2,1);
    for j=1:LX-2
       D(j)=0.5.*mu.*U1(t-1,j)+(1-mu).*U1(t-1,j+1)+0.5.*mu.*U1(t-1,j+2); end
    % ---- Thomas algorithm -----------
    R=zeros(LX-2,1); Y=zeros(LX-2,1); R(1)=C(1)/B(1); Y(1)=D(1)/B(1);
    for j=2:LX-2
       R(j)=C(j)/(B(j)-R(j-1)*A(j));
       Y(j)=(D(j)-Y(j-1)*A(j))/(B(j)-R(j-1)*A(j)); end
    U1(t,LX-1)=Y(LX-2);
    for j=LX-3:-1:1
       U1(t,j+1)=Y(j)-R(j)*U1(t,j+2); end
    U1(t,1)=0;U1(t,LX)=0; end
figure(1), mesh(Taxis,Xaxis,U1), xlabel('T'), ylabel('X'), zlabel('U')
% --------------- dx=0.05; dt=0.01 -------------------
```

```
dx=0.05; dt=0.01; X=0:dx:1; T=0:dt:2; LX=length(X); LT=length(T);
[Xaxis,Taxis]=meshgrid(X,T); U2=zeros(LT,LX); mu=dt./dx./dx; U2(1,:)=u0(X);
A=-0.5.*mu.*ones(LX-2,1); B=(1+mu).*ones(LX-2,1); C=-0.5.*mu.*ones(LX-2,1);
A(1)=0; C(LX-2)=0;
for t=2:LT
    D=zeros(LX-2,1);
    for j=1:LX-2
        D(j)=0.5.*mu.*U2(t-1,j)+(1-mu).*U2(t-1,j+1)+0.5.*mu.*U2(t-1,j+2); end
    % ---- Thomas algorithm -----------
    R=zeros(LX-2,1); Y=zeros(LX-2,1); R(1)=C(1)/B(1); Y(1)=D(1)/B(1);
    for j=2:LX-2
        R(j)=C(j)/(B(j)-R(j-1)*A(j));
        Y(j)=(D(j)-Y(j-1)*A(j))/(B(j)-R(j-1)*A(j)); end
    U2(t,LX-1)=Y(LX-2);
    for j=LX-3:-1:1
        U2(t,j+1)=Y(j)-R(j)*U2(t,j+2); end
    U2(t,1)=0;U2(t,LX)=0; end
figure(2), mesh(Taxis,Xaxis,U2), xlabel('T'), ylabel('X'), zlabel('U')
% ------ for t=0, 0.02, 0.04, 0.06, 0.08, 0.10 ------
figure(3), plot(X,U1(1,:),X,U2(1,:)),xlabel('T'), ylabel('X'),legend('Satisfying Maximum
    Principle','Not Satisfying Maximum Principle')
figure(4), plot(X,U1(21,:),X,U2(3,:)),xlabel('T'), ylabel('X'),legend('Satisfying Maximum
    Principle','Not Satisfying Maximum Principle')
figure(5), plot(X,U1(41,:),X,U2(5,:)),xlabel('T'), ylabel('X'),legend('Satisfying Maximum
    Principle','Not Satisfying Maximum Principle')
figure(6), plot(X,U1(61,:),X,U2(7,:)),xlabel('T'), ylabel('X'),legend('Satisfying Maximum
    Principle','Not Satisfying Maximum Principle')
figure(7), plot(X,U1(81,:),X,U2(9,:)),xlabel('T'), ylabel('X'),legend('Satisfying Maximum
    Principle','Not Satisfying Maximum Principle')
figure(8), plot(X,U1(101,:),X,U2(11,:)),xlabel('T'), ylabel('X'),legend('Satisfying Maximum
    Principle','Not Satisfying Maximum Principle')
```

Chapter 2 Hyperbolic Equations

2.1 Simple Linear Hyperbolic Equations and Euler Schemes

Suppose the hyperbolic equation can be written as $F(Du, u, x) = 0$. In general, we cannot expect a classical solution, i.e., a globally smooth solution, due to the nature of a wave, which may include a "spike" which is not differentiable. However, in practice, we define **WEAK SOLUTIONS**, e.g., viscosity solutions or entropy solutions (vanishing viscosity solutions). For a general hyperbolic equation, we can still use the method of characteristics to solve the problem locally, and obtain the locally smooth solution. E.g., let $z(s) \triangleq u(\vec{x}(s))$, $\vec{p(s)} \triangleq Du(\vec{x}(s))$ be a local definition of a general hyperbolic equation, its behavior along the characteristic curve $\{\vec{x}(s)\}$ can be presented as follows. D is a partial derivation operator, $\vec{x}$ is the vector of independent variables, s is the parameter along the characteristic curve. A diagram of general characteristic curve is shown in Fig.2.1

$$\begin{cases} \dot{\vec{x}}(s) = D_{\vec{p}}F\left(\vec{p}(s), u(s), \vec{x}(s)\right) \\ \dot{z}(s) = D_{\vec{p}}F\left(\vec{p}(s), u(s), \vec{x}(s)\right) \cdot \vec{p}(s) \\ \dot{\vec{p}}(s) = -D_{\vec{x}}F - (D_u F)\vec{p}(s) \end{cases} \tag{2.1}$$

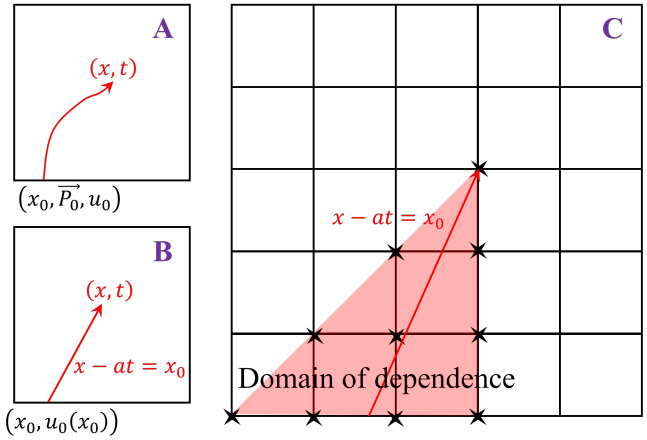

Figure 2.1: Diagrams of a general characteristics (A), a linear characteristics (B), and a domain of dependence under a given mesh grid (C).

Example 2.1 (Local Linearization of a Hyperbolic Equation)
Consider $F\left(\vec{p}(s), u(s), \vec{x}(s)\right) = F(u_t, u_x, u_y, u, t, x, y)$, s.t. $z(s) = u$, $\vec{p} = [u_t, u_x, u_y]^T$ and $\vec{x} = [t, x, y]^T$. Then, the first equation in Eq.2.1 can represent the geometry of the characteristic curve.

$$\frac{d\vec{x}(s)}{ds} = \left[\frac{dt}{ds}, \frac{dx}{ds}, \frac{dy}{ds}\right]^T = \left[\frac{\partial F}{\partial u_t}, \frac{\partial F}{\partial u_x}, \frac{\partial F}{\partial u_y}\right]^T$$

The second equation in Eq.2.1 can be treated as a local version of the PDE, with the implicit function theorem.

$$\frac{dz(s)}{ds} = \left[\frac{\partial F}{\partial u_t}, \frac{\partial F}{\partial u_x}, \frac{\partial F}{\partial u_y}\right][u_t, u_x, u_y]^T = 0 \Rightarrow \frac{\partial F}{\partial u_t}u_t + \frac{\partial F}{\partial u_x}u_x + \frac{\partial F}{\partial u_y}u_y = 0$$

The third equation in Eq.2.1 can be verified using the total differentiation of F, with the assumption that $D_{\vec{p}}F$

is arbitrary.

$$\begin{aligned}\frac{dF}{ds} &= D_{\vec{p}}F \cdot \dot{\vec{p}}(s) + D_u F \cdot \dot{z}(s) + D_{\vec{x}}F \cdot \dot{\vec{x}}(s) \\ &= D_{\vec{p}}F \cdot \dot{\vec{p}}(s) + D_u F D_{\vec{p}}F \cdot \vec{p} + D_{\vec{x}}F \cdot D_{\vec{p}}F = D_{\vec{p}}F \cdot \left[\dot{\vec{p}}(s) + (D_u)F\vec{p} + D_{\vec{x}}F\right] = 0 \\ \Rightarrow \quad \dot{\vec{p}}(s) &= -D_{\vec{x}}F - (D_u F)\vec{p}\end{aligned}$$

▲

The ODE system Eq.2.1 is satisfied along the characteristic line, and the characteristic line is also know as the **DOMAIN OF DEPENDENCE (DOD)** of an object point on the characteristic line, since all of the information is gained from this characteristic line. The characteristic line can be linear, or a general curve.

In this chapter, we will focus on the following model problem (a transfer equation or an advection equation) with specified BCs and ICs.

$$u_t + a(x,t)u_x = 0 \tag{2.2}$$

The characteristics can be written as

$$\begin{cases} \dfrac{dx}{dt} = a(x,t) \\ \dfrac{du}{dt} = \dfrac{\partial u}{\partial t} + \dfrac{\partial u}{\partial x}\dfrac{dx}{dt} = u_t + a(x,t)u_x = 0 \end{cases}$$

Suppose $a(x,t) = a$ is a constant, then the characteristics becomes a linear function, $x - at = x_0$, and $u(x,t) = u_0(x_0) = u_0(x - at)$, since the u values along the characteristic line is constant. The linear characteristics is shown in Fig.2.1.

2.1.1 Stability and Courant-Friedrich-Lewy Condition

We apply FDM to solve Eq.2.2 when $a(x,t) = a$ is a constant, and the first scheme is the **EXPLICIT EULER SCHEME**. When designing the FDM scheme, we want to use the point close to the characteristics to compute the numerical solution, which is also referred to as the **UPWIND SCHEME** in the spatial domain. For the explicit Euler scheme

1. If $a > 0$, define $\nu = a\Delta t/\Delta x$, we have
$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + a\frac{U_j^n - U_{j-1}^n}{\Delta x} \Rightarrow U_j^{n+1} = U_j^n - \nu\left(U_j^n - U_{j-1}^n\right) = (1-\nu)U_j^n + \nu U_{j-1}^n$$
2. If $a < 0$, define $\nu = a\Delta t/\Delta x$, we have
$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + a\frac{U_{j+1}^n - U_j^n}{\Delta x} \Rightarrow U_j^{n+1} = U_j^n - \nu\left(U_{j+1}^n - U_j^n\right) = (1+\nu)U_j^n - \nu U_{j+1}^n$$

An graphic reason of using the upwind scheme to compute U_j^n is that the DODs for the numerical schemes, e.g., the shaded area demarcated by $U_{j-1}^n, U_j^n, U_j^{n+1}$ in Fig.2.1 when $a > 0$, must cover the DOD for the PDE, i.e., the characteristics.

Theorem 2.1 (Courant-Friedrich-Lewy Condition, CFL Condition)

For a convergent numerical scheme, the DOD of the PDE must be contained in the DOD of the numerical scheme. ♡

Based on the CLF condition, $\nu = a\Delta t/\Delta x$ must satisfy $|\nu| \leq 1$ to ensure the convergence. ν is also known as

the **CFL NUMBER**. Because of convergence, the numerical solution will be stable. Thus, $|\nu| \leq 1$ also serves as a stability condition.

We now verify the numerical stability using the Fourier analysis. WLOG, suppose $a > 0$, and write the Fourier mode as $U_j^n = \lambda^n e^{ikj\Delta x}$. Sending it in the numerical scheme yields

$$\begin{aligned} & \lambda^{n+1} e^{ikj\Delta x} = (1-\nu)\lambda^n e^{ikj\Delta x} + \nu\lambda^n e^{ik(j-1)\Delta x} \\ \Rightarrow \quad & \lambda(k) = (1-\nu) + \nu e^{-ik\Delta x} = (1-\nu) + \nu\cos k\Delta x - i\nu \sin k\Delta x \\ \Rightarrow \quad & |\lambda(k)| = \sqrt{1 - 4\nu(1-\nu)\sin^2\frac{k}{2}\Delta x} \end{aligned}$$

Since we need $|\lambda(k)| < 1$, i.e., $\lambda(k)^2 < 1$, and $0 \leq \nu \leq 1$ can ensure our demand. Similarly, if $a < 0$, we have

$$|\lambda(k)| = \sqrt{1 + 4\nu(1+\nu)\sin^2\frac{k}{2}\Delta x}$$

and that will require $-1 \leq \nu \leq 0$ to ensure $|\lambda(k)| < 1$. Therefore, the results from Fourier stability analysis coincide the CFL condition.

Remark *(Central Difference Scheme is not Stable)*
The central difference scheme for x is not stable

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + a\frac{U_{j+1}^n - U_{j-1}^n}{2\Delta x} = 0 \Rightarrow U_j^{n+1} = U_j^n - \frac{\nu}{2}\left(U_{j+1}^n - U_{j-1}^n\right)$$

Suppose the Fourier mode is $U_j^n = \lambda^n e^{ikj\Delta x}$, and then use it in the central difference scheme, we have

$$\lambda(k) = 1 - \frac{\nu}{2}\left(e^{ik\Delta x} - e^{-ik\Delta x}\right) = 1 - \nu(i\sin k\Delta x)$$

Thus, $|\lambda(k)| \geq 1$, and the scheme is unstable.

Remark *The CFL condition is not an iff condition. For a convergent scheme, it must satisfy the CFL condition, while the reversal may not be satisfied. Thus, the CFL condition is a necessary but not sufficient condition.*

2.1.2 Convergence

Suppose $a > 0$, the truncation error is

$$\begin{aligned} T_j^n &= \frac{1}{\Delta t}[u(x_j, t_{n+1}) - u(x_j, t_n)] - \frac{a}{\Delta x}[u(x_j, t_n) - u(x_{j-1}, t_n)] \\ &= \frac{1}{\Delta t}\left[u_t(x_j,t_n)\Delta t + \frac{1}{2}u_{tt}(x_j,t_n)\Delta t^2 + ...\right] + \frac{a}{\Delta x}\left[u_x(x_j,t_n)\Delta x + \frac{1}{2}u_{xx}(x_j,t_n)\Delta x^2 + ...\right] \\ &\sim O(\Delta x + \Delta t) \to 0 \text{ as } \Delta t, \Delta x \to 0 \end{aligned}$$

Let $e_j^n = U_j^n - u(x_j, t_n)$, then

$$e_j^{n+1} = (1-\nu)e_j^n + \nu e_{j-1}^n - \Delta t T_j^n$$

Due to the CFL condition, we already have $0 \leq \nu \leq 1$, and the triangular inequality can be applied. Assume $E^n = \max_{i,j}|e_{i,j}^n|$ and $\overline{T} = \max_{i,j}|T_{i,j}^n|$.

$$\begin{aligned} & |e_{i,j}^{n+1}| \leq E^n + \Delta t|T_j^n| \leq E^n + \Delta t\overline{T} \\ \Rightarrow \quad & E^{n+1} \leq E^n + \Delta t\overline{T} \\ \Rightarrow \quad & E^{n+1} \leq n\Delta t\overline{T} \leq t_F\overline{T} \sim O(\Delta t + \Delta x) \end{aligned}$$

Thus, we have a good local estimation. However, that may not be a good global estimation. E.g., given a step function as the IC, then in the "step-jump" territory, if the characteristics intersect the mesh grid right at the

mesh nodes, or outside the "step-jump" discontinuous region, the numerical scheme can provide good results. Otherwise, the numerical scheme may not be good enough, i.e., the sharpness of the "step-jump" might be lost.

2.1.3 Dissipation and Dispersion

We describe two types of errors when using FDM to solve hyperbolic PDEs. First, recall what we have learned

1. The upwind scheme can be summarized as
$$U_j^{n+1} = \begin{cases} (1-\nu)U_j^n + \nu U_{j-1}^n & \text{if } a_j^n > 0, \nu = \dfrac{a_j^n \Delta t}{\Delta x} > 0 \\ (1+\nu)U_j^n - \nu U_{j+1}^n & \text{if } a_j^n < 0, \nu = \dfrac{a_j^n \Delta t}{\Delta x} < 0 \end{cases} \tag{2.3}$$
where a_j^n can be called as the velocity of propagation.
2. The CFL number should satisfy $|\nu| \leq 1$. The CFL condition is a necessary condition, that is if the numerical scheme is convergent, then the CFL condition is satisfied.

In the exact solution, as the wave propagates, we can expect that the "shape" of the wave maintains, while the position of the wave changes, particularly for the simple wave propagation in Eq.2.2 with a as a constant. Theoretically, there should be no change of the amplitude. The basic solution to the hyperbolic equation is $u(x,t) = e^{i(kx+\omega t)}, \omega = -ak$. When we go to the next time step $t \to t + \Delta t$, that mode will transfer to $u(x, t+\Delta t) = e^{i[kx+\omega(t+\Delta t)]} = e^{i(kx+\omega t)} e^{i\omega \Delta t}$. From the individual mode in the exact solution, we observe that the amplitude of the wave does not change, and the phase is changed.

However, in numerical solutions, such ideal situations may not occur. Recall the solution mode assumed in Fourier analysis is $U_j^n = \lambda^n e^{ikj\Delta}$, and suppose $a > 0$ and the upwind scheme in Eq.2.3 is used. We obtain
$$\lambda(k) = 1 - \nu\left(1 - e^{-ik\Delta t}\right)$$
In this equation, we can naturally write $\lambda = |\lambda| e^{i\theta}$, where $|\lambda|$ is the amplitude and $\theta = \arg(\lambda)$ is the phase angle of λ. Thus, when time increases from t to $t + \Delta t$, we have
$$\begin{aligned} U_j^n &= \left[|\lambda|^n e^{in\theta}\right] e^{i(kj\Delta t)} = |\lambda|^n e^{i(n\theta + kj\Delta t)} \\ \xrightarrow{t \to t+\Delta t} \quad U_j^{n+1} &= |\lambda|^{n+1} e^{i[(n+1)\theta + kj\Delta t]} = |\lambda|^n e^{i(n\theta+kj\Delta t)} |\lambda| e^{i\theta} = U_j^n |\lambda| e^{i\theta} \end{aligned}$$
We observe $e^{i\theta}$ indicates the change of phase as time progresses. Comparing with the exact solution, we should expect $|\lambda| = 1$ and $\theta = \omega \Delta t$, s.t. the numerical solution can match the exact solution.

Unfortunately, in numerical solutions, errors in both the amplitude and phase occurs, which are referred to as

1. **DAMPING/DISSIPATION**: errors in the changes of amplitude.
2. **DISPERSION**: errors in the changes of phase.

We start with the damping. Consider the numerical scheme when $a > 0$. In the relation of λ, $\lambda(k) = 1 - \nu\left(1 - e^{-ik\Delta t}\right)$, if $\nu = 1$, $\lambda(k) = e^{ik\Delta t}$ and $|\lambda| = 1$. Hence, there is no damping error. However, if $0 < \nu < 1$, $|\lambda| < 1$, then damping will occur.

As for the dispersion, since $\lambda(k) = |\lambda| e^{i\theta} = 1 - \nu\left(1 - e^{-ik\Delta t}\right)$, separate the real part and the imaginary part
$$\begin{cases} \text{Re}: & |\lambda| \cos\theta = (1-\nu) + \nu \cos k\Delta x \\ \text{Im}: & |\lambda| \sin\theta = -\nu \cos k\Delta x \end{cases} \Rightarrow \theta = \arg\lambda = -\arctan\left[\frac{\nu \cos k\Delta x}{(1-\nu) + \nu \cos k\Delta x}\right]$$
Let $\xi = k\Delta x$, where small ξ corresponds to the low frequency portion.

Lemma 2.1 (Expansion of arctan Function)

If q can be expressed as powers of p in the following form, as $p \to 0$,

$$q \sim c_1 p + c_2 p^2 + c_3 p^3 + c_4 p^4 + \dots$$

then

$$\arctan q \sim c_1 p + c_2 p^2 + \left(c_3 - \frac{1}{3}c_1^3\right) p^3 + \left(c_4 - c_1^2 c_2\right) p^4 + \dots$$

♡

Using Lemma 2.1, $\arg\lambda$ can be simplified

$$\begin{aligned}\arg\lambda &= -\arctan\left[\nu\left(\xi - \frac{1}{6}\xi^3 + \dots\right)\left(1 - \frac{1}{2}\nu\xi^2 + \dots\right)^{-1}\right] \\ &= -\arctan\left[\nu\xi - \frac{1}{6}\nu(1-3\nu)\xi^3 + \dots\right] = -\nu\xi\left[1 - \frac{1}{6}(1-\nu)(1-2\nu)\xi^2 + \dots\right]\end{aligned}$$

We can obtain the exact phase change during wave propagation, if

$$\arg\lambda = -\nu\xi = -\frac{a\Delta t}{\Delta x}k\Delta x = -ak\Delta t = \omega t$$

However, we have a long expansion, which can be referred to as the error terms. Using the derivation above, it can be shown that the relatively error in phase is of order ξ^2.

Remark *How bad the damping error can be in the sense of ξ? Recall that $|\lambda|^2 = 1 - 4\nu(1-\nu)\sin^2\frac{1}{2}k\Delta x$. The error of amplitude is of order ξ^2. Hence globally, the magnitude of λ has the error of order ξ.*

We summary the error analyses as follows. For $0 < \nu < 1$, there will be two parts of errors. A diagram for the damping error is shown in Fig.2.2

1. Damping/Dissipation: error in the change of amplitude, relative to $|\lambda|^2$, is of order $O(\xi^2)$.
2. Dispersion: error in the change of phase, relative to $\arg\lambda$, is of order $O(\xi^2)$.

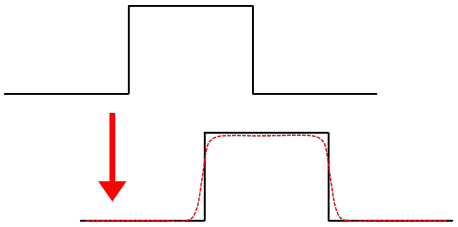

Figure 2.2: The change of amplitude shown with the dash line.

Remark *Notice that we already have the convergence analysis, a natural question is that since the numerical solution converges, where the damping and dispersion error come? The reason is that when we perform convergence analysis, we fix a point (x_j, t_n) and let $\Delta x, \Delta t \to 0$. However, numerically we cannot let $\Delta x, \Delta t \to 0$. Therefore, we need to consider the damping and dispersion as a result of not being able to do time iteration with $\Delta x, \Delta t \to 0$. On the other hand, let us still look the bright side, i.e., the maximum value principle is always conserved with the upwind scheme.*

2.2 High Order Approximation And Vanishing Viscosity Solution

2.2.1 From Explicit Euler Scheme to Lax-Wendroff Scheme

In order to resolve the issues of damping and dispersion, a better scheme is needed. The upwind scheme uses linear interpolation to compute the numerical solution at the next time step. Therefore, it is possible to apply a high-order interpolation. We first introduce the **LAX-WENDROFF SCHEME**, where a quadratic interpolation is adopted.

$$\begin{aligned} & U_j^{n+1} = \frac{1}{2}\nu(1+\nu)U_{j-1}^n + (1-\nu^2)U_j^n - \frac{1}{2}\nu(1-\nu)U_{j+1}^n \\ \Rightarrow \quad & U_j^{n+1} = U_j^n - \nu\Delta_{0x}U_j^n + \frac{1}{2}\nu^2\delta_x^2 U_j^n \end{aligned} \tag{2.4}$$

Proof We provide a derivation for the Lax-Wendroff scheme. Suppose the exact solution exists, where we will use the PDE $u_t = -au_x$, $u_{xt} = u_{tx} = -(au_x)_x$ and the central difference $\Delta_{0x}U_j^n = [U_{j+1}^n - U_{j-1}^n]/2$, $\delta_x U_j^n = U_{j+1/2}^n - U_{j-1/2}^n$.

$$\begin{aligned} u(x,t+\Delta t) &= u(x,t) + u_t(x,t)\Delta t + \frac{1}{2}u_{tt}(x,t)\Delta t^2 \\ &= u(x,t) + [-a(x,t)u_x(x,t)]\Delta t + \frac{1}{2}\left[-a(x,t)u_x(x,t)\right]_t \Delta t^2 \\ &= u(x,t) + [-a(x,t)u_x(x,t)]\Delta t + \frac{1}{2}\left[-a_t(x,t)u_x(x,t) - a(x,t)[u_t(x,t)]_x\right]\Delta t^2 \\ &= u(x,t) + [-a(x,t)u_x(x,t)]\Delta t + \frac{1}{2}\left[-a_t(x,t)u_x(x,t) + a(x,t)[a(x,t)u_x(x,t)]_x\right]\Delta t^2 \end{aligned}$$

Then, discretize the Taylor expansion, we have

$$U_j^{n+1} = U_j^n - a_j^n\Delta t\frac{\Delta_{0x}U_j^n}{\Delta x} + \frac{1}{2}\Delta t^2\left[-(a_t)_j^n\frac{\Delta_{0x}U_j^n}{\Delta x} + a_j^n\frac{\delta_x(a_j^n\delta_x U_j^n)}{\Delta^2}\right]$$

Suppose a is constant (or a varies slowly), then

$$\begin{aligned} & U_j^{n+1} = U_j^n - a\Delta t\frac{\Delta_{0x}U_j^n}{\Delta x} + \frac{1}{2}\Delta t^2\left[a^2\frac{\delta_x^2 U_j^n}{\Delta^2}\right] \\ \Rightarrow \quad & U_j^{n+1} = U_j^n - \nu\Delta_{0x}U_j^n + \frac{1}{2}\nu^2\delta_x^2 U_j^n \end{aligned}$$

Or we can expand the scheme as Eq.2.4. ■

Observing that at least one of the coefficients in Eq.2.4, especially based on the second equations, is negative. Thus, it does not satisfy the Maximum Principle, s.t. oscillation may occur.

The Fourier analysis for the Lax-Wendroff scheme is quite straight forward. Suppose $U_j^n = \lambda^n e^{ijk\Delta x}$ and recall that $\xi = k\Delta x$, and we can solve that

$$\lambda(k) = 1 - 2\nu^2\sin^2\frac{1}{2}k\Delta x - i\nu\sin k\Delta x = 1 - 2\nu^2\sin^2\frac{1}{2}\xi - i\nu\sin\xi$$

And the modulus of $\lambda(k)$

$$|\lambda(k)|^2 = 1 - 4\nu^2(1-\nu^2)\sin^4\frac{1}{2}k\Delta x$$

Therefore, if $\nu = 1$, $|\lambda(k)| = 1$. If $0 < \nu < 1$, $|\lambda(k)| < 1$. For the question above, the error of damping is given as $O(\xi^4)$, comparing to $O(\xi^2)$ for the upwind scheme.

However, the relative error of dispersion is still given by $O(\xi^2)$, since

$$\begin{cases} \text{Re}: & 1-2\nu^2\sin^2(k\Delta x/2) \\ \text{Im}: & -\nu\sin k\Delta x \end{cases} \Rightarrow \arg\lambda = -\arctan\left[\frac{\nu\sin k\Delta x}{1-2\nu^2\sin^2(k\Delta x/2)}\right]$$

Therefore

$$\arg\lambda \sim \nu\xi\left[1-\frac{1}{6}(1-\nu^2)\xi^2+...\right]$$

And we cannot say the Lax-Wendroff scheme provide advantages in fixing the dispersion.

Remark *(Summary of Euler Schemes)*
For the upwind scheme, the truncation error is of the order $O(\Delta t+\Delta x)$. For Lax-Wendroff scheme, the truncation error is of the order $O(\Delta t+\Delta x^2)$. CFL condition for both method should be $|\nu|=|a\Delta t/\Delta x|<1$ since they are one-step forward schemes (draw a mesh grid to check it). Two kinds of "shape errors" discussed are damping/dissipation for amplitudes and dispersion for phases of the waves. We suppose the dispersion is relatively hard to remove.

2.2.2 Lax-Wendroff Scheme and Conservation Laws

Suppose $f(u)$ is sufficiently smooth, and a conservation law can be written as follows. Note that in Eq.2.5, only u is the argument of $f(u)$.

$$\begin{aligned} & u_t+[f(u)]_x=0 \\ \Rightarrow\quad & u_t+a(u)u_x=0,\ a(u)=f'(u) \end{aligned} \tag{2.5}$$

The Lax-Wendroff scheme for Eq.2.5 can be roughly motivated as follows, where in the second step, we follow the assumption that a is nearly constant w.r.t. time.

$$\begin{aligned} u(x,t+\Delta t) &= u(x,t)+u_t(x,t)\Delta t+\frac{1}{2}u_{tt}u(x,t)\Delta t^2 \\ &\approx u(x,t)-[f(u)]_x\Delta t+\frac{1}{2}\left[a(u)[f(u)]_x\right]_x\Delta t^2 \end{aligned}$$

Use the central differences w.r.t. x, and the discretized numerical scheme is

$$U_j^{n+1}=U_j^n-\frac{\Delta t}{\Delta x}\Delta_{0x}f\left(U_j^n\right)+\frac{1}{2}\frac{\Delta t^2}{\Delta x^2}\delta_x\left[a(U)\delta_x f\left(U_j^n\right)\right] \tag{2.6}$$

Recall that there is already a $1/2$ within the Δ_{0x} so the denominator is just Δx rather than $2\Delta x$, see Eq.1.7.

To apply Eq.2.6, we need the characteristic speed $a(U)=a\left(U_{j\pm1/2}^n\right)$, s.t. the central difference δ_x can be fulfilled, where $a(u)=dx/dt$. There exists two methods for obtaining that.

$$a\left(U_{j\pm1/2}^n\right)=\frac{1}{2}\left[a\left(U_j^n\right)+a\left(U_{j\pm1}^n\right)\right],\ \text{or, } a\left(U_{j\pm1/2}^n\right)=\frac{\Delta_{\pm x}f(U_j^n)}{\Delta_{\pm x}U_j^n}$$

In general, we prefer the second one, for it can capture the relationship between $f(u)$ and $a(u)$.

Example 2.2 (Burgers' Equation)
Suppose the Burgers' equation for an invisvid flow

$$u_t+\left[\frac{1}{2}u^2\right]_x=0\Longleftrightarrow u_t+uu_x=0 \tag{2.7}$$

Solve it with characteristics, where we have

$$\begin{cases} \dfrac{dx}{dt} = u \\ \dfrac{du}{dt} = u_t + u_x \dfrac{dx}{dt} = u_t + uu_x = 0 \end{cases}$$

Therefore, the solution can be written as follows, where the wave amplitude maintained along characteristics. However, the tricky issue is the characteristics are not straight lines but related to the solution itself.

$$u(x,t) = u_0\left(x - tu(x,t)\right)$$

Try a very smooth IC, $u(x,0) = u_0(x) = \exp[-10(4x-1)^2]$. Then, we may find a **SHOCK** in the solution domain, where two characteristics intersect. Since the two characteristics hold two different (in general) initial values, right at the point the shock starts, the function (solution) will be multi-valued. That is one common issue for the nonlinear problems, and the points where shock appears become singularities.

An illustrative diagram is shown in Fig.2.3. We observe that at t_c, the classical solution breaks down, and if $t \geq t_c$, the PDE $u_t + a(u)u_x = 0$ is not valid any more, for the derivative u_x becomes not well defined. In general, a weak solution is need in such a problem, and here we propose the viscosity solution. ▲

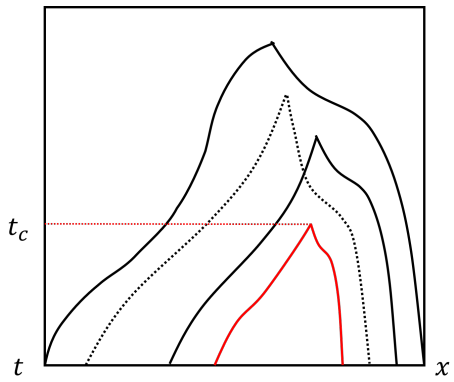

Figure 2.3: Intersection of characteristics in the solution domain. t_c dndicates the earliest occurrence of shocks

SINGULAR PERTURBATION AND VANISHING VISCOSITY

The viscous Burgers' equation can be written as

$$u_t + uu_x = \varepsilon u_{xx} \tag{2.8}$$

$0 < \varepsilon \ll 1$ is the small viscosity parameter. We obtain solution u_ε, and $u_\varepsilon \to u$ in some weak sense. Then u is one solution of $u_t + uu_x = 0$.

To understand the relationship between the viscous equation and the Lax-Wendroff scheme, we adopted the model problem with a constant wave speed a, and derive the Lax-Wendroff scheme

$$\begin{aligned} & u_t + au_x = \varepsilon u_{xx} \\ \Rightarrow \quad & U_j^{n+1} = U_j^n - \frac{a\Delta t}{2\Delta x}\left(U_{j+1}^n - U_{j-1}^n\right) + \frac{a^2\Delta t^2}{2\Delta x^2}\left(U_{j+1}^n - 2U_j^n + U_{j-1}^n\right) \\ & = U_j^n - a\Delta t\frac{U_{j+1}^n - U_{j-1}^n}{2\Delta x} + \frac{a^2\Delta t^2}{2}\frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{\Delta x^2} \end{aligned}$$

Let $\varepsilon = a^2\Delta t/2$, we obtain

$$\frac{U_j^{n+1}-U_j^n}{\Delta t}+a\frac{U_{j+1}^n-U_{j-1}^n}{2\Delta x}=\frac{a^2\Delta t}{2}\cdot\frac{U_{j+1}^n-2U_j^n+U_{j-1}^n}{\Delta x^2}$$
$$\Rightarrow\quad \frac{U_j^{n+1}-U_j^n}{\Delta t}+a\frac{U_{j+1}^n-U_{j-1}^n}{2\Delta x}=\varepsilon\cdot\frac{U_{j+1}^n-2U_j^n+U_{j-1}^n}{\Delta x^2}$$

Thus, essentially, the Lax-Wendroff scheme solves the viscous version of the original PDE.

For the intersection of characteristics, where shocks exist, we rewrite the upwind scheme and the Lax-Wendroff scheme from the conservation law shown in Eq.2.5, i.e., $u_t+[f(u)]_x=0$.

1. The upwind scheme from the original PDE becomes
$$U_j^{n+1}=U_j^n-\frac{1}{2}\frac{\Delta t}{\Delta x}\left[\left(1-\text{sign}\left(A_{j+1/2}^n\right)\right)\Delta_{+x}F_j^n+\left(1+\text{sign}\left(A_{j-1/2}^n\right)\right)\Delta_{-x}F_j^n\right]$$
2. The Lax-Wendroff scheme, essentially corresponding to the viscous solution, becomes
$$U_j^{n+1}=U_j^n-\frac{1}{2}\frac{\Delta t}{\Delta x}\left[\left(1-\frac{\Delta t}{\Delta x}A_{j+1/2}^n\right)\Delta_{+x}F_j^n+\left(1+\frac{\Delta t}{\Delta x}A_{j-1/2}^n\right)\Delta_{-x}F_j^n\right]$$

where $F_j^n=f\left(U_j^n\right)$, and $A_{j\pm1/2}^n=\Delta_{\pm x}F_j^n/\Delta_{\pm x}U_j^n$ are the characteristic speed.

Suppose the PDE is in a vector form, i.e., $\vec{u}_t+[\vec{f}(\vec{u})]_x=0$. Let $\vec{u}_t=-[\vec{f}(\vec{u})]_x$, and $\vec{u}_{tt}=-\vec{f}_{xt}=-\vec{f}_{tx}=-\left(A\vec{u}_t\right)_x=\left(A\vec{f}_x\right)_x$, where $A=\partial_{\vec{u}}\vec{f}(\vec{u})$. The Lax-Wendroff scheme can be written as

$$\vec{U}_j^{n+1}=\vec{U}_j^n-\frac{\Delta t}{\Delta x}\Delta_{0x}\vec{f}\left(\vec{U}_j^n\right)+\frac{1}{2}\frac{\Delta t^2}{\Delta x^2}\delta_x\left[A\left(\vec{U}_j^n\right)\delta_x\vec{f}\left(\vec{U}_j^n\right)\right]$$

2.2.3 Other Useful Schemes

We will demonstrate some useful numerical schemes with a as a constant.

LEAP-FROG SCHEME

In explicit Euler scheme, using central difference will results in an unstable scheme, i.e., the following scheme is not stable based on Fourier analysis.

$$\frac{U_j^{n+1}-U_j^n}{\Delta t}+a\frac{U_{j+1}^n-U_{j-1}^n}{2\Delta x}=0$$

However, we can apply central differences in the time domain, which results in the leap-frog scheme, i.e.,

$$\frac{U_j^{n+1}-U_j^{n-1}}{2\Delta t}+a\frac{U_{j+1}^n-U_{j-1}^n}{2\Delta x}=0 \tag{2.9}$$

The scheme has the order of accuracy $O(\Delta t^2+\Delta x^2)$. One challenge is that the scheme needs one additional step to obtain the numerical solution at the second time step, where we may use the Lax-Wendroff scheme in Eq.2.4.

LAX-FRIEDRICHS SCHEME

We start with the unstable scheme

$$\frac{U_j^{n+1}-U_j^n}{\Delta t}+a\frac{U_{j+1}^n-U_{j-1}^n}{2\Delta x}=0\Rightarrow U_j^{n+1}=U_j^n-\frac{\nu}{2}\left(U_{j+1}^n-U_{j-1}^n\right)$$

Replace $U_j^n=(U_{j+1}^n+U_{j-1}^n)/2$, we have

$$U_j^{n+1}=\frac{U_{j+1}^n+U_{j-1}^n}{2}-\frac{\nu}{2}\left(U_{j+1}^n-U_{j-1}^n\right) \tag{2.10}$$

To show why Lax–Friedrichs scheme works, rearrange the equation

$$U_j^{n+1} = U_j^n + \frac{1}{2}\left(U_{j+1}^n - 2U_j^n + U_{j-1}^n\right) - \frac{a\Delta t}{2\Delta x}\left(U_{j+1}^n - U_{j-1}^n\right)$$
$$\Rightarrow \quad \frac{U_j^{n+1} - U_j^n}{\Delta t} + a\frac{U_{j+1}^n - U_{j-1}^n}{2\Delta x} = \frac{\Delta x^2}{2\Delta t}\frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{\Delta x^2}$$

It satisfies the CFL condition and it is stable. Comparing to the viscous equation, we have

$$u_t + au_x = \varepsilon u_{xx} \Rightarrow \varepsilon = \frac{\Delta x^2}{2\Delta t}$$

BEAM-WARMING SCHEME

Start with the Lax-Wendroff scheme in Eq.2.4, i.e.,

$$U_j^{n+1} = U_j^n - \frac{a\Delta t}{2\Delta x}\left(U_{j+1}^n - U_{j-1}^n\right) + \frac{a^2\Delta t^2}{2\Delta x^2}\left(U_{j+1}^n - 2U_j^n + U_{j-1}^n\right)$$

And we use the upwind approximations for the second and the third terms in RHS to obtain the Beam-Warming scheme,

$$\begin{cases} U_j^{n+1} = U_j^n - \dfrac{a\Delta t}{2\Delta x}\left(3U_j^n - 4U_{j-1}^n + U_{j-2}^n\right) + \dfrac{a^2\Delta t^2}{2\Delta x^2}\left(U_j^n - 2U_{j-1}^n + U_{j-2}^n\right), & a > 0 \\ U_j^{n+1} = U_j^n - \dfrac{a\Delta t}{2\Delta x}\left(-3U_j^n + 4U_{j+1}^n - U_{j+2}^n\right) + \dfrac{a^2\Delta t^2}{2\Delta x^2}\left(U_j^n - 2U_{j+1}^n + U_{j+2}^n\right), & a < 0 \end{cases} \tag{2.11}$$

The accuracy order is of $O(\Delta t + \Delta x^2)$, and the CFL condition is $|\nu| = |a\Delta t/\Delta x| \leq 2$. The Beam-Warming scheme can be considered as an upwind version of Lax-Wendroff scheme.

2.2.4 Characteristics Tracing and Interpolations

Characteristics tracing means solving the equation along the characteristic curves. Suppose the problem is $u_t + au_{xx} = 0$, where $a > 0$ and constant. First search along the characteristic line

$$\frac{dx}{dt} = a, \ \frac{du}{dt} = 0$$

We may observe $u(x^*, t_n) = U_j^{n+1}$ along one characteristic curve. The idea is to interpolate $u(x^*, t_n)$ with $\{U_j^n\}_{j=1}^J$. A diagram of characteristics tracing is shown in Fig.2.4. There exists a range of examples in doing the interpolations.

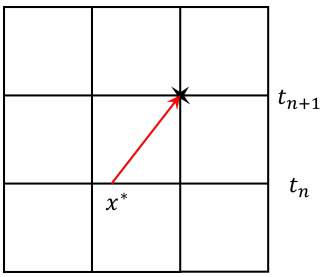

Figure 2.4: A diagram of characteristics tracing

LINEAR INTERPOLATION

Since $x^* = x_j - a\Delta t$ and $a > 0$, the linear polynomial interpolation with U_{j-1}^n and U_j^n can be written as

$$P_1(x) = U_j^n + \frac{x - x_j}{\Delta x}\left(U_j^n - U_{j-1}^n\right)$$

Hence

$$U_j^{n+1} = u(x^*, t_n) \approx P_1(x^*) = U_j^n - \frac{a\Delta t}{\Delta x}\left(U_j^n - U_{j-1}^n\right) = U_j^n - \nu\left(U_j^n - U_{j-1}^n\right)$$

Thus, we obtain the upwind scheme again, with truncation error of the order $O(\Delta t + \Delta x)$.

QUADRATIC INTERPOLATION I

Suppose we use the quadratic interpolation with nodes $\{U^n_{j-1}, U^n_j, U^n_{j+1}\}$. The interpolation results can be written as

$$U^{n+1}_j \approx P_2(x^*) = \frac{1}{2}\nu(1+\nu)U^n_{j-1} + (1-\nu^2)U^n_j - \frac{1}{2}\nu(1-\nu)U^n_{j+1}$$

That is the Lax-Wendroff scheme, with truncation error of the order $O(\Delta t + \Delta x^2)$. Recall that the Lax-Wendroff scheme is designed to mitigate the damping errors.

QUADRATIC INTERPOLATION II

Suppose we use the quadratic interpolation with nodes $\{U^n_{j-2}, U^n_{j-1}, U^n_j\}$. The interpolation results can be written as

$$U^{n+1}_j \approx P_2(x^*) = \frac{1}{2}\nu(\nu-1)U^n_{j-2} + (2\nu-\nu^2)U^n_{j-1} - \left(1 - \frac{3}{2}\nu + \frac{1}{2}\nu^2\right)U^n_j$$

That is the Beam-Warming scheme, with truncation error of the order $O(\Delta t + \Delta x^2)$.

Remark *In general, we can choose stencils (points on the t_n level) to construct polynomial interpolations to approximate $u(x^*, t_n)$. High-order interpolations are doable, while CFL condition must be satisfied when choosing the stencils.*

Remark *Forward Euler schemes usually requires the CFL condition, i,e, $|a\Delta t/\Delta x| \leq C$ for some constant C, s.t. $\Delta t = O(\Delta x)$, which is OK for many applications. In order to choose a larger time step size, one way is to design implicit schemes, where the CFL condition can be automatically satisfied and no restriction on Δt or Δx is needed.*

2.3 Implicit Schemes

The implicit schemes for hyperbolic equations are similar to the ones for parabolic equations, where u_x or U_{xx} is approximated at the time level t^{n+1}. For our model problem shown in Eq.2.2, suppose $a > 0$ is a constant, and there exists a variety of designs of the implicit schemes.

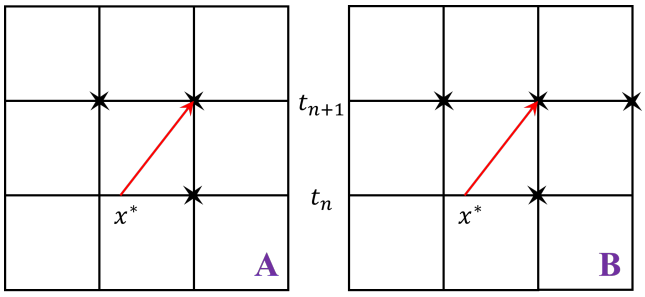

Figure 2.5: The stencils of the upwind scheme (A) and the backward-time central difference scheme (B)

UPWIND SCHEME

The upwind scheme with a constant $a > 0$ can be written as

$$\frac{U^{n+1}_j - U^n_j}{\Delta t} + a\frac{U^{n+1}_j - U^{n+1}_{j-1}}{\Delta x} = 0 \tag{2.12}$$

The CFL condition is satisfied automatically, which can be observed with Fig.2.5. Using Fourier analysis, assume the Fourier mode is $U_j^n = \lambda^n e^{ikj\Delta x}$, where we can solve

$$\lambda(k) = \frac{1}{1 + \nu(1 - \cos k\Delta x + i \sin k\Delta x)} \tag{2.13}$$

Define $\xi = k\Delta x$, we have

$$|\lambda(k)|^2 = \frac{1}{[1 - \nu(1 - \cos \xi)]^2 + \nu^2 \sin^2 \xi} \leq 1, \ \forall \nu$$

Therefore, the implicit upwind scheme is unconditionally stable. The truncation error can be computed at (x_j, t_{n+1}), which has the order of $O(\Delta t + \Delta x)$.

BACKWARD-TIME CENTRAL DIFFERENCE SCHEME

The backward-time central difference scheme with a constant $a > 0$ can be written as

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + a\frac{U_{j+1}^{n+1} - U_{j-1}^{n+1}}{2\Delta x} = 0 \tag{2.14}$$

Recall that for the corresponding explicit time scheme, it is unconditionally unstable. However, the CFL condition can be satisfied automatically for the implicit scheme. A diagram for the stencil is shown in Fig.2.5. The Fourier analysis with the assumed mode $U_j^n = \lambda^n e^{ikj\Delta x}$ yields

$$\lambda(k) = \frac{1}{1 + i\nu \sin k\Delta x} \tag{2.15}$$

Define $\xi = k\Delta x$, we have

$$|\lambda(k)|^2 = \frac{1}{1 + \nu^2 \sin^2 \xi} \leq 1, \ \forall \nu$$

So the scheme is unconditionally stable. The truncation error can be computed at (x_j, t_{n+1}), which has the order of $O(\Delta t + \Delta x^2)$.

LAX-WENDROFF IMPLICIT SCHEME

Recall the explicit Lax-Wendroff scheme is

$$U_j^{n+1} = \frac{1}{2}\nu(1 + \nu)U_{j-1}^n + (1 - \nu^2)U_j^n - \frac{1}{2}\nu(1 - \nu)U_{j+1}^n$$

Change the value at t_n to t_{n+1}, we have the implicit Lax-Wendroff scheme

$$-\frac{1}{2}\nu(1 + \nu)U_{j-1}^{n+1} + (1 + \nu^2)U_j^{n+1} + \frac{1}{2}\nu(1 - \nu)U_{j+1}^{n+1} = U_j^n \tag{2.16}$$

Clearly, the CFL condition is satisfied automatically, and equipped with the Fourier analysis with the assumed mode $U_j^n = \lambda^n e^{ikj\Delta x}$,

$$\lambda(k) = \frac{1}{1 + 2\nu^2 \sin^2 \frac{1}{2}k\Delta x + i\nu \sin k\Delta x} \tag{2.17}$$

It is obvious that $|\lambda(k)| \leq 1, \ \forall \nu$, s.t. the scheme is unconditionally stable. By computing the truncation error at (x_j, t_{n+1}), the scheme has an accuracy of order $O(\Delta t + \Delta x^2)$.

CRANK-NICOLSON SCHEME

Rearrange the explicit and implicit central difference scheme, the C-N scheme can be written as

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{a}{2}\left[\frac{U_{j+1}^n - U_{j-1}^n}{2\Delta x} + \frac{U_{j+1}^{n+1} - U_{j-1}^{n+1}}{2\Delta x}\right] = 0 \tag{2.18}$$

The CFL condition can be automatically satisfied, and with the Fourier analysis, we have

$$\lambda(k) = \frac{1 - \frac{i}{2}\nu \sin k\Delta x}{1 + \frac{i}{2}\nu \sin k\Delta x} = \frac{1 - \frac{i}{2}\nu \sin \xi}{1 + \frac{i}{2}\nu \sin \xi} \tag{2.19}$$

Thus, $|\lambda(k)| = 1$, and the scheme is unconditionally stable. However, we should be aware that $|\lambda(k)| = 1$ could be dangerous, for during the computation, the running error may cause $|\lambda(k)| > 1$. By computing the truncation error at $(x_j, t_{n+1/2})$, the scheme has an accuracy of order $O(\Delta t^2 + \Delta x^2)$.

If we would like to add stability to the C-N scheme, we may enlarge the weight of the implicit portion. However, that will make the scheme not symmetric between t_n and t_{n+1}, and hence we loss the second order accuracy in time.

Remark *(MATLAB Demonstration)*
Consider the following hyperbolic problem

$$\begin{cases} \textit{PDE:} & u_t + au_{xx} = 0, (x,t) \in [0,1] \times [0, t_F] \\ \textit{IC:} & u(x,0) = u_0(x) \\ \textit{BC:} & \textit{some compatible BCs} \end{cases}$$

For the compatible BCs,

1. *$a > 0$ and constant, the characteristics are from left, so we need BC at the left side, i.e., $u(0,t) = g(t)$,*
2. *$a < 0$ and constant, the characteristics are from right, so we need BC at the right side, i.e., $u(1,t) = h(t)$*

We can draw a diagram to present the relation between the characteristics and the BCs, and we omit the details here. For some application, we may need periodic BCs, such as $u(0,t) = u(1,t)$ for problem repeated in the spatial domain.

A sample (pseudo) code for the explicit upwind scheme with $a > 0$ can be shown as follows. Recall the numerical scheme is shown as $U_j^{n+1} = U_j^n - \nu\left(U_j^n - U_{j-1}^n\right)$

```
% ------------ Explicit Upwind Scheme, a>0 --------------------
u=zeros(N,J), u(1,x)=u0(x);
for n=2:N, j=1:J
      if j==1 u(n,j)=g(tn) end
      if j>1  u(n,j)=u(n-1,j)-nu*(u(n-1,j)-u(n-1,j-1)) end end
```

If we apply the backward-time central difference scheme

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + a\frac{U_{j+1}^{n+1} - U_{j-1}^{n+1}}{2\Delta x} = 0 \Longleftrightarrow -\frac{\nu}{2}U_{j-1}^{n+1} + U_j^{n+1} + \frac{\nu}{2}U_{j+1}^{n+1} = U_j^n$$

Therefore, the implicit scheme can be reduced into a linear system $A\vec{u}^{n+1} = \vec{b}$, where A is a tri-diagonal system. For $j = 0$, we use the given BC since $a > 0$. The tricky issue is the computation of the value at J, for the lack of information at $J + 1$. $J + 1$ is known as a **GHOST POINT**, *which does not exist actually. The ghost point can be obtained by linear interpolation with function values at $J - 1$ and J.*

Remark *(Periodic Boundary Conditions)*
Consider the explicit time Lax–Friedrichs, one can use U_{J-1} as U_{-1}, and use U_1 as U_{J+1}. Thus, for $j = 0, 1, 2, .., J$, we have

$$\left.\begin{aligned} U_0^{n+1} &= \left(\frac{1}{2} - \frac{\nu}{2}\right)U_1^n + \left(\frac{1}{2} + \frac{\nu}{2}\right)U_{-1}^n \\ U_J^{n+1} &= \left(\frac{1}{2} - \frac{\nu}{2}\right)U_{J+1}^n + \left(\frac{1}{2} + \frac{\nu}{2}\right)U_{J-1}^n \end{aligned}\right\} = \left(\frac{1}{2} - \frac{\nu}{2}\right)U_1^n + \left(\frac{1}{2} + \frac{\nu}{2}\right)U_{J-1}^n$$

Normal Lax–Friedrichs scheme can be applied for $j = 1, 2, ..., J - 1$.

The implicit Lax–Friedrichs scheme can be defined by using values at time t_{n+1} for the spatial derivative. In this case, we transfer the implicit Lax–Friedrichs scheme into a linear system, and the coefficient matrix will not be tri-diagonal with the periodic BC. In general, the form of the coefficient matrix can be shown as follows, and Gauss elimination or Gauss-Seidal iteration can be used to solve such a linear system.

$$\begin{pmatrix} \times & \times & \dots & \dots & \dots & \times \\ \times & \times & \times & \dots & \dots & \dots \\ \dots & \times & \times & \times & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \times & \times & \times \\ \times & \dots & \dots & \dots & \times & \times \end{pmatrix}$$

2.4 2D Hyperbolic Problems

Consider the following model problem

$$\begin{cases} u_t + au_x + bu_y = 0 \\ u(x, y, 0) = u_0(x, y) \end{cases} \tag{2.20}$$

$a,\ b$ are constant. Let $\nu_x = a\Delta t/\Delta x,\ \nu_y = b\Delta t/\Delta y$. The characteristics of the 2D model problem is as follows,

$$\frac{dx}{dt} = a,\ \frac{dy}{dt} = b,\ \frac{du}{dt} = 0 \tag{2.21}$$

Along the characteristic line, the solution to the 2D problem can be written as $u(x, y, t) = u_0(x - at, y - bt)$. A domain of dependence is shown in Fig.2.6. And we present several numerical schemes as follows.

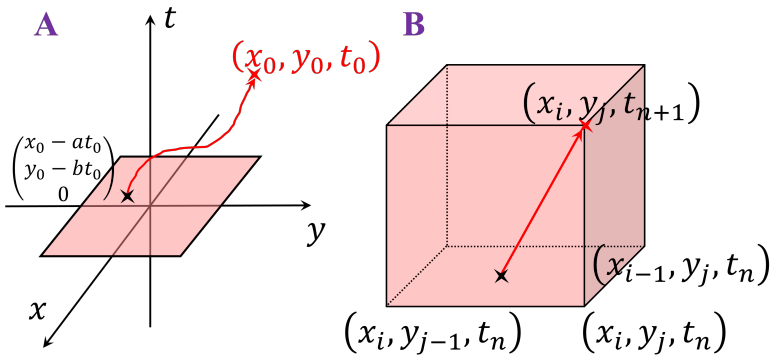

Figure 2.6: The diagram of characteristics and the domain of dependence (A) and the stencils used in the upwind scheme (B)

EXPLICIT UPWIND SCHEME

Suppose $a, b > 0$, then the explicit upwind scheme can be written as

$$\frac{U_{i,j}^{n+1} - U_{i,j}^n}{\Delta t} + a\frac{U_{i,j}^n - U_{i-1,j}^n}{\Delta x} + b\frac{U_{i,j}^n - U_{i,j-1}^n}{\Delta y} = 0 \tag{2.22}$$

The stencils and a characteristic line are shown in Fig.2.6. The characteristics are straight lines shown via Eq.2.21. Referring to the CFL condition, we need $\max\{|\nu_x|, |\nu_y|\} \leq 1$.

To apply the Fourier analysis, assume $U_{i,j}^n = \lambda^n e^{\tau k_x i\Delta x} e^{\tau k_y j\Delta y}$, where $\tau = \sqrt{-1}$ and $\vec{k} = (k_x, k_y)$. Then, we

can solve for $\lambda(\vec{k})$ from the numerical scheme in Eq.2.22.

$$\begin{aligned}\lambda(\vec{k}) &= 1-\nu_x\left(1-e^{-\tau k_x\Delta x}\right)-\nu_y\left(1-e^{-\tau k_y\Delta y}\right)\\ |\lambda(\vec{k})|^2 &= \left(1-2\nu_x\sin^2\frac{\xi}{2}-2\nu_y\sin^2\frac{\eta}{2}\right)^2+(\nu_x\sin\xi+\nu_y\sin\eta)^2\end{aligned} \tag{2.23}$$

We assume $\xi = k_x\Delta x, \eta = k_y\Delta y$, and we want to have $|\lambda(\vec{k})| \le 1$. Taking the derivative of $|\lambda(\vec{k})|^2$ w.r.t. $\vec{k}$, we have the maxima of $|\lambda(\vec{k})|^2$ located at $(\pm\pi,\pm\pi),(\pm\pi,0),(0,\pm\pi),(0,0)$. Compute the values of $|\lambda(\vec{k})|^2$ at all of those points, we obtain

$$\begin{aligned}|\lambda(\pm\pi,\pm\pi)|^2 &= (1-2\nu_x-2\nu_y)^2 &\Rightarrow\quad |\nu_x+\nu_y|\le 1\\ |\lambda(\pm\pi,0)|^2 &= (1-2\nu_x)^2 &\Rightarrow\quad |\nu_x|\le 1\\ |\lambda(0,\pm\pi)|^2 &= (1-2\nu_y)^2 &\Rightarrow\quad |\nu_y|\le 1\\ |\lambda(0,0)|^2 &= 1\end{aligned}$$

Thus, the stability condition will be $|\nu_x+\nu_y| \le 1$. Notice that the computation above is based on $a, b > 0$. Thus, in general case, we need $|\nu_x| + |\nu_y| \le 1$. We emphasize that the stability condition is more restrict comparing to the CFL condition, i.e., $\max\{|\nu_x|,|\nu_y|\} \le 1$. The truncation error can be computed using Taylor expression at (x_i, y_j, t_n), which is $|T^n_{i,j}| = O(\Delta t+\Delta x+\Delta y)$.

EXPLICIT LAX-FRIEDRICHS SCHEME

Analogy to the 1D case, the central difference scheme is not stable, i.e.,

$$U^{n+1}_{i,j} = U^n_{i,j}-\frac{\nu_x}{2}\left(U^n_{i+1,j}-U^n_{i-1,j}\right)-\frac{\nu_y}{2}\left(U^n_{i,j+1}-U^n_{i,j-1}\right)$$

Replace $U^n_{i,j}$ with an average of value over the nearest four points

$$U^{n+1}_{i,j} = \frac{1}{4}\left(U^n_{i-1,j}+U^n_{i+1,j}+U^n_{i,j-1}+U^n_{i,j+1}\right)-\frac{\nu_x}{2}\left(U^n_{i+1,j}-U^n_{i-1,j}\right)-\frac{\nu_y}{2}\left(U^n_{i,j+1}-U^n_{i,j-1}\right) \tag{2.24}$$

After adopting the Fourier analysis, we can solve that

$$\begin{aligned}\lambda(\vec{k}) &= \frac{1}{2}(\cos\xi+\cos\eta)-\tau(\nu_x\sin\xi+\nu_y\sin\eta),\ \tau=\sqrt{-1}\\ |\lambda(\vec{k})|^2 &= 1-(\sin^2\xi+\sin^2\eta)\left[\frac{1}{2}-(\nu_x^2+\nu_y^2)\right]-\frac{1}{4}(\cos\xi-\cos\eta)^2-(\nu_x\sin\eta-\nu_y\sin\xi)^2\end{aligned} \tag{2.25}$$

To ensure $|\lambda(\vec{k})|^2 \le 1$, we require

$$|\lambda(\vec{k})|^2 \le 1-(\sin^2\xi+\sin^2\eta)\left[\frac{1}{2}-(\nu_x^2+\nu_y^2)\right] \le 1 \Rightarrow \nu_x^2+\nu_y^2 \le \frac{1}{2}$$

That is more strict than $\max\{|\nu_x|,|\nu_y|\} \le 1$ as required by the CFL condition. By computing the truncation error at (x_i, y_j, t_n), the scheme is of the order $O(\Delta t+\Delta x+\Delta y)$.

2.4.1 Alternative Direction Implicit Scheme

We propose the design of alternative direction schemes in this section. However, comparing to ADI, the alternative direction schemes here may not necessarily implicit. Start with the model problem Eq.2.20, and perform Taylor expansion w.r.t. time, t. Let $Au = A_1u + A_2u = -au_x - bu_y$, where A_1, A_2 should be considered as "operators".

$$\begin{aligned}u(t+\Delta t) &= u(t)+\Delta t u_t(t)+O(\Delta t^2) = u(t)-\Delta t(au_x+bu_y)+O(\Delta t^2)\\ &= (1+\Delta tA_1)(1+\Delta tA_2)u(t)-(\Delta t)^2A_1A_2u(t)+O(\Delta t^2)\end{aligned}$$

Thus, we have $u(t+\Delta t) \approx (1+\Delta A_1)(1+\Delta A_2)u(t)$, or $U_{i,j}^{n+1} = (1+\Delta A_1)(1+\Delta A_2)U_{i,j}^n$. Decompose it into a "two-step" format,

$$\begin{cases} U^{n+1/2} = (1+\Delta t A_2)U^n \\ U^{n+1} = (1+\Delta t A_1)U^{n+1/2} \end{cases} \tag{2.26}$$

$U^{n+1/2}$ is a very "formal" label, where $n+1/2$ does not necessary mean a specific time step. We can also observe that based on the Δt, since there is no $\Delta t/2$ appears in this equation indicating half time steps.

We provide one explicit alternative direction scheme and three ADI (implicit) schemes as follows.

LAX-WENDROFF SCHEME, EXPLICIT IN TIME

The alternative direction version of the Lax-Wendroff scheme for the 2D case is

$$\begin{cases} U_{i,j}^{n+1/2} = U_{i,j}^n - \nu_x \Delta_{0x} U_{i,j}^n + \dfrac{1}{2}\nu_x^2 \delta_x^2 U_{i,j}^n \\ U_{i,j}^{n+1} = U_{i,j}^{n+1/2} - \nu_y \Delta_{0y} U_{i,j}^{n+1/2} + \dfrac{1}{2}\nu_y^2 \delta_y^2 U_{i,j}^{n+1/2} \end{cases} \tag{2.27}$$

Each step in Eq.2.27 is analogous to the 1D expression in Eq.2.6. By cancelling the intermediate stage $U_{i,j}^{n+1/2}$, we get the full scheme. By computing the truncation error at the intermediate step, $(x_i, y_j, t_{n+1/2})$, we should obtain a result similar to the 1D scheme, i.e., $\left|T_{i,j}^{n+1/2}\right| = O(\Delta t^2 + \Delta x^2 + \Delta y^2)$

Applying the Fourier analysis, we have

$$\lambda(\vec{k}) = \left(1 - \tau\nu_x \sin\xi - 2\nu_x^2 \sin^2\frac{\xi}{2}\right)\left(1 - \tau\nu_y \sin\eta - 2\nu_y^2 \sin^2\frac{\eta}{2}\right) \tag{2.28}$$

After some calculation, the stability implies $\max\{|\nu_x|, |\nu_y|\} \leq 1$, same as the CFL condition.

LOCALLY ONE-DIMENSION SCHEME (LOD, ADI)

This is our first ADI scheme. The numerical formulation can be shown as follows

$$\begin{cases} (1+\nu_x \Delta_{0x}) U_{i,j}^{n+1/2} = U_{i,j}^n \\ (1+\nu_y \Delta_{0y}) U_{i,j}^{n+1} = U_{i,j}^{n+1/2} \end{cases} \tag{2.29}$$

Cancel the intermediate step at $(i, j, n+1/2)$, we have

$$\left(U_{i,j}^{n+1} - U_{i,j}^n\right) + \nu_x \Delta_{0x} U_{i,j}^{n+1} + \nu_y \Delta_{0y} U_{i,j}^{n+1} + \nu_x \nu_y \Delta_{0x} \Delta_{0y} U_{i,j}^{n+1} = 0$$

By computing the truncation error at the intermediate step, $(x_i, y_j, t_{n+1/2})$, we should obtain $\left|T_{i,j}^{n+1/2}\right| = O(\Delta t^2 + \Delta x^2 + \Delta y^2)$ since apparently, only central differences are applied.

Applying the Fourier analysis, we have the following results for the growth factor, and we have $|\lambda(\vec{k})| \leq 1$.

$$\lambda(\vec{k}) = \frac{1}{(1+\tau\nu_x \sin\xi)(1+\tau\nu_y \sin\eta)} \tag{2.30}$$

CRANK-NICOLSON SCHEME (C-N, ADI)

The numerical scheme is shown as follows

$$\left(U_{i,j}^{n+1} - U_{i,j}^n\right) + \frac{\nu_x}{2}\Delta_{0x}\left(U_{i,j}^n + U_{i,j}^{n+1}\right) + \frac{\nu_y}{2}\Delta_{0y}\left(U_{i,j}^n + U_{i,j}^{n+1}\right) = 0 \tag{2.31}$$

Similar to the C-N scheme for parabolic problems, it is unconditionally stable. However, referring to Eq.2.19, we can image this "stability" is not stable, i.e., the C-N scheme is at the margin of the stability territory. Computing the truncation error at $(x_i, y_j, t_{n+1/2})$ yields $\left|T_{i,j}^{n+1/2}\right| = O(\Delta t^2 + \Delta x^2 + \Delta y^2)$

In order to rewrite the C-N scheme into an ADI version, we rearrange the original C-N scheme into

$$\begin{aligned}
&\left(1+\frac{\nu_x}{2}\Delta_{0x}+\frac{\nu_y}{2}\Delta_{0y}\right)U_{i,j}^{n+1}=\left(1-\frac{\nu_x}{2}\Delta_{0x}-\frac{\nu_y}{2}\Delta_{0y}\right)U_{i,j}^{n}\\
\Rightarrow\quad &\left(1+\frac{\nu_x}{2}\Delta_{0x}\right)\left(1+\frac{\nu_y}{2}\Delta_{0y}\right)U_{i,j}^{n+1}=\left(1-\frac{\nu_x}{2}\Delta_{0x}\right)\left(1-\frac{\nu_y}{2}\Delta_{0y}\right)U_{i,j}^{n}\\
&\qquad\qquad+\frac{\nu_x\nu_y}{4}\Delta_{0x}\Delta_{0y}\left(U_{i,j}^{n+1}-U_{i,j}^{n}\right)+O(\Delta t^2)\\
\Rightarrow\quad &\left(1+\frac{\nu_x}{2}\Delta_{0x}\right)\left(1+\frac{\nu_y}{2}\Delta_{0y}\right)U_{i,j}^{n+1}\approx\left(1-\frac{\nu_x}{2}\Delta_{0x}\right)\left(1-\frac{\nu_y}{2}\Delta_{0y}\right)U_{i,j}^{n}
\end{aligned}$$

Decompose it and obtain the ADI scheme as

$$\begin{cases}
\left(1+\frac{\nu_x}{2}\Delta_{0x}\right)U_{i,j}^{n+1/2}=\left(1-\frac{\nu_y}{2}\Delta_{0y}\right)U_{i,j}^{n}\\
\left(1+\frac{\nu_y}{2}\Delta_{0y}\right)U_{i,j}^{n+1}=\left(1-\frac{\nu_x}{2}\Delta_{0x}\right)U_{i,j}^{n+1/2}
\end{cases}\tag{2.32}$$

BEAM-WARMING SCHEME (ADI)

The ADI version of the Beam-warming scheme can be written as

$$\begin{cases}
\left(1+\frac{\nu_x}{2}\Delta_{0x}\right)U_{i,j}^{*}=\left(1-\frac{\nu_x}{2}\Delta_{0x}\right)\left(1-\frac{\nu_y}{2}\Delta_{0y}\right)U_{i,j}^{n}\\
\left(1+\frac{\nu_y}{2}\Delta_{0y}\right)U_{i,j}^{n+1}=U_{i,j}^{*}
\end{cases}\tag{2.33}$$

By applying the Fourier analysis, we have $|\lambda(\vec{k})|=1$, which is expected by comparing to its form with the C-N scheme and referring to Eq.2.19. So, it is unconditionally stable, but such a stability is dangerous. Computing the truncation error at (x_i,y_j,t_*) yields $O(\Delta t^2+\Delta x^2+\Delta y^2)$. Although we do not explicitly write $U_{i,j}^{*}$ as $U_{i,j}^{n+1/2}$ here, based on the appearance of $\nu_x/2$ and $\nu_y/2$, we see the intermediate step $U_{i,j}^{*}$ is essentially $U_{i,j}^{n+1/2}$.

2.5 Consistency, Convergence, Stability and Lax Equivalent Theorem

Given the general form of the PDE as follows

$$\begin{cases}
\dfrac{\partial u}{\partial t}=\mathcal{L}(u), & \Omega\times(0,t_F]\\
g(u)=g_0, & \text{on } \partial\Omega\\
u(x,0)=u_0(x), & \text{on } \Omega \text{ when } t=0
\end{cases}\tag{2.34}$$

And the following assumptions are adopted

1. Ω is a bounded domain, and $\partial\Omega$ is the boundary of Ω.
2. $g(u)$ represents the BC and $u_0(x)$ represents the IC.
3. $\mathcal{L}(u)$ is a **LINEAR OPERATOR**, adn involves the partial derivatives of u w.r.t. the spatial domain. $\mathcal{L}(u)$ does not explicitly involves t.

E.g., if $u_t=u_{xx}$, we have $\mathcal{L}(u)=u_{xx}$; if $u_t+au_x=0$, we have $\mathcal{L}(u)=-au_x$.

The problem is **WELL-POSED** if

1. Solution exists for all data u_0, where $\|u_0\|$ is bounded.
2. $\exists K$ as a constant, and $\forall$ pairs of solutions u and v, $\|u(t_n)-v(t_n)\|\le K\|u_0-v_0\|$, with $t_n\le t_F$.

In general, FDM is a two-step work, i.e., to discretize of the PDE and to solve the difference equation system. For discratization, e.g., in 2D spatial domain, the time and spatial steps can be presented as $(\Delta t,\Delta x,\Delta y)$, and the mesh points can be written as (t_n,x_i,y_j). Denote $\vec{U}^n=U_{i,j}^n$ as the numerical solution at level t_n, while denote $\vec{u}^n=u(x_i,y_j,t_n)$ as the discretized values of the exact solution at time t_n.

Definition 2.1 (Norm)

In the discretized space, define the following norms

1. l^∞ **MAXIMUM NORM**: $\|\vec{U}^n\|_\infty = \max_{i,j}\{U^n_{i,j}\}$
2. l^2 **NORM**: $\|U^n\|_2 = \left[\sum_{i,j} |U^n_{i,j}|^2 \cdot v_{i,j}\right]^{1/2}$, *where* $v_{i,j}$ *is the cell volume, e.g.,* $v_{i,j} = \Delta x \Delta y$ *in 2D problems.*

The corresponding L^∞ *and* L^2 *norms for continuous cases can be defined analogically, where* max *can be replaced with* sup *and* $\sum$ *can be replaced with* $\int$.

♣

After the discretization, the PDE can be transformed into a difference equation system, where a general form of the difference schemes can be shown as

$$B_1\vec{U}^{n+1} = B_0\vec{U}^n + \vec{F}^n \tag{2.35}$$

where B_1, B_0 and $\vec{F}^n$ depend on the FDM schemes and the PDE, i.e., the linear operator $\mathcal{L}(u)$. Assume B_1 is invertible, then

$$\vec{U}^{n+1} = B_1^{-1}\left[B_0\vec{U}^n + \vec{F}^n\right]$$

Suppose B_1 is **UNIFORMLY WELL-CONDITIONED**, i.e., $\exists K$ as an constant, s.t. $\|B_1^{-1}\| \leq K\Delta t$. Under such problem settings, we present the following general version of the theoretical issues we discussed before.

CONSISTENCY:

Consistency is defined as "if the discretized difference equation consists with the original continuous PDE".

$$B_1\vec{U}^{n+1} - \left[B_0\vec{U}^n + \vec{F}^n\right] \to \frac{\partial u}{\partial t} - \mathcal{L}(u), \text{ as } \Delta t, \Delta x, \Delta y \to 0 \tag{2.36}$$

After defining the truncation error by replacing the numerical solution by the exact solution in FDM, i.e.,

$$T^n = B_1\vec{u}^{n+1} - \left[B_0\vec{u}^n + \vec{F}^n\right]$$

Then, the consistency of the difference scheme with the PDE means, for sufficiently smooth solution u

$$T^n_{i,j} \to 0, \text{ as } \Delta t, \Delta x, \Delta y \to 0, \forall i, j$$

Denote h as the size of the spatial mesh grid. For the order of accuracy, suppose p, q are the largest positive numbers, s.t. the following estimation can be satisfied with sufficiently smooth solution u, i.e.,

$$\left|T^n_{i,j}\right| \leq C[\Delta t^p + h^q], \text{ as } \Delta t, h \to 0, \forall i, j$$

Then, the scheme is called to have the order of accuracy p in Δt and q in h.

CONVERGENCE:

The numerical scheme provides convergent approximation to the PDE, if $\|\vec{U}^n - \vec{u}^n\| \to 0$ as $\Delta t, h \to 0$, for $n\Delta t = t \in [0, t_F)$, and for $\forall u_0$ that makes the PDE problem well-posed, i.e., the solution is unique in this case.

Convergence of a finite difference numerical scheme can be understood with the following two aspects:

1. The linear system has a solution, i.e. $B_1\vec{U}^{n+1} = B_0\vec{U}^n + \vec{F}^n$.
2. The numerical solution converge to the exact solution.

STABILITY:

The scheme is said to be stable, if two solutions $\vec{U}^n$ and $\vec{V}^n$ of the scheme has the same non-homogeneous term $\vec{F}^n$ (for force vector) but with different ICs $\vec{U}^0$ and $\vec{V}^0$, s.t. $\exists$ constant K independent to the ICs and mesh size,

and the following inequality is satisfied

$$\left\|\vec{V}^n - \vec{U}^n\right\| \leq K \left\|\vec{V}^0 - \vec{U}^0\right\|, \, n\Delta t \leq t_F \Longleftrightarrow \left\|\left(B_1^{-1}B_0\right)^n\right\| \leq K, \, n\Delta t \leq t_F$$

The equivalent relationship can be shown as follows. Suppose $\vec{U}^{n+1} = B_1^{-1}(B_0\vec{U}^n + \vec{F}^n)$ and $\vec{V}^{n+1} = B_1^{-1}(B_0\vec{V}^n + \vec{F}^n)$. Thus

$$\left(\vec{U}^{n+1} - \vec{V}^{n+1}\right) = B_1^{-1}B_0\left(\vec{U}^n - \vec{V}^n\right) = \left(B_1^{-1}B_0\right)^n\left(\vec{U}^0 - \vec{V}^0\right)$$

If recursive relation is with a bounded coefficient $\left\|\left(B_1^{-1}B_0\right)^n\right\| \leq K$, then let $\vec{W} = \vec{V} - \vec{U}$, and $\vec{W}^{n+1} = \left(B_1^{-1}B_0\right)^n\vec{W}^0$. Since $\vec{W}^0$ and $\left\|\left(B_1^{-1}B_0\right)^n\right\|$ are bounded, $\vec{W}^{n+1}$ is bounded.

Theorem 2.2 (Lax Equivalent Theorem)

Given a consistent difference approximation to a well-posed linear evolutionary problem, which is uniformly solvable in the sense $\|B_1^{-1}\| \leq K\Delta t$ for some constant K, i.e., the conditional number is small, the stability of the scheme is necessary and sufficient for convergence. ♡

Proof We emphasize the $\Rightarrow$ direction.

$\Rightarrow$: Use the numerical and exact solution

$$\begin{cases} B_1\vec{U}^{n+1} = B_0\vec{U}^n + \vec{F}^n \\ B_1\vec{u}^{n+1} = B_0\vec{u}^n + \vec{F}^n + \vec{T}^n \end{cases}$$

Take the difference between the two equations and use iterative expressions

$$\begin{aligned} \vec{U}^{n+1} - \vec{u}^{n+1} &= B_1^{-1}B_0\left(\vec{U}^n - \vec{u}^n\right) - B_1^{-1}\vec{T}^n \\ &= B_1^{-1}B_0\left[B_1^{-1}B_0\left(\vec{U}^{n-1} - \vec{u}^{n-1}\right) - B_1^{-1}\vec{T}^{n-1}\right] - B_1^{-1}\vec{T}^n \\ \vec{U}^n - \vec{u}^n &= -B_1^{-1}\vec{T}^{n-1} - B_1^{-1}B_0B_1^{-1}\vec{T}^{n-2} - \ldots - \left(B_1^{-1}B_0\right)^{n-1}B_1^{-1}\vec{T}^0 \end{aligned}$$

Recall the two critical conditions used in FDM

Well-posed (for the PDE): $\left\|\left(B_1^{-1}B_0\right)^n\right\| \leq K, \, n\Delta t \leq t_F$

Well-conditioned (for the linear system): $\left\|B_1^{-1}\right\| \leq K_1\Delta t$

Under those two conditions, we have $\left\|\left(B_1^{-1}B_0\right)^m B_1^{-1}\right\| \leq K_1K\Delta t$ for any $m \leq n$. Thus,

$$\left\|\vec{U}^n - \vec{u}^n\right\| \leq K_1K\Delta t\sum_{m=0}^{n-1}\left\|\vec{T}^m\right\| \leq K_1Kt_F\max_{m=1,2,\ldots,n}\left\|\vec{T}^m\right\|$$

Using the consistency, we have the convergence.

$\Leftarrow$: This can be proved by triangle inequality and uniformly solvable condition. A more simple observation is that in numerical PDE, we usually use consistency and stability to imply convergence, so convergence looks "stronger". We can use convergence and triangle inequalities to establish stability if the original PDE has continuous dependence to the initial condition (the PDE is not ill-posed or chaotic). ■

Since stability is quite useful, we provide more description about the stability. First, note that let $\vec{W}^n = \vec{U}^n - \vec{V}^n$, the definition of stability is equivalent to "if $B_1\vec{W}^{n+1} = B_0\vec{W}^n$ and $n\Delta t \leq t_F$, then $\|\vec{W}^n\| \leq K\|\vec{W}^0\|$". Some tools can be used to judge stability

1. Fourier analysis is usually an "iff" condition with constant coefficients.

2. Maximum principle is usually a "sufficient" condition with varying coefficients.

Remark *(Maximum Principle v.s. Stability)*

1. *Some schemes do not satisfy the maximum principle are actually stable in the maximum norm.*
2. *Maximum principle is seldom available nor even appropriate for hyperbolic problems. I.e., for hyperbolic PDEs, we usually cannot have maximum principle, e.g., Lax-Wendroff scheme.*

For a Fourier mode, $\vec{k}$, from the Fourier analysis and the finite difference scheme, we have

$$\hat{B}_1(\vec{k})\hat{U}^{n+1}(\vec{k}) = \hat{B}_0(\vec{k})\hat{U}^n(\vec{k})$$

Let $\hat{G}(\vec{k}) = \left[\hat{B}_1^{-1}\hat{B}_0\right]$ be the **AMPLIFICATION MATRIX**, and the stability in l^2 norm is equivalent to

$$\left|\left[\hat{G}(\vec{k})\right]^n\right| \leq K,\ \forall \vec{k} \text{ and } n\Delta t \leq t_F$$

$|\cdot|$ is the matrix norm.

Theorem 2.3 (von Neumann Condition)

A necessary condition for stability is that there exists a constant K', s.t.,

$$\left|\lambda(\vec{k})\right| \leq 1 + K'\Delta t,\ \forall \vec{k},\ n\Delta t \leq t_F$$

For every eigenvalue $\lambda(\vec{k})$ of the amplification matrix $\hat{G}(\vec{k})$. ♡

Proof Just need to show $\left|\lambda(\vec{k})\right| \leq K$. Using the assumption

$$\left|\lambda(\vec{k})\right|^n \leq \left(1 + K'\Delta t\right)^n \approx \left(1 + K'n\Delta t\right) + O(\Delta t^2)$$
$$\Rightarrow \left|\lambda(\vec{k})\right|^n \approx \left(1 + K't_F\right) + O(\Delta t^2) \sim 1 + K't_F = K$$

Thus, $\left|\left[\hat{G}(\vec{k})\right]^n\right|$ is bounded, which is an equivalent condition for stability. ■

Remark *(Dissipation and Dispersion)*

1. *We define the dissipation of PDE solutions as that the Fourier modes do not grow (stable) w.r.t. time and at least one mode decays. We say the numerical scheme is of* **NON-DISSIPATION** *if the Fourier modes neither decay nor grow.*
2. *We define dispersion of PDEs as that the Fourier modes of difference wave length (or different wave numbers) propagate at different speed.*

2.6 Exercises

✍ **Exercise 2.1** Consider the following 1D problem

$$u_t(x,t) + u_x(x,t) = 0$$
$$u(x,0) = u_0(x) = \begin{cases} x, & x \in (0,1] \\ 2-x, & x \in [1,2) \end{cases}$$

1. Write down the explicit upwind scheme, the explicit Lax-Friedrichs scheme, the C-N scheme, and the explicit Lax-Wendroff scheme. Find the order to accuracy and the stability condition for these schemes.

2. Solve the PDE with the explicit schemes mentioned above with a periodic BC, i.e., $u(0,t) = u(2,t)$. Take (i) $\Delta t/\Delta x = 1/2$, $\Delta x = 2/100$, and (ii) $\Delta t/\Delta x = 3/2$, $\Delta x = 2/100$. Plot numerical solutions and exact solutions at $t = [0, 2, 4, 16, 32, 64]\Delta t$. Make comparisons and comment on your numerical results. (e.g., dissipation, dispersion, stability, etc).
3. Check the order of accuracy for the explicit Lax-Friedrichs scheme and the C-N scheme by comparing the numerical solutions with the exact solutions: choose $\Delta t/\Delta x = 1/2$ and the refinement path $\{\Delta x = (2/100, 1/100, 1/200, 1/400)\}$, check the accuracy of the numerical solutions with both maximum norm error and l_2 norm error at $t = 0.2$.
4. Repeat the third part with a smooth IC $u_0(x) = \sin \pi x$ and comment on the results.

Solve

1. *We provide accuracy and stability analyses for the selected schemes as follows.*
   EXPLICIT UPWIND SCHEME: *Recall the scheme is*

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x}(U_j^n - U_{j-1}^n) = U_j^n - \nu(U_j^n - U_{j-1}^n)$$

Compute the truncation error at (x_j, t_n)*, then*

$$\begin{aligned}
T_j^n =& \frac{1}{\Delta t}\left[u(x_j, t_{n+1}) - u(x_j, t_n)\right] + \frac{1}{\Delta x}\left[u(x_j, t_n) - u(x_{j-1}, t_n)\right] \\
=& \frac{1}{\Delta t}\left[u(x_j, t_n) + u_t(x_j, t_n)\Delta t + \frac{1}{2}u_{tt}(x_j, t_n)\Delta t^2 + \ldots - u(x_j, t_n)\right] \\
&+ \frac{1}{\Delta x}\left[u(x_j, t_n) + u_x(x_j, t_n)\Delta x + \frac{1}{2}u_{xx}(x_j, t_n)\Delta x^2 + \ldots - u(x_j, t_n)\right] \\
=& u_t(x_j, t_n) + \frac{1}{2}u_{tt}(x_j, t_n)\Delta t + O(\Delta t^2) + u_x(x_j, t_n) + \frac{1}{2}u_{xx}(x_j, t_n)\Delta x + O(\Delta x^2) + \ldots \sim O(\Delta t + \Delta x)
\end{aligned}$$

Take the Fourier mode $U_j^n = \lambda^n e^{ikj\Delta x}$*, we obtain*

$$\lambda^{n+1} e^{ikj\Delta x} = \lambda^n e^{ikj\Delta x} - \nu\left(\lambda^n e^{ikj\Delta x} - \lambda^{n+1} e^{ik(j-1)\Delta x}\right)$$

s.t. we can solve for λ*,*

$$\lambda(k) = 1 - \nu\left(1 - e^{-ik\Delta x}\right),\ |\lambda(k)|^2 = 1 - 4\nu(1-\nu)\sin^2\frac{k}{2}\Delta x$$

If we want $|\lambda(k)|^2 \le 1$*, we need* $0 \le \nu \le 1$*. That matches the CFL condition.*

EXPLICIT LAX-FRIEDRICHS SCHEME: *Recall the scheme is*

$$U_j^{n+1} = \frac{1}{2}\left(U_{j+1}^n + U_{j-1}^n\right) - \frac{\Delta t}{2\Delta x}\left(U_{j+1}^n - U_{j-1}^n\right) = \frac{1}{2}\left(U_{j+1}^n + U_{j-1}^n\right) - \frac{\nu}{2}\left(U_{j+1}^n - U_{j-1}^n\right)$$

Compute the truncation error at (x_j, t_n)*, then*

$$\begin{aligned}
T_j^n =& \frac{1}{\Delta t}\left[u(x_j, t_{n+1}) - \frac{1}{2}u(x_{j-1}, t_n) - \frac{1}{2}u(x_{j+1}, t_n)\right] + \frac{1}{2\Delta x}\left[u(x_j, t_n) - u(x_{j-1}, t_n)\right] \\
=& \frac{1}{\Delta t}\left[u(x_j, t_n) + u_t(x_j, t_n)\Delta t + \frac{1}{2}u_{tt}(x_j, t_n)\Delta t^2 + \ldots - u(x_j, t_n) - \frac{1}{2}u_{xx}(x_j, t_n)\Delta x^2 - \ldots\right] \\
&+ \frac{1}{2\Delta x}\left[u(x_j, t_n) + u_x(x_j, t_n)\Delta x + \frac{1}{2}u_{xx}(x_j, t_n)\Delta x^2 + \frac{1}{6}u_{xxx}(x_j, t_n)\Delta x^3 + \ldots\right. \\
&\left. - u(x_j, t_n) + u_x(x_j, t_n)\Delta x - \frac{1}{2}u_{xx}(x_j, t_n)\Delta x^2 + \frac{1}{6}u_{xxx}(x_j, t_n)\Delta x^3 - \ldots\right] \\
=& u_t(x_j, t_n) + \frac{1}{2}u_{tt}(x_j, t_n)\Delta t - \frac{1}{2}u_{xx}(x_j, t_n)\frac{\Delta x^2}{\Delta t} + u_x(x_j, t_n) + \frac{1}{6}u_{xxx}(x_j, t_n)\Delta x^2 + \ldots \\
=& O\left(\Delta t + \frac{\Delta x^2}{\Delta t} + \Delta x^2\right) \sim O(\Delta t + \Delta x)
\end{aligned}$$

This scheme has the first order accuracy on time and space, if we assume $\Delta t \sim \Delta x$*. Take the Fourier mode* $U_j^n = \lambda^n e^{ikj\Delta x}$*, we obtain*

$$\lambda^{n+1} e^{ikj\Delta x} = \frac{1}{2}\left(\lambda^n e^{ik(j+1)\Delta x} + \lambda^n e^{ik(j-1)\Delta x}\right) - \frac{\nu}{2}\left(\lambda^n e^{ik(j+1)\Delta x} - \lambda^n e^{ik(j-1)\Delta x}\right)$$

s.t. we can solve for λ*,*

$$\lambda(k) = \cos k\Delta x - i\nu \sin k\Delta x,\ |\lambda(k)|^2 = \cos^2 k\Delta x + \nu^2 \sin^2 k\Delta x$$

If we want $|\lambda(k)|^2 \leq 1$, we need $0 \leq \nu \leq 1$, which matches the CFL condition.

C-N SCHEME: *Recall the scheme is*

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{4\Delta x}\left[(U_{j+1}^n - U_{j-1}^n) + (U_{j+1}^{n+1} - U_{j-1}^{n+1})\right] = U_j^n - \frac{\nu}{4}\left[(U_{j+1}^n - U_{j-1}^n) + (U_{j+1}^{n+1} - U_{j-1}^{n+1})\right]$$

Compute the truncation error at $(x_j, t_{n+1/2})$, then

$$\begin{aligned}
T_j^{n+1/2} =& \frac{1}{\Delta t}\left[u(x_j, t_{n+1}) - u(x_j, t_n)\right] + \frac{1}{4\Delta x}\left[(u(x_{j+1}, t_n) + u(x_{j-1}, t_n)) + (u(x_{j+1}, t_{n+1}) + u(x_{j-1}, t_{n+1}))\right] \\
=& \left[u_t(x_j, t_{n+1/2}) + \frac{1}{24}u_{ttt}(x_j, t_{n+1/2})\Delta t^2 + ...\right] \\
&+ \left[u_x(x_j, t_{n+1/2}) + \frac{1}{8}u_{xtt}(x_j, t_{n+1/2})\Delta t^2 + \frac{1}{6}u_{xxx}(x_j, t_{n+1/2})\Delta x^2 + \frac{1}{48}u_{xxxtt}(x_j, t_{n+1/2})\Delta x^2\Delta t^2 + ...\right] \\
\sim& O(\Delta t^2 + \Delta x^2)
\end{aligned}$$

Take the Fourier mode $U_j^n = \lambda^n e^{ikj\Delta x}$, we obtain,

$$\lambda(k) = \frac{1 - \frac{i}{2}\nu \sin k\Delta x}{1 + \frac{i}{2}\nu \sin k\Delta x}$$

Thus, $|\lambda(k)| = 1$, and the scheme is unconditionally stable, but risky since $|\lambda|$ is on the boundary.

EXPLICIT LAX-WENDROFF SCHEME: *Recall the scheme is*

$$\begin{aligned}
U_j^{n+1} =& \frac{\Delta t}{2\Delta x}\left[1 + \frac{\Delta t}{\Delta x}\right]U_{j-1}^n + \left[1 - \frac{\Delta t^2}{\Delta x^2}\right]U_j^n - \frac{\Delta t}{2\Delta x}\left[1 - \frac{\Delta t}{\Delta x}\right]U_{j+1}^n \\
=& \frac{\nu}{2}(1+\nu)U_{j-1}^n + (1-\nu^2)U_j^n - \frac{\nu}{2}(1-\nu)U_{j+1}^n
\end{aligned}$$

Or we can rewrite it as

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{U_j^{n+1} - U_{j-1}^n}{2\Delta x} - \frac{\Delta t}{2}\frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{\Delta x^2} = 0$$

Compute the truncation error at (x_j, t_n), then

$$\begin{aligned}
T_j^n =& \frac{1}{\Delta t}\left[u(x_j, t_{n+1}) - u(x_j, t_n)\right] + \frac{1}{2\Delta x}\left[u(x_{j+1}, t_n) - u(x_{j-1}, t_n)\right] - \frac{\Delta t}{2\Delta x^2}\left[u(x_{j+1}, t_n) - 2u(x_j, t_n) + u(x_{j-1}, t_n)\right] \\
=& \left[u_t(x_j, t_n) + \frac{1}{2}u_{tt}(x_j, t_n)\Delta t + O(\Delta t^2) + ...\right] + \left[u_x(x_j, t_n) + \frac{1}{6}u_{xxx}(x_j, t_n)\Delta x^2 + O(\Delta x^4) + ...\right] \\
&- \frac{\Delta t}{2\Delta x^2}\left[u_{xx}(x_j, t_n)\Delta x^2 + O(\Delta x^4)\right] = \left[\frac{1}{2}u_{tt} - \frac{1}{2}u_{xx}\right]\Delta t + \frac{1}{6}u_{xxx}\Delta x^2 + ... \sim O(\Delta t + \Delta x^2)
\end{aligned}$$

Take the Fourier mode $U_j^n = \lambda^n e^{ikj\Delta x}$, we obtain,

$$\lambda(k) = 1 - 2\nu^2 \sin^2\frac{1}{2}k\Delta x - i\nu \sin k\Delta x,\ |\lambda(k)|^2 = 1 - 4\nu^2(1-\nu^2)\sin^4\frac{1}{2}k\Delta x$$

Thus, $|\lambda(k)| \leq 1$ if $0 < \nu \leq 1$.

2. *We omit the plots but summary some important observations.*

EXPLICIT UPWIND SCHEME: *If $\nu = 5$, the solution is stable. If $\nu = 1.5$, the solution is not stable. The dissipation error is obvious, particularly a the maxima and the minima of the wave. The dispersion error is not obvious in the numerical results.*

EXPLICIT LAX-FRIEDRICHS SCHEME: *If $\nu = 5$, the solution is stable. If $\nu = 1.5$, the solution is not only unstable, but also not satisfies the maximum principle, which can be observed from the maximum and the minimum points of the wave. The dissipation error is obvious, particularly a the maxima and the minima of the wave when $\nu = 0.5$. The dispersion error is not obvious in the numerical results.*

C-N SCHEME: *For both $\nu = 0.5, 1.5$, the solutions are stable. Dissipation errors can be observed for both ν, particularly for the scheme with $\nu = 1.5$, where the dissipation error occurs not only near the maxima and minima of the wave, but also within the increasing and decreasing portions. The dispersion error can be observed in the results with $\nu = 1.5$ (Δt is relatively large).*

EXPLICIT LAX-WENDROFF SCHEME: *For $\nu = 1.5$, the numerical solution is hyper-unstable, while the solution under $\nu = 0.5$ is stable. Dissipation and dispersion errors can be observed for $\nu = 0.5$. For comparison, the magnitude of dissipation error is smaller than the upwind scheme, but the dispersion error is relatively large comparing with the other three schemes.*

3. *Although we omit the plots, we mention that the maximum norm (l^∞) and l^2 norm of the differences between the numerical solutions and the exact solutions in normal and log-log scales. As the mesh size decreases, the error decreases. However, it does not follow the order of truncation or convergence errors. One reason is that the exact solution is not smooth, s.t. the derivatives of the exact solution is not bounded. Recall that the smoothness of the exact solution is the prerequisites of the error analyses, while the absolute convergence is the prerequisites of the Fourier analyses. Thus, in this example, the order of accuracy from theoretical analyses may not be achieved.*
4. *Similar to the results from the previous problem, as the mesh size decreases, the error decreases. In this example, the exact solution, as well as the IC, are continuous and smooth. Therefore, the actual order or accuracy matches the theoretical derivations.*

▲

Remark *In Sections 3 and 4, we applied the refinements in the spatial domain. What about Δt? Since Δt and Δx may have different orders of accuracy, then what is the order of accuracy of the entire solutions? In order to avoid such trouble, we first fixed ν, and make comparisons at te same final time point, i.e., $t = 0.2$.*

Appendix. The MATLAB code

```
% ------------ Math 517 Chap.2 Exercise 1 (Section 2) --------
% --- Upwind Scheme, nu=0.5, dx=0.02, dt=0.01 ---
clear all; close all; clc;
u0=@(x) x*(x>=0&&x<1)+(2-x)*(x>=1&&x<=2);
Dx=0.02; nu=0.5; Dt=nu*Dx; T=0:Dt:3; X=0:Dx:2; tF=length(T); J=length(X); U1=zeros(tF,J);
[Xaxis,Taxis]=meshgrid(X,T);
for j=1:J; U1(1,j)=u0(X(j)); end
for n=2:tF
    U1(n,1)=U1(n-1,1)-nu*(U1(n-1,1)-U1(n-1,J-1));
    for j=2:J
       U1(n,j)=U1(n-1,j)-nu*(U1(n-1,j)-U1(n-1,j-1)); end; end
figure(1), mesh(Taxis,Xaxis,U1), xlabel('T'), ylabel('X'), zlabel('U')
% --- Lax-Friedrichs Scheme, nu=0.5, dx=0.02, dt=0.01 ---
clear all; clc;
u0=@(x) x*(x>=0&&x<1)+(2-x)*(x>=1&&x<=2);
Dx=0.02; nu=0.5; Dt=nu*Dx; T=0:Dt:3; X=0:Dx:2; tF=length(T); J=length(X); U1=zeros(tF,J);
[Xaxis,Taxis]=meshgrid(X,T);
for j=1:J; U1(1,j)=u0(X(j)); end
for n=2:tF
    U1(n,1)=0.5*(U1(n-1,2)+U1(n-1,J-1))-0.5*nu*(U1(n-1,2)-U1(n-1,J-1));
    U1(n,J)=0.5*(U1(n-1,2)+U1(n-1,J-1))-0.5*nu*(U1(n-1,2)-U1(n-1,J-1));
    for j=2:J-1
       U1(n,j)=0.5*(U1(n-1,j+1)+U1(n-1,j-1))-0.5*nu*(U1(n-1,j+1)-U1(n-1,j-1)); end; end
figure(2), mesh(Taxis,Xaxis,U1), xlabel('T'), ylabel('X'), zlabel('U')
% --- C-N Scheme, nu=0.5, dx=0.02, dt=0.01 ---
clear all; clc; error=0.0001;
u0=@(x) x*(x>=0&&x<1)+(2-x)*(x>=1&&x<=2);
Dx=0.02; nu=0.5; Dt=nu*Dx; T=0:Dt:3; X=0:Dx:2; tF=length(T); J=length(X); U1=zeros(tF,J); D=
    zeros(J,1);
[Xaxis,Taxis]=meshgrid(X,T);
for j=1:J; U1(1,j)=u0(X(j)); end
for n=2:tF
    D(1)=U1(n-1,1)-0.25*nu*U1(n-1,2)+0.25*nu*U1(n-1,J-1);
    D(J)=U1(n-1,1)-0.25*nu*U1(n-1,2)+0.25*nu*U1(n-1,J-1);
    for j=2:J-1; D(j)=U1(n-1,j)-0.25*nu*U1(n-1,j+1)+0.25*nu*U1(n-1,j-1); end
```

```
    % --- Gauss Seidel (iteration with "order")---
    UU=ones(1,J);
    while max(abs(UU-U1(n,:))./U1(n,:))>error
        UU=U1(n,:);
        U1(n,1)=D(1)-0.25*nu*UU(2)+0.25*nu*UU(J-1);
        for j=2:J-1; U1(n,j)=D(j)-0.25*nu*UU(j+1)+0.25*nu*U1(n,j-1); end
        U1(n,J)=D(J)-0.25*nu*U1(n,2)+0.25*nu*U1(n,J-1); end; end
figure(3), mesh(Taxis,Xaxis,U1), xlabel('T'), ylabel('X'), zlabel('U')
% --- Lax-Wendroff Scheme, nu=0.5, dx=0.02, dt=0.01 ---
clear all; clc;
u0=@(x) x*(x>=0&&x<1)+(2-x)*(x>=1&&x<=2);
Dx=0.02; nu=0.5; Dt=nu*Dx; T=0:Dt:3; X=0:Dx:2; tF=length(T); J=length(X); U1=zeros(tF,J);
[Xaxis,Taxis]=meshgrid(X,T);
for j=1:J; U1(1,j)=u0(X(j)); end
for n=2:tF
    U1(n,1)=0.5*nu*(1+nu)*U1(n-1,J-1)+(1-nu*nu)*U1(n-1,1)-0.5*nu*(1-nu)*U1(n-1,2);
    U1(n,J)=0.5*nu*(1+nu)*U1(n-1,J-1)+(1-nu*nu)*U1(n-1,J)-0.5*nu*(1-nu)*U1(n-1,2);
    for j=2:J-1
        U1(n,j)=0.5*nu*(1+nu)*U1(n-1,j-1)+(1-nu*nu)*U1(n-1,j)-0.5*nu*(1-nu)*U1(n-1,j+1);
    end; end
figure(4), mesh(Taxis,Xaxis,U1), xlabel('T'), ylabel('X'), zlabel('U')

% ------------ Math 517 Chap.2 Exercise 1 (Sections 3 and 4) --------
clear all; close all; clc; error=0.000001;
u0=@(x) x.*(x>=0&&x<1)+(2-x).*(x>=1&&x<=2); % for 4, use u0=@(x) sin(pi.*x);
% --- C-N Scheme ---
nu=0.5; DDx=[0.02,0.01,0.005,0.0025];
for i=1:4
    Dx=DDx(i); Dt=nu*Dx; T=0:Dt:0.2; X=0:Dx:2;
    tF=length(T); J=length(X); U1=zeros(tF,J); D=zeros(J,1);
    for j=1:J; U1(1,j)=u0(X(j)); end
    for n=2:tF
        D(1)=U1(n-1,1)-0.25*nu*U1(n-1,2)+0.25*nu*U1(n-1,J-1);
        D(J)=U1(n-1,J)-0.25*nu*U1(n-1,2)+0.25*nu*U1(n-1,J-1);
        for j=2:J-1; D(j)=U1(n-1,j)-0.25*nu*U1(n-1,j+1)+0.25*nu*U1(n-1,j-1); end
        % --- Gauss Seidel (iteration with "order")---
        UU=zeros(1,J); U1(n,:)=U1(n-1,:);
        while max(abs(UU-U1(n,:))./U1(n,:))>error
            UU=U1(n,:);
            U1(n,1)=D(1)-0.25*nu*UU(2)+0.25*nu*UU(J-1);
            for j=2:J-1; U1(n,j)=D(j)-0.25*nu*UU(j+1)+0.25*nu*U1(n,j-1); end
            U1(n,J)=D(J)-0.25*nu*U1(n,2)+0.25*nu*U1(n,J-1);
        end; end
    Uexact=MyUex(X,0.2); % for 4, use Uexact=MyUexSin(X,0.2);
    LinfCN(i)=max(abs(Uexact-U1(tF,:)));
    L2CN(i)=sqrt(Dx*((Uexact-U1(tF,:))*(Uexact-U1(tF,:))')); end
% --- Lax-Friedrichs Scheme ---
for i=1:4
```

```
    Dx=DDx(i); Dt=nu*Dx; T=0:Dt:0.2; X=0:Dx:2;
    tF=length(T); J=length(X); U2=zeros(tF,J);
    for j=1:J; U2(1,j)=u0(X(j)); end
    for n=2:tF
        U2(n,1)=0.5*(U2(n-1,2)+U2(n-1,J-1))-0.5*nu*(U2(n-1,2)-U2(n-1,J-1));
        U2(n,J)=0.5*(U2(n-1,2)+U2(n-1,J-1))-0.5*nu*(U2(n-1,2)-U2(n-1,J-1));
        for j=2:J-1
            U2(n,j)=0.5*(U2(n-1,j+1)+U2(n-1,j-1))-0.5*nu*(U2(n-1,j+1)-U2(n-1,j-1)); end; end
    Uexact=MyUex(X,0.2); % for 4, use Uexact=MyUexSin(X,0.2);
    LinfLF(i)=max(abs(Uexact-U2(tF,:)));
    L2LF(i)=sqrt(Dx*((Uexact-U2(tF,:))*(Uexact-U2(tF,:))')); end
% --- plots ---
figure(1), subplot(2,2,1), hold on, box on, title('l_{\infty} norm')
yyaxis left, plot(DDx,LinfCN,'b*-'), ylabel('Error of Crank-Nicolson')
yyaxis right, plot(DDx,LinfLF,'r.-'), ylabel('Error of Lax-Friedrichs')
subplot(2,2,2), hold on, box on, title('l_2 norm')
yyaxis left, plot(DDx,L2CN,'b*-'), ylabel('Error of Crank-Nicolson')
yyaxis right, plot(DDx,L2LF,'r.-'), ylabel('Error of Lax-Friedrichs')
subplot(2,2,3), loglog(DDx,LinfCN,'b*-',DDx,LinfLF,'r.-')
title('log(l_{\infty} norm)'), legend('Error of Crank-Nicolson','Error of Lax-Friedrichs')
subplot(2,2,4), loglog(DDx,L2CN,'b*-',DDx,L2LF,'r.-')
title('log(l_2 norm)'), legend('Error of Crank-Nicolson','Error of Lax-Friedrichs')

% --- Auxiliary Function, Exact Solution for Sections 1~3 ---
function uexact=MyUex(X,t)
ll=X-t; n=length(ll); uexact=zeros(1,n);
u10=@(x) x.*(x>=0&&x<1)+(2-x).*(x>=1&&x<=2);
for i=1:n
% A simple way to use mod operator
    ll(i)=mod(ll(i),2.0);
% A tedious looping way, may need to move more than one "2" to trace the solution back to
    the [0,2] interval backwards along the characteristics.
    while ~(ll(i)>=0&&ll(i)<=2)
        if ll(i)<0; ll(i)=ll(i)+2;
        else; ll(i)=ll(i)-2; end; end
    uexact(i)=u10(ll(i)); end

% --- Auxiliary Function, Exact Solution for Section 4 ---
function uexact=MyUexSin(X,t)
ll=X-t; n=length(ll); uexact=zeros(1,n);
u10=@(x) sin(pi.*x);
for i=1:n
% A simple way to use mod operator
    ll(i)=mod(ll(i),2.0);
% A tedious looping way, may need to move more than one "2" to trace the solution back to
    the [0,2] interval backwards along the characteristics.
    while ~(ll(i)>=0&&ll(i)<=2)
        if ll(i)<0; ll(i)=ll(i)+2;
```

```
    else; ll(i)=ll(i)-2; end; end
uexact(i)=u10(ll(i)); end
```

Chapter 3 Non-Linear Equations and Conservation Laws

3.1 Conservation Laws and The Scalar Formulation

CONSERVATION LAWS (CLs) are used to describe the model of physics, such as mass, energy, momentum, etc. Especially nonlinear PDEs belongs to the framework of CLs, where the behaviors of both the exact solution and the numerical solution are affected by the non-linearity. The general form of a CL can be shown as follows, where we assume that $\vec{F}(\vec{u})$ is convex, i.e., the Hessian $\vec{F}''(\vec{u}) > 0$.

$$\begin{cases} \dfrac{\partial \vec{u}}{\partial t} + \dfrac{\partial}{\partial x}\vec{F}(\vec{u}) = 0 \\ \vec{u}(x,0) = \vec{u}_0(x) \end{cases}, \quad \begin{cases} \vec{u}_t + \vec{F}'(\vec{u})\vec{u}_x = 0 \\ \vec{u}(x,0) = \vec{u}_0(x) \end{cases} \tag{3.1}$$

The first equation system in Eq.3.1 is called the **CONSERVATIVE FORM**, while the second equation system is called the **NON-CONSERVATIVE FORM**. By integrating of the conservative form, we obtain the **INTEGRAL FORM** shown in Eq.3.2.

$$\begin{aligned} & \int_{t_1}^{t_2}\int_a^b \left[\vec{u}_t dxdt + \left[\vec{F}(\vec{u})\right]_x\right] dxdt = 0 \\ \Rightarrow \quad & \begin{cases} \displaystyle\int_{t_1}^{t_2}\int_a^b \vec{u}_t dxdt = \int_a^b \left[\vec{u}(x,t_2) - \vec{u}(x,t_1)\right] dx \\ \displaystyle\int_{t_1}^{t_2}\int_a^b \left[\vec{F}(\vec{u})\right]_x dxdt = \int_{t_1}^{t_2} \left[\vec{F}(\vec{u}(b,t)) - \vec{F}(\vec{u}(a,t))\right] dx \end{cases} \\ \Rightarrow \quad & \int_a^b \vec{u}(x,t_2)dx - \int_a^b \vec{u}(x,t_1)dx = -\int_{t_1}^{t_2} \vec{F}(\vec{u}(b,t))dt + \int_{t_1}^{t_2} \vec{F}(\vec{u}(a,t))dt \end{aligned} \tag{3.2}$$

Remark *(Physical Interpretation of Eq.3.2)*
The change in the amount of materials in the interval $[a,b]$ during the time interval $[t_1,t_2]$, is equal to the material fluxes across the boundary $x=a$ and $x=b$ during that time interval.

We will consider strictly hyperbolic CLs, i.e., $\vec{F}$ has distinct real eigenvalues. One example is the in-viscid Burgers' equation, $u_t+(u^2/2)_x = 0$. The non-conservative form of the in-viscid Burgers' equation is $u_t+uu_x = 0$. Both of the conservative form and non-conservative form may be used in the following sections. Recall the viscous Burgers' equation is $u_t + (u^2/2)_x = \varepsilon u_{xx}$.

Consider the scalar CLs. By using the method of characteristics, the characteristics curve should be

$$\begin{cases} \dfrac{d}{dt}x(t) = F'[u(x(t),t)] \\ \dfrac{d}{dt}u(x(t),t) = u_x\dfrac{dx(t)}{dt} + u_t = F'(u)u_x + u_t = 0 \end{cases}$$

The solution is constant along characteristic curves. Since u and $F'(u)$ are constant along the characteristics, the characteristic curves should be straight lines. Thus, t is the only independent variable along those straight lines.

Proposition 3.1 (Implicit Solution of Scalar CLs)
If u is a sufficiently smooth solution to the IVP $u_t + \partial_x F(u) = 0$ and the IC $u(x,0) = u_0(x)$, $\forall t > 0, x \in \mathbb{R}$, then u will satisfy $u(x,t) = u_0\left[x - F'[u(x(t),t)]\right]$. Thus, the solution is defined implicitly. ♠

Since u is constant along characteristics and $\frac{dx(t)}{dt} = F'[u(x(t),t)] = F'(u_0)$. Then the characteristics must be straight lines. Tracing backwards to the level of IC, we should have $x(t) = F'(u_0)t + C$.

Remark *It is easy to see the characteristics of $u_t + au_x = 0$. However, for a nonlinear case, the characteristics generally have a range of slopes, i.e., $1/F'(u_0)$ for different positions x in the IC. However, each characteristic is a straight line.*

Example 3.1 (Characteristics of In-viscid Burgers' Equation)
Consider the in-viscid Burgers' equation

$$u_t + \left(u^2/2\right)_x = 0,\ x \in [0,1],\ u_0(x) = \sin 2\pi x$$

The characteristics is

$$x(t) = F'(u_0)t + C = (\sin 2\pi x_0)t + x_0$$

A series of characteristics are shown in 3.1. We observe that characteristics intersect (concentrating along a vertical line in a 3D means intersecting in 2D under vertical projection). Then the solution becomes double (or multiple) valued at the points where characteristics intersect. ▲

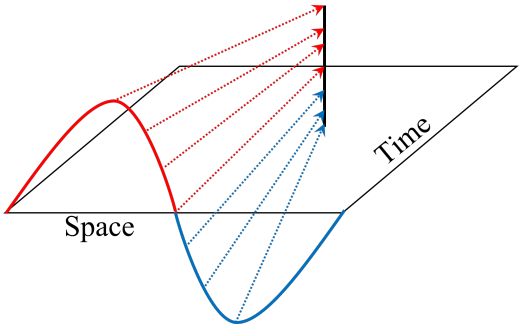

Figure 3.1: Illustrative Diagram of the Intersections of Characteristics of In-viscid Burgers' Equation

In the previous example, denote the time at which the characteristics first intersect by $t = T_b$ and refer it to as the **BREAKING POINT**. Such a scenario is not acceptable for many physical situations. We then hope the solution to the in-viscid Burgers' equation $u_t + (u^2/2)_x = 0$ to be the limits of the solution to the viscid Burgers' equation $u_t + (u^2/2)_x = \varepsilon u_{xx}$ as $\varepsilon \to 0$, rather than the breaking wave solution.

We will obtain solutions with discontinuity at the breaking point (not classical solution any more), and the way we pursue is to find the **WEAK SOLUTION** to the in-viscid Burgers' equation.

E.g., given the CL in a conservative form

$$\begin{cases} \vec{u}_t + \left[\vec{F}(\vec{u})\right]_x = 0, & \mathbb{R} \times (0,\infty) \\ \vec{u}(x,0) = \vec{u}_0(x), & x \in \mathbb{R} \end{cases}$$

Taking integral by a test function with a compact support, vanishing BCs and vanishing terminal time condition at $t = T$, i.e.

$$\phi : \overline{supp(\phi)} \subseteq [a,b] \times [0,T],\ \phi(a,t) = \phi(b,t) = 0,\ \phi(x,T) = 0 \tag{3.3}$$

In general we should have $\phi(x,0)\neq 0$ to represent the non-homogeneous ICs. Then, we have

$$\begin{aligned}0&=\int_0^\infty\int_{-\infty}^\infty\left[u_t+F(u)_x\right]\phi(x,t)dxdt=\int_0^T\int_a^b\left[u_t+F(u)_x\right]\phi(x,t)dxdt\\&=\int_a^b\left[(u\phi)_0^T-\int_0^T u\phi_t dt\right]dx+\int_0^T\left[[F(u)\phi]_a^b-\int_a^b F(u)\phi_x dx\right]dt\\&=-\int_a^b u(x,0)\phi(x,0)dx-\int_a^b\int_0^T u\phi_t dtdx-\int_0^T\int_a^b F(u)\phi_x dxdt\end{aligned}$$

Therefore, we obtain **WEAK FORM** which will generate weak solutions

$$\int_0^\infty\int_{-\infty}^\infty\left[u\phi_t+F(u)\phi_x\right]dxdt+\int_{-\infty}^\infty u_0\phi_0 dx=0,\ u_0=u(x,0),\ \phi_0=\phi(x,0)\tag{3.4}$$

Remark *(Relation between Classical Solutions and Weak Solutions)*

1. *If u is a classical solution to the original problem, then u satisfies the integral (weak) form for any test function ϕ.*
2. *If u is continuously differentiable and satisfies the integral (weak) form for any test function ϕ, then u is a classical solution to the original problem.*

Definition 3.1 (Weak Solution)

If u satisfies the integral (weak) form for any test function $\phi\in\mathcal{C}^1$, u is said to be a weak solution to the original IVP. A weak solution does not need to be a classical solution. ♣

Example 3.2 (In-viscid Burgers' Equation with Intersecting Characteristics)

Consider the following in-viscid Burgers' equation, with the (intersecting) characteristics shown in Fig.3.2.

$$\begin{cases}u_t+\left(u^2/2\right)_x=0\\u_0(x)=\begin{cases}1, & x\leq 0\\0, & x>0\end{cases}\end{cases}$$

A weak solution to the original IVP is

$$u(x,t)=\begin{cases}1, & x\leq t/2\\0, & x>t/2\end{cases}$$

Let us verify that $u(x,t)$ is the weak solution. $\forall\phi(x,t)\in\mathcal{L}^2\cap\mathcal{C}^1$, and $\phi(-\infty,t)=\phi(\infty,t)=\phi(x,\infty)=0$, then

$$\begin{aligned}&\int_0^\infty\int_{-\infty}^\infty[u\phi_t+F(u)\phi_x]dxdt+\int_{-\infty}^\infty u_0\phi(x,0)dx\\&=\underbrace{\int_{-\infty}^0\left[\int_0^\infty\phi_t dt\right]dx+\int_0^\infty\left[\int_{2x}^\infty\phi_t dt\right]dx}_{\text{two regions with } u=1}+\frac{1}{2}\int_0^\infty\left[\int_{-\infty}^{t/2}\phi_x dx\right]dt+\int_{-\infty}^0\phi_0 dx\\&=\int_{-\infty}^0-\phi_0 dx+\int_0^\infty-\phi(x,2x)dx+\frac{1}{2}\int_0^\infty\phi\left(\frac{1}{2}t,t\right)dt+\int_{-\infty}^0\phi_0 dx\\&=\underbrace{\int_0^\infty-\phi(x,2x)dx+\frac{1}{2}\int_0^\infty\phi\left(\frac{1}{2}t,t\right)dt}_{\text{names of integral variables do not matter}}=\int_{x=t/2,x\geq 0}-\phi(x,t)dx+\int_{x=t/2.x\geq 0}\phi(x,t)d\frac{t}{2}=0\end{aligned}$$

The characteristics w.r.t. the weak solution is given in Fig.3.2. Note that a discontinuity along the curve $x=t/2$ is shown and the speed of propagation for such discontinuity is $dx/dt=1/2$. ▲

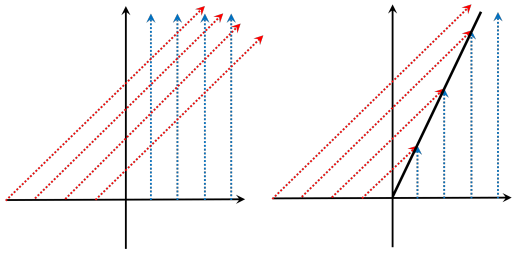

Figure 3.2: Diagram of the In-viscid Burgers' Equation (Left) and the Weak Solution (Right)

Remark *(Shock)*
A discontinuity of a piecewise continuous weak solution is call a **SHOCK** *if the characteristics on both side of the discontinuity impinge on the curve of discontinuity w.r.t. time. In the example above, "In-viscid Burgers' Equation with Intersecting Characteristics", the discontinuity will be a shock if* $a_L > s > a_R$*, where* s *is the propagation speed of the discontinuity along its curve in the direction of increasing time, and* $a_L = F'(u_L)$, $a_R = F'(u_R)$.

Example 3.3 (In-viscid Burgers' Equation with Decentralized Characteristics)
Consider the in-viscid Burger's equation, with the (decentralized) characteristics shown in Fig.3.3.

$$\begin{cases} u_t + \left(u^2/2\right)_x = 0 \\ u_0(x) = \begin{cases} 0, & x \le 0 \\ 1, & x > 0 \end{cases} \end{cases}$$

Obviously, one weak solution to the original IVP is

$$u(x,t) = \begin{cases} 0, & x \le t/2 \\ 1, & x > t/2 \end{cases}$$

The characteristics of this weak solution is shown in Fig.3.3. In this case, $a_L = 0, a_R = 1, s = 1/2$. Thus, the discontinuity is not a shock since the characteristics on both side of the discontinuity do not impinge. Moreover, it is not an entropy solution. Recall that an equivalent name for the entropy solution is "vanishing viscosity solution". Therefore, for the decentralized characteristics, there should be a smooth change between the two adjacent decentralized characteristics, in accordance to "viscosity".

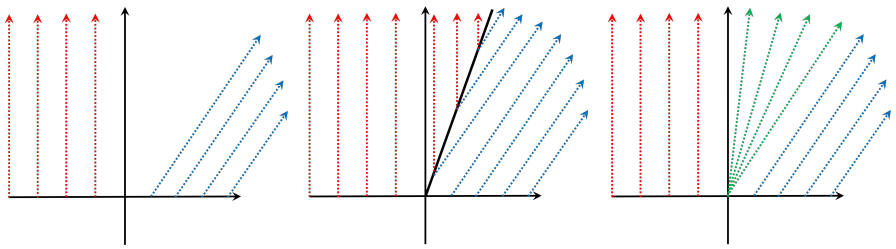

Figure 3.3: Diagram of the In-viscid Burgers' Equation (Left) and the Characteristics of Two Weak Solutions (Middle and Right)

Another weak solution to the original IVP is

$$u(x,t) = \begin{cases} 0, & x < 0 \\ x/t, & 0 \le x \le t \\ 1, & x > t \end{cases}$$

We verify that it is indeed a weak solution.

$$
\begin{aligned}
&\int_0^\infty \int_{-\infty}^\infty [u\phi_t + F(u)\phi_x] dxdt + \int_{-\infty}^\infty u_0\phi(x,0)dx \\
&= \int_0^\infty \left[\int_x^\infty \frac{x}{t}\phi_t dt\right] dx + \int_0^\infty \left[\int_0^x \phi_t dt\right] dx + \frac{1}{2}\int_0^\infty \left[\int_0^t \left(\frac{x}{t}\right)^2 \phi_x dx\right] dt + \frac{1}{2}\int_0^\infty \left[\int_t^\infty \phi_x dx\right] dt + \int_0^\infty \phi_0 dx \\
&= \int_0^\infty \left[-\phi(x,x) + \int_x^\infty \frac{x}{t^2}\phi dt\right] dx + \int_0^\infty [\phi(x,x) - \phi(x,0)]\, dx + \int_0^\infty \left[\frac{1}{2}\phi(t,t) - \int_0^t \frac{x}{t^2}\phi dx\right] dt + \frac{1}{2}\int_0^\infty -\phi(t,t)dt + \int_0^\infty \phi_0 dx \\
&= \left[\int_0^\infty \int_x^\infty \frac{x}{t^2}\phi dt dx - \int_0^\infty \int_0^t \frac{x}{t^2}\phi dx dt\right] + \left[\int_0^\infty -\phi(x,x)dx + \int_0^\infty \phi(x,x)dx + \frac{1}{2}\int_0^\infty (t,t)dt - \frac{1}{2}\int_0^\infty \phi(x,x)dx\right]_{x=t} \\
&\quad + \left[-\int_0^\infty \phi(x,0)dx + \int_0^\infty \phi_0 dx\right] = 0
\end{aligned}
$$

The characteristics of this weak solution is shown in Fig.3.3. As given above, the weak solution is continuous and continuously differentiable except for $x = 0$ and $x = t$. We call it the **RAREFACTION WAVE**.

This example indicates that the weak solutions to a given IVP may not be unique, which is "not good". However, we want to obtain the most "appropriate" one, i.e., the entropy solution. ▲

We are interested in the solutions $u = u(x,t)$ that are smooth except across one or more curves in the $x-t$ plane, where they have jump discontinuities. Recall in the in-viscid Burgers' equation, $F(u) = 1/2$ or 0. Thus, we can verify the following condition.

Proposition 3.2 (Rankine-Hugoniot Jump Condition)

Let C be a smooth curve in the $x-t$ plane, written as $x_c = x_c(t)$, across which u, a weak solution to the IVP, has a jump discontinuity. Let $P = (x_0, t_0), x_0 > 0$ be on C, $s = dx_c(t_0)/dt$, and u_L, u_R be the values to the left and right of P, respectively. Then

$$[u_L - u_R]\frac{dx_c(t_0)}{dt} = F(u_L) - F(u_R)$$

Let $[f] = f_L - f_R$, then the equation above can be written as

$$[u]s = [F],\ s = [F]/[u]$$

where s is the speed of propagation of the discontinuity.

♠

3.2 Entropy Conditions

3.2.1 Entropy Condition I

Recall that the IVP for CLs can be written as $u_t + F(u)_x = 0$, $u(x,0) = u_0(x)$. For the weak solution, for all test functions ϕ with compact supports and $\phi(x,0) \neq 0$ for the ICs, and the weak solution should satisfy

$$\int_0^\infty \int_{-\infty}^\infty [u\phi_t + F(u)\phi_x]\, dxdt + \int_{-\infty}^\infty u_0(x)\phi_0(x)dx = 0,\ \forall\phi$$

The discontinuity of the weak solution may be due to the discontinuity of the ICs, the intersection of the characteristics, or other reasons. Thus, the weak solution may not be unique, while the classical solution does not exist.

In applications, we need to define some approaches to choose the desired solutions. Hence, we may need to jump out of the world of PDE and CLs, and include more physical consideration.

1. Vanishing viscosity solutions:
$$u_t^\varepsilon + F(u^\varepsilon)_x = \varepsilon u_{xx}^\varepsilon$$
where $u^\varepsilon \to u$ as $\varepsilon \to 0$, hence we obtain the weak solution that "looks like" a diffusive process.
2. For steady physical situations, we must ensure the uniqueness of the solution, i.e., some physical information, such as entropy, need to be included.

Definition 3.2 (Entropy Condition I)

The solution to the integral form, i.e., the weak solution $u = u(x,t)$, containing a discontinuity propagation speed $s = dx_c/dt$ is said to satisfy the entropy condition I if
$$F'(u_L) > s > F'(U_R)$$

Entropy condition I is hard to implement in general, because it is difficult to locate the discontinuity. Thus, entropy condition I is not that useful.

Proposition 3.3 (Uniqueness of Entropy Solution with Entropy Condition I)

Suppose that $F(u)$ is convex, and the solution u to the IVP
$$\begin{cases} u_t + F(u)_x = 0, & x \in \mathbb{R}, t > 0 \\ u(x,0) = u_0(x), & x \in \mathbb{R} \end{cases}$$
satisfies the entropy condition I across all jumps. Then u uniquely satisfies the entropy condition I and u is a vanishing viscosity solution.

Definition 3.3 (Entropy Condition I_{nc})

The solution to the integral form, i.e., the weak solution $u = u(x,t)$, containing a discontinuity propagation speed $s = dx_c/dt$ is said to satisfy the entropy condition I_{nc} if
$$\frac{F(u) - F(u_L)}{u - u_L} \geq \frac{F(u_R) - F(u_L)}{u_R - u_L}$$
$\forall u$ between u_L and u_R.

3.2.2 Entropy Condition II

The entropy condition II is established based on the **ENTROPY FUNCTION** $S = S(u)$ and the **ENTROPY FLUX FUNCTION** $\Phi = \Phi(u)$. The two functions are connected via
$$S(u)_t + \Phi(u)_x = 0 \tag{3.5}$$
for smooth solution u with the assumption $S''(u) > 0$. Thus, we have one additional conservative law, i.e., the conservation of entropy. Therefore
$$\begin{cases} S'(u)u_t + \Phi'(u)u_x = 0 \\ u_t + F'(u)u_x = 0 \end{cases} \Rightarrow \Phi'(u) = F'(u)S'(u)$$
That may induce many solutions, s.t. the purpose here is to use the entropy function and the entropy flux function to ensure we can select the correct solution.

Proposition 3.4 (Entropy Function and Vanishing Viscosity Solution)

Suppose u is a vanishing viscosity solution of the IVP shown in Prop.3.3, and it has a related entropy function S, which is convex and $S \geq 0$. Then $S(u)_t + \Phi(u)_x \leq 0$ in a weakly sense. That is, for all positive test functions ϕ, we have the weakly form

$$\int_0^\infty \int_{-\infty}^\infty [S(u)\phi_t + \Phi(u)\phi_x]\, dxdt + \int_{-\infty}^\infty S(u(x,0))\phi(x,0)dx \geq 0$$

Definition 3.4 (Entropy Condition II)

The solution to the integral form, i.e., the weak solution $u = u(x,t)$ is said to satisfy the entropy condition II if there exists an entropy function S and an entropy flux function Φ for which u satisfies the following inequalities in a weakly sense

$$S(u)_t + \Phi(u)_x \leq 0 \Rightarrow \int_0^\infty \int_{-\infty}^\infty [S(u)\phi_t + \Phi(u)\phi_x]\, dxdt + \int_{-\infty}^\infty S(u_0(x))\phi_0(x)dx \geq 0,\ \forall \phi$$

We need to transfer the entropy conditions to a way that can be used. Therefore, in the following proposition, we establish the connection between the entropy condition and the vanishing viscosity solution.

Proposition 3.5 (Entropy Condition and Vanishing Viscosity Solution)

Consider the IVP shown in Prop.3.3, with a convex $F(u)$, and a solution to this IVP contains a weak shock. Suppose the entropy condition II is satisfied for $u = u(x,t)$, with the entropy function S strictly convex, then u is the unique solution to the IVP that satisfies the entropy condition II and the vanishing viscosity solution.

Remark *(Entropy Condition I_a)*

If $F(u)$ is convex, then **ENTROPY CONDITION I$\Longleftrightarrow$ENTROPY CONDITION I$_a$**. *If there exists a constant $E > 0$, s.t.*

$$\frac{u(x+a,t) - u(x,t)}{a} < \frac{E}{t},\ \forall a > 0, t > 0, x \in \mathbb{R} \tag{3.6}$$

1. *If there exists a unique solution that satisfies entropy condition I_a, this solution is a vanishing viscosity solution.*
2. *Suppose the IVP has a related entropy conservation law with a convex S. If u is a piecewise continuous solution, then across each discontinuity u satisfies*

$$s\,[S(u_L) - S(u_R)] - [\Phi(u_L) - \Phi(u_R)] \leq 0$$

where s is the propagation speed of discontinuities. This can be used as an entropy condition.

In summary, the logic path of obtaining the weak solution is ”solution⇒weak solution (jump condition)⇒weak solution (entropy condition)”.

3.2.3 Riemann Problem and Examples

A **RIEMANN PROBLEM** is the following IVP, where $F(u)$ is uniformly convex and $\mathcal{C}^2$. Denote $G = [F'(u)]^{-1}$.

$$\begin{cases} u_t + F(u)_x = 0 \\ u(x,0) = u_0(x) = \begin{cases} u_L, & x < 0 \\ u_R, & x \geq 0 \end{cases}, u_L \neq u_R \end{cases} \tag{3.7}$$

Theorem 3.1 (Entropy Solution to Riemann Problem)

The solutions to Riemann problem can be shown in Fig.3.4.

1. *If $u_L > u_R$,the unique entropy solution to Riemann problem 3.7 is*

$$u(x,t) = \begin{cases} u_L, & \text{if } x/t < s \\ u_R, & \text{if } x/t > s \end{cases}, t > 0, x \in \mathbb{R}, s = \frac{F(u_L) - F(u_R)}{u_L - u_R}$$

2. *If $u_L < u_R$,the unique entropy solution to Riemann problem 3.7 is*

$$u(x,t) = \begin{cases} u_L, & \text{if } x/t < F'(u_L) \\ G(x/t), & \text{if } F'(u_L) < x/t < F'(u_R) \\ u_R, & \text{if } x/t > F'(u_R) \end{cases}, t > 0$$

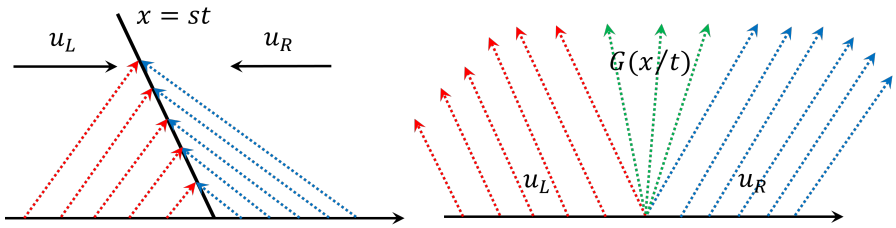

Figure 3.4: Entropy Solutions to Riemann Problem, Shock Wave (Left) and Rarefaction Wave (Right)

Example 3.4 (Burgers' Equation)

Find the entropy solution $u(x,t)$ to the solution

$$\begin{cases} u_t + uu_x = 0 \\ u(x,0) = u_0(x) = \begin{cases} 1, & x < 0 \\ 2, & 0 < x < 1 \\ 0, & x > 1 \end{cases}, \forall t > 0 \end{cases}$$

Check the piecewise IC,

1. $x = 0$: The IC is 1 on the left side and 2 on the right side. For an entropy solution, it can only jump down across shocks. Thus, there exists rarefaction wave.
2. $x = 1$: The IC is 2 on the left side and 0 on the right side. Thus, there exists a shock with the speed given by the Rankine-Hugoniot jump condition, i.e., Prop.3.2. $(2+0)/2 = 1$, since $F(u) = u^2/2$.

So for small $0 \leq t \leq 1$, we have

$$u(x,t) = \begin{cases} 1, & x \leq t \\ x/t, & t < x < 2t \\ 2, & 2t < x < 1+t \\ 0, & x \geq 1+t \end{cases}, \; 0 \leq t \leq 1$$

Then, there exists an interesting phenomenon when the characteristics with speed 2 starting at $x = 0$ hits the shock. That happens with $2t = t + 1 \Rightarrow t = 1$. Right after that, the shock is built from characteristics of the rarefaction fan (region). Then, for the shock curve $x(t)$, we have $x(t = 1) = 2$, and the Rankine-Hugoniot jump condition, i.e., Prop.3.2 implies

$$s = x'(t) = \frac{(x/t)^2/2 - 0}{x/t - 0} = \frac{1}{2} \cdot \frac{x(t)}{t} \Rightarrow x(t) = 2\sqrt{t}$$

Then we have the next part of this solution

$$u(x,t) = \begin{cases} 1, & x \leq t \\ x/t, & t < x < 2\sqrt{t} \\ 0, & x \geq 2\sqrt{t} \end{cases}, \; 1 \leq t \leq 4$$

Then, the next interesting phenomenon occurs when the characteristics with speed 1 hits the characteristics with speed 0. That happens when $t = 2\sqrt{t} \Rightarrow t = 4$, since $t \geq 0$. After that, the shocks travels with speed $(1+0)/2 = 1/2$, and we have the last part of this solution

$$u(x,t) = \begin{cases} 1, & x \leq 2 + t/2 \\ 0, & x > 2 + t/2 \end{cases}, \; t \geq 4$$

▲

3.3 Numerical Solution to Conservation Laws

In the numerical solution to the CLs, we use both the analytic and qualitative information. To optimally solve the PDE, we need to

1. Resolve shocks, contact discontinuities and rarefaction waves (fans), and approximate the propagation speed of shocks, etc.
2. Select the entropy solutions (the vanishing viscosity solutions) from a bundle of weak solutions.

The purpose of studying numerical techniques are to

1. Find the stable and consistent schemes.
2. Include the consideration of dissipation and dispersion, etc.

However, our purpose cannot be easily fulfilled. There are some possibilities of numerical solutions if the numerical schemes are not carefully developed, such as

1. A numerical solution but not weak solution, etc.
2. A weak solution but not a vanishing viscosity solution, etc.

For the numerical schemes for CLs, consider the IVP as

$$u_t + [F(u)]_x = 0$$

Different from the FDM scheme established previously, the FDM for CLs are based on cells rather than nodes, i.e., $I_j = (x_j - \Delta x/2, x_j + \Delta x/2)$, refer to Fig.3.5. And we obtain a piecewise constant function $\overline{U}^n(x)$ on x-space as the approximation of the solution, i.e.,

$$U_j^n \approx \frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} u(x,t_n)dx, \quad \overline{U}^n(x) = U_j^n, \quad x_j - \frac{\Delta x}{2} = x_{j-1/2} \le x \le x_{j+1/2} = x_j + \frac{\Delta x}{2}$$

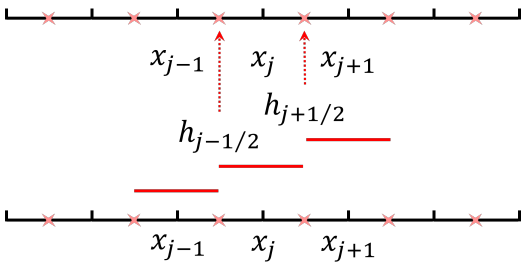

Figure 3.5: Cell Average Used in FDMs for CLs

Then, we say that U_j^n converges to the analytic solution $u = u(x,t_n)$ at $t = t_n$, i.e., $\|\overline{U}^n - u(x,t_n)\|_* \to 0$ as $\Delta x \to 0$. $*$ represents some approximating norm depends on the functional space, e.g., the Lebesgue spaces $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_{1,loc}$, etc.

3.3.1 Consistency of Conservation Laws

In a general form, the CLs have the following form

$$u_t = \mathcal{L}(u) + F \tag{3.8}$$

where F is an non-homogeneous term (source or sink). For the differential scheme, let

$$G_j^n = \mathcal{L}_j^n(U_j^n),\ \vec{U}^{n+1} = \vec{Q}(\vec{U}^n) + \Delta t\vec{G}^n$$

where $\vec{U}^n = \{U_j^n\}$, $\vec{G}^n = \{G_j^n\}$, and $\vec{Q}(\vec{U}^n)$ is the source or sink term.

Definition 3.5 (Consistency I)

A finite difference scheme $G_j^n = \mathcal{L}_j^n(U_j^n)$ is point-wisely consistent with the CL in Eq.3.8 at (x,t) if

$$T_j^n = \mathcal{L}_j^n\left(u(j\Delta x, n\Delta t)\right) - G_j^n \to 0$$

as $\Delta t, \Delta x \to 0$ and $(j\Delta x, n\Delta t) \to (x,t)$, where u is a solution to the CL.

Definition 3.6 (Consistency II)

A finite difference scheme $G_j^n = \mathcal{L}_j^n(U_j^n)$ is consistent with the CL w.r.t. norm $\|\cdot\|$, if a solution of the CL, u, satisfies

$$\Delta t\vec{T}^n = \vec{u}^{n+1} - Q(\vec{u}^n) - \Delta t\vec{G}^n, \text{ where } \left\|\vec{T}^n\right\| \to 0, \text{ as } \Delta x, \Delta t \to 0$$

3.3.2 Conservative Schemes

Following the consistency of the finite difference schemes of a CL, we begin to discuss some conservative schemes. Start with the conservation law in a conservative form in Eq.3.1, i.e., $u_t + [F(u)]_x = 0$. Take

integration from t_n to t_{n+1} on the cell $(x_{j-1/2}, x_{j+1/2})$, we have

$$\int_{t_n}^{t_{n+1}} \int_{x_{j-1/2}}^{x_{j+1/2}} u_t(x,t)dxdt + \int_{t_n}^{t_{n+1}} \int_{x_{j-1/2}}^{x_{j+1/2}} [F(u(x,t))]_x dxdt = 0$$

$$\Rightarrow \quad \int_{x_{j-1/2}}^{x_{j+1/2}} u(x,t_{n+1})dx - \int_{x_{j-1/2}}^{x_{j+1/2}} u(x,t_n)dx + \int_{t_n}^{t_{n+1}} \left[F(u(x_{j+1/2},t))\right] dt - \int_{t_n}^{t_{n+1}} \left[F(u(x_{j-1/2},t))\right] dt = 0$$

Using the definition of the cell average

$$\Delta x \left(U_j^{n+1} - U_j^n\right) + \int_{t_n}^{t_{n+1}} \left[F(u(x_{j+1/2},t))\right] dt - \int_{t_n}^{t_{n+1}} \left[F(u(x_{j-1/2},t))\right] dt = 0$$

Based on the above formulation, we consider the schemes of the form

$$U_j^{n+1} = U_j^n - \nu \left[h_{j+1/2}^n - h_{j-1/2}^n\right], \ \nu = \Delta t/\Delta x$$

Define the function h as the **NUMERICAL FLUX FUNCTION**, and $h_{j+1/2}^n, h_{j-1/2}^n$ approximate the fluxes at $x = x_{j-1/2}, x = x_{j+1/2}$, respectively. Theoretically, we have

$$\Delta t h_{j+1/2}^n = \int_{t_n}^{t_{n+1}} \left[F\left(u\left(x_{j+1/2},t\right)\right)\right] dt, \ \Delta t h_{j-1/2}^n = \int_{t_n}^{t_{n+1}} \left[F\left(u\left(x_{j-1/2},t\right)\right)\right] dt$$

Two approximations can be written as

TWO-POINT APPROXIMATION : $h_{j+1/2}^n = h\left(U_j^n, U_{j+1}^n\right), \ h_{j-1/2}^n = h\left(U_{j-1}^n, U_j^n\right)$

MULTI-POINT APPROXIMATION : $h_{j+1/2}^n = h\left(U_{j-p}^n, ..., U_{j+q}^n\right), \ h_{j-1/2}^n = h\left(U_{j-p-1}^n, ..., U_{j+q-1}^n\right)$

Remark *(Terminology)*

1. *Combining the* $U_j^{n+1} = U_j^n - \nu \left[h_{j+1/2}^n - h_{j-1/2}^n\right]$ *and the numerical flux functions yields a* **CONSERVATIVE DIFFERENCE SCHEME**.
2. *If the numerical flux functions are given in the two-point form, then it is called a* **THREE-POINT SCHEME**.

Now, we present the consistency of the three-point scheme. Recall that "consistency" is to insert the exact solution to the numerical scheme and estimate the difference.

$$\begin{aligned}
&u(x_j,t_{n+1}) - u(x_j,t_n) + \nu\left[h(u(x_j,t_n),u(x_{j+1},t_n)) - h(u(x_{j-1},t_n),u(x_j,t_n))\right] \\
&= u(x_j,t_n) + u_t(x_j,t_n)\Delta t + O(\Delta t^2) - u(x_j,t_n) + \nu\left[h(u(x_j,t_n),u(x_{j+1},t_n)) - h(u(x_{j-1},t_n),u(x_j,t_n))\right] \\
&= u_t(x_j,t_n)\Delta t + O(\Delta t^2) + \nu\left[h(u(x_j,t_n),u(x_j,t_n)) + h_2(u(x_j,t_n),u(x_j,t_n))(u(x_{j+1},t_n) - u(x_j,t_n)) + ...\right] \\
&\quad - \nu\left[h(u(x_j,t_n),u(x_j,t_n)) + h_1(u(x_j,t_n),u(x_j,t_n))(u(x_{j-1},t_n) - u(x_j,t_n)) + ...\right] \\
&= u_t(x_j,t_n)\Delta t + O(\Delta t^2) + \nu\left[h_2(u(x_j,t_n),u(x_j,t_n))(u(x_j,t_n) + u_x(x_j,t_n)\Delta x + O(\Delta x^2) - u(x_j,t_n) + ...\right] \\
&\quad - \nu\left[h_1(u(x_j,t_n),u(x_j,t_n))(u(x_j,t_n) - u_x(x_j,t_n)\Delta x + O(\Delta x^2) - u(x_j,t_n) + ...\right]
\end{aligned}$$

h_1, h_2 represent the partial derivatives w.r.t. the first and the second argument. In the following derivation, we merge the partial derivatives using the chain rule.

$$\begin{aligned}
&u(x_j,t_{n+1}) - u(x_j,t_n) + \nu\left[h(u(x_j,t_n),u(x_{j+1},t_n)) - h(u(x_{j-1},t_n),u(x_j,t_n))\right] \\
&= u_t(x_j,t_n)\Delta t + \nu\left[h_1 + h_2\right] u_x(x_j,t_n)\Delta x + O(\Delta t^2) + O(\Delta t\Delta x) \\
&= u_t(x_j,t_n)\Delta t + \nu\left[h_1(u(x_j,t_n),u(x_j,t_n)) + h_2(u(x_j,t_n),u(x_j,t_n))\right] u_x(x_j,t_n)\Delta x + O(\Delta t^2) + O(\Delta t\Delta x) \\
&= u_t(x_j,t_n)\Delta t + \nu h_x(u(x_j,t_n),u(x_j,t_n))\Delta x + O(\Delta t^2) + O(\Delta t\Delta x) \\
&= \left[u_t(x_j,t_n) + h_x(u(x_j,t_n),u(x_j,t_n))\right]\Delta t + O(\Delta t^2) + O(\Delta t\Delta x) \\
&= \left[u_t + h_x(u,u)\right]_{(x_j,t_n)}\Delta t + O(\Delta t^2) + O(\Delta t\Delta x)
\end{aligned}$$

We observe that the consistency requires that

$$h(u,u) = F(u)$$
$$\text{and} \quad T_j^n = [u_t + h_x(u,u)]_{(x_j,t_n)} + O(\Delta t + \Delta x) = [u_t + F(u)_x]_{(x_j,t_n)} + O(\Delta t + \Delta x)$$

Proposition 3.6 (Consistency)

1. *If $h(u,u) = F(u)$, the three-point difference scheme is consistent with the IVP.*
2. *If $h(u,...,u) = F(u)$, the multi-point difference scheme is consistent with the IVP.*

♠

Remark *The proof of the Prop.3.6, shown in the derivations above, assumes differentiability. The same consistency results with h can be achieved if h is Lipschitz continuous w.r.t. each argument.*

Theorem 3.2 (Lax-Wendroff)

Suppose U_j^n is a discrete solution using a consistent, conservative difference approximation to a given CL as an IVP. If $U_j^n \to u$ w.r.t the $\mathcal{L}_{1,loc}$ norm as $\Delta x, \Delta t \to 0$, then $u = u(x,t)$ is a weak solution to the IVP.

♡

Proof Using the conservative difference scheme

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{h_{j+1/2}^n - h_{j-1/2}^n}{\Delta x} = 0$$

Take the summation with $\phi_j^n = \phi(j\Delta x, n\Delta t)$, as $\phi \in C_0^1$ is the test function.

$$\sum_{n=0}^{\infty}\sum_{j=-\infty}^{\infty} \phi_j^n \frac{U_j^{n+1} - U_j^n}{\Delta t} + \phi_j^n \frac{h_{j+1/2}^n - h_{j-1/2}^n}{\Delta x} = 0$$

$$\xrightarrow{\text{summation by part}} \quad -\sum_{n=0}^{\infty}\sum_{j=-\infty}^{\infty}\left[U_j^{n+1}\frac{\phi_j^{n+1} - \phi_j^n}{\Delta t} + h_{j+1/2}^n \frac{\phi_{j+1}^n - \phi_j^n}{\Delta x}\right] - \sum_{j=-\infty}^{\infty}\frac{U_j^0\phi_j^0}{\Delta t} = 0$$

Multiply $\Delta t \Delta x$ on both sides of the equation and let $\Delta t, \Delta x \to 0$.

$$-\int_0^{\infty}\int_{-\infty}^{\infty}[u\phi_t + h(u,...,u)\phi_x]\,dxdt - \int_{-\infty}^{\infty} u_0\phi_0 dx = 0$$

Using the condition of consistency, $F(u) = h(u,...,u)$. Using the fact $U_j^n \to u$, we have

$$-\int_0^{\infty}\int_{-\infty}^{\infty}[u\phi_t + F(u)\phi_x]\,dxdt - \int_{-\infty}^{\infty} u_0\phi_0 dx = 0$$

Therefore, u is the weak solution. ■

Remark *Thm.3.2 only ensures the solution to be a weak solution. However, using a consistent, conservative difference scheme does not guarantee the vanishing viscosity solution (or the entropy solution). That means, more information is needed.*

Remark *(Some Digressions Regarding to the Proof of Thm.3.2)*

1. **ABEL TRANSFORM** *(summation by part):*
$$\sum_{k=q}^{p} a_k(b_{k+1} - b_k) = a_p b_{p+1} - a_q b_q - \sum_{k=q+1}^{p} b_k(a_k - a_{k-1})$$
2. *Thm.3.2 does not claim that the only way to obtain a weak solution is to use a conservative scheme.*

3. *In general, we would choose conservative schemes using Thm.3.2. The simplest approach is to use* $U_j^{n+1} = U_j^n - \nu\left(h_{j+1/2}^n - h_{j-1/2}^n\right)$ *with some appropriate numerical flux functions.*
4. *We can get a weak solution by using a consistent, conservative difference scheme, based on Thm.3.2. However, we are much interested in an entropy solution.*

Example 3.5 (Some Conservative Schemes)

1. **FTFS (FORWARDS TIME FORWARD SPACE) SCHEME:**

   Consider the FTFS Euler scheme

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{F(U_{j+1}^n) - F(U_j^n)}{\Delta x} = 0$$

   Thus, comparing with the cell average formulation

$$\Delta x(U_j^{n+1} - U_j^n) + \int_{t_n}^{t_{n+1}} \left[F(u(x_{j+1/2}, t))\right] dt - \int_{t_n}^{t_{n+1}} \left[F(u(x_{j-1/2}, t))\right] dt = 0$$

   That is to say we approximate

$$\Delta t h_{j+1/2}^n = \int_{t_n}^{t_{n+1}} \left[F(u(x_{j+1/2}, t))\right] dt \approx \Delta t F(u(x_{j+1})) \quad \Rightarrow \quad h_{j+1/2}^n = F(u(x_{j+1/2})) \approx F_{j+1}^n$$

$$\Delta t h_{j-1/2}^n = \int_{t_n}^{t_{n+1}} \left[F(u(x_{j-1/2}, t))\right] dt \approx \Delta t F(u(x_j)) \quad \Rightarrow \quad h_{j-1/2}^n = F(u(x_{j-1/2})) \approx F_j^n$$

   Thus, FTFS is conservative naturally, i.e.,

$$U_j^{n+1} = U_j^n + \nu\left[h_{j+1/2}^n - h_{j-1/2}^n\right]$$

2. **LAX-WENDROFF SCHEME:**

   To obtain the Lax-Wendroff scheme, choose the approximation

$$h_{i+1/2}^n = \frac{1}{2}\left[F_{j+1}^n + F_j^n\right] - \frac{\nu}{2} A_{j+1/2}\left[F_{j+1}^n - F_j^n\right], \; A_{j+1/2} = F'\left(\frac{U_{j+1}^n + U_j^n}{2}\right)$$

   Then, the Lax-Wendroff scheme can be written as

$$U_j^{n+1} = U_j^n - \nu\left[\frac{1}{2}\left(F_{j+1}^n + F_j^n\right) - \frac{\nu}{2} A_{j+1/2}\left(F_{j+1}^n - F_j^n\right) - \frac{1}{2}\left(F_j^n + F_{j-1}^n\right) + \frac{\nu}{2} A_{j-1/2}\left(F_j^n - F_{j-1}^n\right)\right]$$

   Lax-Wendroff scheme is conservative and three-point.

3. **LAX-FRIEDRICHS SCHEME:**

   To obtain the Lax-Friedrichs scheme, choose the approximation

$$h_{i+1/2}^n = \frac{1}{2}\left[F_{j+1}^n + F_j^n\right] - \frac{1}{2\nu}\left[U_{j+1}^n - U_j^n\right]$$

   Then, the Lax-Friedrichs scheme can be written as

$$\begin{aligned} U_j^{n+1} &= U_j^n - \nu\left[\frac{1}{2}\left(F_{j+1}^n + F_j^n\right) - \frac{1}{2\nu}\left(U_{j+1}^n - U_j^n\right) - \frac{1}{2}\left(F_j^n + F_{j-1}^n\right) + \frac{1}{2\nu}\left(U_j^n - U_{j-1}^n\right)\right] \\ &= U_j^n + \frac{1}{2}\left(U_{j+1}^n - 2U_j^n + U_{j-1}^n\right) - \frac{\nu}{2}\left(F_{j+1}^n + F_{j-1}^n\right) \\ &= \frac{1}{2}\left(U_{j+1}^n + U_{j-1}^n\right) - \frac{\nu}{2}\left(F_{j+1}^n - F_{j-1}^n\right) \end{aligned}$$

4. **MAC-CORMACK SCHEME:**

   Recall that the Mac-Cormack scheme is a two-step Lax-Wendroff scheme. The idea behind Mac-Cormack is the "prediction-correction" procedure.

$$\begin{aligned} &\text{Prediction,} \quad U_j^* = U_j^n - \nu[F(U_{j+1}^n) - F(U_j^n)] \\ &\text{Correction,} \quad U_j^{n+1} = \frac{1}{2}\left[U_j^n + U_j^* - \nu[F(U_j^*) - F(U_{j-1}^*)]\right] \end{aligned}$$

We can choose a very "ugly" numerical fluxes

$$h^n_{j+1/2} = \frac{1}{2}\left[F^n_{j+1} + F(U^*_j)\right],\ h^n_{j-1/2} = \frac{1}{2}\left[F^n_j + F(U^*_{j-1})\right]$$

with $U^*_j = U^n_j - \nu[F(U^n_{j+1}) - F(U^n_j)]$. Then we obtain the Mac-Cormack scheme.

5. **BEAM-WARMING SCHEME:**

To obtain the Beam-Warming scheme, choose the approximation

$$h^n_{j+1/2} = \frac{1}{2}\left(F^n_j + F(U^*_j)\right) + \frac{1}{2}\left(F^n_j - F^n_{j-1}\right),\ U^*_j = U^n_j - \nu\left(F^n_j - F^n_{j-1}\right)$$

Then, the Beam-Warming scheme is

$$\begin{aligned}
U^{n+1}_j &= U^n_j - \nu[h^n_{j+1/2} - h^n_{j-1/2}] \\
&= U^n_j - \nu\left[\frac{1}{2}\left(F^n_j + F^*_j\right) + \frac{1}{2}\left(F^n_j - F^n_{j-1}\right) - \frac{1}{2}\left(F^n_{j-1} + F^*_{j-1}\right) - \frac{1}{2}\left(F^n_{j-1} - F^n_{j-2}\right)\right] \\
&= \frac{1}{2}U^n_j + \left[\frac{1}{2}U^n_j - \frac{\nu}{2}\left(F^n_j - F^n_{j-1}\right)\right] - \frac{\nu}{2}\left(F^*_j - F^*_{j-1}\right) - \frac{\nu}{2}\left(F^n_j - 2F^n_{j-1} + F^n_{j-2}\right) \\
&= \frac{1}{2}\left(U^n_j + U^*_j\right) - \frac{\nu}{2}\left(F^*_j - F^*_{j-1}\right) - \frac{\nu}{2}\left(F^n_j - 2F^n_{j-1} + F^n_{j-2}\right)
\end{aligned}$$

under the assumption that F is positive, indicating we are using the upwind fashion.

▲

3.3.3 Discrete Conservation Law

Recall that using a consistent, conservative difference scheme does not guarantee to get the vanishing viscosity solution, or say the entropy solution. That means, more information is needed. Thus, we shift to the **DISCRETE CONSERVATION LAW**. The analytical form of a CL is

$$\int_a^b u(x,t_2)dx - \int_a^b u(x,t_1)dx = -\left[\int_{t_1}^{t_2} F(u(b,t))dt - \int_{t_1}^{t_2} F(u(a,t))dt\right]$$

Numerically, we expect the conservative scheme to satisfy some analogous form. Therefore, sum the formula $U^{n+1}_j = U^n_j - \nu\left(h^n_{j+1/2} - h^n_{j-1/2}\right)$ over j from j_1 to j_2 and n from n_1 to n_2.

$$\sum_{n=n_1}^{n_2}\sum_{j=j_1}^{j_2} U^{n+1}_j = \sum_{n=n_1}^{n_2}\sum_{j=j_1}^{j_2}\left[U^n_j - \nu\left(h^n_{j+1/2} - h^n_{j-1/2}\right)\right]$$

Rearrange this equation, we have

$$\sum_{j=j_1}^{j_2}\left(U^{n_2+1}_j - U^{n_1}_j\right)\Delta x = -\sum_{n=n_1}^{n_2}\left(h^n_{j_2+1/2} - h^n_{j_1-1/2}\right)\Delta t \tag{3.9}$$

This equation referred to as the **SUMMATION FORM OF A CONSERVATION LAW**.

The CFL condition of the above conservative scheme is shown as follows. Consider the problem

$$\begin{cases} u_t + [F(u)]_x = 0,\ x \in \mathbb{R}, t > 0 \\ u(x,0) = u_0(x),\ x \in \mathbb{R} \end{cases}$$

and a three-point scheme

$$U^{n+1}_j = Q(U^n_{j-1}, U^n_j, U^n_{j+1})$$

For a given point $(j\Delta x, (n+1)\Delta t)$, as in the linear case, the numerical domain of dependence is $[(j-n-$

$1)\Delta x, (j+n+1)\Delta x]$. If u is smooth, $u(x,t) = u_0\left(x - F'(u(x(t),t))t\right)$, i.e., we need

$$\begin{aligned} & (j-n-1)\Delta x \le j\Delta x - F'(u(j\Delta x,(n+1)\Delta t))(n+1)\Delta t \le (j+n+1)\Delta x \\ \Rightarrow \quad & -1 \le -\nu F'(u(j\Delta x,(n+1)\Delta t)) \le 1 \\ \Rightarrow \quad & \nu\left|F'(u(j\Delta x,(n+1)\Delta t))\right| \le 1 \end{aligned}$$

Therefore, it is sufficient to have

$$\nu \max|F'| \le 1$$

And the discrete, nonlinear CFL condition can be given as

$$\nu \max_{j,n}\left|a^n_{j+1/2}\right| \le 1,\ a^n_{j+1/2} = \begin{cases} \dfrac{F_{j+1}-F_j}{U^n_{j+1}-U^n_j}, & \text{if } U^n_{j+1}-U^n_j \neq 0 \\ F'(U^n_j) = F'(U^n_{j+1}), & \text{if } U^n_{j+1}-U^n_j = 0 \end{cases}$$

Remark *(CFL condition)*

1. *For a scheme with the form $U^{n+1}_j = Q(U^n_{j-1}, U^n_j)$, the CFL condition is $0 \le \nu \max F' \le 1$;*
2. *For a scheme with the form $U^{n+1}_j = Q(U^n_j, U^n_{j+1})$, the CFL condition is $-1 \le \nu \max F' \le 0$;*

We finish this section with some analysis based on the entropy. Recall that the entropy function $S = S(u)$ and the entropy flux function $\Phi = \Phi(u)$ satisfy

$$\Phi'(u) = F'(u)S'(u)$$

Both $S = S(u)$ and $\Phi = \Phi(u)$ are not unique.

Definition 3.7 (Numerical Entropy Flux)

The numerical entropy flux function $\Psi^n_{j+1/2}$ based on Φ can be defined in either of the following two ways

$$\begin{aligned} &\textit{Two-point:} && \Psi^n_{j+1/2} = \Psi(U^n_j, U^n_{j+1}) \\ &\textit{Multi-point:} && \Psi^n_{j+1/2} = \Psi(U^n_{j-p}, ..., U^n_{j+q}) \end{aligned}$$

For consistency, we require that $\Psi(U, ..., U) = \Phi(U)$.

♣

Theorem 3.3 (Discrete Entropy Condition II)

Let S and Φ be the entropy function and the entropy flux function for the CL $u_t + [F(u)]_x = 0$, and let $\Phi^n_{j+1/2}$ be a numerical entropy flux function that consistent with the entropy flux function Φ. In addition, suppose that a solution to the difference scheme satisfies the discrete entropy condition

$$S(U^{n+1}_j) \le S(U^n_j) - \nu\left[\Psi^n_{j+1/2} - \Psi^n_{j-1/2}\right]$$

If $U^n_j \to u$ is $\mathcal{L}_{1,loc}$ as $\Delta x, \Delta t \to 0$, then u satisfies the entropy condition II shown in Def.3.4.

♡

3.4 Entropy Scheme and Monotone Scheme

Recall the CL and the corresponding numerical scheme.

$$\begin{aligned} & u_t + [F(u)]_x = 0 \\ & U^{n+1}_j = U^n_j - \nu\left[h^n_{j+1/2} - h^n_{j-1/2}\right] \end{aligned} \tag{3.10}$$

Definition 3.8 (Entropy Scheme, E-Scheme)

The numerical scheme $U_j^{n+1} = U_j^n - \nu\left[h_{j+1/2}^n - h_{j-1/2}^n\right]$ *is called an entropy scheme (E-scheme) if*

1. $h_{j+1/2}^n \le F(U)$ *for all* $U \in [U_j, U_{j+1}]$, *if* $U_j < U_{j+1}$.
2. $h_{j+1/2}^n \ge F(U)$ *for all* $U \in [U_{j+1}, U_j]$, *if* $U_{j+1} < U_j$.

Similar conditions apply for $h_{j-1/2}^n$.

♣

Remark *In general, Checking E-Scheme may be difficult.*

Definition 3.9 (Monotone Scheme, M-Scheme)

A difference scheme with the following form

$$U_j^{n+1} = Q(U_{j-p-1}^n, ..., U_{j+q}^n)$$

is said to be a monotone scheme if the function Q *is a monotonically increasing function w.r.t each of its arguments.*

♣

Example 3.6 (Some M-Schemes)

1. **FTBS (FORWARD TIME AND SPACE) SCHEME:**

   For $u_t + au_x = 0$, consider the scheme

   $$U_j^{n+1} = Q(U_{j_1}^n, U_j^n) = U_j^n - a\nu(U_j^n - U_{j-1}^n)$$

   Then we have,

   $$\frac{\partial Q}{\partial U_{j-1}^n} = a\nu, \ \frac{\partial Q}{\partial U_j^n} = 1 - a\nu$$

   If the CFL condition is satisfied, i.e., $0 \le a\nu \le 1$, then

   $$\frac{\partial Q}{\partial U_{j-1}^n} \ge 0, \ \frac{\partial Q}{\partial U_j^n} \ge 0$$

   Therefore, for the FTBS scheme, the CFL condition is equivalent to the M-Scheme

2. **LAX-FRIEDRICHS SCHEME:**

   Consider the numerical scheme

   $$U_j^{n+1} = Q(U_{j-1}^n, U_j^n, U_{j+1}^n) = \frac{1}{2}\left(U_{j+1}^n + U_{j-1}^n\right) - \frac{\nu}{2}\left(F_{j+1}^n - F_{j-1}^n\right)$$

   Then we have,

   $$\frac{\partial Q}{\partial U_{j-1}^n} = \frac{1}{2} + \frac{1}{2}\nu F'(U_{j-1}^n), \ \frac{\partial Q}{\partial U_j^n} = 0, \ \frac{\partial Q}{\partial U_{j+1}^n} = \frac{1}{2} - \frac{1}{2}\nu F'(U_{j+1}^n)$$

   To make it as an M-Scheme, we need $-1 \le \nu F' \le 1$, which coincide with the CFL condition for the Lax-Friedrichs Scheme.

3. **LAX-WENDROFF SCHEME:**

   For $u_t + au_x = 0$, consider the numerical scheme

   $$U_j^{n+1} = Q(U_{j-1}^n, U_j^n, U_{j+1}^n) = U_j^n - \frac{a\nu}{2}(U_{j+1}^n - U_{j-1}^n) + \frac{a^2\nu^2}{2}(U_{j+1}^n - 2U_j^n + U_{j-1}^n)$$

   Then we have,

   $$\frac{\partial Q}{\partial U_{j-1}^n} = \frac{a\nu}{2} + \frac{a^2\nu^2}{2}, \ \frac{\partial Q}{\partial U_j^n} = 1 - a^2\nu^2, \ \frac{\partial Q}{\partial U_{j+1}^n} = -\frac{a\nu}{2} + \frac{a^2\nu^2}{2}$$

To make it as an M-Scheme, we need $a\nu(1+a\nu) \geq 0, (1-a\nu)(1+a\nu) \geq 0, -a\nu(1-a\nu) \geq 0$. Suppose $a\nu \neq 0$, then the three conditions cannot be met simultaneously. Therefore, the Lax-Wendroff scheme is not an M-Scheme, but the Lax-Wendroff scheme has an accuracy higher than the first order.

▲

Definition 3.10 (Total Variation Decreasing, TVD)

A difference scheme is said to be total variation decreasing (TVD), if the solution produced by the scheme satisfies

$$TV(\vec{U}^{n+1}) \leq TV(\vec{U}^n), \qquad \forall n$$

$$\text{or, } \sum_{j=-\infty}^{\infty} |U_{j+1}^{n+1} - U_j^{n+1}| \leq \sum_{j=-\infty}^{\infty} |U_{j+1}^n - U_j^n|, \quad \forall n$$

where the total variation is denote as

$$TV(\vec{U}^n) = \sum_{j=-\infty}^{\infty} |U_{j+1}^n - U_j^n|$$

♣

Note that when the IC, $u_0(x)$, is smooth, or the **TOTAL VARIATION (TV)** is small, we can usually have $TV(\vec{U}^0) \geq TV(\vec{U}^1) \geq TV(\vec{U}^2) \geq ...$ High order scheme may cause oscillations in the numerical solution, which induces TV increasing, such as the Lax-Wendroff scheme. Note that the Lax-Wendroff scheme does not satisfy the maximum principle either.

Example 3.7 (FTBS Scheme is TVD)

Consider the IVP $u_t + au_x = 0, a > 0$ with the FTBS scheme

$$U_j^{n+1} = U_j^n - a\nu(U_j^n - U_{j-1}^n)$$

Then it can be shown that FTBS Scheme is TVD as follows

$$\begin{aligned} TV(\vec{U}^{n+1}) &= \sum_{j=-\infty}^{\infty} |U_{j+1}^{n+1} - U_j^{n+1}| = \sum_{j=-\infty}^{\infty} |(U_{j+1}^n - U_j^n) - a\nu(U_{j+1}^n - 2U_j^n + U_{j-1}^n)| \\ &= \sum_{j=-\infty}^{\infty} |(1-a\nu)(U_{j+1}^n - U_j^n) + a\nu(U_j^n - U_{j-1}^n)| \\ &\leq (1-a\nu) \sum_{j=-\infty}^{\infty} |U_{j+1}^n - U_j^n| + a\nu \sum_{j=-\infty}^{\infty} |U_j^n - U_{j-1}^n| = TV(\vec{U}^n) \end{aligned}$$

The absolute value is a convex function, and assume that the CFL condition is satisfied, i.e., $0 \leq a\nu \leq 1$. ▲

Definition 3.11 (Essentially Non-oscillatory, ENO)

A difference scheme is said to be essential non-oscillatory (ENO), if the solution produced by the scheme satisfies

$$\sum_{j=-\infty}^{\infty} |U_{j+1}^{n+1} - U_j^{n+1}| \leq \sum_{j=-\infty}^{\infty} |U_{j+1}^n - U_j^n| + O(\Delta x^P)$$

For $\forall n \geq 0$ and some P.

♣

Definition 3.12 (Equivalent Forms of Difference Schemes)

1. **Incremental Form, I-Form**

$$\begin{aligned} U_j^{n+1} &= U_j^n + C_{j+1/2}^n \Delta_{+x} U_j^n - D_{j-1/2}^n \Delta_{-x} U_j^n \\ &= U_j^n + C_{j+1/2}^n (U_{j+1}^n - U_j^n) - D_{j-1/2}^n (U_j^n - U_{j-1}^n) \end{aligned}$$

where $C_{j+1/2}^n$ and $D_{j-1/2}^n$ depend on U_j^n and its neighbors. The way to transform between the I-form and the conservative form is shown as follows

$$h_{j+1/2} = \begin{cases} F(U_j^n) - \dfrac{1}{\nu} C_{j+1/2}^n \Delta_{+x} U_j^n \\ F(U_{j+1}^n) - \dfrac{1}{\nu} D_{j+1/2}^n \Delta_{+x} U_j^n \end{cases} \iff \begin{cases} C_{j+1/2}^n = -\nu \dfrac{h_{j+1/2}^n - F_j^n}{U_{j+1}^n - U_j^n} \\ D_{j+1/2}^n = -\nu \dfrac{h_{j+1/2}^n - F_{j+1}^n}{U_{j+1}^n - U_j^n} \end{cases}$$

To show the equivalence between the I-form and the conservative form, just plug everything into the conservative form $U_j^{n+1} = U_j^n - \nu(h_{j+1/2}^n - h_{j-1/2}^n)$.

2. **Q-Form**

$$\begin{aligned} U_j^{n+1} &= U_j^n - \frac{\nu}{2}(F_{j+1}^n - F_{j-1}^n) - \frac{1}{2}\Delta_{+x}(Q_{j-1/2}^n \Delta_{-x} U_j^n) \\ &= U_j^n - \frac{\nu}{2}(F_{j+1}^n - F_{j-1}^n) - \frac{1}{2}\left[Q_{j+1/2}^n (U_{j+1}^n - U_j^n) - Q_{j-1/2}^n (U_j^n - U_{j-1}^n)\right] \end{aligned}$$

where Q is called the numerical viscosity coefficient since it associates with the second order difference. The way to transform between Q-form and conservative form is shown as follows

$$h_{j+1/2}^n = \frac{1}{2}(F_j^n + F_{j+1}^n) - \frac{1}{2\nu} Q_{j+1/2}^n \Delta_{+x} U_j^n$$

To show the equivalence between the Q-form and the conservative form, just plug everything into the conservative form $U_j^{n+1} = U_j^n - \nu(h_{j+1/2}^n - h_{j-1/2}^n)$.

3. **I-Form$\Longleftrightarrow$Q-Form**

   (a). *I-Form$\Rightarrow$Q-Form:*

$$\begin{cases} C_{j+1/2}^n = \dfrac{1}{2}(Q_{j+1/2}^n - \nu a_{j+1/2}^n) \\ D_{j+1/2}^n = \dfrac{1}{2}(Q_{j+1/2}^n + \nu a_{j+1/2}^n) \end{cases}, \qquad a_{j+1/2}^n = \begin{cases} \dfrac{\Delta_{+x} F_j^n}{\Delta_{+x} U_j^n}, & \Delta_{+x} U_j^n \neq 0 \\ F'(U_j^n), & \Delta_{+x} U_j^n = 0 \end{cases}$$

   (b). *Q-Form$\Rightarrow$I-Form:*

$$\begin{cases} Q_{j+1/2}^n = \nu \dfrac{F_j^n + F_{j+1}^n - 2h_{j+1/2}^n}{\Delta_{+x} U_j^n} \\ Q_{j+1/2}^n = C_{j+1/2}^n + D_{j+1/2}^n \end{cases}$$

♣

Proposition 3.7 (E-Scheme with CFL Condition Implies Discrete Entropy Condition)

Consider the IVP $u_t + F(u)_x = 0$ as a CL with the associate entropy function S and entropy flux function Φ. Let U_j^n be an approximation to the solution, obtained by an E-scheme

$$U_j^{n+1} = U_j^n - \nu\left[h_{j+1/2}^n - h_{j-1/2}^n\right]$$

Then, if U_j^n satisfies the condition

$$\nu\left|(F_j^n - h_{j+1/2}^n) + (F_{j+1}^n - h_{j+1/2}^n)\right| \leq \frac{1}{2}\left|U_{j+1}^n - U_j^n\right|$$

i.e., the CFL condition or equivalently, the condition $|Q_{j+1/2}^n| \leq 1/2$, see Def.3.12. Then, there exists a

numerical flux function $\Phi^n_{j+1/2}$, s.t., U^n_j satisfies the discrete entropy condition

$$S(U^{n+1}_j) \leq S(U^n_j) - \nu \left[\Phi^n_{j+1/2} - \Phi^n_{j-1/2}\right]$$

Remark *If the solution of the E-scheme converges to a function u as $\Delta x, \Delta t \to 0$, then u will be a weak solution and a vanishing viscosity solution (entropy solution) of the CL $u_t + F(u)_x = 0$. We cannot ask for much more.*

E-schemes admit many nice properties in that they will not introduce dispersive wiggles into the solution. So we claim some of the properties in the following propositions.

Proposition 3.8 (TVD of I-Forms)

Consider a difference scheme in I-form, if $C^n_{j+1/2} \geq 0$, $D^n_{j+1/2} \geq 0$, and $C^n_{j+1/2} + D^n_{j+1/2} \leq 1$, for $\forall j$, then the scheme is TVD.

Proof Recall that the I-form

$$U^{n+1}_j = U^n_j + C^n_{j+1/2}(U^n_{j+1} - U^n_j) - D^n_{j-1/2}(U^n_j - U^n_{j-1})$$

Then, plug this into the TV formulation

$$\begin{aligned}
&TV(\vec{U}^{n+1}) = \sum_{j=-\infty}^{\infty} |U^{n+1}_{j+1} - U^{n+1}_j| \\
&= \sum_{j=-\infty}^{\infty} \left|U^n_{j+1} + C^n_{j+3/2}(U^n_{j+2} - U^n_{j+1}) - D^n_{j+1/2}(U^n_{j+1} - U^n_j) - U^n_j - C^n_{j+1/2}(U^n_{j+1} - U^n_j) + D^n_{j-1/2}(U^n_j - U^n_{j-1})\right| \\
&\leq \sum_{j=-\infty}^{\infty} C^n_{j+3/2}|U^n_{j+2} - U^n_{j+1}| + \sum_{j=-\infty}^{\infty} (1 - C^n_{j+1/2} - D^n_{j+1/2})|U^n_{j+1} - U^n_j| + \sum_{j=-\infty}^{\infty} D^n_{j-1/2}|U^n_{j+2} - U^n_{j+1}| \\
&= \sum_{j=-\infty}^{\infty} |U^n_{j+1} - U^n_j| = TV(\vec{U}^n)
\end{aligned}$$

■

Proposition 3.9 (TVD of E-Schemes)

A conservative E-scheme that satisfies the following condition is TVD.

$$\nu \left|(F^n_j - h^n_{j+1/2}) + (F^n_{j+1} - h^n_{j+1/2})\right| \leq \left|\Delta_{+x} U^n_j\right|$$

Proof Use the relation

$$C^n_{j+1/2} = -\nu \frac{h^n_{j+1/2} - F^n_j}{U^n_{j+1} - U^n_j},\ D^n_{j+1/2} = -\nu \frac{h^n_{j+1/2} - F^n_{j+1}}{U^n_{j+1} - U^n_j}$$

Based on the definition of the E-scheme, Def.3.8, we have $C^n_{j+1/2} \geq 0$, $D^n_{j+1/2} \geq 0$. Combining the relation above and the requirement in the proposition, we have $C^n_{j+1/2} + D^n_{j+1/2} \leq 1$. Then, the proof follows Prop.3.8. ■

Proposition 3.10 (First Order Accuracy of E-Schemes)

E-scheme is at most first order accuracy.

Remark *Although E-scheme is only of the first order accuracy, it is good enough for many applications.*

Proposition 3.11 (Sufficient Condition for E-scheme)

Three-point, conservative, M-schemes are E-schemes.

♠

Proof First, recall that the requirements for the E-scheme, as listed in Def.3.8, are

1. $h^n_{j+1/2} \leq F(U)$ for all $U \in [U_j, U_{j+1}]$, if $U_j < U_{j+1}$.
2. $h^n_{j+1/2} \geq F(U)$ for all $U \in [U_{j+1}, U_j]$, if $U_{j+1} < U_j$.

for a conservative scheme $U^{n+1}_j = U^n_j - \nu \left[h^n_{j+1/2} - h^n_{j-1/2}\right]$.

Second, recall that the three-point can be written as

$$U^{n+1}_j = Q = U^n_j - \nu \left[h(U^n_{j+1}, U^n_j) - h(U^n_j, U^n_{j+1})\right]$$

use the monotone property (recall that h_1, h_2 are partial derivatives with respect to the first and the second arguments).

$$\begin{cases} \dfrac{\partial Q}{\partial U^n_{j+1}} = -\nu h_1 \geq 0 \\ \dfrac{\partial Q}{\partial U^n_j} = 1 - \nu h_2 + \nu h_1 \geq 0 \Rightarrow -\dfrac{1}{\nu} \leq h_1 \leq 0 \\ \dfrac{\partial Q}{\partial U^n_{j+1}} = -\nu h_2 \geq 0 \end{cases}$$

Then, we can compare $F(U^n_j) = h(U^n_j, U^n_j)$ and $h(U^n_{j+1}, U^n_j)$ by using $-1/\nu \leq h_1 \leq 0$, and the result follows. ■

Proposition 3.12 (TVD of M-Schemes)

Conservative M-schemes are TVD.

♠

Theorem 3.4 (Discrete Entropy Condition of M-Scheme)

A conservative M-scheme satisfies the discrete entropy condition

$$S(U^{n+1}_j) \leq S(U^n_j) - \nu \left[\Phi^n_{j+1/2} - \Phi^n_{j-1/2}\right]$$

with entropy function $S(u) = |u - c|$, entropy flux function $\Phi = sign(u - c)|F(u) - F(c)|$, and the induced numerical entropy flux function is

$$\psi^n_{j+1/2} = h^n_{j+1/2}\left(\widetilde{U}^n_{j-p}, ..., \widetilde{U}^n_{j+q}\right) - h^n_{j+1/2}\left(\widetilde{\widetilde{U}}^n_{j-p}, ..., \widetilde{\widetilde{U}}^n_{j+q}\right)$$

where $\widetilde{U}^n_j = \max\{c, U^n_j\}$, $\widetilde{\widetilde{U}}^n_j = \min\{c, U^n_j\}$, and Φ^n_{j+1} is consistent with Φ.

♡

Remark *Thm.3.4 implies that if we have a conservative M-scheme for which U^n_j converges to u as $\Delta x, \Delta t \to 0$ in a $\mathcal{L}_{1,loc}$ space, then u will be an entropy solution, a.k.a. the vanishing viscosity solution.*

Proposition 3.13 (Truncation Error of Conservative Schemes)

The truncation error of a conservative finite difference scheme of the form $U_j^{n+1} = a(U_{j-p+1}^n, ..., U_{j+q}^n)$ *is given by*

$$\Delta t T_j^n = -(\Delta t)^2 \left\{[q(u)u_x]_x\right\}_{u=U_j^n} + O\left(\Delta t(\Delta t^2 + \Delta x^2)\right)$$

where $u = u(x,t)$ *is a solution to the CL* $u_x + F(u)_x = 0$*, and*

$$q(u) = \frac{1}{2}\left[\frac{1}{\nu}\sum_{-(p+1)}^{q} j^2 Q_j(u, ..., u) - [F'(u)]^2\right]$$

Proposition 3.14 (First Order Accuracy of M-Schemes)

M-schemes are almost always only of the first order accuracy.

Remark *Prop.3.14 implies that if we need high-order schemes, we should consider neither linear scheme nor three-point schemes. And we will see it is impossible to obtain a TVD scheme of a second order accuracy.*

Proposition 3.15 (First Order Accuracy of Linear TVD Schemes)

Linear TVD difference schemes are at most first order accurate.

Proof Consider a linear finite difference scheme $U_j^n = \sum_{k=-p}^{q} a_k U_{j+k}^n$. Based on TVD, we can prove that $a_k \geq 0$. Hence, a linear TVD scheme is a M-scheme. Thus, it has a first order accuracy at most. ■

Remark *(Summary of Conservation and Monotone)*

Based on the definition of entropy conditions and the discussion in this section, we arrive the following results

1. *Conservative scheme* $\Rightarrow$ *weak solution.*
2. *Conservative M-scheme* $\Rightarrow$ *entropy solution.*
3. *M-schemes are simple but of the first order accuracy at most.*

Definition 3.13 (Incremental TVD)

A difference scheme in I-form is said to be incremental TVD if

$$C_{j+1/2}^n \geq 0,\ D_{j+1/2}^n \geq 0,\ C_{j+1/2}^n + D_{j+1/2}^n \leq 1$$

Remark *The definition of incremental TVD, in Def.3.13, is just for convenient. There may be other TVD schemes that do not satisfy the incremental TVD condition. The name "incremental TVD" can be consider as the "TVD of an incremental form (I-form). Therefore, when we see the definition, it looks like the requirements for the TVD with respect to the I-form, rather than the general TVD defintion shown in Def.3.10.*

Proposition 3.16 (Sufficient Condition for Incremental TVD)

A conservative scheme is incremental TVD iff the numerical viscosity coefficient Q *satisfies,*

$$\nu\left|a_{j+1/2}^n\right| \leq Q_{j+1/2}^n \leq 1, \quad \forall j$$

where

$$a_{j+1/2}^n = \begin{cases} \dfrac{\Delta_{+x} F_j^n}{\Delta_{+x} U_j^n}, & \Delta_{+x} U_j^n \neq 0 \\ F'(U_j^n), & \Delta_{+x} U_j^n = 0 \end{cases}$$

Proof Recall the definition of $C_{j+1/2}^n$ and $D_{j+1/2}^n$ in the I-form

$$C_{j+1/2}^n = \frac{1}{2}\left(Q_{j+1/2}^n - \nu a_{j+1/2}^n\right), \; D_{j+1/2}^n = \frac{1}{2}\left(Q_{j+1/2}^n + \nu a_{j+1/2}^n\right)$$

Then, the proof follows the definition of incremental TVD of the I-form. ■

Proposition 3.17 (First Order Accuracy with Incremental TVD, I)

A three-point, conservative scheme, which is incremental TVD, is at most first order accurate.

Proof We start with the I-form. Using the definition of incremental TVD shown in Def.3.13, the monotone property of the scheme can be derived. Therefore, the scheme is of the first order accuracy at most. ■

Proposition 3.18 (First Order Accuracy with with Incremental TVD, II)

At smooth extrema that are not sonic points, an scheme with incremental TVD is at most first order accurate.

3.4.1 Godunov Monotone Scheme

Although monotone schemes are of the first order accuracy at most, they can usually serve as the building box of high-order schemes. An important monotone scheme is known as the **Godunov Scheme**. Consider the Burgers' equation shown in Eq.2.7 with an IC $u = u_0$, and we recall it here

$$u_t + \left[\frac{1}{2}u^2\right]_x = 0, \; F'(u) = u, \; u = u_0$$

After some time period t, the shape of the solution is shown in Fig.3.6. As time processes, the solution at or near A and B will travel faster than the solution at or near C. Does there exist a numerical scheme that can capture the features of the solution? We will need to introduce the idea of Godunov scheme in Fig.3.6.

The idea of Godunov scheme is

1. At t_0, approximate U^0 as a piecewise constant step function. Then solve the CL as a sequence of Riemann problems locally to $t_1 \Rightarrow U^1$.
2. At t_1, approximate U^1 as a piecewise constant step function. Then solve the CL as a sequence of Riemann problems locally to $t_2 \Rightarrow U^2$.
3. At t_2, approximate U^2 as a piecewise constant step function. Then solve the CL as a sequence of Riemann problems locally to $t_3 \Rightarrow U^3$.
4. Repeat the procedure as t approaches t_F.

Thus, Godunov scheme converts the CL as a series of local Riemann problems, and the exact solution to those Riemann problems in Thm.3.1 can be used locally.

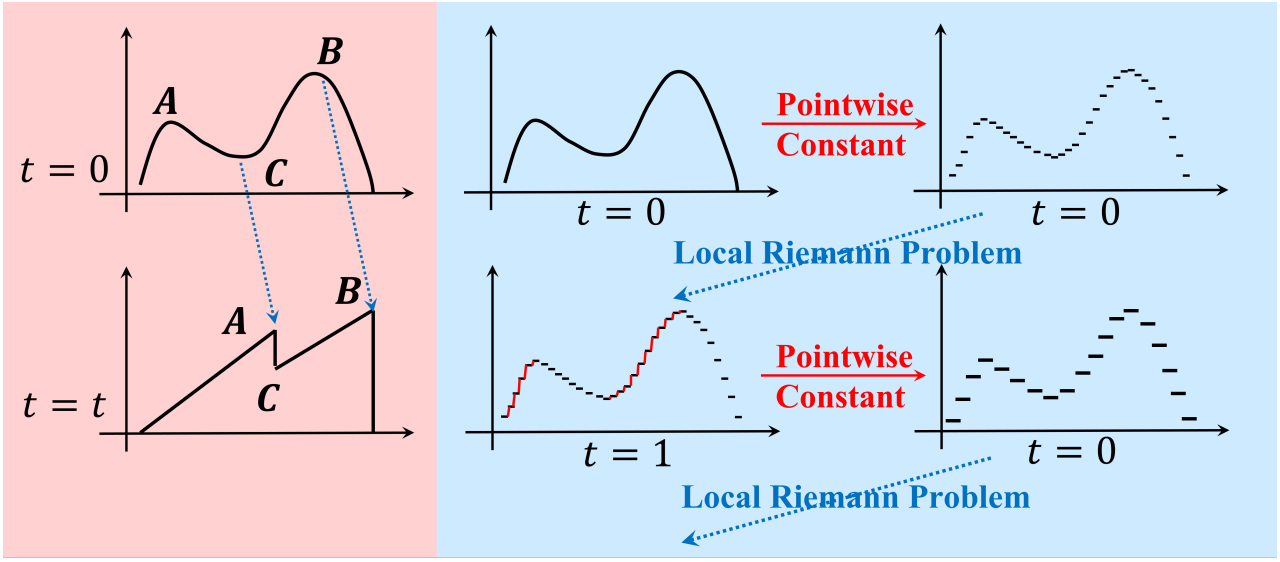

Figure 3.6: The Solution to Burgers' Equation (Left) and The Idea of Godunov Monotone Scheme (Right)

In general, consider the CL

$$\begin{cases} u_t + [F(u)]_x = 0,\ x \in \mathbb{R}, t > 0 \\ u(x,0) = u_0(x) \end{cases}$$

Let $\overline{U}^n(x)$ be the piecewise constant approximation to the solution at $t_n = n\Delta t$, i.e.,

$$\overline{U}^n(x) = U_j^n, \text{ if } x \in \left(x_{j-1/2}, x_{j+1/2}\right)$$

Potential discontinuities of $\overline{U}^n(x)$ will occur at $x_{j\pm1/2}$. Then, with the given $\overline{U}^n(x)$, let $\overline{U}(x,t)$ be the solution to CL with $\overline{U}(x,t_n) = \overline{U}^n(x)$ for $t > t_n$. We want to use $\overline{U}(x,t)$ to approximate the solution at t_{n+1}, i.e., U_j^{n+1} or $\overline{U}^{n+1}$

$$U_j^{n+1} = \frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} \overline{U}(x, t_{n+1})dx$$

Then, $\overline{U}^{n+1}(x)$, which is exactly the same as U_j^{n+1} at x_j, is the piecewise constant function, and the next loop can be started with $\overline{U}^{n+1}(x)$.

Locally, the Riemann problems associated with the cells centered at $x_{j-1/2}$ for $j = -\infty, ..., \infty$ can be shown as follows. Note that Fig.3.5 shows the averaging cell $\left(x_{j-1/2}, x_{j+1/2}\right)$, but here the cells should be $[x_{j-1}, x_j]$, where one potential discontinuity appears right at $x_{j-1/2}$

$$\begin{cases} u_t + [F(u)]_x = 0,\ x \in [x_{j-1}, x_j], t > t_n \\ u(x,t_n) = \begin{cases} u_{j-1}^n, & \text{if } x < x_{j-1/2} \\ u_j^n, & \text{if } x > x_{j-1/2} \end{cases} \end{cases}$$

Then, the series of local Riemann problems can be connected to formulate a Riemann problem over the whole domain, with the piecewise IC at $t = t_n$.

$$\begin{cases} u_t + [F(u)]_x = 0,\ x \in \mathbb{R}, t > t_n \\ u(x,t_n) = \begin{cases} u_{j-1}^n, & \text{if } x < x_{j-1/2} \\ u_j^n, & \text{if } x > x_{j-1/2} \end{cases} \end{cases}$$

Integrate the connected local Riemann problem for $x \in (x_{j-1/2}, x_{j+1/2})$, $t = t_n \to t_{n+1}$, and $\overline{U}^n(x)$ is the IC,

$$\int_{x_{j-1/2}}^{x_{j+1/2}} \left(\overline{U}(x, t_{n+1}) - \overline{U}(x, t_n)\right) dx + \int_{t_n}^{t_{n+1}} \left[F(\overline{U}(x_{x+1/2}, t)) - F(\overline{U}(x_{x-1/2}, t))\right] dt = 0$$

Then, we obtain

$$U_j^{n+1} = U_j^n - \frac{1}{\Delta x} \int_{t_n}^{t_{n+1}} \left[F(\overline{U}(x_{x+1/2}, t)) - F(\overline{U}(x_{x-1/2}, t))\right] dt$$

Note that F, or actually $\overline{U}(x_{x\pm1/2}, t)$ at $t = t_n$, could be a jump. But as $t > t_n$, the jump will be moved as a shock or vanished as a rarefaction wave.

We can show that Godunov scheme is conservative by defining a numerical flux,

$$h_{j\pm1/2} = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \left[F(\overline{U}(x_{j\pm1/2}, t))\right] dt$$

$$\Rightarrow \quad U_j^{n+1} = U_j^n - \nu[h_{j+1/2}^n - h_{j-1/2}^n]$$

Recall that a numerical flux function is consistent with the flux if $h(U, U) = F(U)$, and that is also a necessary condition for the consistency of Godunov scheme.

Proposition 3.19 (Local Riemann Problem Implies Global Entropy Solution)

If the entropy solution to the local Riemann problem are chosen in the construction of Godunov scheme, then there exists a numerical entropy flux function for which Godunov scheme will satisfy the discrete entropy condition. If the numerical solution converges to a function u as $\Delta t, \Delta x \to 0$, then u will be the entropy solution to the CL.

♠

Remark Prop.3.19 implies that it is important to use the entropy solution to the local Riemann problem, shown in Thm.3.1. Then we can obtain an ideal solution to the global CL problem.

3.5 High Resolution Schemes

Up to now, we have three requirements or objectives to a conservation law, which are ensuring the conservative property, obtaining the entropy solution, and resolving the shocks. To give a solution with a higher order accuracy, we will need to ensure some good properties or schemes:

1. Conservative property: or we need TVD (or relaxed TVD requirements with ENO scheme if necessary).
2. Entropy conditions.
3. Nonlinear schemes or schemes with the second or higher order accuracy, but no three-point schemes.

We will introduce three approaches

1. Flux-limiter methods.
2. Slope-limiter methods.
3. Modified-flux methods.

3.5.1 Flux-Limiter Method

We write the numerical flux function of the scheme as

$$h_{j+1/2}^n = h_{L,j+1/2}^n + \phi_j^n \left[h_{H,j+1/2}^n - h_{L,j+1/2}^n\right] \tag{3.11}$$

To understand Eq.3.11, we need to first understand why we need both low order schemes and high order schemes.

1. With high order schemes, we can obtain solutions with a high accuracy.
2. With low order schemes, we can obtain solutions with nice properties, such as monotone, TVD, and etc.

Thus, we have a rule of thumb

1. If the exact solution is good and smooth, we prefer the high order schemes.
2. If the exact solution has shocks, we prefer the low order schemes at or near the shocks, where the high order schemes may produce some oscillations.

In Eq.3.11, h_L, h_H are the numerical flux functions of a low order scheme and a high order scheme, respectively. ϕ_j^n is a coefficient to be determined. Thus, the combined scheme leverages with smooth property of the low order schemes and the accuracy of the high order schemes.

Two extreme (degraded) cases of Eq.3.11 are

1. $\phi_j^n \to 1 \Rightarrow$ high order schemes in smooth regions.
2. $\phi_j^n \to 0 \Rightarrow$ low order schemes near steep gradient or discontinuities.

Alternatively, we also write Eq.3.11 in the form

$$h_{j+1/2}^n = h_{H,j+1/2}^n - (1-\phi_j^n)\left[h_{H,j+1/2}^n - h_{L,j+1/2}^n\right] \tag{3.12}$$

Example 3.8 (One-way Wave Equation)
Consider the one-way wave equation

$$u_t + au_x = 0, a > 0$$

Let

$$\begin{aligned} h_{j+1/2}^n &= aU_j^n + \frac{1}{2}a(1-a\nu)\Delta_{+x}U_j^n \\ &= aU_j^n + \left[aU_j^n + \frac{1}{2}a(1-a\nu)\Delta_{+x}U_j^n - aU_j^n\right] = h_{L,j+1/2}^n + \left[h_{H,j+1/2}^n - h_{L,j+1/2}^n\right] \end{aligned}$$

We observe that the two numerical fluxes are

$$\begin{cases} h_{L,j+1/2}^n = aU_j^n, & \text{FTBS Scheme} \\ h_{H,j+1/2}^n = aU_j^n + \frac{1}{2}a(1-a\nu)\Delta_{+x}U_j^n, & \text{Lax-Wendroff Scheme} \end{cases}$$

In general, we can write

$$h_{j+1/2}^n = h_{L,j+1/2}^n + \phi_j^n\left[h_{H,j+1/2}^n - h_{L,j+1/2}^n\right] = aU_j^n + \phi_j^n\frac{1}{2}a(1-a\nu)\Delta_{+x}U_j^n$$

▲

In Eq.3.11, ϕ_j^n is referred to as the **FLUX-LIMITER**, chosen to be non-negative. Our desire is to make $\phi_j^n \to 1$ in smooth regions of the solution and $\phi_j^n \to 0$ near discontinuities. A common approach is to write ϕ_j^n as $\phi_j^n = \phi(\theta_j^n)$, where θ_j^n is referred to as the **SMOOTHNESS PARAMETER**, and will be define as

$$\theta_j^n = \frac{\Delta_{-x}U_j^n}{\Delta_{+x}U_j^n} \tag{3.13}$$

That means when the solution pattern does not change a lot, we have $\theta_j^n \to 1$, and otherwise, θ_j^n can be quite different from 1.

Remark *(A Complete Version of One-way Wave Scheme)*
In the one-way wave equation, we can write the numerical flux into the following cases based on the smoothness

parameter

$$
\begin{aligned}
&U_j^{n+1} = U_j^n - \nu\left[h_{j+1/2}^n - h_{j-1/2}^n\right] \\
=&U_j^n - \nu\left[aU_j^n + \phi_j^n\frac{1}{2}a(1-a\nu)\Delta_{+x}U_j^n - aU_{j-1}^n - \phi_{j-1}^n\frac{1}{2}a(1-a\nu)\Delta_{+x}U_{j-1}^n\right] \\
=&U_j^n - a\nu\Delta_{-x}U_j^n - \frac{1}{2}a\nu(1-a\nu)\phi_j^n\Delta_{+x}U_j^n + \frac{1}{2}a\nu(1-a\nu)\phi_{j-1}^n\Delta_{+x}U_{j-1}^n \\
=&U_j^n - a\nu\Delta_{-x}U_j^n - \frac{1}{2}a\nu(1-a\nu)\Delta_{-x}\left(\phi_j^n\Delta_{+x}U_j^n\right) \\
=&\begin{cases}
U_j^n - \nu\left[aU_j^n - aU_{j-1}^n\right], & \phi = 0 \\
U_j^n - \nu\left[aU_j^n + \frac{1}{2}a(1-a\nu)\Delta_{+x}U_j^n - aU_{j-1}^n - \frac{1}{2}a(1-a\nu)\Delta_{+x}U_{j-1}^n\right], & \phi = 1 \\
U_j^n - \nu\left[aU_j^n + \frac{\Delta_{-x}U_j^n}{\Delta_{+x}U_j^n}\frac{1}{2}a(1-a\nu)\Delta_{+x}U_j^n - aU_{j-1}^n - \frac{\Delta_{-x}U_{j-1}^n}{\Delta_{+x}U_{j-1}^n}\frac{1}{2}a(1-a\nu)\Delta_{+x}U_{j-1}^n\right], & \phi(\theta) = \theta
\end{cases}
\end{aligned}
$$

We observe that the schemes corresponding to the three θ cases are the upwind scheme, Lax-Wendroff scheme and beam-warming scheme.

Now, we have understood the motivation of the flux-limiter method. The goal is to choose ϕ, s.t., the scheme is TVD and of the second order accuracy. Recall the definition of incremental TVD for an I-form, Def.3.13, where if the scheme of an I-form is incremental TVD, then it is TVD. Note that the incremental TVD in Def.3.13 is a short-cut to determine TVD with an I-form. We first write the flux-limiter scheme (FTBS and Lax-Wendroff) into an I-form

$$
\begin{aligned}
U_j^{n+1} &= U_j^n - \frac{1}{2}a\nu(1-a\nu)\phi_j^n\Delta_{+x}U_j^n - \left[a\nu - \frac{1}{2}a\nu(1-a\nu)\phi_{j-1}^n\right]\Delta_{-x}U_j^n \\
C_{j+1/2}^n &= -\frac{1}{2}a\nu(1-a\nu)\phi_j^n \\
D_{j-1/2}^n &= a\nu - \frac{1}{2}a\nu(1-a\nu)\phi_{j-1}^n
\end{aligned}
$$

If the CFL condition is satisfied, we will need $0 \leq a\nu \leq 1$, which can be seen either based on the PDE or based on the CFL condition of the discrete CL in Eq.3.9. We will also need $\phi_j^n \to 1$ as the solution is smooth. However, we will have $C_{j+1/2}^n < 0$ under those conditions. Therefore, the flux-limiter scheme show above is not incremental TVD.

If we write the flux-limiter scheme (FTBS and Lax-Wendroff) into another I-form,

$$
\begin{aligned}
U_j^{n+1} &= U_j^n - \left\{a\nu - \frac{1}{2}a\nu(1-a\nu)\phi_{j-1}^n + \frac{1}{2}a\nu(1-a\nu)\phi_j^n\frac{\Delta_{+x}U_j^n}{\Delta_{-x}U_j^n}\right\}\Delta_{-x}U_j^n \\
C_{j+1/2}^n &= 0 \\
D_{j-1/2}^n &= a\nu - \frac{1}{2}a\nu(1-a\nu)\phi_{j-1}^n + \frac{1}{2}a\nu(1-a\nu)\phi_j^n\frac{\Delta_{+x}U_j^n}{\Delta_{-x}U_j^n} \\
&= a\nu - \frac{1}{2}a\nu(1-a\nu)\phi_{j-1}^n + \frac{1}{2}a\nu(1-a\nu)\phi_j^n\frac{\phi(\theta_j^n)}{\theta_j^n}
\end{aligned}
$$

Therefore, it is necessary to show that $0 \leq D^n_{j-1/2} \leq 1$, i.e.,

$$
\begin{aligned}
& 0 \leq a\nu - \frac{1}{2}a\nu(1-a\nu)\phi^n_{j-1} + \frac{1}{2}a\nu(1-a\nu)\phi^n_j \frac{\phi(\theta^n_j)}{\theta^n_j} \leq 1 \\
\Longleftrightarrow \quad & \begin{cases} a\nu - \frac{1}{2}a\nu(1-a\nu)\phi^n_{j-1} + \frac{1}{2}a\nu(1-a\nu)\phi^n_j \frac{\phi(\theta^n_j)}{\theta^n_j} \geq 0 \\ a\nu - \frac{1}{2}a\nu(1-a\nu)\phi^n_{j-1} + \frac{1}{2}a\nu(1-a\nu)\phi^n_j \frac{\phi(\theta^n_j)}{\theta^n_j} \leq 1 \end{cases} \\
\Longleftrightarrow \quad & \begin{cases} (1-a\nu)\left[\phi(\theta^n_{j-1}) - \frac{\phi(\theta^n_j)}{\theta^n_j}\right] \leq 2 \\ a\nu\left[\phi(\theta^n_{j-1}) - \frac{\phi(\theta^n_j)}{\theta^n_j}\right] \geq -2 \end{cases} \\
\Longleftrightarrow \quad & \left|\phi(\theta^n_{j-1}) - \frac{\phi(\theta^n_j)}{\theta^n_j}\right| \leq 2, \ \forall j
\end{aligned}
$$

Then $0 \leq D^n_{j-1/2} \leq 1$, and $|C_{j+1/2} + D_{j-1/2}| \leq 1$. Hence, the scheme is TVD. Moreover, since we always have $C_{j+1/2} = 0$, we can have $0 \leq D^n_{j+1/2} \leq 1$, and $|C_{j+1/2} + D_{j+1/2}| \leq 1$. Thus, the scheme is incremental TVD.

The equation

$$
\left|\phi(\theta^n_{j-1}) - \frac{\phi(\theta^n_j)}{\theta^n_j}\right| \leq 2, \ \forall j \tag{3.14}
$$

serves as the guideline to choose ϕ the limiter function.

Remark *(Some Notations)*

1. *The first requirement is $\phi(\theta) = 0$ for $\theta \leq 0$. In another word, we truncate the negative part of $\phi(\theta)$.*
2. *It is necessary to require $0 \leq \phi(\theta)/\theta \leq 2$ and $0 \leq \phi(\theta) \leq 2$. In another word, if the two conditions are satisfied, we can ensure Eq.3.14 or the TVD of the scheme (or incremental TVD based on the I-form).*
3. *We observe that being incremental TVD or not depends on the I-form representation.*

Proposition 3.20 (Bounded Flux-Limiter Implies Consistency)

If the flux-limiter function ϕ is bounded, then the difference scheme

$$
U^{n+1}_j = U^n_j - a\nu\Delta_{-x}U^n_j - \frac{1}{2}a\nu(1-a\nu)\Delta_{-x}\left(\phi^n_j\Delta_{+x}U^n_j\right)
$$

is consistent with the one-way wave equation. If $\phi(1) = 1$, and ϕ is Lipschitz continuous at $\theta = 1$, then the difference scheme is of the second order accuracy on the smooth portion of the solution with u_x bounded from 0.

♠

Example 3.9 (Examples of Flux-Limiters)

We will observe that some flux-limiters, e.g. the first one, can be TVD but not of the second order accuracy; some can achieve the second order accuracy, from the second one to the fifth one; some are even not TVD. The flux-limiters are shown in Fig. 3.7.

1. $\phi(\theta) = \min\{2\theta, 2\}, \theta \geq 0$, which is TVD but not of the second order accuracy, since $\phi(1) = 2 \neq 1$.
2. **SUPERBEE LIMITER:** $\phi(\theta) = \max\{0, \min\{1, 2\theta\}, \min\{\theta, 2\}\}$, which is TVD and of the second order accuracy.

3. **VAN LEER LIMITER:** $\phi(\theta) = (|\theta| + \theta)/(1 + |\theta|)$, which is TVD and of the second order accuracy.
4. **C-O LIMITER:** $\phi(\theta) = \max\{0, \min\{\theta, \psi\}\}$, $1 \leq \psi \leq 2$, which is TVD and of the second order accuracy.
5. **BW-LW (BEAM-WARMING & LAX-WENDROFF) LIMITER:** $\phi(\theta) = \max\{0, \min\{\theta, 1\}\}$, which is TVD and of the second order accuracy.
6. $\phi(\theta) = 1$, and the scheme is reduced to the Lax-Wendroff scheme, which is not TVD.
7. $\phi(\theta) = \theta$, and the scheme is reduced to the Beam-Warming scheme, which is not TVD.

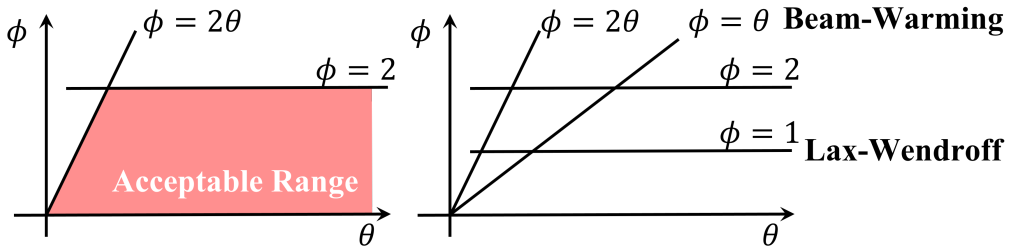

Figure 3.7: The Acceptable Range and Two Second Order Schemes Which Are Not TVD.

▲

Definition 3.14 (Symmetry of Flux-Limiter)

A flux-limiter function is said to be symmetric if

$$\frac{\phi(\theta)}{\theta} = \phi\left(\frac{1}{\theta}\right)$$

♣

For a general one-way wave equation, a could be positive or negative. Then the application of the upwind scheme as the low order scheme should be

$$h^n_{L,j+1/2} = \frac{1}{2}a\left(U^n_j + U^n_{j+1}\right) - \frac{1}{2}|a|\left(U^n_{j+1} - U^n_j\right)$$

And by using the Lax-Wendroff scheme as the high order scheme, a high resolution scheme can be created with the following numerical flux function

$$h^n_{j+1/2} = h^n_{L,j+1/2} + \frac{1}{2}\phi^n_j a[\text{sign}(a) - a\nu]\Delta_{+x}U^n_j,\ \text{sign}(a) = \begin{cases} 1, & a > 0 \\ 0, & a = 0 \\ -1, & a < 0 \end{cases}$$

Note that the sign(a) function is not necessary for the Lax-Wendroff scheme, since it is symmetric. However, in order to give the TVD condition using the I-form, we need to do that to satisfy Eq.3.14, and

$$\phi^n_j = \phi(\theta^n_j),\ \theta^n_j = \begin{cases} \Delta_{-x}U^n_j/\Delta_{+x}U^n_j, & a > 0 \\ \Delta_{+x}U^n_{j+1}/\Delta_{+x}U^n_j, & a < 0 \end{cases}$$

Using the definitions for ϕ and the smoothness parameter θ are reasonable. That is because we are trying to evaluate the numerical flux within (x_j, x_{j+1}), or in another word, right at $x_{j+1/2}$. When $a > 0$, we check backwards to the previous interval (x_{j-1}, x_j), and evaluate the numerical flux with x_{j-1}, x_j, x_{j+1}. When $a < 0$, we check forwards to the next interval (x_{j+1}, x_{j+2}), and evaluate the numerical flux with x_j, x_{j+1}, x_{j+2}.

Up to now, we have defined some conservative, TVD schemes with the second order accuracy in the smooth parts. We would like to know if those schemes give the expected entropy solutions (or vanishing viscosity

solution) to the original equation. At this moment, we only have the property of conservative TVD. It may be hard to prove the entropy solution analytically, while numerically, we will see entropy solutions in general. However, exception do exist.

For a CL, one of the analogies is to use the nonlinear upwind scheme as the low order scheme

$$U_j^{n+1} = \begin{cases} U_j^n - \nu\Delta_{+x}F_j^n, & \text{if } a_{j+1/2}^n > 0 \\ U_j^n - \nu\Delta_{-x}F_j^n, & \text{if } a_{j+1/2}^n < 0 \end{cases}, \text{ with } a_{j+1/2}^n = \begin{cases} \dfrac{\Delta_{+x}F_j^n}{\Delta_{+x}U_j^n}, & \text{if } \Delta_{+x}U_j^n \neq 0 \\ F'(U_j^n), & \text{if } \Delta_{+x}U_j^n = 0 \end{cases}$$

and use the nonlinear Lax-Wendroff scheme as the high order scheme

$$U_j^{n+1} = U_j^n - \nu\Delta_{0x}F_j^n + \frac{\nu^2}{2}\Delta_{-x}\left[(a_{j+1/2}^n)^2\Delta_{+x}U_j^n\right]$$

I.e., we can explicitly list the low order and high order numerical fluxes as

$$h_{L,j+1/2}^n = \frac{1}{2}[F_j^n + F_{j+1}^n] - \frac{1}{2}\left|a_{j+1/2}^n\right|\Delta_{+x}U_j^n$$
$$h_{H,j+1/2}^n = \frac{1}{2}[F_j^n + F_{j+1}^n] - \frac{\nu}{2}\left(a_{j+1/2}^n\right)^2\Delta_{+x}U_j^n$$

The combined numerical flux can be written as

$$h_{j+1/2}^n = h_{L,j+1/2}^n + \phi_j^n\frac{\left|a_{j+1/2}^n\right|}{2}\left[a - \nu\left|a_{j+1/2}^n\right|\right]\Delta_{+x}U_j^n$$

with the same smoothness parameter defined above, i.e.,

$$\theta_j^n = \begin{cases} \Delta_{-x}U_j^n/\Delta_{+x}U_j^n, & a > 0 \\ \Delta_{+x}U_{j+1}^n/\Delta_{+x}U_j^n, & a < 0 \end{cases}$$

By using combined numerical flux, it is obvious to see the reason we have to use $\text{sign}(a)$ or $\text{sign}(a_{j+1/2}^n)$ in the Lax-Wendroff scheme. E.g., suppose $\Delta_{+x}U_j^n \neq 0$,

$$\frac{1}{2}\left|a_{j+1/2}^n\right|\Delta_{+x}U_j^n = \frac{1}{2}\left|\frac{\Delta_{+x}F_j^n}{\Delta_{+x}U_j^n}\right|\Delta_{+x}U_j^n = \Delta_{+x}F_j^n\text{sign}(a_{j+1/2}^n)$$

Remark *Using a variety of combinations of the low order and high order schemes will give a range of high resolution schemes. However, in many cases, it is usually difficult to theoretically prove the TVD or the entropy solution. Thus, in general, we perform numerical tests. E.g.,*

$$h_{j+1/2}^n = h_{j+1/2}^E - \frac{1}{2}\phi(\theta_j^+)\left[\left(h_{j+1/2}^E - F_{j+1}^n\right) + \frac{\nu}{\Delta_{+x}U_j^n}\left(h_{j+1/2}^E - F_{j+1}^n\right)^2\right]$$
$$- \frac{1}{2}\phi(\theta_{j-1}^-)\left[\left(h_{j+1/2}^E - F_j^n\right) + \frac{\nu}{\Delta_{+x}U_j^n}\left(h_{j+1/2}^E - F_j^n\right)^2\right]$$

where h^E is some E-schemes or M-schemes, and θ_j^+ and θ_{j-1}^- are defined as

$$\theta_j^+ = \frac{\Delta_{+x}U_j^n}{\Delta_{+x}U_{j-1}^n} \times \frac{\Delta_{+x}U_{j-1}^n + \nu\left(h_{j-1/2}^E - F_j^n\right)}{\Delta_{+x}U_j^n + \nu\left(h_{j+1/2}^E - F_{j+1}^n\right)} \times \frac{h_{j-1/2}^E - F_j^n}{h_{j+1/2}^E - F_{j+1}^n}$$

$$\theta_{j-1}^- = \frac{\Delta_{+x}U_{j-2}^n}{\Delta_{+x}U_{j-1}^n} \times \frac{\Delta_{+x}U_{j-1}^n + \nu\left(h_{j-1/2}^E - F_{j-1}^n\right)}{\Delta_{+x}U_{j-2}^n + \nu\left(h_{j-3/2}^E - F_{j-2}^n\right)} \times \frac{h_{j-1/2}^E - F_{j-1}^n}{h_{j-3/2}^E - F_{j-2}^n}$$

If the CFL condition, i.e., $\nu|F'| \leq 2/3$ is satisfied, the above scheme is TVD.

3.5.2 Slope-Limiter Method

The slope-limiter method is similar to the Godunov scheme. However, instead of using piecewise constant approximation as it is in Godunov, piecewise linear approximation is used here. An illustrative comparison between the Godunov scheme and the Slope-limiter method is shown in Fig.3.8.

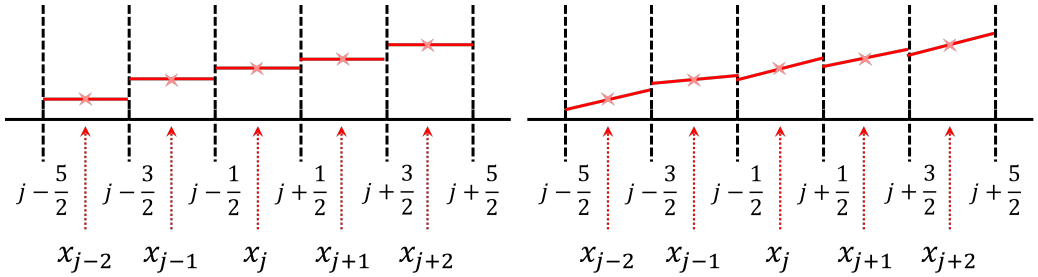

Figure 3.8: The Comparison between the Godunov Scheme and the Slope-Limiter Method.

E.g., at $t = t_0$, use U_j^n to define a piecewise linear approximation to our solution, i.e., $\overline{U}^n$. Then $\overline{U}^n$ becomes a piecewise linear function, s.t. $\overline{U}^n(j\Delta x) = U_j^n$. The slope of the piecewise linear function on the cell of x_j, i.e., $(x_{j-1/2}, x_{j+1/2})$ is given by σ_j^n, which will be determined later. That is to say, the linear approximation on $(x_{j-1/2}, x_{j+1/2})$ will be

$$\overline{U}^n(x) = U_j^n + \sigma_j^n(x - x_j^n) \tag{3.15}$$

Then march $t_n \to t_{n+1}$, we solve the CL as

$$\begin{cases} u_t + [F(u)]_x = 0,\ t > 0,\ x \in \mathbb{R} \\ u_0(x) = \overline{U}^n(x) \end{cases}$$

Remark *Since the piecewise linear approximation is used, the local Riemann scheme is not applicable.*

Example 3.10 (One-way Wave Equation)
Consider the one-way wave equation

$$u_t + au_x = 0,\ a > 0$$

For $a > 0$, to compute $\overline{U}$ on $(x_{j-1/2}, x_{j+1/2})$, we must solve the following local problem.

$$\begin{cases} u_t + au_x = 0 \\ u(x, t_n) = u_0(x) = \begin{cases} U_{j-1}^n + \sigma_{j-1}^n(x - x_{j-1}), & x_{j-3/2} \le x \le x_{j-1/2} \\ U_j^n + \sigma_j^n(x - x_j), & x_{j-1/2} \le x \le x_{j+1/2} \end{cases} \end{cases} \tag{3.16}$$

So at $t = t_{n+1}$, using the method of characteristics,

$$u(x, t_{n+1}) = u_0(x - a\Delta t) = \begin{cases} U_{j-1}^n + \sigma_{j-1}^n(x - x_{j-1} - a\Delta t), & x_{j-3/2} \le x - a\Delta t \le x_{j-1/2} \\ U_j^n + \sigma_j^n(x - x_j - a\Delta t), & x_{j-1/2} \le x - a\Delta t \le x_{j+1/2} \end{cases}$$

A demonstration of the characteristics is shown in Fig.3.9.

In general, if $a > 0$, suppose that the piecewise linear function is left-continuous at each point, particularly at the discontinuous points, such as at $x_{j-1/2}$, $x_{j+1/2}$,

$$U_{j-1/2} = U^- = U_{j-1} + \sigma_{j-1}^n \frac{\Delta x}{2},\ U_{j+1/2} = U^- = U_j + \sigma_j^n \frac{\Delta x}{2}$$

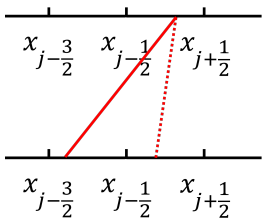

Figure 3.9: A Characteristics Based on the Slope-Limiter Method.

Recall that the slope-limiter method is a "one time step" method. Then, we take the integral $t \to t_n$ for that cell

$$
\begin{aligned}
0 =& \int_{x_{j-1/2}}^{x_{j+1/2}} \int_{t_n}^{t_{n+1}} [u_t + a u_x] dx dt = \int_{x_{j-1/2}}^{x_{j+1/2}} \int_{0}^{\Delta t} [u_t + a u_x] dx dt \\
=& \int_{x_{j-1/2}}^{x_{j+1/2}} [u(x, \Delta t) - u(x, 0)] dx + a \int_0^{\Delta t} [u(x_{j+1/2}, t) - u(x_{j-1/2}, t)] dt \\
\approx& [U_j^{n+1} - U_j^n] \Delta x + a \int_0^{\Delta t} \left[U_j^n + \sigma_j^n (x_{j+1/2} - at - x_j) - U_{j-1}^n - \sigma_{j-1}^n (x_{j-1/2} - at - x_{j-1}) \right] dt \\
=& [U_j^{n+1} - U_j^n] \Delta x + a \int_0^{\Delta t} \left[U_j^n + \sigma_j^n \left(\frac{\Delta x}{2} - at \right) - U_{j-1}^n - \sigma_{j-1}^n \left(\frac{\Delta x}{2} - at \right) \right] dt \\
=& [U_j^{n+1} - U_j^n] \Delta x + a \Delta t \Delta_{-x} U_j^n + a \left[\frac{1}{2} \Delta x \Delta t - \frac{1}{2} \Delta t^2 \right] \Delta_{-x} \sigma_j^n
\end{aligned}
$$

Hence, we see the reason of using linear function to substitute $u(x_{j\pm1/2}, t)$ is that by CFL condition, $x_{j\pm1/2} - at$ should fall into a continuous linear cell or interval, where no jump discontinuous points occur, as shown by the characteristics in Fig.3.9. Rearrange this equation, we can obtain the numerical scheme for $a > 0$.

$$U_j^{n+1} = U_j^n - a\nu \Delta_{-x} U_j^n - \frac{1}{2} a\nu(1 - a\nu) \Delta x \Delta_{-x} \sigma_j^n$$

In a similar way, assume the right-continuous at $x_{j-1/2}$, $x_{j+1/2}$, $x_{j+3/2}$ and so on, and we can have a numerical scheme for $a < 0$.

$$U_j^{n+1} = U_j^n - a\nu \Delta_{+x} U_j^n + \frac{1}{2} a\nu(1 + a\nu) \Delta x \Delta_{+x} \sigma_j^n$$

In summary, we have

$$U_j^{n+1} = \begin{cases} U_j^n - a\nu \Delta_{-x} U_j^n - \frac{1}{2} a\nu(1 - a\nu) \Delta x \Delta_{-x} \sigma_j^n, & a > 0 \\ U_j^n - a\nu \Delta_{+x} U_j^n + \frac{1}{2} a\nu(1 + a\nu) \Delta x \Delta_{+x} \sigma_j^n, & a < 0 \end{cases}$$

Let

$$h_{j+1/2}^n = \begin{cases} a U_j^n + \frac{1}{2} a(1 - a\nu) \Delta x \sigma_j^n, & a > 0 \\ a U_{j+1}^n - \frac{1}{2} a(1 + a\nu) \Delta x \sigma_j^n, & a < 0 \end{cases}$$

Thus, the scheme is conservative, and in general, $h_{j+1/2}^n$ can be written as follows with a special definition of the Boolean identity function $\mathbb{I}$.

$$h_{j+1/2}^n = \frac{a}{2}(U_j^n + U_{j+1}^n) - \frac{|a|}{2} \Delta_{+x} U_j^n + \frac{a}{2} [\text{sign}(a) - a\nu] \Delta x \sigma_{j+\mathbb{I}(a<0)}^n, \ \mathbb{I}(a < 0) = \begin{cases} 0, & a > 0 \\ 1, & a < 0 \end{cases}$$

▲

Remark *If $a > 0$ and let $\sigma_j^n = \Delta_{+x} U_j^n / \Delta x$, the scheme becomes the Lax-Wendroff scheme.*

So far we have seen an example of using piecewise linear function to approximate the solution. That implies the "(linear) slope". However, from a linguistic perspective, we have not seen the word "limiter" yet. That comes when we pursue the TVD property of the linear approximations.

Since the Godunov scheme is TVD, the slope-limiter method can also be TVD. Therefore, the goal is to choose σ, s.t. the slope-limiter method can be of the second order accuracy, TVD or can even produce an entropy solution. Thus, we propose some restrictions for the slope σ, and hence, this method becomes a "slope-limiter". The TV of a slope-limiter can be upper bounded by the TV of the Godunov scheme, shown in Fig.3.10.

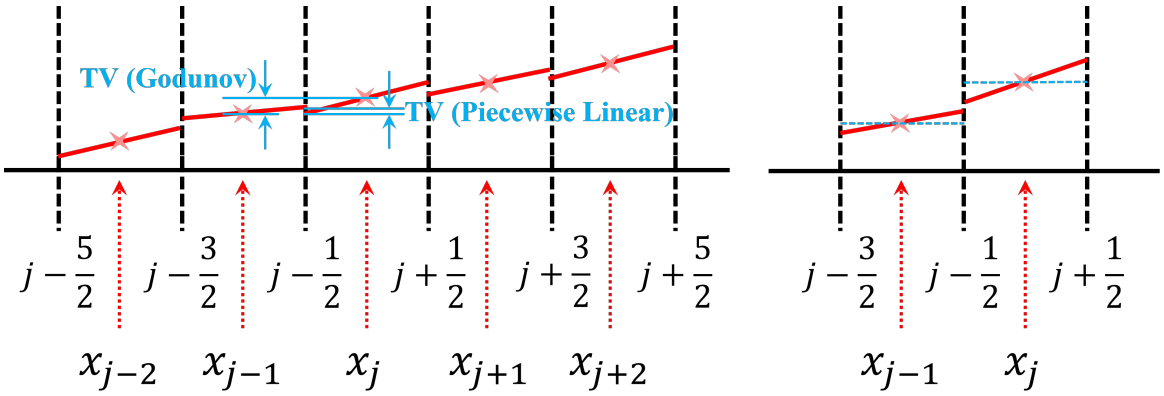

Figure 3.10: A Comparison of TV between Godunov Scheme and Slope-Limiter Scheme, Also Refer to Fig.3.8. The Right Diagram Emphasizes That The Jump Discontinuities of Piecewise Constant And Piecewise Linear Approximations.

Proposition 3.21 (TVD of Slope-Limiter)

If the slopes $\sigma_j^n, j = -\infty, ..., \infty$ are chosen, s.t. the TV of the piecewise linear approximation to the solution is $\leq$ the TV of the piecewise constant approximation provided by the Godunov scheme, then the slope-limiter scheme is TVD.

Based on Prop.3.21, it is important to have the hypothesis that by appropriately choosing the slopes, the TVD property can be ensured or even optimized for a given numerical problem. For the one-way wave equation, the variation between x_{j-1} and x_j can be $\leq$ the variation of the piecewise constant approximation, if the following equation is satisfied.

$$\lim_{x \to x_{j-1/2}^-} \overline{U}^n(x) \leq \lim_{x \to x_{j-1/2}^+} \overline{U}^n(x)$$

Also refer to the right diagram in Fig.3.9.

Using Prop.3.21, it is necessary to choose σ_j^n carefully to satisfy the equation above. One possible choice is the **MINMOD SLOPE-LIMITER**

$$\sigma_j^n = \frac{1}{\Delta x}\text{minmod}\left\{\Delta_{+x}U_j^n, \Delta_{-x}U_j^n\right\}, \ \text{minmod}\{a,b\} = \begin{cases} a, & \text{if } |a| < |b| \text{ and } ab > 0 \\ b, & \text{if } |a| > |b| \text{ and } ab > 0 \\ 0, & \text{if } ab < 0 \end{cases} \tag{3.17}$$

Recall that σ_j is the slope of the piecewise linear approximation within $(x_{j-1/2}, x_{j+1/2})$, centered at x_j. Usually, σ_j is related to the computation of numerical flux $h_{j+1/2}^n$.

Now, we use the slope-limiter scheme for the nonlinear CL. Different from the Godunov scheme, there is no local solver with an exact solution. Thus, we need to first solve

$$\begin{cases} u_t + [F(u)]_x = 0, & x \in \mathbb{R}, t > 0 \\ u(x, t_n) = \overline{U}^n, & \text{for } \overline{U}^n(x, t_n) = U_j^n + \sigma_j^n (x - x_j) \end{cases}$$

By taking the integral form

$$\int_{x_{j-1/2}}^{x_{j+1/2}} \left[\overline{U}(x, t_{n+1}) - \overline{U}(x, t_n)\right] dx + \int_{t_n}^{t_{n+1}} \left[F(\overline{U}(x_{j+1/2}, t)) - F(\overline{U}(x_{j-1/2}, t))\right] dt = 0$$

We can obtain the following numerical scheme.

$$U_j^{n+1} = U_j^n - \frac{1}{\Delta x} \int_{t_n}^{t_{n+1}} \left[F(\overline{U}(x_{j+1/2}, t)) - F(\overline{U}(x_{j-1/2}, t))\right] dt$$

Let us define the following numerical flux as a extension of the Godunov scheme

$$h_{x_{j+1/2}} = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \left[F(\overline{U}(x_{j+1/2}, t))\right] dt$$

Since $\overline{U}(x_{j+1/2}, t)$ and $F(\overline{U})$ are not a constant. A natural question is, how can we compute or approximate $h_{j\pm1/2}^n$? Define that $\sigma_j^n = \frac{1}{\Delta x}\text{minmod}\left\{\Delta_{+x} U_j^n, \Delta_{-x} U_j^n\right\}$, and $U_j^{\pm} = U_j^n \pm 1/2 \Delta x \sigma_j^n$ as the values on the cell boundaries but inside the cell, i.e., the "interior limits" of the piecewise linear lines. An illustrative scheme is shown in Fig.3.11.

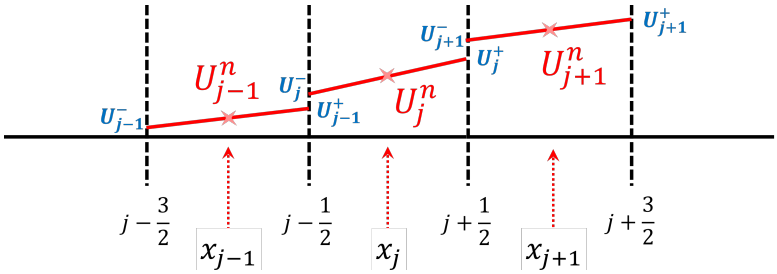

Figure 3.11: Illustrations of the Boundary Values for the Cells.

To ensure the TV of piecewise linear approximation is $\leq$ the TV of piecewise constant approximation, we need

$$U_j^- \leq U_j^+ \leq U_{j+1}^- \leq U_{j+1}^+, \text{ or, } U_j^- \geq U_j^+ \geq U_{j+1}^- \geq U_{j+1}^+$$

Back to our model problem, $u_t + F'(u)u_x = 0$. If $F'(u)$ is piecewise constant with respect to u, we may reduce the problem to $u_t + au_x = 0$. That is our goal at this step to simplify the problem. Since we already have some ideas how U changes in the x domain, let $G(u)$ be a piecewise linear interpolation of $F(u)$ at the four points, $U_j^{\pm}$, $U_{j+1}^{\pm}$, shown in Fig.3.12, and set

$$G_j' = \begin{cases} \dfrac{F(U_j^+) - F(U_j^-)}{U_j^+ - U_j^-}, & \text{if } \sigma_j^n \neq 0 \\ F'(U_j^n), & \text{if } \sigma_j^n = 0 \end{cases} \Rightarrow G(u) = \begin{cases} F(U_j^+) + (u - U_j^+)G_j', & u \in \left[U_j^-, U_j^+\right] \\ F(U_{j+1}^-) + (u - U_{j+1}^-)G_{j+1}', & u \in \left[U_{j+1}^-, U_{j+1}^+\right] \end{cases}$$

In the numerical scheme, use $G(u)$ instead of $F(u)$, and need to solve the "new" problem

$$\begin{cases} u_t + G_x = 0 \\ u(x, t_n) = \overline{U}^n \end{cases} \Rightarrow \begin{cases} u_t + G'u_x = 0 \\ u(x, t_n) = \overline{U}^n \end{cases}$$

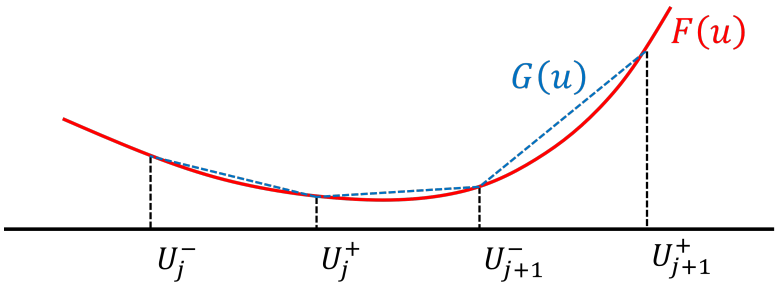

Figure 3.12: $G(u)$ as A Piecewise Linear Interpolation of $F(u)$.

Then, G' is piecewise constant, since G is piecewise linear. Thus, the above equation is solvable using the method of characteristics. Additionally, we want that G is a sufficiently good approximation of F, s.t.,

$$F' > 0 \Rightarrow G' > 0$$
$$F' < 0 \Rightarrow G' < 0$$

by choosing Δx sufficiently small. Hence, referring to $\overline{U}^n(x) = U_j^n + \sigma_j^n(x - x_j)$, we have

$$\overline{U}(x_{j+1/2}, t) = \begin{cases} U_j^+ - (t - t_n)\sigma_j^n G_j', & \text{if } F' > 0 \\ U_{j+1}^- - (t - t_n)\sigma_{j+1}^n G_{j+1}', & \text{if } F' < 0 \end{cases}$$

Then $h_{x_{j+1/2}}$ can be approximated by

$$\tilde{h}_{j+1/2}^n = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} G\left(\overline{U}(x_{j+1/2}, t)\right) dt = \begin{cases} F(U_j^+) - \frac{1}{2}\Delta t \sigma_j^n (G_j')^2, & \text{if } F' > 0 \\ F(U_{j+1}^-) - \frac{1}{2}\Delta t \sigma_{j+1}^n (G_{j+1}')^2, & \text{if } F' < 0 \end{cases} \tag{3.18}$$

Thus, we can use the CL

$$U_j^{n+1} = U_j^n - \nu\left(h_{j+1/2} - h_{j-1/2}\right)$$

Remark *We provide a simple derivation for Eq.3.18 when $F' > 0$,*

$$\begin{aligned} &G\left(\overline{U}(x_{j+1/2}, t)\right) = F(U_j^+) + (u - U_j^+)G_j' \\ =&F(U_j^+) + \left[U_j^+ - (t - t_n)\sigma_j^n G_j' - U_j^+\right] G_j' = F(U_j^+) - (t - t_n)\sigma_j^n \left(G_j'\right)^2 \end{aligned}$$

Hence

$$\frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} G\left(\overline{U}(x_{j+1/2}, t)\right) dt = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \left[F(U_j^+) - (t - t_n)\sigma_j^n \left(G_j'\right)^2\right] dt$$

Then Eq.3.18 can be shown by taking this integral.

Remark *Given CFL condition, $\nu|F'| \leq 1/2$, satisfied. Then the above scheme is of the second order on smooth sections of the solution which are not local extrema.*

If $G_j' G_{j+1}' \leq 0$, define

$$G_{j+1/2}' = \begin{cases} [F(U_{j+1}^-) - F(U_j^+)]/(U_{j+1}^- - U_j^+), & \text{if } U_{j+1}^- \neq U_j^+ \\ F'(U_j^+), & \text{if } U_{j+1}^- = U_j^+ \end{cases}$$

and the numerical flux can be estimated as

1. If $G_j' > 0$, $G_{j+1/2}'(U_{j+1}^- - U_j^+) = 0$, and $G_{j+1}' < 0$, set

$$\tilde{h}_{j+1/2}^n = \begin{cases} F(U_j^+) - \frac{1}{2}\Delta t \sigma_j^n (G_j')^2, & \text{if } \sigma_j^n (G_j')^2 \geq \sigma_{j+1}^n (G_{j+1}')^2 \\ F(U_{j+1}^-) - \frac{1}{2}\Delta t \sigma_{j+1}^n (G_{j+1}')^2, & \text{otherwise} \end{cases} \tag{3.19}$$

i.e. the numerical flux is limited by the smaller one.

2. Otherwise, set

$$\tilde{h}^n_{j+1/2} = \begin{cases} F(U^+_j) - \frac{1}{2}\Delta t \sigma^n_j (G'_j)^2, & \text{if } G'_j \geq 0 \text{ and } G'_{j+1/2} \geq 0 \\ F(U^-_{j+1}) - \frac{1}{2}\Delta t \sigma^n_{j+1} (G'_{j+1})^2, & \text{if } G'_{j+1} \leq 0 \text{ and } G'_{j+1/2} \leq 0 \\ F(U_0), & \text{if } G'_j < 0 \text{ and } G'_{j+1} > 0 \end{cases} \tag{3.20}$$

where $U_0 = \min\{\max\{U^+_j, U_s\}, U^-_{j+1}\}$, and U_s is call as a sonic point.

In conclusion, the scheme should be applied based on the

1. If $G'_j G'_{j+1} > 0$, use Eq.3.18;
2. Else use Eq.3.19 or Eq.3.20.

3.5.3 Modified-Flux Method

So far, some smart approaches are established to approximate the numerical flux. However, for the **MODIFIED-FLUX METHOD**, we modify the flux function of the PDE, or in another word, we modify the CL. The keys for the modified-flux method is that the flux function of the CL, rather than the numerical flux, is modified.

Given the CL

$$u_t + [F(u)]_x = 0$$

Let h^L be a conservative, three-point, incrementally TVD scheme, e.g., E-scheme or M-scheme (which is of the first order). Let Q^L denote the coefficient of numerical viscosity in the Q-form of the scheme, recall Def.3.12, where

$$h^L_{j+1/2} = \frac{1}{2}\left(F^n_j + F^n_{j+1}\right) - \frac{1}{2\nu} Q^L_{j+1/2} \Delta_{+x} U^n_j$$

If incrementally TVD, we require $\nu \left|a_{j+1/2}\right| \leq Q^L_{j+1/2} \leq 1$ (can be proved if by converting it to the I-form). The scheme can approximate the CL to the first order. Denote the leading term in the truncation error as $\Delta t \left\{[q(u)u_x]_x\right\}_{u=u^n_j}$, according to Prop.3.13. Then, we can add this term to the CL,

$$u_t + [F(u)]_x = \Delta t \left[q(u)u_x\right]_x$$

Then the "first order scheme" can approximate the solution to the modified equation to the second order.

Thus, the idea of the modified-flux method is to apply first order scheme to solve a modified CL

$$u_t + [F^M(u)]_x = 0 \tag{3.21}$$

where F^M has been chosen s.t. the first order approximation to Eq.3.21 will be a second order approximate to the original CL, $u_t + [F(u)]_x = 0$.

To give a first example of the second order schemes, start with h^{LW} (Lax-Wendroff numerical flux) and Q^{LW} shown as follows

$$\begin{cases} h^{LW}_{j+1/2} = \left[F^n_j + F^{n+1}_j\right] - \frac{\nu}{2}(a^n_{j+1/2})^2 \Delta_{+x} U^n_j \\ Q^{LW}_{j+1/2} = \nu^2 (a^n_{j+1/2})^2 \end{cases}$$

We apply the first order scheme for h^L and Q^L, with F replaced by F^M, i.e.,

$$F^M_j = F^n_j + \frac{1}{\nu} g_j, \quad g_j = g(U_{j-1}, U_j, U_{j+1})$$

g is a function to be determined. Let $a^M_{j+1/2}$ defined based on F^M, rather than F. Then, the modified numerical flux can be given as

$$h^M_{j+1/2} = \frac{1}{2}\left[F^M_{j+1} + F^M_j\right] - \frac{1}{2\nu} Q^M_{j+1/2} \Delta_{+x} U^n_j$$

where $Q^M_{j+1/2}$ is Q^L with $a^n_{j+1/2}$ replaced by $a^M_{j+1/2}$, i.e.,

$$h^M_{j+1/2} = 1/2\left[F^M_{j+1} + F^M_j\right] + \frac{1}{2\nu}\left[g_{j+1} + g_j - Q^L\left(\nu a^n_{j+1/2} + \frac{\Delta_{+x} g_j}{\Delta_{+x} U^n_j}\right)\Delta_{+x} U^n_j\right]$$

Proposition 3.22 (Second Order Accuracy of h^M)

Suppose Q^L is Lipschitz continuous and g_j satisfies

$$g_j + g_{j+1} = \left[Q^L(\nu a^n_{j+1/2}) - \nu^2 (a^n_{j+1/2})^2\right]\Delta_{+x} U^n_j + O(\Delta x^2)$$
$$\Delta_{+x} g_i = O(\Delta x^2)$$

Then the difference scheme with $h^M_{j+1/2}$ is second order accurate on smooth sections of the solution.

The function g can be defined as follows,

$$g_j = \begin{cases} \text{sign}\left\{\tilde{g}_{j+1/2}\right\} \min\left\{|\tilde{g}_{j+1/2}|, |\tilde{g}_{j-1/2}|\right\}, & \text{when } \tilde{g}_{j+1/2}\tilde{g}_{j-1/2} \geq 0 \\ 0, & \text{else, i.e. } \tilde{g}_{j+1/2}\tilde{g}_{j-1/2} \leq 0 \end{cases} \tag{3.22}$$
$$\text{where} \quad \tilde{g}_{j+1/2} = \frac{1}{2}\left[Q^L(\nu a^n_{j+1/2}) - \nu^2 (a^n_{j+1/2})^2\right]\Delta_{+x} U^n_j$$

Remark *If g is defined as in Eq.3.22, then the requirements in the above proposition is satisfied.*

Proposition 3.23 (Second Order Accuracy of The Scheme)

Difference scheme

$$U^{n+1}_j = U^n_j - \nu \Delta_{-x} h^M_{j+1/2}$$
$$\textit{where} \quad h^M_{j+1/2} = \frac{1}{2}\left[F^M_{j+1} + F^M_j\right] + \frac{1}{2\nu}\left[g_{j+1} + g_j - Q^L\left(\nu a^n_{j+1/2} + \frac{\Delta_{+x} g_j}{\Delta_{+x} U^n_j}\right)\Delta_{+x} U^n_j\right]$$

and with g defined as in Eq.3.22 has the second order accuracy for the solution segments away from the extrema of the solutions.

Lemma 3.1 (Boundedness of g)

Let g_i be defined as in Eq.3.22, then we have

$$\left|\frac{\Delta_{+x} g_j}{\Delta_{+x} U^n_j}\right| \leq \frac{1}{2}\left|Q^L(\nu a^n_{j+1/2}) - (a^n_{j+1/2})^2\right|$$

Proposition 3.24

Suppose that Q^L satisfies $|\nu a^n_{j+1/2}| \leq Q^L_{j+1/2} < 1$, $\forall j$, and g_j is defined in Eq.3.22, then the follwong

scheme is TVD.

$$h^M_{j+1/2} = \frac{1}{2}\left[F^M_{j+1} + F^M_j\right] + \frac{1}{2\nu}\left[g_{j+1} + g_j - Q^L\left(\nu a^n_{j+1/2} + \frac{\Delta_{+x} g_j}{\Delta_{+x} U^n_j}\right)\Delta_{+x} U^n_j\right]$$

3.6 Brief Introduction to The Implicit Schemes

Start with the integral form defined based on the cell average

$$\Delta x\left(U^{n+1}_j - U^n_j\right) + \int_{t_n}^{t_{n+1}} F(U(x_{j+1/2}, t))dt - \int_{t_n}^{t_{n+1}} F(U(x_{j-1/2}, t))dt = 0$$

For explicit schemes, we approximate the integral terms as $\Delta t\left(h^n_{j+1/2} - h^n_{j-1/2}\right)$, at $t = t_n$. More detailed, we approximate the integrals as

$$\Delta t\left[F(U(x_{j+1/2}, t)) - F(U(x_{j-1/2}, t))\right] + O(\Delta t^2)$$

Then the design of explicit schemes follows as discussed previously.

Similarly, we can approximate the integrals at $t = t_{n+1}$,

$$\Delta t\left[F(U(x_{j+1/2}, t_{n+1})) - F(U(x_{j-1/2}, t_{n+1}))\right] + O(\Delta t^2)$$

which will lead to an implicit scheme.

A different quadrature rule will lead to a **CRANK-NICOLSON** type scheme, e.g., using a trapezoidal rule

$$\begin{aligned}\Delta x\left(U^{n+1}_j - U^n_j\right) &+ \frac{\Delta t}{2}\left[F(U(x_{j+1/2}, t_n)) - F(U(x_{j-1/2}, t_n))\right] \\ &+ \frac{\Delta t}{2}\left[F(U(x_{j+1/2}, t_{n+1})) - F(U(x_{j-1/2}, t_{n+1}))\right] + O(\Delta t^3)\end{aligned}$$

Then Crank-Nicolson scheme can be written as

$$\begin{aligned}\Delta x\left(U^{n+1}_j - U^n_j\right) &+ \frac{\Delta t}{2}\left[F(U(x_{j+1/2}, t_n)) - F(U(x_{j-1/2}, t_n))\right] \\ &+ \frac{\Delta t}{2}\left[F(U(x_{j+1/2}, t_{n+1})) - F(U(x_{j-1/2}, t_{n+1}))\right] = 0\end{aligned}$$

3.7 Differential Schemes for 2D Conservation Laws

Analogy to the linear PDE, $u_t + au_x + bu_y = 0$, a 2D scalar CL can be defined as

$$u_t + [F(u)]_x + [G(u)]_y = 0 \tag{3.23}$$

Assume the CL is associated with appropriate ICs and BCs, and the TVD property is defined to avoid any bad oscillation near the local extrema. TV in a 2D spatial domain is defined as

$$TV(\vec{U}^n) = \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\left[\Delta x\left\|U^n_{i+1,j} - U^n_{i,j}\right\| + \Delta y\left\|U^n_{i,j+1} - U^n_{i,j}\right\|\right] \tag{3.24}$$

Proposition 3.25 (Goodman & Le Veque)

Most of the 2D TVD schemes are at most first order accurate, except for a small number of trivial cases.

Remark *For higher order schemes, they may not be TVD. Instead, consider the ENO property of a scheme shown as follows,*

$$TV(\vec{U}^{n+1}) \leq TV(\vec{U}^{n}) + O(\Delta x^p) + O(\Delta y^q)$$

Partition the computational domain with the mesh grid (x_i, y_j), $i, j = -\infty, ..., \infty$, where mesh size is defined as $\Delta x, \Delta y$. The cell associated with (x_i, y_j) is $(x_{i-1/2}, x_{i+1/2}) \times (y_{j-1/2}, y_{j+1/2})$, shown in Fig.3.13. Approximate the numerical solution at $t_n = n\Delta t$ as $U^n_{i,j} \approx u(x_i, y_j.t_n)$. $U^n_{i,j}$ will be the "cell average" approximation to the solution.

$$U^n_{i,j} = \frac{1}{\Delta x \Delta y} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} u(x, t, t_n) dxdy$$

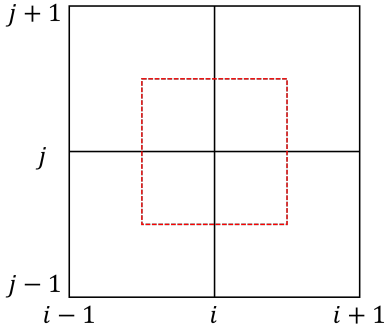

Figure 3.13: Cell Average Based on the 2D Discretization Grid.

The integral form of the original CL becomes

$$\begin{aligned}
&\int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \left[u(x, y, t_{n+1}) - u(x, y, t_n)\right] dxdy \\
+ &\int_{t_n}^{t_{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} \left[F(u(x_{i+1/2}, y, t)) - u(x_{i-1/2}, y, t)\right] dydt \\
+ &\int_{t_n}^{t_{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \left[G(u(x, y_{j+1/2}, y, t)) - G(u(x, y_{j-1/2}, y, t))\right] dxdt = 0
\end{aligned}$$

Using the idea of cell-average

$$\begin{aligned}
U^{n+1}_{i,j} = U^n_{i,j} &- \frac{1}{\Delta x \Delta y} \int_{t_n}^{t_{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} \left[F(u(x_{i+1/2}, y, t)) - u(x_{i-1/2}, y, t)\right] dydt \\
&- \frac{1}{\Delta x \Delta y} \int_{t_n}^{t_{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \left[G(u(x, y_{j+1/2}, y, t)) - G(u(x, y_{j-1/2}, y, t))\right] dxdt
\end{aligned}$$

Then we can write the 2D conservative scheme as

$$U^{n+1}_{i,j} = U^n_{i,j} - \nu_x \left[p^n_{i+1/2,j} - p^n_{i-1/2,j}\right] - \nu_y \left[q^n_{i,j+1/2} - q^n_{i,j-1/2}\right]$$

where $\nu_x = \Delta t/\Delta x$, $\nu_y = \Delta t/\Delta y$, and p, q are the x-direction and y-direction numerical flux functions, respectively. In another word, we need to use

$$\begin{aligned}
&- \nu_x \Delta_{-x} p^n_{i+1/2,j}, && \text{approximate the integral w.r.t } F \\
&- \nu_y \Delta_{-y} q^n_{i+1/2,j}, && \text{approximate the integral w.r.t } G
\end{aligned}$$

In general, p depends on $U_{i-p,j+l}, ..., U_{i+q,j+l}$, $l = -r, ...r$. But in many cases, especially in the one-step explicit scheme, $r = 0$. Similarly, q depends on $U_{i+l',j-p}, ..., U_{i+l',j+q}$, $l' = -r', ...r'$. But in many cases,

especially in the one-step explicit scheme, $r' = 0$. In Prop.3.26, we give the definition of consistency, and in the following example, we provide some practical ways to choose p and q.

Proposition 3.26 (Consistency of The 2D Schemes)

If $p(U, ..., U) = F(U)$, $q(U, ..., U) = G(U)$, then the conservative scheme

$$U_{i,j}^{n+1} = U_{i,j}^n - \nu_x \left[p_{i+1/2,j}^n - p_{i-1/2,j}^n\right] - \nu_y \left[q_{i,j+1/2}^n - q_{i,j-1/2}^n\right]$$

consists with the original problem

$$u_t + [F(u)]_x + [G(u)]_y = 0$$

Example 3.11 (Some Example Schemes)

1. **FTBS (FORWARD TIME BACKWARD SPACE) SCHEME**

$$p_{i+1/2,j}^n = F_{i,j}^n,\ q_{i+1/2,j}^n = G_{i,j}^n$$

2. **GENERALIZED UPWIND SCHEME**

$$p_{i+1/2,j}^n = \frac{1}{2}\left(F_{i,j}^n + F_{i+1,j}^n\right) - \frac{1}{2}\left|a_{i+1/2,j}^n\right| \Delta_{+x} U_{i,j}^n$$
$$q_{i,j+1/2}^n = \frac{1}{2}\left(G_{i,j}^n + G_{i,j+1}^n\right) - \frac{1}{2}\left|b_{i,j+1/2}^n\right| \Delta_{+y} U_{i,j}^n$$

where, e.g.,

$$a_{i+1/2,j} = \begin{cases} \dfrac{\Delta_{+x} F_{i,j}}{\Delta_{+x} U_{i,j}}, & \Delta_{+x} U_{i,j} \neq 0 \\ F'(U_{i,j}), & \Delta_{+x} U_{i,j} = 0 \end{cases},\ b_{i+1/2,j} = \begin{cases} \dfrac{\Delta_{+y} G_{i,j}}{\Delta_{+y} U_{i,j}}, & \Delta_{+y} U_{i,j} \neq 0 \\ G'(U_{i,j}), & \Delta_{+y} U_{i,j} = 0 \end{cases}$$

3. **LAX-FRIEDRICHS SCHEME**

$$p_{i+1/2,j}^n = \frac{1}{2}\left(F_{i,j}^n + F_{i+1,j}^n\right) - \frac{1}{4\nu_x}\Delta_{+x} U_{i,j}^n$$
$$q_{i,j+1/2}^n = \frac{1}{2}\left(G_{i,j}^n + G_{i,j+1}^n\right) - \frac{1}{4\nu_y}\Delta_{+y} U_{i,j}^n$$

4. **LAX-WENDROFF SCHEME**

$$p_{i+1/2,j}^n = \frac{1}{2}\left(F_{i,j}^n + F_{i+1,j}^n\right) - \frac{\nu_x}{2} A_{i+1/2,j} \Delta_{+x} F_{i,j}^n$$
$$q_{i,j+1/2}^n = \frac{1}{2}\left(G_{i,j}^n + G_{i,j+1}^n\right) - \frac{\nu_x}{2} B_{i,j+1/2} \Delta_{+y} U_{i,j}^n$$

where, e.g.,

$$A_{i+1/2,j} = F'\left(\frac{U_{i,j}^n + U_{i+1,j}^n}{2}\right),\ B_{i,j+1/2} = G'\left(\frac{U_{i,j}^n + U_{i,j+1}^n}{2}\right)$$

5. **ALTERNATIVE DIRECTION (AD) LAX-WENDROFF SCHEME**

$$p_{i+1/2,j}^n = \frac{1}{2}\left(F_{i,j}^n + F_{i+1,j}^n\right) - \frac{\nu_x}{2}\left|a_{i+1/2,j}^n\right|^2 \Delta_{+x} U_{i,j}^n$$
$$q_{i,j+1/2}^{n+1/2} = \frac{1}{2}\left(G_{i,j}^{n+1/2} + G_{i,j+1}^{n+1/2}\right) - \frac{\nu_y}{2}\left|b_{i,j+1/2}^{n+1/2}\right|^2 \Delta_{+y} U_{i,j}^{n+1/2}$$

Especially for the linear case

$$p_{i+1/2,j}^n = a U_{i,j}^n + \frac{1}{2} a(1 - a\nu_x)\Delta_{+x} U_{i,j}^n$$
$$q_{i,j+1/2}^{n+1/2} = b U_{i,j}^{n+1/2} + \frac{1}{2} b(1 - b\nu_y)\Delta_{+y} U_{i,j}^{n+1/2}$$

The **LOCALLY ONE DIMENSIONAL (LOD)** schemes will be

$$U_{i,j}^{n+1/2} = U_{i,j}^{n} - \nu_x \Delta_{-x} p_{i+1/2,j}^{n}$$
$$U_{i,j}^{n+1} = U_{i,j}^{n+1/2} - \nu_y \Delta_{-y} q_{i,j+1/2}^{n+1/2}$$

▲

Remark *(Factorization Approximation)*
Consider the original CL $u_t = -(F_x + G_y)$*, and its integral form* $U^{n+1} - U^n = -\int_{t_n}^{t_{n+1}} (F_x + G_y)\, dt$*. We can make the following manipulation,*

$$\begin{aligned} U^{n+1} &= U^n - (F_x + G_y)\,\Delta t + O(\Delta t^2) \\ &= U^n - \Delta t F'(u^n) u_x^n - \Delta t G'(u^n) u_y^n + O(\Delta t^2) \\ &= U^n - \Delta t F'(u^n) u_x^n - \Delta t G'(u^n) u_y^n + \Delta t^2 \left[G'(u^n)\frac{\partial}{\partial y}\right]\left[F'(u^n)\frac{\partial}{\partial x}\right] u^n + O(\Delta t^2) \end{aligned}$$

Thus,

$$u^{n+1} = \left[1 - \Delta t G'^n(u^n)\frac{\partial}{\partial y}\right]\left[1 - \Delta t F'(u^n)\frac{\partial}{\partial x}\right] u^n + O(\Delta t^2)$$

Thus, we have a difference scheme in a "split form",

$$\begin{cases} u^{n+1/2} = \left[1 - \Delta t F'^n(u^n)\dfrac{\partial}{\partial x}\right] u^n \\ u^{n+1} = \left[1 - \Delta t G'^n(u^n)\dfrac{\partial}{\partial y}\right] u^{n+1/2} \end{cases}$$

Remark *(Separate Differentiation w.r.t* x *&* y *in The Local 1D Scheme)*
Let us consider a **LOCAL ONE DIMENSIONAL (LOD)** *scheme for the original problem* $u_t + [F(u)]_x + [G(u)]_y = 0$ *in the following form*

$$\begin{cases} U_{i,j}^{n+1/2} = U_{i,j}^{n} - \nu_x \Delta_{+x} p_{i+1/2,j}^{n} \\ U_{i,j}^{n+1} = U_{i,j}^{n+1/2} - \nu_y \Delta_{+y} q_{i,j+1/2}^{n+1/2} \end{cases}$$

p *and* q *are numerical fluxes in* x *and* y*. We could use any* p *and* q *we have learned before. E.g., the AD Lax-Wendroff scheme in the previous example.*

Example 3.12 Consider the linear case $u_t + a u_x + b u_y = 0$. The AD Lax-Wendroff scheme can be deduced via

$$\begin{cases} p_{i+1/2,j}^{n} = a U_{i,j}^{n} + \dfrac{1}{2} a(1 - a\nu_x)\Delta_{+x} U_{i,j}^{n} \\ q_{i,j+1/2}^{n} = b U_{i,j}^{n+1/2} + \dfrac{1}{2} b(1 - b\nu_y)\Delta_{+y} U_{i,j}^{n+1/2} \end{cases}$$

▲

We will not provide detailed discussion for the 2D high resolution scheme, but we leave a simple example here. For $a > 0$, $b > 0$, we can write

$$p_{i+1/2,j}^{n} = a U_{i,j}^{n} + \frac{1}{2}\phi_{i,j}^{x,n} a(1 - a\nu_x)\Delta_{+x} U_{i,j}^{n}$$
$$q_{i,j+1/2}^{n} = b U_{i,j}^{n} + \frac{1}{2}\phi_{i,j}^{y,n} b(1 - b\nu_y)\Delta_{+y} U_{i,j}^{n}$$

where ϕ^x, ϕ^y are the flux limiter functions,

$$\phi_{i,j}^{x,n} = \phi_{i,j}^{x,n}\left(\theta_{i,j}^{x,n}\right),\ \phi_{i,j}^{y,n} = \phi_{i,j}^{y,n}\left(\theta_{i,j}^{y,n}\right),\ \text{with} \quad \theta_{i,j}^{x,n} = \frac{\Delta_{-x} U_{i,j}^{n}}{\Delta_{+x} U_{i,j}^{n}}, \theta_{i,j}^{y,n} = \frac{\Delta_{-y} U_{i,j}^{n}}{\Delta_{+y} U_{i,j}^{n}}$$

Thus, the high resolution scheme is established for individual dimensions. However, we cannot expect TVD for a 2D spatial domain.

3.8 Exercises

Exercise 3.1 Solve the 1D Burgers' equation

$$\begin{cases} u_t(x,t) + \frac{1}{2}\left[u^2(x,t)\right]_x = 0, & (x,t) \in [0,2] \times [0,2] \\ u(x,0) = u_0(x) = \sin(x), & x \in [0,2] \\ u(0,t) = u(2,t), & \text{Periodic BC in } x \end{cases}$$

with the conservative schemes

$$u_j^{n+1} = u_j^n - \nu\left(h_{j+1/2}^n - h_{j-1/2}^n\right)$$

1. Show that the Lax-Friedrichs scheme with the following numerical flux

$$h_{j+1/2}^n = \frac{1}{2}\left(F_{j+1}^n + F_j^n\right) - \frac{1}{2\nu}\left(u_{j+1}^n - u_j^n\right)$$

   is consistent with the Burgers' equation, and is monotonic if the CFL condition is satisfied.
2. Show that the Lax-Wendroff scheme

$$h_{j+1/2}^n = \frac{1}{2}\left(F_{j+1}^n + F_j^n\right) - \frac{\nu}{2}A_{j+1/2}\left(F_{j+1}^n - F_j^n\right),\ A_{j+1/2} = F'\left(\frac{u_{j+1}^n + u_j^n}{2}\right) = \frac{u_{j+1}^n + u_j^n}{2}$$

   is consistent with the Burgers' equation, but it is not monotonic.
3. Solve the Burgers' equation with the schemes in 1 and 2, with $\Delta x = 1/100$, $\Delta t = \Delta x/2$. Plot the solution at $t = 0.5, 1.0, 1.5, 2.0$. Comment on the numerical solution.
4. Use the numerical fluxes defined in 1 and 2 as h_L and h_H respectively, and design a high resolution scheme as it is in the flux-limiter method,

$$h_{j+1/2}^n = h_{L,j+1/2}^n + \phi_j^n\left(h_{H,j+1/2}^n - h_{L,j+1/2}^n\right)$$

   where the flux-limiter $\phi_j^n = \phi(\theta_j^n) = \phi\left(\frac{u_j^n - u_{j-1}^n}{u_{j+1}^n - u_j^n}\right)$ is given by the Superbee limiter

$$\phi(\theta) = \max\{0, \min\{1, 2\theta\}, \min\{\theta, 2\}\}$$

   Solve the Burgers' equation by the high resolution scheme with $\Delta x = 1/100$, $\Delta t = \Delta x/2$. Plot the solution at $t = 0.5, 1.0, 1.5, 2.0$. Comment on the numerical solution.

Solve

1. *For* **CONSISTENCY**, *we have*

$$\begin{aligned} \Delta t T_j^n &= u(x_j, t_{n+1}) - u(x_j, t_n) + \nu\left[\frac{1}{2}\left(\frac{1}{2}u^2(x_{j+1}, t_n) + \frac{1}{2}u^2(x_j, t_n)\right) - \frac{1}{2\nu}\left(u(x_{j+1}, t_n) - u(x_j, t_n)\right)\right] \\ &\quad - \nu\left[\frac{1}{2}\left(\frac{1}{2}u^2(x_j, t_n) + \frac{1}{2}u^2(x_{j-1}, t_n)\right) - \frac{1}{2\nu}\left(u(x_j, t_n) - u(x_{j-1}, t_n)\right)\right] \\ &= u(x_j, t_{n+1}) - u(x_j, t_n) - \frac{1}{2}\left(u(x_{j+1}, t_n) - u(x_j, t_n)\right) + \frac{1}{2}\left(u(x_j, t_n) - u(x_{j-1}, t_n)\right) \\ &\quad \nu\left[\frac{1}{2}\left(\frac{1}{2}u^2(x_{j+1}, t_n) - \frac{1}{2}u^2(x_{j-1}, t_n)\right)\right] \\ &= \underbrace{u(x_j, t_{n+1}) - \frac{1}{2}u(x_{j+1}, t_n) - \frac{1}{2}u(x_{j-1}, t_n)}_{=I_1} + \underbrace{\frac{\Delta t}{4\Delta x}\left[u^2(x_{j+1}, t_n) - u^2(x_{j-1}, t_n)\right]}_{=I_2} \end{aligned}$$

The two portions can be evaluated as follows

$$
\begin{aligned}
I_1 &= u_t(x_j,t_n)\Delta t - \frac{1}{2}u_x x(x_j,t_n)\Delta x^2 + O(\Delta t^2) + O(\Delta x^3)\\
I_2 &= \frac{\Delta t}{\Delta x}\left[\frac{1}{4}u^2(x_j,t_n) + \frac{1}{2}uu_x(x_j,t_n)\Delta x + \frac{1}{4}(uu_{xx}+u_x^2)\Delta x^2 + O(\Delta x^3)\right]\\
&\quad - \frac{\Delta t}{\Delta x}\left[\frac{1}{4}u^2(x_j,t_n) - \frac{1}{2}uu_x(x_j,t_n)\Delta x + \frac{1}{4}(uu_{xx}+u_x^2)\Delta x^2 + O(\Delta x^3)\right]\\
&= \frac{\Delta t}{\Delta x}[uu_x(x_j,t_n)\Delta x + O(\Delta x^3)] = \frac{\Delta t}{2\Delta x}\frac{\partial}{\partial x}u^2(x_j,t_n)\Delta x + \frac{\Delta t}{\Delta x}O(\Delta x^3) = \frac{1}{2}\left[u^2(x_j,t_n)\right]_x \Delta t + O(\Delta t\Delta x^2)
\end{aligned}
$$

Thus, plug I_1, I_2 back to the truncation error equation

$$
T_j^n = \frac{I_1+I_2}{\Delta t} = u_t(x_j,t_n) + \frac{1}{2}[u^2(x_j,t_n)]_x + O(\Delta t) + O\left(\frac{\Delta x^2}{\Delta t}\right) \sim O(\Delta t) + O\left(\frac{\Delta x^2}{\Delta t}\right)
$$

Suppose that $\Delta t \sim \Delta x$, i.e., Δt is of the same order as Δx. Then $T_j^n \sim O(\Delta t + \Delta x)$. I.e., the scheme is consistent and the truncation error is of the first order.

For **MONOTONE**, *we first recall the conservative scheme as follows*

$$
U_j^{n+1} = \frac{U_{j+1}^n + U_{j-1}^n}{2} - \frac{\nu}{2}\left[\frac{1}{2}(U_{j+1}^n)^2 - \frac{1}{2}(U_{j-1}^n)^2\right]
$$

The CFL condition should be $\nu|F'| = |\nu U_j^n| \le 1$, due the flux term in the Burgers' equation. Thus

$$
\frac{\partial U_j^{n+1}}{U_{j+1}^n} = \frac{1}{2} - \frac{\nu}{2}U_{j+1}^n \ge 0,\ \frac{\partial U_j^{n+1}}{U_{j+n}^n} = 0,\ \frac{\partial U_j^{n+1}}{U_{j-1}^n} = \frac{1}{2} + \frac{\nu}{2}U_{j-1}^n \ge 0
$$

Thus, the CFL condition implies monotone.

2. *For* **CONSISTENCY**, *we have*

$$
\begin{aligned}
\Delta t T_j^n &= u(x_j,t_{n+1}) - u(x_j,t_n) + \nu\left[\frac{1}{2}\left(\frac{1}{2}u^2(x_{j+1},t_n) + \frac{1}{2}u^2(x_j,t_n)\right) - \frac{\nu}{2}A_{j+1/2}\left(\frac{1}{2}u^2(x_{j+1},t_n) - \frac{1}{2}u^2(x_j,t_n)\right)\right]\\
&\quad - \nu\left[\frac{1}{2}\left(\frac{1}{2}u^2(x_j,t_n) + \frac{1}{2}u^2(x_{j-1},t_n)\right) - \frac{\nu}{2}A_{j-1/2}\left(\frac{1}{2}u^2(x_j,t_n) - \frac{1}{2}u^2(x_{j-1},t_n)\right)\right]\\
&= \underbrace{u(x_j,t_{n+1}) - u(x_j,t_n)}_{=I_1}\\
&\quad + \underbrace{\frac{\nu}{4}\left[u^2(x_{j+1},t_n) - u^2(x_{j-1},t_n)\right] + \frac{\nu^2}{4}\left[A_{j-1/2}\left(u^2(x_j,t_n) - u^2(x_{j-1},t_n)\right) - A_{j+1/2}\left(u^2(x_{j+1},t_n) - u^2(x_j,t_n)\right)\right]}_{=I_2}
\end{aligned}
$$

The two portions can be evaluated as follows

$$
\begin{aligned}
I_1 &= u_t(x_j,t_n)\Delta t + \frac{1}{2}u_t t(x_j,t_n)\Delta t^2 + O(\Delta t^3)\\
I_2 &= \frac{\nu}{4}\left[u^2(x_j,t_n) - 2uu_x(x_j,t_n)\Delta x + (uu_x x + u_x^2)\Delta x^2 - u^2(x_j,t_n) + 2uu_x(x_j,t_n)\Delta x - (uu_x x + u_x^2)\Delta x^2 + O(\Delta x^3)\right]\\
&\quad + \frac{\nu^2}{4}\left[\frac{u(x_j,t_n)+u(x_{j-1},t_n)}{2}\left[u^2(x_j,t_n) - u^2(x_{j-1},t_n)\right] - \frac{u(x_{j+1},t_n)+u(x_j,t_n)}{2}\left[u^2(x_{j+1},t_n) - u^2(x_j,t_n)\right]\right]\\
&= \frac{\nu}{4}\left[4uu_x(x_j,t_n)\Delta x + O(\Delta x^3)\right]\\
&\quad + \frac{\nu^2}{8}\left[2u(x_j,t_n) - u_x\Delta x + \frac{1}{2}u_{xx}\Delta x^2 + O(\Delta x^3)\right]\left[2uu_x\Delta x - (uu_{xx}+u_x^2)\Delta x^2 + O(\Delta x^3)\right]\\
&\quad - \frac{\nu^2}{8}\left[2u(x_j,t_n) + u_x\Delta x + \frac{1}{2}u_{xx}\Delta x^2 + O(\Delta x^3)\right]\left[2uu_x\Delta x + (uu_{xx}+u_x^2)\Delta x^2 + O(\Delta x^3)\right]\\
&= \nu uu_x(x_j,t_n)\Delta x + \nu O(\Delta x^3) - \frac{\nu^2}{2}u^2(x_j,t_n)u_{xx}(x_j,t_n)\Delta x^2 - \nu^2 u(x_j,t_n)u_x^2(x_j,t_n)\Delta x^2 + \nu^2 O(\Delta x^3)\\
&= uu_x(x_j,t_n)\Delta t - \frac{\nu^2}{2}u^2(x_j,t_n)u_{xx}(x_j,t_n)\Delta t^2 - \nu^2 u(x_j,t_n)u_x^2(x_j,t_n)\Delta t^2 + O(\Delta x^2\Delta t + \Delta x\Delta t^2)
\end{aligned}
$$

Thus, suppose $\Delta t \sim \Delta x$, we have

$$T_j^n = \frac{I_1 + I_2}{\Delta t} = u_t + uu_x + \frac{1}{2}\left[u_{tt} - u^2 u_{xx} - 2uu_x^2\right]\Delta t + O(\Delta x^2 + \Delta t^2)$$

In additional, suppose the solution is smooth, we can directly take the partial derivative of the original PDE w.r.t. time

$$0 = u_{tt} + u_t u_x + uu_{xt} = u_{tt} - u_x\left[\frac{1}{2}u^2\right]_x - u\left[\frac{1}{2}u^2\right]_{xx} = u_{tt} - u^2 u_{xx} - 2uu_x^2$$

Therefore, we cannot only ensure the consistency of the scheme, but also observe $T_j^n \sim O(\Delta t^2 + \Delta x^2)$.

For **MONOTONE**, *we need to present the conservative scheme explicitly*

$$\begin{aligned}
U_j^{n+1} &= U_j^n - \nu\left[\frac{1}{2}\left(F_{j+1}^n + F_j^n\right) - \frac{\nu}{2}A_{j+1/2}\left(F_{j+1}^n - F_j^n\right)\right] + \nu\left[\frac{1}{2}\left(F_j^n + F_{j-1}^n\right) - \frac{\nu}{2}A_{j-1/2}\left(F_j^n - F_{j-1}^n\right)\right] \\
&= U_j^n - \frac{\nu}{2}\left[\left(F_{j+1}^n + F_j^n\right) - \left(F_j^n + F_{j-1}^n\right)\right] + \frac{\nu^2}{2}A_{j+1/2}\left(F_{j+1}^n - F_j^n\right) - \frac{\nu^2}{2}A_{j-1/2}\left(F_j^n - F_{j-1}^n\right) \\
&= U_j^n - \frac{\nu}{2}\left[\left(F_{j+1}^n - F_j^n\right) + \left(F_j^n - F_{j-1}^n\right)\right] + \frac{\nu^2}{2}A_{j+1/2}\left(F_{j+1}^n - F_j^n\right) - \frac{\nu^2}{2}A_{j-1/2}\left(F_j^n - F_{j-1}^n\right) \\
&= U_j^n + \left(\frac{\nu^2}{2}A_{j+1/2} - \frac{\nu}{2}\right)\left(F_{j+1}^n - F_j^n\right) - \left(\frac{\nu^2}{2}A_{j-1/2} + \frac{\nu}{2}\right)\left(F_j^n - F_{j-1}^n\right) \\
&= U_j^n + \frac{\nu A_{j+1/2}}{2}\left(\nu A_{j+1/2} - 1\right)\left(U_{j+1}^n - U_j^n\right) + \frac{\nu A_{j-1/2}}{2}\left(\nu A_{j-1/2} + 1\right)\left(U_{j-1}^n - U_j^n\right)
\end{aligned}$$

In the derivation above, we implicitly use the definitions of $F(u) = u^2$ and $A_{j\pm1/2}$. Since the CFL condition is $\left|\nu A_{j+1/2}^n\right| \le 1$, we have

$$\begin{aligned}
\frac{\partial U_j^{n+1}}{\partial U_{j+1}^n} &= \frac{\nu A_{j+1/2}}{2}\left(\nu A_{j+1/2} - 1\right) \ge 0 \\
\frac{\partial U_j^{n+1}}{\partial U_j^n} &= 1 - \frac{\nu A_{j+1/2}}{2}\left(\nu A_{j+1/2} - 1\right) - \frac{\nu A_{j-1/2}}{2}\left(\nu A_{j-1/2} + 1\right) \ge 0 \\
\frac{\partial U_j^{n+1}}{\partial U_{j-1}^n} &= \frac{\nu A_{j-1/2}}{2}\left(\nu A_{j-1/2} + 1\right) \ge 0
\end{aligned}$$

However, we need to consider the numerical scheme centered at x_{j+1} and x_{j-1}. Therefore, by shifting the scheme to U_{j+1}^{n+1} and U_{j-1}^{n+1}, we also require

$$\begin{aligned}
\frac{\partial U_{j+1}^{n+1}}{\partial U_j^n} &= \frac{\nu A_{j+1/2}}{2}\left(\nu A_{j+1/2} + 1\right) \ge 0 \\
\frac{\partial U_{j-1}^{n+1}}{\partial U_j^n} &= \frac{\nu A_{j-1/2}}{2}\left(\nu A_{j-1/2} - 1\right) \ge 0
\end{aligned}$$

WLOG, assume $\nu A_{j+1/2} \ne 0$. Then, in one of the equations, $\nu A_{j+1/2} > 0$ implies $\nu A_{j+1/2} \ge 1$, while in the other one, $\nu A_{j+1/2} < 0$ implies $\nu A_{j+1/2} \le -1$. Similar issues can be found in $\nu A_{j-1/2}$. That already contradicts to the CFL condition.

Moreover, no matter how we choose $\nu A_{j+1/2}$, we can show that one of $\partial U_{j+1}^{n+1}/\partial U_{j+1}^n$ and $\partial U_j^{n+1}/\partial U_j^n$ is not greater than 0, since

$$\begin{aligned}
\frac{\partial U_{j+1}^{n+1}}{\partial U_{j+1}^n} &= 1 - \frac{\nu A_{j+3/2}}{2}\left(\nu A_{j+3/2} - 1\right) - \frac{\nu A_{j+1/2}}{2}\left(\nu A_{j+1/2} + 1\right) \le 1 - \frac{\nu A_{j+1/2}}{2}\left(\nu A_{j+1/2} + 1\right) \le 0, \\
&\qquad \text{if } \nu A_{j+1/2} > 0, \nu A_{j+1/2} \ge 1 \\
\frac{\partial U_j^{n+1}}{\partial U_j^n} &= 1 - \frac{\nu A_{j+1/2}}{2}\left(\nu A_{j+1/2} - 1\right) - \frac{\nu A_{j-1/2}}{2}\left(\nu A_{j-1/2} + 1\right) \le 1 - \frac{\nu A_{j+1/2}}{2}\left(\nu A_{j+1/2} - 1\right) \le 0, \\
&\qquad \text{if } \nu A_{j+1/2} < 0, \nu A_{j+1/2} \le -1
\end{aligned}$$

Similar reasoning can be applied to $\nu A_{j-1/2}$. Thus the conservative scheme is not a monotone scheme.

3. *The numerical results are shown in Fig.3.14. The results shows that the Lax-Friedrichs scheme can present the shock at $X = 1$ and the rarefaction waves at $X = 0, X = 2$ at correct locations. However, the shock is "smoothed" by the Lax-Friedrichs scheme. That is to say, the shock is not a sharp one. Thus, damping or dissipation error can be observed at the shock.*

The Lax-Wendroff scheme can present the rarefaction waves at $X = 0, X = 2$. However, it leads to oscillations at the shock.

That is because the Lax-Wendroff scheme is not monotone, and generally it cannot give the entropy solution.

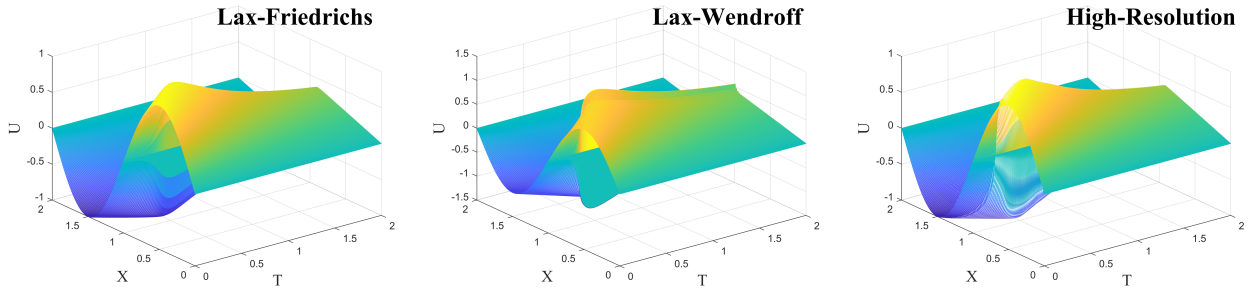

Figure 3.14: Simulation Results with Lax-Friedrichs Scheme, Lax-Wendroff Scheme and High-Resolution Scheme.

4. *The numerical results are shown in Fig.3.14, and the comparisons at specific time steps are shown in Fig.3.15. Note that the flux limiter $\phi(\theta)$ should be calculated based on the upwind direction, i.e.,*

$$h^n_{j+1/2} = h^n_{L,j+1/2} + \phi^n_j \left(h^n_{H,j+1/2} - h^n_{L,j+1/2}\right), \begin{cases} \theta = \dfrac{u^n_j - u^n_{j-1}}{u^n_{j+1} - u^n_j}, & A_{j+1/2} > 0 \\ \theta = \dfrac{u^n_{j+2} - u^n_{j+1}}{u^n_{j+1} - u^n_j}, & A_{j+1/2} < 0 \end{cases}$$

As discussed above, Lax-Friedrichs scheme can generate shock and rarefaction waves, i.e., it is an entropy scheme and provides the entropy solution. However. it induces "damping or dissipation" error at the shock. Lax-Wendroff scheme leads to oscillations at the shock, since it is not monotone or TVD. High-resolution scheme leverages the advantages from both schemes, and it has a sharper shock compared to the Lax-Friedrichs scheme, and it produces relatively small oscillations. In the segments with rarefaction waves, all of the three schemes have similar results.

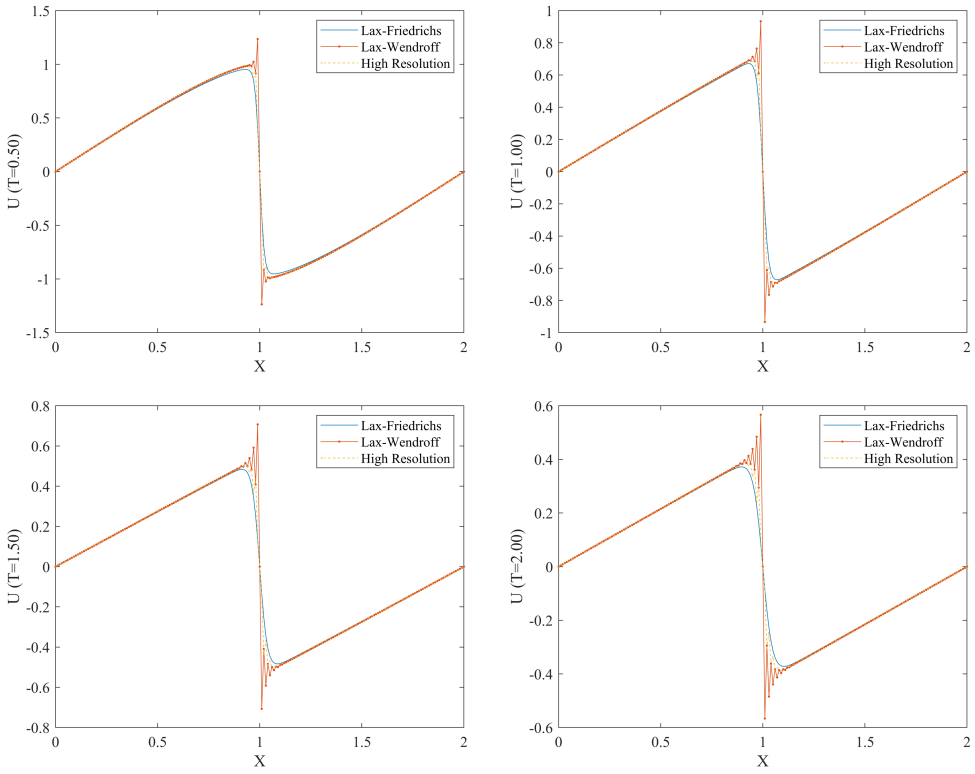

Figure 3.15: The Comparison among Lax-Friedrichs Scheme, Lax-Wendroff Scheme and High-Resolution Scheme.

▲

Appendix. The MATLAB code

```
% ------------ Math 517 Chap.3 Exercise 1 --------
clear all; close all; clc
uu0=@(x) sin(pi*x); DX=0.01;nu=0.5;DT=nu*DX; X=0:DX:2; T=0:DT:2;
[Xaxis,Taxis]=meshgrid(X,T); J=length(X); tF=length(T);
% ------------- Lax-Friedrichs ------------------
U1=zeros(tF,J);
for i=1:J
    U1(1,i)=uu0(X(i)); end
for t=2:tF
    hposi=0.5*(0.5*U1(t-1,2)*U1(t-1,2)+0.5*U1(t-1,1)*U1(t-1,1))-(0.5/nu)*(U1(t-1,2)-U1(t
        -1,1));
    hnega=0.5*(0.5*U1(t-1,1)*U1(t-1,1)+0.5*U1(t-1,J-1)*U1(t-1,J-1))-(0.5/nu)*(U1(t-1,1)-U1(t
        -1,J-1));
    U1(t,1)= U1(t-1,1)-nu*(hposi-hnega);
    for i=2:(J-1)
        hposi=0.5*(0.5*U1(t-1,i+1)*U1(t-1,i+1)+0.5*U1(t-1,i)*U1(t-1,i))-(0.5/nu)*(U1(t-1,i+1)
            -U1(t-1,i));
        hnega=0.5*(0.5*U1(t-1,i)*U1(t-1,i)+0.5*U1(t-1,i-1)*U1(t-1,i-1))-(0.5/nu)*(U1(t-1,i)-
            U1(t-1,i-1));
        U1(t,i)= U1(t-1,i)-nu*(hposi-hnega); end
    hposi=0.5*(0.5*U1(t-1,2)*U1(t-1,2)+0.5*U1(t-1,J)*U1(t-1,J))-(0.5/nu)*(U1(t-1,2)-U1(t-1,J
        ));
    hnega=0.5*(0.5*U1(t-1,J)*U1(t-1,J)+0.5*U1(t-1,J-1)*U1(t-1,J-1))-(0.5/nu)*(U1(t-1,J)-U1(t
        -1,J-1));
    U1(t,J)= U1(t-1,J)-nu*(hposi-hnega); end
figure(1)
mesh(Taxis,Xaxis,U1), xlabel('T'),ylabel('X'),zlabel('U'),
set(get(gca, 'YLabel'), 'FontName','Times New Roman','FontSize', 16)
set(get(gca, 'XLabel'), 'FontName','Times New Roman','FontSize', 16)
set(get(gca, 'ZLabel'), 'FontName','Times New Roman','FontSize', 16)
% ------------- Lax Wendroff --------------------
U2=zeros(tF,J);
for i=1:J
    U2(1,i)=uu0(X(i)); end
for t=2:tF
    Aposi=0.5*(U2(t-1,1)+U2(t-1,2));Anega=0.5*(U2(t-1,J-1)+U2(t-1,1));
    hposi=0.5*(0.5*U2(t-1,2)*U2(t-1,2)+0.5*U2(t-1,1)*U2(t-1,1))-(0.5*nu)*Aposi*(0.5*U2(t
        -1,2)*U2(t-1,2)-0.5*U2(t-1,1)*U2(t-1,1));
    hnega=0.5*(0.5*U2(t-1,1)*U2(t-1,1)+0.5*U2(t-1,J-1)*U2(t-1,J-1))-(0.5*nu)*Anega*(0.5*U2(t
        -1,1)*U2(t-1,1)-0.5*U2(t-1,J-1)*U2(t-1,J-1));
    U2(t,1)= U2(t-1,1)-nu*(hposi-hnega);
    for i=2:(J-1)
        Aposi=0.5*(U2(t-1,i)+U2(t-1,i+1));Anega=0.5*(U2(t-1,i-1)+U2(t-1,i));
        hposi=0.5*(0.5*U2(t-1,i+1)*U2(t-1,i+1)+0.5*U2(t-1,i)*U2(t-1,i))-(0.5*nu)*Aposi*(0.5*
            U2(t-1,i+1)*U2(t-1,i+1)-0.5*U2(t-1,i)*U2(t-1,i));
```

```
        hnega=0.5*(0.5*U2(t-1,i)*U2(t-1,i)+0.5*U2(t-1,i-1)*U2(t-1,i-1))-(0.5*nu)*Anega*(0.5*
            U2(t-1,i)*U2(t-1,i)-0.5*U2(t-1,i-1)*U2(t-1,i-1));
        U2(t,i)= U2(t-1,i)-nu*(hposi-hnega); end
    Aposi=0.5*(U2(t-1,J)+U2(t-1,2));Anega=0.5*(U2(t-1,J-1)+U2(t-1,J));
    hposi=0.5*(0.5*U2(t-1,2)*U2(t-1,2)+0.5*U2(t-1,J)*U2(t-1,J))-(0.5*nu)*Aposi*(0.5*U2(t
        -1,2)*U2(t-1,2)-0.5*U2(t-1,J)*U2(t-1,J));
    hnega=0.5*(0.5*U2(t-1,J)*U2(t-1,J)+0.5*U2(t-1,J-1)*U2(t-1,J-1))-(0.5*nu)*Anega*(0.5*U2(t
        -1,J)*U2(t-1,J)-0.5*U2(t-1,J-1)*U2(t-1,J-1));
    U2(t,J)= U2(t-1,J)-nu*(hposi-hnega); end
figure(2)
mesh(Taxis,Xaxis,U2), xlabel('T'),ylabel('X'),zlabel('U'),
set(get(gca, 'YLabel'), 'FontName','Times New Roman','FontSize', 16)
set(get(gca, 'XLabel'), 'FontName','Times New Roman','FontSize', 16)
set(get(gca, 'ZLabel'), 'FontName','Times New Roman','FontSize', 16)
% ------------- High Resolution Scheme --------------------
UH=zeros(tF,J);
for i=1:J
    UH(1,i)=uu0(X(i)); end
for t=2:tF
    for i=1:J
        if(i==J)
            Aposi=0.5*(UH(t-1,i)+UH(t-1,2));
        else
            Aposi=0.5*(UH(t-1,i)+UH(t-1,i+1));
        end
        if(Aposi>0)
            if(i==1)
                theta=(UH(t-1,i)-UH(t-1,J-1))/(UH(t-1,i+1)-UH(t-1,i));
            elseif(i==J)
                theta=(UH(t-1,i)-UH(t-1,i-1))/(UH(t-1,2)-UH(t-1,i));
            else
                theta=(UH(t-1,i)-UH(t-1,i-1))/(UH(t-1,i+1)-UH(t-1,i));
            end
        else
            if(i==J)
                theta=(UH(t-1,3)-UH(t-1,2))/(UH(t-1,2)-UH(t-1,i));
            elseif(i==(J-1))
                theta=(UH(t-1,2)-UH(t-1,J))/(UH(t-1,J)-UH(t-1,i));
            else
                theta=(UH(t-1,i+2)-UH(t-1,i+1))/(UH(t-1,i+1)-UH(t-1,i));
            end
        end
        phi=max([0,min([1,2*theta]),min([theta,2])]);
        if(i==J)
            hLposi=0.5*(0.5*UH(t-1,2)*UH(t-1,2)+0.5*UH(t-1,i)*UH(t-1,i))-(0.5/nu)*(UH(t-1,2)-
                UH(t-1,i));
            hHposi=0.5*(0.5*UH(t-1,2)*UH(t-1,2)+0.5*UH(t-1,i)*UH(t-1,i))-(0.5*nu)*Aposi*(0.5*
                UH(t-1,2)*UH(t-1,2)-0.5*UH(t-1,i)*UH(t-1,i));
```

```
        else
            hLposi=0.5*(0.5*UH(t-1,i+1)*UH(t-1,i+1)+0.5*UH(t-1,i)*UH(t-1,i))-(0.5/nu)*(UH(t
                -1,i+1)-UH(t-1,i));
            hHposi=0.5*(0.5*UH(t-1,i+1)*UH(t-1,i+1)+0.5*UH(t-1,i)*UH(t-1,i))-(0.5*nu)*Aposi
                *(0.5*UH(t-1,i+1)*UH(t-1,i+1)-0.5*UH(t-1,i)*UH(t-1,i));
        end
        hposi=hLposi+phi*(hHposi-hLposi);
        if(i==1)
            Anega=0.5*(UH(t-1,J-1)+UH(t-1,i));
        else
            Anega=0.5*(UH(t-1,i-1)+UH(t-1,i));
        end
        if(Anega>0)
            if(i==1)
                theta=(UH(t-1,J-1)-UH(t-1,J-2))/(UH(t-1,i)-UH(t-1,J-1));
            elseif(i==2)
                theta=(UH(t-1,1)-UH(t-1,J-1))/(UH(t-1,i)-UH(t-1,1));
            else
                theta=(UH(t-1,i-1)-UH(t-1,i-2))/(UH(t-1,i)-UH(t-1,i-1));
            end
        else
            if(i==J)
                theta=(UH(t-1,2)-UH(t-1,i))/(UH(t-1,i)-UH(t-1,i-1));
            elseif(i==1)
                theta=(UH(t-1,i+1)-UH(t-1,i))/(UH(t-1,i)-UH(t-1,J-1));
            else
                theta=(UH(t-1,i+1)-UH(t-1,i))/(UH(t-1,i)-UH(t-1,i-1));
            end
        end
        phi=max([0,min([1,2*theta]),min([theta,2])]);
        if(i==1)
            hLnega=0.5*(0.5*UH(t-1,i)*UH(t-1,i)+0.5*UH(t-1,J-1)*UH(t-1,J-1))-(0.5/nu)*(UH(t
                -1,i)-UH(t-1,J-1));
            hHnega=0.5*(0.5*UH(t-1,i)*UH(t-1,i)+0.5*UH(t-1,J-1)*UH(t-1,J-1))-(0.5*nu)*Anega
                *(0.5*UH(t-1,i)*UH(t-1,i)-0.5*UH(t-1,J-1)*UH(t-1,J-1));
        else
            hLnega=0.5*(0.5*UH(t-1,i)*UH(t-1,i)+0.5*UH(t-1,i-1)*UH(t-1,i-1))-(0.5/nu)*(UH(t
                -1,i)-UH(t-1,i-1));
            hHnega=0.5*(0.5*UH(t-1,i)*UH(t-1,i)+0.5*UH(t-1,i-1)*UH(t-1,i-1))-(0.5*nu)*Anega
                *(0.5*UH(t-1,i)*UH(t-1,i)-0.5*UH(t-1,i-1)*UH(t-1,i-1));
        end
        hnega=hLnega+phi*(hHnega-hLnega);
        UH(t,i)= UH(t-1,i)-nu*(hposi-hnega); end; end
figure(3)
mesh(Taxis,Xaxis,UH), xlabel('T'),ylabel('X'),zlabel('U'),
set(get(gca, 'YLabel'), 'FontName','Times New Roman','FontSize', 16)
set(get(gca, 'XLabel'), 'FontName','Times New Roman','FontSize', 16)
set(get(gca, 'ZLabel'), 'FontName','Times New Roman','FontSize', 16)
```

```
% ---------------------- Plot at specific times -------------------------------
% ---------------------- t=0.50 -----------------------------
figure(4)
plot(X,U1(101,:),'-',X,U2(101,:),'.-',X,UH(101,:),'--')
xlabel('X'),ylabel('U (T=0.50)')
legend('Lax-Friedrichs','Lax-Wendroff','High Resolution')
set(get(gca, 'YLabel'), 'FontName','Times New Roman','FontSize', 16),
set(get(gca, 'XLabel'), 'FontName','Times New Roman','FontSize', 16),
set(gca, 'FontName','Times New Roman','FontSize', 12)
% ---------------------- t=1.00 -----------------------------
figure(5)
plot(X,U1(201,:),'-',X,U2(201,:),'.-',X,UH(201,:),'--')
xlabel('X'),ylabel('U (T=1.00)')
legend('Lax-Friedrichs','Lax-Wendroff','High Resolution')
set(get(gca, 'YLabel'), 'FontName','Times New Roman','FontSize', 16),
set(get(gca, 'XLabel'), 'FontName','Times New Roman','FontSize', 16),
set(gca, 'FontName','Times New Roman','FontSize', 12)
% ---------------------- t=1.50 -----------------------------
figure(6)
plot(X,U1(301,:),'-',X,U2(301,:),'.-',X,UH(301,:),'--')
xlabel('X'),ylabel('U (T=1.50)')
legend('Lax-Friedrichs','Lax-Wendroff','High Resolution')
set(get(gca, 'YLabel'), 'FontName','Times New Roman','FontSize', 16),
set(get(gca, 'XLabel'), 'FontName','Times New Roman','FontSize', 16),
set(gca, 'FontName','Times New Roman','FontSize', 12)
% ---------------------- t=2.00 -----------------------------
figure(7)
plot(X,U1(401,:),'-',X,U2(401,:),'.-',X,UH(401,:),'--')
xlabel('X'),ylabel('U (T=2.00)')
legend('Lax-Friedrichs','Lax-Wendroff','High Resolution')
set(get(gca, 'YLabel'), 'FontName','Times New Roman','FontSize', 16),
set(get(gca, 'XLabel'), 'FontName','Times New Roman','FontSize', 16),
set(gca, 'FontName','Times New Roman','FontSize', 12)
```

Chapter 4 Hamilton-Jacobi Equation

4.1 Introduction to Viscosity Solutions of Hamilton-Jacobi Equation

We are interested in solving the following **HAMILTON-JACOBI (HJ)** equation with pre-specified IC

$$\begin{aligned} &\text{Common Version:} && \phi_t + H\left(\phi_{x_1}, ..., \phi_{x_d}\right) = 0,\ \phi(x,0) = \phi_0(x) \\ &\text{Simple Version:} && \phi_t + H\left(\nabla\phi\right) = 0,\ \phi(x,0) = \phi_0(x) \\ &\text{General Version:} && \phi_t + H\left(x, \phi, \nabla\phi\right) = 0,\ \phi(x,0) = \phi_0(x) \end{aligned} \tag{4.1}$$

where

1. H is call the **HAMILTONIAN** function. In general, H is nonlinear, but we usually request that H is at least Lipschitz continuous.
2. In some applications, H could also depend on ϕ, x and t, i.e., $H = H(x, \phi, \nabla\phi)$. However, the main difficulty for numerical solutions is the nonlinear dependency of H on the gradient of ϕ.
3. HJ equations often appear in applications, such as image processing and computer vision, control and differential games, geometrical optics, and even classical mechanics.
4. In general, classical solution to Eq.4.1 does not exist.

At least for 1D cases, the HJ equation is equivalent to the CL. In order to see that, take the partial derivative of the PDE w.r.t. x

$$\partial_x\phi_t + \partial_x H\left(\phi_x\right) = 0, \quad \partial_x\phi(x,0) = \partial_x\phi_0(x)$$

Let $u = \phi_x$, we can identify

$$u_t + \partial_x H\left(u\right) = 0,\ u(x,0) = u_0(x) \tag{4.2}$$

Since we want to make analogies to the CL, we recall some properties regarding to Eq.4.2.

1. Strong $\mathcal{C}^1$ solution generically does not exist for arbitrary time points t even if the IC $u_0(x)$ is smooth.
2. Weak solution is defined by an integral procedure. For piecewise smooth solutions, by requiring the solution to satisfy the PDE at all $\mathcal{C}^1$ points and to satisfy a Rankine-Hugoniot jump condition, i.e., Prop.3.2.
$$H(u^+) - H(u^-) = s(u^+ - u^-)$$
along a discontinuity curve in the $x-t$ space, where s is the slope of the curve, i.e., the propagation speed.
3. Weak solutions are not unique. The unique, physically relevant weak solution, a.k.a., the entropy solution, is the one satisfying certain entropy conditions. For piecewise smooth solutions, to test if a weak solution is the entropy solution, one must test the **OLEINIK ENTROPY CONDITION**, i.e., entropy condition I in Def.3.2,
$$\frac{H(u) - H(u^-)}{u - u^-} \geq s \geq \frac{H(u^+) - H(u)}{u^+ - u}$$
for all u between u^- and u^+, along a discontinuity curve in the $x - t$ space. For a convex conservation law $H''(u) \succ 0$, the Oleinik entropy condition simplifies to the **LAX ENTROPY CONDITION**,
$$u^- > u^+$$
That is, the solution can only "jump down".

Because of the "equivalent" between the HJ equation and CL, we can infer the difficulties of the HJ equation in

the way similar to the CL.

Thus, for the HJ equation, i.e., the common version in Eq.4.1, we can have the following claims.

1. Strong $\mathcal{C}^1$ solution generically does not exist for arbitrary time points t even if the initial condition $\phi^0(x)$ is smooth.
2. Among the Lipschitz continuous but not necessarily C^1 segments or solutions, the unique viscosity solution is defined via the concept of **VISCOSITY SUB-SOLUTION** and **VISCOSITY SUPER-SOLUTION**, shown in Def.4.1.
3. The viscosity solution is unique. According to the Lax entropy condition, for convex Hamiltonian $H'' \succ 0$, the kinks, i.e., discontinuities in the first derivatives, of the viscosity solution can only point upwards $(\phi_x^- > \phi_x^+)$

Definition 4.1 (Viscosity Sub-, Super-, Solutions)

1. *ϕ is called a viscosity sub-solution of the HJ equation if, for any smooth function ψ, at each local maximum point $(\overline{x}, \overline{t})$ of $\phi - \psi$, we have the inequality*
$$\psi_t(\overline{x}, \overline{t}) + H\left(\psi_{x_1}(\overline{x}, \overline{t}), ..., \psi_{x_d}(\overline{x}, \overline{t})\right) \leq 0$$
2. *ϕ is called a viscosity super-solution of the HJ equation if, for any smooth function ψ, at each local minimum point $(\overline{x}, \overline{t})$ of $\phi - \psi$, we have the inequality*
$$\psi_t(\overline{x}, \overline{t}) + H\left(\psi_{x_1}(\overline{x}, \overline{t}), ..., \psi_{x_d}(\overline{x}, \overline{t})\right) \geq 0$$
3. *ϕ is called the viscosity solution to the HJ equation if it is both a viscosity sub-solution and a viscosity super-solution. Viscosity solution is unique.*

♣

Proposition 4.1 (Viscosity Sub-, Super-Solutions)

Let ϕ be a classical solution to Eq.4.1 and ψ be any $\mathcal{C}^2$ function. Then if $\phi - \psi$ has a local maximum (local minimum) at a point $(\overline{x}, \overline{t})$, then
$$\psi_t(\overline{x}, \overline{t}) + H\left(\nabla\psi\right) \leq (\geq) 0$$

♠

Proof ϕ is the classical solution, and hence it is the varnishing viscosity solution to
$$\phi_t(\overline{x}, \overline{t}) + H\left(\nabla\phi\right) = \varepsilon\nabla^2\phi,\ \varepsilon \to 0$$
Suppose $\phi - \psi$ has a local maximum at $(\overline{x}, \overline{t})$, and use the maximum principle
$$\nabla\phi = \nabla\psi,\ \nabla^2\phi \leq \nabla^2\psi$$
Hence
$$\phi_t(\overline{x}, \overline{t}) + H\left(\nabla\phi\right) = \varepsilon\nabla^2\phi \leq \varepsilon\nabla^2\psi = \psi_t(\overline{x}, \overline{t}) + H\left(\nabla\psi\right),\ \varepsilon \to 0$$
Similar to the local minimum case. ■

4.2 1D First Order Monotone Schemes

For simplicity, We first consider 1D case with periodic BCs and uniform meshes. Various generalizations will be discussed later. The simple case is shown as follows

$$\phi_t + H(\phi_x) = 0,\ 0 \le x < 1 \tag{4.3}$$

$[0, 1]$ is covered by a uniform mesh $\{x_i\}_{i\in\mathbb{N}} = \{i\Delta x\}_{i\in\mathbb{N}}$. The approximation of $\phi(x_i, t)$ is denoted by $\phi_i(t)$, or just ϕ_i. Discretizing x first, we use the standard notation,

$$\Delta_{+x}\phi_i = \phi_{i+1} - \phi_i,\ \Delta_{-x}\phi_i = \phi_i - \phi_{i-1}$$

Definition 4.2 (First Order Monotone Scheme)

The first order monotone schemes are defined as schemes of the form

$$\frac{d}{dt}\phi_i = -\widehat{H}\left(\frac{\Delta_{-x}\phi_i}{\Delta x}, \frac{\Delta_{+x}\phi_i}{\Delta x}\right)$$

where $\widehat{H}$ is called a **NUMERICAL HAMILTONIAN**, *which is Lipschitz continuous w.r.t. both arguments and consistent with the Hamiltonian H in the PDE, i.e.,*

$$\widehat{H}(u, u) = H(u)$$

A monotone numerical Hamiltonian $\widehat{H}$ is one which is monotonically non-decreasing in the first argument and monotonically non-increasing in the second argument, symbolically represented as $\widehat{H}(\uparrow, \downarrow)$. ♣

Remark *if $\widehat{H}$ is a monotone numerical Hamiltonian, then $\widehat{H}$ is increasing in ϕ_i and decreasing in ϕ_{i-1} and ϕ_{i+1}, i.e.,*

$$\widehat{H}\left(\frac{\Delta_{-x}\phi_i}{\Delta x}, \frac{\Delta_{+x}\phi_i}{\Delta x}\right) = \widehat{H}\left(\frac{\phi_i - \phi_{i-1}}{\Delta x}, \frac{\phi_{i+1} - \phi_i}{\Delta x}\right)$$

After the spatial discretization, we shift to the full first order scheme. Start with the **SEMI-DISCRETE (CONTINUOUS-IN-TIME) FORM**

$$\frac{d}{dt}\phi_i = -\widehat{H}\left(\frac{\Delta_{-x}\phi_i}{\Delta x}, \frac{\Delta_{+x}\phi_i}{\Delta x}\right) \tag{4.4}$$

When a monotone numerical Hamiltonian is adopted, the scheme above becomes a monotone schemes, a.k.a., a semi-discrete (continuous in time) form of the monotone scheme.

The fully discrete scheme can be obtained by using the forward Euler scheme in the time domain. It is still called a monotone scheme, e.g.,

$$\phi_i^{n+1} = \phi_i^n - \Delta t\widehat{H}\left(\frac{\Delta_{-x}\phi_i}{\Delta x}, \frac{\Delta_{+x}\phi_i}{\Delta x}\right) \tag{4.5}$$

The CFL condition will also be applied.

Proposition 4.2 (Properties of Monotone Schemes)

The monotone schemes have the following properties.

1. *monotone schemes are stable w.r.t. $\mathcal{L}^\infty$ norm;*
2. *solutions using monotone schemes can converge to the viscosity solution of the PDE;*
3. *the error between the numerical solution of the monotone scheme and the exact viscosity solution*

of the PDE, measured in $\mathcal{L}^\infty$ norm, is at least one half order, i.e., $O(\sqrt{\Delta x})$. ♠

Remark *(Notations Regarding to Monotone Schemes)*

1. *The relatively low accuracy, i.e., $O(\sqrt{\Delta x})$, is not a particular concern for viscosity solutions containing kinks. In fact, for many cases, estimations with errors of one half order is optimal.*
2. *However, it is an unfortunate fact that for smooth solutions, the monotone schemes cannot provide a solution with accuracy higher than the first order. This is indeed a serious concern, as we hope the scheme can reach high order accuracy for smooth solutions, or in smooth regions of non-smooth solutions. Monotone schemes would not be able to achieve this.*
3. *Nevertheless, the importance of monotone schemes is that they are often used as building blocks for high order schemes. All the high order schemes are built upon first order monotone schemes. Recall the low order numerical flux functions used in the high resolution schemes for CLs.*

Example 4.1 **(Monotone Numerical Hamiltonians)**

1. **LAX-FRIEDRICHS:**
$$\widehat{H}^{LF}(u^-, u^+) = H\left(\frac{u^- + u^+}{2}\right) - \frac{1}{2}\alpha(u^+ - u^-)$$
α is defined to make the numerical Hamiltonian to be a monotone scheme
$$\alpha = \max_{A \le u \le B} |H'(u)|$$
H' is the derivative of H w.r.t u if it is $\mathcal{C}^1$, or the Lipschitz constant of H if it is Lipschitz continuous but not $\mathcal{C}^1$. It can be easily shown that $\widehat{H}^{LF}$ is a monotone scheme for $A \le u \le B$.
2. **MODIFIED LAX-FRIEDRICHS:**
$$\widehat{H}^{LFa}(u^-, u^+) = \frac{1}{2}\left[H(u^-) + H(u^+) - \alpha(u^+ - u^-)\right]$$
where α is also defined as
$$\alpha = \max_{A \le u \le B} |H'(u)|$$
3. **LOCAL LAX-FRIEDRICHS:**
$$\widehat{H}^{LLF}(u^-, u^+) = H\left(\frac{u^- + u^+}{2}\right) - \frac{1}{2}\alpha(u^-, u^+)(u^+ - u^-)$$
where
$$\alpha(u^-, u^+) = \max_{u \in I(u^-, u^+)} |H'(u)|,\ I(a,b) = [\min(a,b), \max(a,b)]$$
This is also a monotone Hamiltonian but it has smaller dissipation than the (global) Lax-Friedrichs Hamiltonian, $\widehat{H}^{LF}$. Recall that α corresponds to the viscosity term in the HJ equation, and it is smaller than the general Lax-Friedrichs scheme. Thus, we have less viscosity for this case.
4. **GODUNOV:**
$$\widehat{H}^{G}(u^-, u^+) = \text{ext}_{u \in I(u^-, u^+)} H(u)$$
where the function ext() is defined by
$$\text{ext}_{u \in I(a,b)} = \begin{cases} \min\limits_{a \le u \le b}, & \text{if } a \le b \\ \max\limits_{b \le u \le a}, & \text{if } b \le a \end{cases}$$

5. **ROE:**

$$\widehat{H}^{RF}(u^-,u^+) = \begin{cases} H(u^*), & \text{case 1} \\ H\left(\frac{u^-+u^+}{2}\right) - \frac{1}{2}\alpha(u^-,u^+)(u^+-u^-), & \text{case 2} \end{cases}$$

where

$$u^* = \begin{cases} u^-, & \text{if } H'(u) \geq 0 \\ u^+, & \text{if } H'(u) \leq 0 \end{cases}$$

Case 1 refers to the situation when $H'(u)$ does not change sign for $u \in I(u^-,u^+)$, and Case 2 refers to the remaining situations. Roe Hamiltonian with local Lax-Friedrichs Hamiltonian (or say entropy flux) is easy to code and has a small numerical viscosity, hence it is quite popular.

6. **OSHER-SETHIAN:**

If the Hamiltonian in the HJ equation is of the form $H(u) = f(u^2)$ and f is a monotone function, then a simple Osher-Sethian Hamiltonian is defined as

$$\widehat{H}^{OS}(u^-,u^+) = f(\overline{u}^2)$$

where $\overline{u}^2$ is defined as

$$\overline{u}^2 = \begin{cases} \left[\min(u^-,0)\right]^2 + \left[\max(u^+,0)\right]^2, & \text{if } f(\downarrow) \\ \left[\min(u^+,0)\right]^2 + \left[\max(u^-,0)\right]^2, & \text{if } f(\uparrow) \end{cases}$$

This numerical Hamiltonian is purely upwind and easy to program. Hence it should be used whenever possible.

▲

We now discuss some generalizations, for non-uniform and non-smooth meshes. We need to make the following change, i.e., replacing the numerical Hamiltonian in Def.4.2

$$\widehat{H}\left(\frac{\Delta_{-x}\phi_i}{\Delta x}, \frac{\Delta_{+x}\phi_i}{\Delta x}\right) \xleftarrow{\text{Replace by}} \widehat{H}\left(\frac{\Delta_{-x}\phi_i}{\Delta x_{i-1/2}}, \frac{\Delta_{+x}\phi_i}{\Delta x_{i+1/2}}\right), \Delta x_{i-1/2} = x_i - x_{i-1}, \Delta x_{i+1/2} = x_{i+1} - x_i$$

Note that $x_{i\pm1/2}$ may not be equal since the mesh is not uniform. There is no change to the numerical Hamiltonian $\widehat{H}$, and all the conclusions discussed above are still valid.

Suppose we finish discussing the numerical Hamiltonian now, and we obtain the semi-discrete scheme for the 1D HJ equation, i.e., $\phi_t + H(\phi_x) = 0$, with pre-specified ICs and period BCs. The semi-discrete scheme is similar to the one shown in Def.4.2,

$$\frac{d\phi_i}{dt} + \widehat{H}(u_i^-,u_i^+) = 0, \text{ where} u_i^- = \frac{\Delta_{-x}\phi_i}{\Delta x}, u_i^+ = \frac{\Delta_{+x}\phi_i}{\Delta x} \tag{4.6}$$

i.e., u_i^- is the left derivative, and u_i^+ is the right derivative, which can be used for a first order scheme. Consistency of the scheme can be expressed as

$$\widehat{H}(u,u) = H(u)$$

As for the time domain, we may use the forward Euler scheme, i.e.,

$$\phi_i^{n+1} = \phi_i^n - \Delta t \widehat{H}\left(\frac{\Delta_{-x}\phi_i}{\Delta x}, \frac{\Delta_{+x}\phi_i}{\Delta x}\right)$$

or the three point scheme, i.e.,

$$\phi_j^{n+1} = Q(\phi_{i-1}^n, \phi_i^n, \phi_{i+1}^n)$$

As α becoming larger and larger in the series of Lax-Friedrichs scheme, the viscosity, and hence the dissipation error, will increases.

If we change the periodic BCs to the non-periodic BCs.

1. If $H' > 0$, $\phi(0,t) = g(t)$
2. If $H' < 0$, $\phi(1,t) = g(t)$

These two BCs are necessary for the PDE. For purely upwind first order schemes, such as Godunov, Roe with Lax-Friedrichs Hamiltonian (or say entropy flux), Osher-Sethian, and etc., there is no need to provide a numerical BC. However, for Lax-Friedrichs scheme, both the left and right derivatives should be used, where numerical BCs with polynomial extrapolations are needed to obtain the BCs one step out of the spatial grid, i.e., at "-1" or "$J+1$".

4.3 2D Schemes

To multi-dimensional space with Cartesian meshes, the numerical Hamiltonian is the only thing that needs to be changed. The new numerical Hamiltonian will depends on more variables. We take the 2D case as an example. The 2D mesh is denoted by (x_i, y_j), $x_i = i\Delta x$, $y_j = j\Delta y$. The numerical approximation to $\phi(x_i, y_j, t)$ is denoted as $\phi_{i,j}$. We adopted the standard notations.

$$\Delta_{\pm x}\phi_{i,j} = \pm(\phi_{i\pm1,j} - \phi_{i,j}),\ \Delta_{\pm y}\phi_{i,j} = \pm(\phi_{i,j\pm1} - \phi_{i,j})$$

A first order monotone scheme is defined as follows.

$$\frac{d}{dt}\phi_{i,j} = -\widehat{H}\left(\frac{\Delta_{-x}\phi_{i,j}}{\Delta x}, \frac{\Delta_{+x}\phi_{i,j}}{\Delta x}; \frac{\Delta_{-y}\phi_{i,j}}{\Delta y}, \frac{\Delta_{+y}\phi_{i,j}}{\Delta y}\right) \tag{4.7}$$

where the numerical Hamiltonian $\widehat{H}$ is Lipschitz continuous w.r.t. all the four arguments, and $\widehat{H}$ is consistent with the Hamiltonian H, i.e.,

$$\widehat{H}(u,u;v,v) = H(u,v)$$

For a monotone numerical Hamiltonian $\widehat{H}$, it is monotonically non-decreasing in the first and third arguments and monotonically non-increasing in the other two. This can be symbolically represented as

$$\widehat{H}(\uparrow,\downarrow;\uparrow,\downarrow)$$

The scheme in Eq.4.7 with a monotone numerical Hamiltonian is called a **MONOTONE SCHEME**. All the conclusions for 1D monotone schemes still hold in 2D or higher dimensional cases.

Example 4.2 (Monotone Numerical Hamiltonians)

1. **LAX-FRIEDRICHS:**

$$\widehat{H}^{LF}(u^-,u^+;v^-,v^+) = H\left(\frac{u^-+u^+}{2}, \frac{v^-+v^+}{2}\right) - \frac{1}{2}\alpha^x(u^+-u^-) - \frac{1}{2}\alpha^y(v^+-v^-)$$

where α^x, α^y are defined to make the numerical Hamiltonian to be a monotone scheme

$$\alpha^x = \max_{A\le u\le B; C\le v\le D}|H_1(u,v)|,\ \alpha^y = \max_{A\le u\le B; C\le v\le D}|H_2(u,v)|$$

where $H_i(u,v)$ is the partial derivative of H w.r.t. the i^{th} argument, or the Lipschitz constant of H with respect to the i^{th} argument. It can be easily shown that $\widehat{H}^{LF}$ is a monotone scheme for $A \le u \le B$, $C \le v \le D$.

2. **MODIFIED LAX-FRIEDRICHS:**

$$\widehat{H}^{LFa}(u^-,u^+;v^-,v^+) = \frac{1}{4}\left[H(u^-,v^-)+H(u^+,v^-)+H(u^-,v^+)+H(u^+,v^+)\right] - \frac{1}{2}\alpha^x(u^+-u^-) - \frac{1}{2}\alpha^y(v^+-v^-)$$

where α^x, α^y are defined as

$$\alpha^x = \max_{A\le x\le B; C\le v\le D} |H_1(u,v)|,\ \alpha^y = \max_{A\le u\le B; C\le v\le D} |H_2(u,v)|$$

3. **LOCAL LAX-FRIEDRICHS:**

$$\widehat{H}^{LLF}(u^-,u^+;v^-,v^+) = H\left(\frac{u^-+u^+}{2},\frac{v^-+v^+}{2}\right) - \frac{1}{2}\alpha^x(u^-,u^+)(u^+-u^-) - \frac{1}{2}\alpha^y(v^-,v^+)(v^+-v^-)$$

where

$$\alpha^x(u^-,u^+) = \max_{x\in I(u^-,u^+); C\le v\le D} |H_1(u,v)|,\ \alpha^y(v^-,v^+) = \max_{A\le x\le B; v\in I(v^-,v^+)} |H_2(u,v)|$$

where: $I(a,b) = [\min(a,b), \max(a,b)]$

We can prove that the local Lax-Friedrichs Hamiltonian $\widehat{H}^{LLF}$ is monotone for $A \le u \le B$ and $C \le v \le D$. The local Lax-Friedrichs Hamiltonian $\widehat{H}^{LLF}$ has smaller dissipation than the global one, i.e., $\widehat{H}^{LF}$, for it has smaller viscosity coefficients $\alpha^x(u^-,u^+)$; $\alpha^y(v^-,v^+)$.

We can image a "more" local Lax-Friedrichs Hamiltonian as

$$\widehat{H}^{LLLF}(u^-,u^+;v^-,v^+) = H\left(\frac{u^-+u^+}{2},\frac{v^-+v^+}{2}\right) - \frac{1}{2}\alpha^x(u^-,u^+;v^-,v^+)(u^+-u^-) - \frac{1}{2}\alpha^y(u^-,u^+;v^-,v^+)(v^+-v^-)$$

where

$$\alpha^x(u^-,u^+;v^-,v^+) = \max_{x\in I(u^-,u^+); v\in I(v^-,v^+)} |H_1(u,v)|$$

$$\alpha^y(u^-,u^+;v^-,v^+) = \max_{x\in I(u^-,u^+); v\in I(v^-,v^+)} |H_2(u,v)|$$

This would be easier to compute and also would have even smaller dissipation than the local Lax-Friedrichs Hamiltonian $\widehat{H}^{LLF}$ defined above. Unfortunately, $\widehat{H}^{LLLF}$ is not a monotone Hamiltonian.

4. **GODUNOV:**

$$\widehat{H}^{G}(u^-,u^+;v^-,v^+) = \text{ext}_{u\in I(u^-,u^+)}\text{ext}_{v\in I(v^-,v^+)}H(u,v)$$

where the function ext() is defined by

$$\text{ext}_{u\in I(a,b)} = \begin{cases} \min\limits_{a\le u\le b}, & \text{if } a\le b \\ \max\limits_{b\le u\le a}, & \text{if } b\le a \end{cases}$$

Notice that in general

$$\min_u \max_v H(u,v) \ne \max_v \min_u H(u,v)$$

Thus, we may obtain different versions of the Godunov type Hamiltonians $\widehat{H}^G$ by changing the order of the min and max operators.

5. **ROE:**

Roe Hamiltonian with entropy flux is given as

$$\widehat{H}^{RF}(u^-,u^+) = \begin{cases} H(u^*,v^*), & \text{case 1} \\ H\left(\dfrac{u^-+u^+}{2},v^*\right) - \dfrac{1}{2}\alpha^x(u^-,u^+)(u^+-u^-), & \text{case 2} \\ H\left(u^*,\dfrac{v^-+v^+}{2}\right) - \dfrac{1}{2}\alpha^y(u^-,u^+)(v^+-v^-), & \text{case 3} \\ \widehat{H}^{LLF}\left(u^-,u^+;v^+,v^-\right), & \text{case 4} \end{cases}$$

(a). Case 1 refers to the situation when $H_1(u,v)$ and $H_2(u,v)$ do not change sign for $u \in I(u^-,u^+)$ and $v \in I(v^-,v^+)$.

(b). Case 2 refers to the remaining situations and when $H_2(u,v)$ does not change sign for $A \le u \le B$ and $v \in I(v^-,v^+)$.

(c). Case 3 refers to the remaining situations and when $H_1(u,v)$ does not change sign for $u \in I(u^-,u^+)$ and $C \le v \le D$.

(d). Case 4 refers to all remaining situations.

$$u^* = \begin{cases} u^-, & \text{if } H_1(u,v) \ge 0 \\ u^+, & \text{if } H_1(u,v) \le 0 \end{cases} \quad v^* = \begin{cases} v^-, & \text{if } H_2(u,v) \ge 0 \\ v^+, & \text{if } H_2(u,v) \le 0 \end{cases}$$

This Roe Hamiltonian with local Lax-Friedrichs entropy flux is easy to code and has a numerical viscosity almost as small as the (much more complicated) Godunov scheme. Hence, Roe Hamiltonia is quite popular.

6. **OSHER-SETHIAN:**

If the Hamiltonian in the HJ equation is of the form $H(u) = f(u^2,v^2)$, where f is a monotone function w.r.t. both arguments, then the simple Osher-Sethian Hamiltonian is defined as

$$\widehat{H}^{OS}(u^-,u^+;v^-,v^+) = f(\overline{u}^2,\overline{v}^2)$$

where $\overline{u}^2$ and $\overline{v}^2$ are defined as

$$\overline{u}^2 = \begin{cases} \left[\min(u^-,0)\right]^2 + \left[\max(u^+,0)\right]^2, & \text{if } f(\downarrow,\cdot) \\ \left[\min(u^+,0)\right]^2 + \left[\max(u^-,0)\right]^2, & \text{if } f(\uparrow,\cdot) \end{cases} \quad \overline{v}^2 = \begin{cases} \left[\min(v^-,0)\right]^2 + \left[\max(v^+,0)\right]^2, & \text{if } f(\cdot,\downarrow) \\ \left[\min(v^+,0)\right]^2 + \left[\max(v^-,0)\right]^2, & \text{if } f(\cdot,\uparrow) \end{cases}$$

This numerical Hamiltonian is purely upwind and easy to program. Hence, it should be used whenever possible. However, not all of the Hamiltonians can be written in the form $f(u^2,v^2)$ with a monotone function f.

▲

4.4 Time Domain Discretization

Time domain discretization can be done using the TVD **RUNGE-KUTTA (RK)** method. For a semi-discrete scheme

$$\frac{du}{dt} = \mathcal{L}(u)$$

where $\mathcal{L}(u)$ is a discretized spatial operator, the third order TVD RK method can be simply stated as,

$$\begin{aligned} u^{(1)} &= u^n + \Delta t\mathcal{L}(u^n) \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t\mathcal{L}(u^{(1)}) \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t\mathcal{L}(u^{(2)}) \end{aligned}$$

4.4.1 Introduction to the Runge-Kutta Method

RK methods are a series of single-step methods, but with multiple stages for an individual step. They are motivated by the Taylor approximations for specific IVPs. RK does not require derivatives of the RHS function f, and is therefore a general-purpose IVP solver. RK is one of the most popular ODE solvers, which was first studied by Carle Runge and Martin Kutta around 1900. Modern developments were mostly done by John Butcher in the 1960s.

In order to provide an illustration to the **SECOND ORDER RK METHOD**, WLOG, we consider a first-order ODE system

$$y'(t) = f(t, y(t))$$

Since we want to construct a second-order method, start with the Taylor expansion,

$$y(t + \Delta t) = y(t) + y'(t)\Delta t + \frac{1}{2}y''(t)(\Delta t)^2 + O(\Delta t^3)$$

The first order derivative $y'(t)$ can be replaced by the RHS function f, and the second order derivative $y''(t)$ can be obtained by differentiating f, i.e.,

$$\begin{aligned} y'(t) &= f(t, y(t)) \\ y''(t) &= f_t(t, y(t)) + f_y(t, y(t))y'(t) = f_t(t, y(t)) + f_y(t, y(t))f(t, y(t)) \end{aligned}$$

Therefore, the Taylor expansion becomes

$$\begin{aligned} y(t + \Delta t) &= y(t) + y'(t)\Delta t + \frac{1}{2}y''(t)(\Delta t)^2 + O(\Delta t^3) \\ &= y(t) + f(t, y(t))\Delta t + \frac{1}{2}\left[f_t(t, y(t)) + f_y(t, y(t))f(t, y(t))\right](\Delta t)^2 + O(\Delta t^3) = I \end{aligned}$$

Using the multivariate Taylor equation

$$f(t + \Delta t, y(t) + \Delta t f(t, y(t))) = f(t, y) + \Delta t f_t(t, y) + \Delta t f_y(t, y)f(t, y) + O(\Delta t^2)$$

we can obtain that

$$\begin{aligned} I &= y(t) + f(t, y(t))\Delta t + \frac{1}{2}\left[f_t(t, y(t)) + f_y(t, y(t))f(t, y(t))\right](\Delta t)^2 + O(\Delta t^3) \\ &= y(t) + \frac{1}{2}f(t, y(t))\Delta t + \frac{1}{2}\Delta t\left[f(t, y(t)) + f_t(t, y(t))\Delta t + f_y(t, y(t))f(t, y(t))\Delta t\right] + O(\Delta t^3) \\ &= y(t) + \frac{1}{2}f(t, y(t))\Delta t + \frac{1}{2}\Delta t\left[f(t + \Delta t, y(t) + \Delta t f(t, y(t))) - O(\Delta t^2)\right] + O(\Delta t^3) \\ &= y(t) + \frac{1}{2}f(t, y(t))\Delta t + \frac{1}{2}f(t + \Delta t, y(t) + \Delta t f(t, y(t)))\Delta t + O(\Delta t^3) \end{aligned}$$

Therefore, we have

$$y(t + \Delta t) = y(t) + \frac{1}{2}f(t, y(t))\Delta t + \frac{1}{2}f(t + \Delta t, y(t) + \Delta t f(t, y(t)))\Delta t + O(\Delta t^3)$$

or we write it into a numerical scheme

$$y^{n+1} = y^n + \Delta t\left(\frac{1}{2}K_1 + \frac{1}{2}K_2\right), \text{where } \begin{cases} K_1 = f(t_n, y^n) \\ K_2 = f(t_n + \Delta t, y^n + \Delta t K_1) \end{cases}$$

This is the classical second order RK method. It is also known as the **HEUNS METHOD** or the **IMPROVED EULER METHOD**. Notice that we implicitly made a time domain integral, so although the residue is $O(\Delta t^3)$, the scheme has the second order accuracy w.r.t. the orignal PDE.

Remark *(Second Order RK Method)*

1. *The K_1 and K_2 are known as the* **STAGES** *of the RK method. They represent different estimates for the slope of the solution. Note that $y^n + \Delta t K_1$ is actually an Euler step with step size Δt, starting from*

(t_n, y^n). *Therefore, K_2 corresponds to another slope, and that slope is obtained f with the arguments corresponding to the values after taking one Euler step. In another word, the arguments of f in K_2 is the solution at $t_n + \Delta t$ with the forward Euler scheme. The second order RK method now consists of a single step with the average of the slopes K_1 and K_2.*

2. *The classical second order RK method can also be interpreted as a prediction-correction where the forward Euler method is used as the predictor for a (implicit) trapezoidal rule.*
3. *Another interesting way to understand the second order RK method is to use the integral*

$$y^{n+1} = y^n + \int_{t_n}^{t_{n+1}} f(t, y(t))dt \approx y^n + \Delta t \left(\frac{1}{2}K_1 + \frac{1}{2}K_2\right)$$

The second order RK method looks like a trapezium rule for evaluating the integral above.

A general explicit second order RK method can be presented as

$$y(t + \Delta t) = y(t) + \Delta t \left(b_1 \widetilde{K}_1 + b_2 \widetilde{K}_2\right) + O(\Delta t^3), \text{ with } \begin{cases} \widetilde{K}_1 = f(t_n, y^n) \\ \widetilde{K}_2 = f(t_n + c_2\Delta t, y^n + \Delta t a_{21} K_1) \end{cases} \tag{4.8}$$

Clearly, this is a generalization of the classical RK method since $b_1 = b_2 = 1/2$ and $c_2 = a_{21} = 1$ yields the classical one. It is customary to arrange the coefficients a_{ij} , b_i, and c_i in a so-called RK or **BUTCHER TABLEAUX** as follows:

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} \tag{4.9}$$

As an example, the the Butcher tableaux for the classical second-order RK method is

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & 1/2 & 1/2 \end{array}$$

Explicit RK methods are characterized by a strictly lower triangular matrix A, i.e., $a_{ij} = 0$ if $j \geq i$. Moreover, c_i and a_{ij} are connected

$$c_i = \sum_{j=1}^{s} a_{ij},\ i = 1, 2, ..., s$$

This says that c_i is the row sum of the ith row of the matrix A. This condition is required for a first order method, which further ensure the consistency. In a simple word, e.g., $\widetilde{K}_2 = f(t_n + c_2\Delta t, y^n + \Delta t a_{21} K_1)$, the first and second arguments in f should advance the same time period. We limit our discussion to the cases satisfying such a requirement.

Example 4.3 (Variations of RK Method)

For an explicit second order RK method, it is necessary to have $a_{11} = a_{12} = a_{22} = c_1 = 0$. We can now study what other combinations of b_1, b_2, c_2 and a_{21} can be used in

$$y(t + \Delta t) = y(t) + \Delta t \left(b_1 \widetilde{K}_1 + b_2 \widetilde{K}_2\right) + O(\Delta t^3)$$

to achieve another second order method.

The bivariate Taylor expansion yields

$$\begin{aligned} f(t + c_2\Delta t, y + \Delta t a_{21}\widetilde{K}_1) =& f(t, y) + c_2\Delta t f_t(t, y) + \Delta t a_{21}\widetilde{K}_1 f_y(t, y) + O(\Delta t^2) \\ =& f(t, y) + c_2\Delta t f_t(t, y) + \Delta t a_{21} f_y(t, y) f(t, y) + O(\Delta t^2) \end{aligned}$$

Then, the general second order RK formulation becomes

$$\begin{aligned} y(t+\Delta t) =& y(t) + \Delta t \left\{ b_1 f(t,y) + b_2 \left[f(t,y) + c_2 \Delta t f_t(t,y) + \Delta t a_{21} f_y(t,y) f(t,y) \right] \right\} + O(\Delta t^3) \\ =& y(t) + \Delta t (b_1 + b_2) f(t,y) + b_2 \Delta t^2 \left[c_2 f_t(t,y) + a_{21} f_y(t,y) f(t,y) \right] + O(\Delta t^3) \end{aligned}$$

In order for this to match the general Taylor expansion we want

$$\begin{cases} b_1 + b_2 = 1 \\ c_2 b_2 = 1/2 \\ a_{21} b_2 = 1/2 \end{cases}$$

Thus, we have a system of three nonlinear equations with four unknowns. One popular choice, beside the classical one, is $b_1 = 0$, $b_2 = 1$, and $c_2 = a_{21} = 1/2$. This leads to the **MODIFIED EULER METHOD**. Sometimes, it is also referred to as the **MIDPOINT RULE**, i.e.,

$$y^{n+1} = y^n + \Delta t K_2, \text{ with } \begin{cases} K_1 = f(t_n, y^n) \\ K_2 = f\left(t_n + \dfrac{\Delta t}{2}, y^n + \dfrac{\Delta t}{2} K_1 \right) \end{cases}$$

Its Butcher tableaux is of the form

0	0	0
1/2	1/2	0
	0	1

▲

Remark *(Explicit Euler Scheme)*
The choice $b_1 = 1$, $b_2 = 0$ leads to the explicit Euler scheme. However, since $c_2 b_2 \neq \frac{1}{2}$ and $a_{21} b_2 \neq \frac{1}{2}$, the explicit Euler scheme does not have second order accuracy.

General explicit RK methods are of the form

$$y^{n+1} = y^n + \Delta t \sum_{j=1}^{s} b_j K_j, \text{with} \begin{cases} K_1 = f\left(t_n, y^n\right) \\ K_2 = f\left(t_n + c_2 \Delta t, y^n + \Delta t a_{21} K_1\right) \\ \cdots\cdots \\ K_s = f\left(t_n + c_s \Delta t, y^n + \Delta t \sum_{j=1}^{s-1} a_{sj} K_j \right) \end{cases}$$

Determining the coefficients, e.g., $\vec{c} = (c_1, ..., c_s)$, $\vec{b} = (b_1, ..., b_s)$, $A = (a_{ij})_{S \times S}$, is rather complicated. Explicit RK methods need the following requirements.

1. The matrix A must be strictly lower triangular.
2. $c_i = \sum_{j=1}^{s} a_{ij}$, $i = 1, ..., s$.
3. We obtain different schemes or methods by choosing different coefficient means.

4.4.2 Higher Order Runge-Kutta Methods

Recall the original IVP

$$\frac{du}{dt} = \mathcal{L}(u), \text{ or } y'(t) = f(t, y(t))$$

Third-order TVD RK method, proposed by Shu-Osher is

$$
\begin{aligned}
u^{(1)} &= u^n + \Delta t \mathcal{L}(u^n) \\
u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t \mathcal{L}(u^{(1)}) \\
u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t \mathcal{L}(u^{(2)})
\end{aligned}
$$

Third-order RK method is

$$
\begin{aligned}
y^{n+1} &= y^n + \Delta t \left[\frac{1}{6}K_1 + \frac{2}{3}K_2 + \frac{1}{6}K_3\right] \\
K_1 &= f\left(t_n, y^n\right) \\
K_2 &= f\left(t_n + \frac{\Delta t}{2}, y^n + \frac{\Delta t}{2}K_1\right) \\
K_3 &= f\left(t_n + \frac{\Delta t}{2}, y^n - \Delta t K_1 + 2\Delta t K_2\right)
\end{aligned}
$$

Fourth-order RK method, with the local truncation error as $O(\Delta^5)$, is

$$
\begin{aligned}
y^{n+1} &= y^n + \Delta t \left[\frac{1}{6}K_1 + \frac{1}{3}K_2 + \frac{1}{3}K_3 + \frac{1}{6}K_4\right] \\
K_1 &= f\left(t_n, y^n\right) \\
K_2 &= f\left(t_n + \frac{\Delta t}{2}, y^n + \frac{\Delta t}{2}K_1\right) \\
K_3 &= f\left(t_n + \frac{\Delta t}{2}, y^n + \frac{\Delta t}{2}K_2\right) \\
K_4 &= f\left(t_n + \Delta t, y^n + \Delta t K_3\right)
\end{aligned}
$$

It is also important to note that the classical RK methods require multiple evaluations of f per time step.

Remark *For each time step, the second-order RK method needs to perform two evaluations of f. For a fourth-order method, there are four evaluations. Thus, higher-order (> 4) RK methods are relatively inefficient. Precise data for higher-order methods may not be known, so it may not be necessary to use high order method either, i.e., the precision of the data governs the overall accuracy. However, certain higher-order methods may still be appropriate to use if we want to construct a RK method, which can adjust the time step size adaptively and keep the local truncation error small.*

4.4.3 Connection to Numerical Integration Rules

We illustrate the connection of Runge-Kutta methods to numerical integration. Consider the IVP,

$$y'(t) = f(t, y(t)),\ y(t_0) = y_0$$

Integrate both sides of the differential equation from t to $t + \Delta t$,

$$y(t_n + \Delta t) - y(t_n) = \int_{t_n}^{t_n + \Delta t} f(\tau, y(\tau)) d\tau$$

Therefore, the solution to the IVP can be obtained by solving the integral equation. The numerical scheme is shown as follows.

1. Use the left Riemann sum

$$\int_{t_n}^{t_n + \Delta t} f(\tau, y(\tau)) d\tau \approx \Delta t f(\tau_0, y(\tau_0)) = \Delta t f(t_n, y(t_n))$$

Since $\tau_0 = t_n$, i.e., the left endpoint of the interval. Thus, the above approximation is equivalent to the Euler's method.

2. Use the trapezoidal rule
$$\int_{t_n}^{t_n+\Delta t} f(\tau, y(\tau))d\tau \approx \frac{\Delta t}{2}\left[f(t_n, y(t_n)) + f(t_n + \Delta t, y(t_n + \Delta t))\right]$$
The corresponding IVP solver is therefore
$$y^{n+1} = y^n + \frac{\Delta t}{2} f(t_n, y(t_n)) + \frac{\Delta t}{2} f(t_n + \Delta t, y(t_n + \Delta t))$$
Note that this is NOT the classical second order RK method, since $y_{n+1} = y(t_n + \Delta t)$ is on the RHS. That means that we have an implicit method, i.e., y_{n+1} does not only appear on the LHS. In order to make the method explicit we can use the Euler's method to replace y_{n+1} on the RHS, i.e., $y^{n+1} = y^n \Delta t f(t_n, y_n)$. Then we end up with
$$y^{n+1} = y^n + \frac{\Delta t}{2} f(t_n, y(t_n)) + \frac{\Delta t}{2} f(t_{n+1}, y^n \Delta t f(t_n, y_n))$$
or
$$y^{n+1} = y^n + \Delta t \left[\frac{1}{2}K_1 + \frac{1}{2}K_2\right], \text{where, } \begin{cases} K_1 = f(t_n, y(t_n)) \\ K_2 = f(t_n + \Delta t, y_n + \Delta t K_1) \end{cases}$$
i.e., the classical second order RK method.

3. Use the midpoint integration rule
$$\int_{t_n}^{t_n+2\Delta t} f(\tau, y(\tau))d\tau \approx 2\Delta t f(t_n + \Delta t, y(t_n + \Delta t))$$
Thus, the explicit midpoint rule is
$$y^{n+2} = y^n + 2\Delta t f(t_{n+1}, y_{n+1})$$
This is not a RK method, but an explicit two step method. In PDEs, such as method reappears as the leap-frog method.

As mentioned above, sometimes the modified Euler method is called the midpoint rule.
$$y^{n+1} = y^n + \Delta t f\left(t_n + \frac{\Delta t}{2}, y_n + \frac{\Delta t}{2} f(t_n, y_n)\right)$$
This can be explained by applying the midpoint integration rule with $a = t$ and $b = t + \Delta t$,
$$\int_{t_n}^{t_n+\Delta t} f(\tau, y(\tau))d\tau \approx \Delta t f\left(t_n + \frac{\Delta t}{2}, y(t_n + \frac{\Delta t}{2})\right)$$
If we represent $y(t + \Delta t/2)$ by its Euler approximation $y(t) + (\Delta t/2) f(t, y)$, then we obtain the modified Euler method stated above.

4. Simpson's rule yields the fourth order RK method in case there is no dependence of f on y.
5. Gauss quadrature leads to the so-called **GAUSS-RUNGE-KUTTA, GRK** or **GAUSS-LEGENDRE, GL** methods. One example of such methods is the implicit midpoint rule encountered earlier,
$$y^{n+1} = y^n + \Delta t f\left(t_n + \frac{\Delta t}{2}, \frac{1}{2}(y^n + y^{n+1})\right)$$
Note that the general implicit RK method is of the form
$$y^{n+1} = y^n + \Delta t \sum_{j=1}^{s} b_j K_j, \text{with } K_j = f\left(t_n + c_j \Delta t, y^n + \Delta t \sum_{i=1}^{j} a_{ji} K_i\right), \forall j = 1, ..., s$$

Thus, the implicit midpoint rule corresponds to

$$y_{n+1} = y_n + \Delta t K_1, \text{ with } K_1 = f\left(t_n + \frac{\Delta t}{2}, y^n + \frac{\Delta t}{2} K_1\right)$$

6. More general implicit RK methods exist. However, their constructions can be more difficult, and can sometimes be linked to collocation methods. Some details are given at the end of Chapter 3 in the Iserles book.

4.5 High Order Finite Difference Schemes

We should notice that the meaning of "high order" should be carefully defined when the solution contains discontinuities in its derivatives. In such situations, "high order accuracy" refers to a formal higher order truncation error in the smooth regions of the solution.

In general, the high order scheme can only be expected in smooth regions away from derivative singularities. However, typical high order methods also have a sharper resolution for the derivative singularities. Thus, high order methods are also referred to as "high resolution" schemes, especially when applied them to CLs.

Recall that in Eq.4.4, the first order monotone scheme can be written as follow,

$$\frac{d}{dt}\phi_i = -\widehat{H}(u_i^-, u_i^+), \text{where } u_i^- = \frac{\Delta_{-x}\phi_i}{\Delta x},\ u_i^+ = \frac{\Delta_{+x}\phi_i}{\Delta x} \tag{4.10}$$

are the first order approximations to the left and right derivatives of ϕ at the location $x = x_i$. That is to say, we can obtain u_j^- as follows.

1. Choose a stencil $S = \{x_{i-1}, x_i\}$ containing two grid points, including x_i and biased to the left, hence to approximate the left derivative.
2. Find the interpolating polynomial $p_1(x)$ which interpolates the function ϕ at each grid point in the stencil S, especially $p_1(x_{i-1}) = \phi_{i-1}$ and $p_1(x_i) = \phi_i$. Since the stencil S contains two points, the interpolation polynomial is of degree 1.
3. Take $u_i^- = p_1'(x_i)$ as the approximation to the left derivative of ϕ at the location $x = x_i$. One should work out the algebra first to obtain the explicit formula before programming, i.e.,

   $$u_i^- = \frac{\Delta_{-x}\phi_i}{\Delta x}$$

   The implementation should be simple. That is to say, in the computer code, it is not necessary to see the appearance of the interpolating polynomial $p_1(x)$. This is polynomial is used solely for the purpose of deriving the scheme theoretically.

Similarly, we do the right derivative u_i^+ by using the stencil $S = \{x_i, x_{i+1}\}$ to construct linear polynomial $p_1(x)$ and let

$$u_i^+ = \frac{\Delta_{+x}\phi_i}{\Delta x}$$

That is at least of the first order accuracy.

The idea of interpretation allow us to design high order schemes for the HJ equations based on the monotone Hamiltonian building block. The scheme is still shown as Eq.4.10. However, u_i^- and u_i^+ are now higher order approximations to the left and right derivatives of ϕ at the location $x = x_i$. E.g., the third order approximation to u_i^- can be obtained as follows.

1. Choose a stencil $S = \{x_{i-2}, x_{i-1}, x_i, x_{i+1}\}$ containing 4 grid points, including x_i and biased to the left. Note that the linear stability for time domain calculations restricts how much one can bias the stencil to one side. The most commonly used **UPWIND STENCIL** is one point biased to the upwind side.
2. Find the polynomial $p_3(x)$ which interpolates the function ϕ at each grid point in the stencil S. The interpolating polynomial $p_3(x)$ is a cubic polynomial.
3. Take $u_i^- = p_3'(x_i)$ as the approximation to the left derivative of ϕ at the location $x = x_i$. The explicit formula is
$$u_i^- = \frac{1}{\Delta x}\left[\frac{1}{6}\phi_{i-2} - \phi_{i-1} + \frac{1}{2}\phi_i + \frac{1}{3}\phi_{i+1}\right]$$
For non-uniform mesh, the coefficients will depend on the local mesh sizes, but not on ϕ. I.e., all the information from ϕ is fully built in via $\phi_{i-2}, \phi_{i-1}, \phi_i, \phi_{i+1}$ rather than the coefficients.

Similarly, we do the right derivative u_i^+ by using the stencil $S = \{x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$ to construct linear polynomial $p_3(x)$ and let
$$u_i^+ = \frac{1}{\Delta x}\left[\frac{1}{6}\phi_{i+2} - \phi_{i+1} + \frac{1}{2}\phi_i + \frac{1}{3}\phi_{i-1}\right]$$
That is also at least of the third order accuracy.

Clearly, we can design arbitrarily high order finite difference schemes of this form. Such schemes are all called **LINEAR SCHEMES**, which refers to the fact that those scheme, when applied to linear PDEs, becomes linear. Or in another word, the coefficients of interpolations are independent to ϕ.

These schemes are excellent for smooth solutions but may generate spurious oscillations when applied to non-differentiable viscosity solutions.

We now discuss those various generalizations.

For non-uniform and non-smooth meshes, the only change occurs in the interpolation procedures for $u_i^\pm$. Since it only involves standard polynomial interpolation, non-uniform and non-smooth meshes do not generate any difficulty. This is quite different from the situation for high order conservative finite difference schemes for CLs, which can only be designed for uniform or smooth non-uniform meshes. For non-periodic BCs, numerical BCs are needed for both the inflow and for outflow boundaries.

To multi-dimensional spatial domain using Cartesian meshes, the scheme in Eq.4.10 can be changed to
$$\frac{d}{dt}\phi_{i,j} = -\widehat{H}\left(u_{i,j}^-, u_{i,j}^+; v_{i,j}^-, v_{i,j}^+\right) \tag{4.11}$$

1. $u_{i,j}^\pm$ are high order approximations to the left and right derivatives of ϕ, w.r.t. x at the location $(x, y) = (x_i, y_j)$
2. $u_{i,j}^\pm$ are high order approximations to the left and right derivatives of ϕ, w.r.t. y at the location $(x, y) = (x_i, y_j)$
3. When computing $u_{i,j}^-$, use the 1D procedure described above and the values $\phi_{l,j}, l = i-2, i-1, i, i+1$, with j fixed.
4. Similarly, when computing $v_{i,j}^-$, use the 1D procedure above and the values $\phi_{i,l}, l = j-2, j-1, j, j+1$, with i fixed.
5. Therefore, the main ingredient of the algorithm, namely the interpolation procedure to obtain high order approximations to the left and right derivatives of ϕ, are the same as the 1D case.

4.6 ENO and WENO Interpolations for Left and Right Derivatives

To design a high order finite difference method for solving the HJ IVP

$$\phi_t + H(\phi_x, \phi_y) = 0,\ \phi(x, y, 0) = \phi_0(x, y) \tag{4.12}$$

on a rectangular mesh, we can proceed as follows.

1. Find a suitable monotone numerical Hamiltonian, $\widehat{H}(u^-, u^+; v^-, v^+)$, s.t. $\widehat{H}(\uparrow, \downarrow; \uparrow, \downarrow)$.
2. Compute high order approximations to the left and right derivatives of ϕ w.r.t. x and y at the grid point $(x, y) = (x_i, y_j)$, in a dimension-by-dimension fashion, and denote them as $u^-_{i,j}, u^+_{i,j}, v^-_{i,j}, v^+_{i,j}$, respectively.
3. Formulate the semi-discrete scheme, similar to the one shown in Eq.4.13
$$\frac{d}{dt}\phi_{i,j} = -\widehat{H}\left(u^-_{i,j}, u^+_{i,j}; v^-_{i,j}, v^+_{i,j}\right) \tag{4.13}$$
and discretize the time domain by high order TVD time discretization, or other time discretization method.

To obtain high order finite difference schemes for HJ equations on Cartesian meshes, we only need to discuss the 1D high order accurate approximations to the left and right derivatives, given the function values on mesh grid points. We could use the general polynomial interpolation procedure, i.e.,

1. Choose a stencil $S = \{x_{i-2}, x_{i-1}, x_i, x_{i+1}, \}$ containing 4 grid points, including x_i and biased to the left, i.e., x_{i-1}. Note that the linear stability for time domain calculation restricts how much one can bias the stencil to one side. The most commonly used **UPWIND STENCIL** is a "one-point" biased stencil towards the upwind side.
2. Find the polynomial $p(x)$ which interpolates the function ϕ at each grid point in the stencil S. That is, $p(x_{i-2}) = \phi_{i-2}$, $p(x_{i-1}) = \phi_{i-1}$, $p(x_i) = \phi_i$, $p(x_{i+1}) = \phi_{i+1}$. Because the stencil S has 4 points, the interpolation polynomial $p(x)$ is a cubic polynomial.
3. Take $u_i^- = p'(x_i)$ as the approximation to the left derivative of ϕ at the location $x = x_i$. The explicit formula is
$$u_i^- = \frac{1}{\Delta x}\left[\frac{1}{6}\phi_{i-2} - \phi_{i-1} + \frac{1}{2}\phi_i + \frac{1}{3}\phi_{i+1}\right]$$
The coefficients depend on the local mesh sizes, but not on ϕ.

The scheme above would lead to linear schemes, i.e., the schemes are linear when applied to linear PDEs. These schemes are excellent for smooth solutions, but may generate spurious oscillations when applied to non-differentiable viscosity solutions. We would like to have approximations which can achieve

1. High order accurate in approximating left and right derivatives in smooth regions.
2. (Essentially) non-oscillatory near kinks, i.e., discontinuities in the derivatives of the function.

The spurious oscillations appear because the interpolation stencil S, i.e., the 4-point set, may include a discontinuity of the derivative. In such cases, not only will the accuracy of the interpolation be completely lost, but also the interpolation tends to generate overshoots or undershoots, especially for approximating derivatives. Recall that the accuracy is proved based on the smoothness assumption of the underlying function over the interpolation stencil.

For fixed stencils, e.g., constantly using two points to the left, the point itself, and one point to the right, the stencils will definitely encounter the discontinuity of the derivative somewhere. One way to overcome this difficulty is to set "limitations" in the approximation, through, e.g., a minmod limiter. That leads to second

order TVD type schemes. A better way is to adopt the **ESSENTIALLY NON-OSCILLATORY (ENO)** or **WEIGHTED ESSENTIALLY NON-OSCILLATORY (WENO)** procedure.

4.6.1 Essentially Non-oscillatory

Suppose we use a polynomial $p(x)$ of degree $k \geq 2$ to interpolate the function ϕ in a 2-point stencil $S = \{x_{i-1}, x_i\}$. First, it is important to notice that, unlike the case in the CLs, here we start with a stencil containing (at least) two points $\{x_{i-1}, x_i\}$, rather than just one point $\{x_i\}$. This is because the first order scheme already uses a linear interpolation with the stencil $\{x_{i-1}, x_i\}$. The correct ENO stencil should contain the stencil for the first order base scheme, which provides correct up-winding and stability.

Remark *(Lagrange Polynomial)*
Recall the Lagrange polynomial at x_0, x_1, x_2, x_3

$$\phi(x_k) = \sum_{l=0}^{3} a_{k,l}\phi_l, \quad k = 0, ..., 3$$

The coefficients can be determined as

$$p_3(x) = \phi(x_0)\frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} + \phi(x_1)\frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}$$
$$+\phi(x_2)\frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} + \phi(x_3)\frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}$$

Then take the derivative

$$\phi'(x_k) = p_3'(x_k) = \sum_{l=0}^{3} c_{k,l}\phi(x_l) \text{ for } k = 0, 1, 2, 3$$

The coefficients $\{c_{k,l}\}$ do not depend on ϕ.

Remark *(Newton Polynomial)*
The Newton polynomial has the property that after constructing an interpolation with certain degree, we can easily add higher order terms. Let

$$N(x) = \phi[x_0] + \phi[x_0, x_1](x - x_0) + ... + \phi[x_0, x_1, ..., x_k](x - x_0)(x - x_1)...(x - x_{k-1})$$

where the **DIVIDED DIFFERENCE** *can be defined as*

$$\phi[i] = \phi(x_i)$$
$$\phi[i, i+1] = \frac{\phi[i+1] - \phi[i]}{x_{i+1} - x_i}$$
$$\phi[i, i+1, i+2] = \frac{\phi[i+1, i+2] - \phi[i+1, i]}{x_{i+2} - x_i}$$
$$.....$$
$$\phi[i, ..., i+n] = \frac{\phi[i+1, ..., i+n] - \phi[i, ..., i+n-1]}{x_{i+n} - x_i}$$

Divided difference can be formulated in the form of Fig.4.1.

The divided difference can be written in a table, and the coefficients in the Newton polynomial can be computed and stored. Thus, the ENO difference can be written as

1. Start with the first order stencil $S^1 = \{x_{i-1}, x_i\}$
2. Decide to add either the left neighbor, x_{i-1}, or the right neighbor, x_{i+1}, into the stencil.

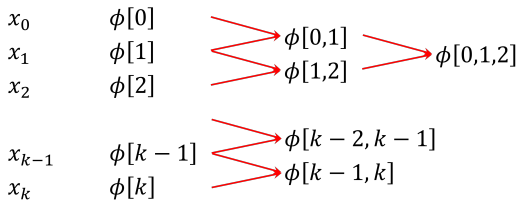

Figure 4.1: The Divided Difference Scheme Used in Newton Polynomials.

We believe the stencil $\{x_{i-2}, x_{i-1}, x_i\}$ is better than $\{x_{i-1}, x_i, x_{i+1}\}$, if

$$|\phi[i-2, i-1, i]| \leq |\phi[i-1, i, i+1]|$$

In order to see that, we define the "left" and "right" interpolation polynomials

$$p_l(x) = p^1(x) + \phi[i-2, i-1, i](x - x_{i-1})(x - x_i), \quad \text{based on } \{x_{i-2}, x_{i-1}, x_i\}$$
$$p_r(x) = p^1(x) + \phi[i-1, i, i+1](x - x_{i-1})(x - x_i), \quad \text{based on } \{x_{i-1}, x_i, x_{i+1}\}$$

where $p^1(x)$ is the first degree interpolation polynomial over $S^1 = \{x_{i-1}, x_i\}$. Notice that $p_l(x) - p^1(x)$ and $p_r(x) - p^1(x)$ are both scalar coefficients for the same function $(x - x_{i-1})(x - x_i)$, and the deviation $p_l(x) - p^1(x)$ is smaller than the deviation $p_r(x) - p^1(x)$.

If we believe the first degree interpolation polynomial $p^1(x)$ is sufficiently good, i.e., it can produce a good first order monotone scheme, we would like to deviate from it as little as possible while keeping the accuracy. Such a preference leads to the ENO choice of $p_l(x)$. Hence in this case, the stencil for the quadratic interpolation polynomial is $S^2 = \{x_{i-2}, x_{i-1}, x_i\}$.

Otherwise, if

$$|\phi[i-2, i-1, i]| \geq |\phi[i-1, i, i+1]|$$

we choose $S^2 = \{x_{i-1}, x_i, x_{i+1}\}$.

3. Repeat this process. Suppose $S^2 = \{x_{i-2}, x_{i-1}, x_i\}$. We could decide to add either the left neighbor, x_{i-3}, or the right neighbor, x_{i+1}, into the stencil. If
$$|\phi[i-3, i-2, i-1, i]| \leq |\phi[i-2, i-1, i, i+1]|$$
then we choose the stencil for the cubic interpolation polynomial as $S^3 = \{x_{i-3}, x_{i-2}, x_{i-1}, x_i\}$. Otherwise, we choose $S^3 = \{x_{i-2}, x_{i-1}, x_i, x_{i+1}\}$.
4. Continue this process until the target stencil S^k is chosen. The interpolation polynomial $p(x)$ on this stencil is the desired one. Then, take $u^- = p'(x_i)$. In practice, we do not need to explicitly compute $p(x)$ and take its derivative. We can simply pre-compute the coefficients for each possible stencil. E.g., if the final ENO stencil is $S^3 = \{x_{i-2}, x_{i-1}, x_i, x_{i+1}\}$, we can directly compute
$$u_i^- = \frac{1}{\Delta x}\left[\frac{1}{6}\phi_{i-2} - \phi_{i-1} + \frac{1}{2}\phi_i + \frac{1}{3}\phi_{i+1}\right]$$
where the coefficients $1/6$, -1, $1/2$ and $1/3$ are pre-computed and stored. For non-uniform meshes, these coefficients depend on the local mesh sizes, but not on ϕ, hence they can still be pre-computed and stored if the mesh is rigid.
5. For computing u^+, the only change to the procedure above is the starting first order stencil, which should be changed to $S^1 = \{x_i, x_{i+1}\}$.

The ENO procedure is simple to code and to compute, even for non-uniform meshes.

Remark *ENO idea is step-by-step method. Thus, one interesting equation could be "Is it possible to choose the*

bad stencil at previous step and obtain better stencil later?" The answer is no. That is because once a bad stencil is chosen, there must be a kink. Then in the later steps, the kink will be maintained in your stencil. Therefore, it is not possible to get less oscillation later.

4.6.2 Weighted Essentially Non-Oscillatory

Consider an illustrative example. If we want to construct a third order interpolation polynomial, is it possible to use all of the potential stencils? The idea for that is if you want to use polynomial with degree of 3. We need to do the computation based on six points. However, if the solution is smooth enough and we have the data from six points, we should be able to do the approximation with the accuracy order of 5. So by using the ENO, we lose some information.

The WENO interpolation procedure is based on the ENO procedure. If we still use the previous example of trying to obtain piecewise cubic polynomial interpolations with the third order accurate for derivatives, and if we insist that the stencil should contain $\{x_{i-1}, x_i\}$, i.e., the stencil for the first order monotone scheme, then we have the following three possible stencils which might be chosen by the ENO procedure, i.e.,

$$S^0 = \{x_{i-3}, x_{i-2}, x_{i-1}, x_i\}$$
$$S^1 = \{x_{i-2}, x_{i-1}, x_i, x_{i+1}\}$$
$$S^2 = \{x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$$

Then the corresponding three third order approximations to the left derivative $u_i^- \approx \phi_x(x_i^-)$ can be

$$u_i^{-,0} = \frac{1}{\Delta x}\left[\frac{1}{3}\Delta_{+x}\phi_{i-3} - \frac{7}{6}\Delta_{+x}\phi_{i-2} + \frac{11}{6}\Delta_{+x}\phi_{i-1}\right]$$
$$u_i^{-,1} = \frac{1}{\Delta x}\left[-\frac{1}{6}\Delta_{+x}\phi_{i-2} + \frac{5}{6}\Delta_{+x}\phi_{i-1} + \frac{1}{3}\Delta_{+x}\phi_i\right]$$
$$u_i^{-,2} = \frac{1}{\Delta x}\left[\frac{1}{3}\Delta_{+x}\phi_{i-1} + \frac{5}{6}\Delta_{+x}\phi_i - \frac{1}{6}\Delta_{+x}\phi_{i+1}\right]$$

In ENO procedure, we need to choose only one of three interpolations and ignore the other two. This is the correct procedure near kinks with discontinuities of the first order derivatives, as one of the three candidate stencils, S^0, S^1 and S^2, may contain a kink. However, in the smooth region, all three stencils are equally good. It is a waste to look at all of them and eventually discard two of them. A greedy algorithm would use all of them, that is the idea of WENO scheme,

$$u_i^- = \omega_0 u_i^{-,0} + \omega_1 u_i^{-,1} + \omega_2 u_i^{-,2} \tag{4.14}$$

Now the question is how to choose the weights ω_0, ω_1 and ω_2. For consistency, $\omega_0 + \omega_1 + \omega_2 = 1$. For stability, we prefer $\omega_0 \geq 0$, $\omega_1 \geq 0$, $\omega_2 \geq 0$, similar to the idea we proposed for the "maximum principle".

Remark *In fact, if $\omega_0 + \omega_1 + \omega_2 = 1$ is satisfied, then any choice of the weights ω_0, ω_1 and ω_2 would lead to at least the third order accuracy, since each $u_i^{-,k}$ already has the third order accurate. There are WENO schemes in the literature of this type. The weighted average does not have a higher order accuracy than the approximations in each sub-stencil.*

A simple algebra reveals that, if we choose the weights ω_0, ω_1 and ω_2 as the so-called **OPTIMAL LINEAR WEIGHTS**, i.e.,

$$\omega_0 = \gamma_0 = 0.1,\ \omega_1 = \gamma_1 = 0.6,\ \omega_2 = \gamma_2 = 0.3$$

Then the approximation

$$u_i^- = \gamma_0 u_i^{-,0} + \gamma_1 u_i^{-,1} + \gamma_2 u_i^{-,2}$$

would be of the fifth order accuracy. In fact, this would be the same approximation obtained from the interpolation polynomial based on the larger stencil. I.e., it is equivalent to constructing a $p_5(x)$ in

$$S = S^0 \cup S^1 \cup S^2 = \{x_{i-3}, x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$$

However, the approximation will be oscillatory near kinks. Therefore, the trick is to require that the choice of the weights ω_0, ω_1 and ω_2 to be the ones, s.t.,

1. If the solution is smooth or in smooth regions, the weights ω_0, ω_1 and ω_2 are very close to the optimal linear weights γ_0, γ_1 and γ_2,
$$\omega_0 = 0.1 + O(\Delta x^2),\ \omega_1 = 0.6 + O(\Delta x^2),\ \omega_2 = 0.3 + O(\Delta x^2)$$
2. If the solution is not smooth or near kinks, and if the stencil S^k contains a kink while at least one of the other two stencils does not, then the corresponding weight ω_k is very small. In fact, we should require $\omega_k = O(\Delta x^4)$ to satisfy the total accuracy.

The key ingredient in designing a nonlinear weight to satisfying the two properties above is a smoothness indicator, which is a measure the smoothness of function being interpolated in the interpolation stencil. The **SMOOTHNESS INDICATOR** is a scaled sum of the squares of the $\mathcal{L}^2$ norms of the second and higher derivatives of the interpolation polynomial on some target intervals.

Remark *In the conservation law, we also use something like the first or higher derivatives as the smoothness limit. Here the smoothness indicator sums the squares of the $\mathcal{L}^2$ norms of the second and higher derivatives.*

These smoothness indicators turn out to be

$$\begin{aligned} IS_0 &= 13(a-b)^2 + 3(a-3b)^2 \\ IS_1 &= 13(b-c)^2 + 3(b+c)^2 \\ IS_2 &= 13(c-d)^2 + 3(3c-d)^2 \end{aligned} \tag{4.15}$$

And the letters a, b, c and d are defined with the central difference operator,

$$a = \frac{\delta^2\phi_{i-2}}{\Delta x},\ b = \frac{\delta^2\phi_{i-1}}{\Delta x},\ c = \frac{\delta^2\phi_i}{\Delta x},\ d = \frac{\delta^2\phi_{i+1}}{\Delta x},\ \text{where } \delta^2\phi = \phi_{i+1} - 2\phi_i + \phi_{i-1}$$

With these smoothness indicators, the nonlinear weights are then defined by

$$\omega_0 = \frac{\widetilde{\omega}_0}{\widetilde{\omega}_0 + \widetilde{\omega}_1 + \widetilde{\omega}_2},\ \omega_1 = \frac{\widetilde{\omega}_1}{\widetilde{\omega}_0 + \widetilde{\omega}_1 + \widetilde{\omega}_2},\ \omega_2 = \frac{\widetilde{\omega}_2}{\widetilde{\omega}_0 + \widetilde{\omega}_1 + \widetilde{\omega}_2},$$

with

$$\widetilde{\omega}_0 = \frac{1}{(\varepsilon + IS_0)^2},\ \widetilde{\omega}_1 = \frac{1}{(\varepsilon + IS_1)^2},\ \widetilde{\omega}_2 = \frac{1}{(\varepsilon + IS_2)^2}$$

ε is a small number to prevent the denominator becoming zero. Typically, $\varepsilon = 10^{-6}$.

Finally, after some algebraic manipulations, we obtain the fifth order WENO approximation to u_i^- as

$$\begin{aligned} u_i^- &= \frac{1}{12\Delta x}\left(-\Delta_{+x}\phi_{i-2} + 7\Delta_{+x}\phi_{i-1} + 7\Delta_{+x}\phi_i - \Delta_{+x}\phi_{i+1}\right) - \Phi^{\text{WENO}}(a,b,c,d) \\ \text{where, } \Phi^{\text{WENO}}(a,b,c,d) &= \frac{1}{3}\omega_0(a - 2b + c) + \frac{1}{6}\left(\omega_2 - \frac{1}{2}\right)(b - 2c + d) \end{aligned} \tag{4.16}$$

with a, b, c and d defined as above.

By symmetry, the approximation to the right derivative u_i^+ is given by

$$u_i^+ = \frac{1}{12\Delta x}\left(-\Delta_{+x}\phi_{i-2} + 7\Delta_{+x}\phi_{i-1} + 7\Delta_{+x}\phi_i - \Delta_{+x}\phi_{i+1}\right) - \Phi^{\text{WENO}}(e,d,c,b) \tag{4.17}$$

where b, c and d defined as above and

$$e = \frac{\delta^2\phi_{i+2}}{\Delta x}$$

4.7 Exercise

Exercise 4.1 Solve the 1D HJ Equation

$$\begin{cases} \phi_t + |\phi_x| = 0 & (x,t) \in [-1,1]\times(0,1] \\ \phi(x,0) = \phi_0(x) = |x| - 0.1 & x \in [-1,1] \\ \text{periodic BC in space} \end{cases}$$

with the scheme

$$\frac{d\phi_i}{dt} + \widehat{H}(u_i^-, u_i^+) = 0$$

1. Use the Lax-Friedrichs numerical Hamiltonian,

$$\widehat{H}^{LF} = H\left(\frac{u^- + u^+}{2}\right) - \frac{1}{2}\alpha(u^+ - u^-), \text{ with } \alpha = \max_{A\le u\le B}|H'(u)|$$

Prove the numerical scheme is consistent and monotone. And solve the equation with the first order Lax-Friedrichs scheme, i.e., forward in time, and first order approximation in space.

2. Use the first order scheme in (1) as the building box, and solve the equation with the second-order RK method and second-order ENO approximations. Recall the second order RK method is

$$\frac{du}{dt} = \mathcal{L}(u) \Rightarrow \begin{cases} u^{(1)} = u^n + \Delta t\mathcal{L}(u^n) \\ u^{n+1} = \frac{1}{2}u^n + \frac{1}{2}\left[u^{(1)} + \Delta t\mathcal{L}(u^{(1)})\right] \Leftrightarrow u^n + \frac{\Delta t}{2}\left[\mathcal{L}(u^n) + \mathcal{L}(u^{(1)})\right] \end{cases}$$

Both steps should include ENO. For (1) and (2), set $\Delta x = 1/100$, and $\Delta t = \Delta x/2$, and plot the numerical solutions at $t = \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Comment on the numerical solutions.

Solve

1. *Suppose that $u = \phi_x$, and $H(u) = |u|$. Then, for* **CONSISTENCY**, *in general, if $\widehat{H}^{LF}(u,u) = H(u)$, then the numerical Hamiltonian is consistent, i.e.*

$$\widehat{H}^{LF}(u,u) = H\left(\frac{u+u}{2}\right) - \frac{1}{2}\alpha(u-u) = H(u)$$

We can also show that

$$\begin{aligned} T =& \widehat{H}^{LF}(u^-,u^+) - H(u) \\ =& H\left(\frac{u^- + u^+}{2}\right) - \frac{1}{2}\alpha(u^+ - u^-) - H(u) \\ =& H\left(\frac{1}{2}\frac{\Delta_{-x}\phi_i}{\Delta x} + \frac{1}{2}\frac{\Delta_{+x}\phi_i}{\Delta x}\right) - \frac{1}{2}\alpha\left(\frac{\Delta_{+x}\phi_i}{\Delta x} - \frac{\Delta_{-x}\phi_i}{\Delta x}\right) - H(\phi_{i,x}) \\ =& H\left[\frac{1}{2\Delta x}\left(\Delta_{-x}\phi_i + \Delta_{+x}\phi_i\right)\right] - \frac{1}{2\Delta x}\alpha\left(\Delta_{+x}\phi_i - \Delta_{-x}\phi_i\right) - H(\phi_{i,x}) \\ =& H\left[\phi_{i,x} + \frac{1}{6}\phi_{i,xxx}\Delta x^2 + O(\Delta x^3)\right] - \frac{1}{2}\alpha\left(\phi_{i,xx}\Delta x + O(\Delta x^3)\right) - H(\phi_{i,x}) \\ \le& H\left(\phi_{i,x}\right) - \frac{1}{2}\alpha\phi_{i,xx}\Delta x + \left|\frac{1}{6}\phi_{i,xxx}\right|\Delta x^2 + O(\Delta x^3) - H(\phi_{i,x}) \sim O(\Delta x) \end{aligned}$$

Therefore, the Lax-Friedrichs numerical Hamiltonian is of the first order accuracy.

For **MONOTONE**, *the definition is that the numerical Hamiltonian is* $\widehat{H}(\uparrow,\downarrow)$, *i.e.,*

$$\widehat{H}_1(u^-,u^+) = \frac{1}{2}H' + \frac{1}{2}\alpha \geq \left(-\frac{1}{2}+\frac{1}{2}\right)\alpha = 0$$
$$\widehat{H}_2(u^-,u^+) = \frac{1}{2}H' - \frac{1}{2}\alpha \leq \left(\frac{1}{2}-\frac{1}{2}\right)\alpha = 0$$

Thus, the numerical Hamiltonian is monotone. Choose $\Delta x = 0.01$, $\Delta t = \Delta x/2$, *then the numerical result is shown in Fig.4.2.*

2. *We combine the second order RK and the second order ENO, the numerical result is shown in Fig.4.2.*

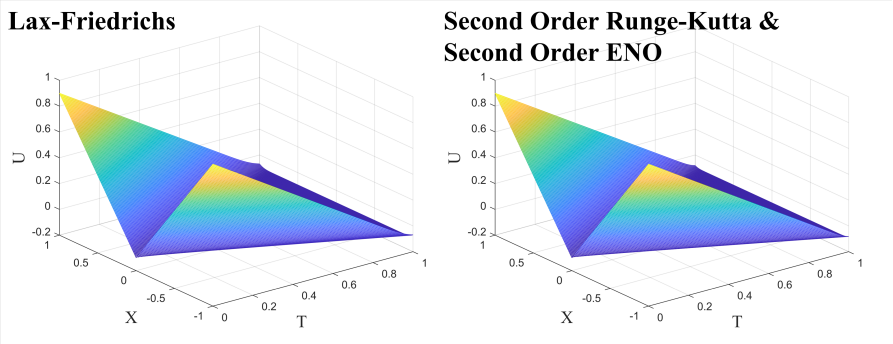

Figure 4.2: Numerical Results Using Lax-Friedrichs Scheme And The Second Order RK and Second Order ENO Scheme.

The comparisons between the Lax-Friedrichs scheme and the second order RK & second order ENO at specific time steps are shown in Fig.4.3.

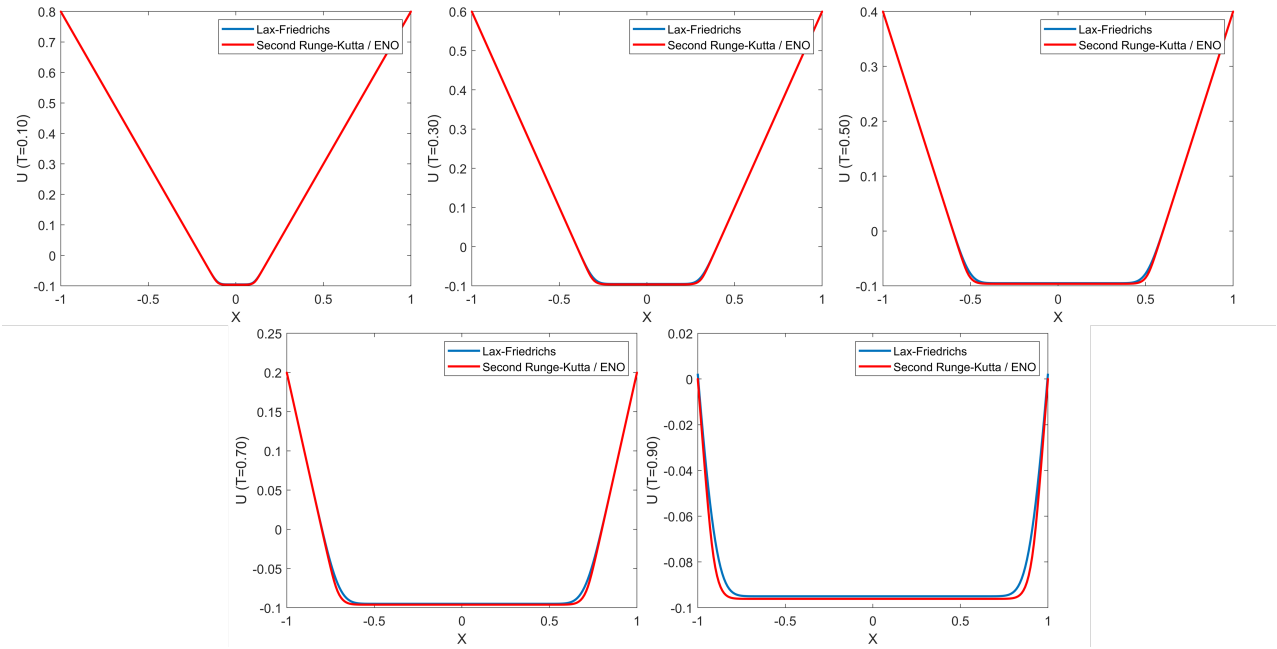

Figure 4.3: The Comparison between Lax-Friedrichs Scheme And The Second Order RK and Second Order ENO Scheme.

The results shows that both of the schemes can give entropy solutions to the original problem. However, the second order RK & second order ENO scheme is of the second order accuracy, while the Lax-Friedrichs scheme has the first order accuracy. The figures show that for higher order scheme, it maintains sharp kinks comparing to the lower order scheme. That can also be understood, since the second order RK & second order ENO scheme has less damping/dissipation error.

▲

Appendix. The MATLAB code

```
% ------------ Math 517 Chap.4 Exercise 1 --------
clear all; close all; clc
uu0=@(x) abs(x)-0.1; DX=0.01; nu=0.5; DT=nu*DX; X=-1:DX:1; T=0:DT:1;
J=length(X); tF=length(T); [Xaxis,Taxis]=meshgrid(X,T);
% ------------- Lax Friedrichs Scheme ------------------
Phi1=zeros(tF,J); alpha=1;
for i=1:J
    Phi1(1,i)=uu0(X(i)); end
for t=2:tF
    for i=1:J
        if(i==1)
            Unega=(Phi1(t-1,1)-Phi1(t-1,J-1))/DX;
        else
            Unega=(Phi1(t-1,i)-Phi1(t-1,i-1))/DX;
        end
        if(i==J)
            Uposi=(Phi1(t-1,2)-Phi1(t-1,J))/DX;
        else
            Uposi=(Phi1(t-1,i+1)-Phi1(t-1,i))/DX;
        end
        H=abs(0.5*(Unega+Uposi))-0.5*(Uposi-Unega);
        Phi1(t,i)= Phi1(t-1,i)-DT*H; end end
figure(1)
mesh(Taxis,Xaxis,Phi1),xlabel('T'),ylabel('X'),zlabel('U'),
set(get(gca, 'YLabel'), 'FontName', 'Times New Roman', 'FontSize', 16)
set(get(gca, 'XLabel'), 'FontName', 'Times New Roman', 'FontSize', 16)
set(get(gca, 'ZLabel'), 'FontName', 'Times New Roman', 'FontSize', 16)
% ------------- 2nd Runge-Kutta / ENO Appro Scheme --------------------
Phi2=zeros(tF,J); alpha=1;
for i=1:J
    Phi2(1,i)=uu0(X(i)); end
for t=2:tF
    % ---------------- First Step Runge-Kutta --------------------
    for i=1:J
        if(i==1)
            U22NegaPosi=((Phi2(t-1,2)-Phi2(t-1,1))-(Phi2(t-1,1)-Phi2(t-1,J-1)))/(DX*DX);
        elseif(i==J)
            U22NegaPosi=((Phi2(t-1,2)-Phi2(t-1,J))-(Phi2(t-1,J)-Phi2(t-1,J-1)))/(DX*DX);
        else
            U22NegaPosi=((Phi2(t-1,i+1)-Phi2(t-1,i))-(Phi2(t-1,i)-Phi2(t-1,i-1)))/(DX*DX);
        end
        if(i==1)
            U22NegaNega=((Phi2(t-1,i)-Phi2(t-1,J-1))-(Phi2(t-1,J-1)-Phi2(t-1,J-2)))/(DX*DX);
        elseif(i==2)
            U22NegaNega=((Phi2(t-1,i)-Phi2(t-1,i-1))-(Phi2(t-1,i-1)-Phi2(t-1,J-1)))/(DX*DX);
        else
            U22NegaNega=((Phi2(t-1,i)-Phi2(t-1,i-1))-(Phi2(t-1,i-1)-Phi2(t-1,i-2)))/(DX*DX);
```

```
end
if (abs(U22NegaNega)<abs(U22NegaPosi))
    if(i==1)
        Unega=(0.5*Phi2(t-1,J-2)-2*Phi2(t-1,J-1)+1.5*Phi2(t-1,i))/DX;
    elseif(i==2)
        Unega=(0.5*Phi2(t-1,J-1)-2*Phi2(t-1,i-1)+1.5*Phi2(t-1,i))/DX;
    else
        Unega=(0.5*Phi2(t-1,i-2)-2*Phi2(t-1,i-1)+1.5*Phi2(t-1,i))/DX;
    end
else
    if(i==1)
        Unega=(0.5*Phi2(t-1,2)-0.5*Phi2(t-1,J-1))/DX;
    elseif(i==J)
        Unega=(0.5*Phi2(t-1,2)-0.5*Phi2(t-1,J-1))/DX;
    else
        Unega=(0.5*Phi2(t-1,i+1)-0.5*Phi2(t-1,i-1))/DX;
    end
end
if(i==1)
    U22PosiNega=((Phi2(t-1,2)-Phi2(t-1,1))-(Phi2(t-1,1)-Phi2(t-1,J-1)))/(DX*DX);
elseif(i==J)
    U22PosiNega=((Phi2(t-1,2)-Phi2(t-1,J))-(Phi2(t-1,J)-Phi2(t-1,J-1)))/(DX*DX);
else
    U22PosiNega=((Phi2(t-1,i+1)-Phi2(t-1,i))-(Phi2(t-1,i)-Phi2(t-1,i-1)))/(DX*DX);
end
if(i==J)
    U22PosiPosi=((Phi2(t-1,3)-Phi2(t-1,2))-(Phi2(t-1,2)-Phi2(t-1,i)))/(DX*DX);
elseif(i==J-1)
    U22PosiPosi=((Phi2(t-1,2)-Phi2(t-1,1))-(Phi2(t-1,1)-Phi2(t-1,J-1)))/(DX*DX);
else
    U22PosiPosi=((Phi2(t-1,i+2)-Phi2(t-1,i+1))-(Phi2(t-1,i+1)-Phi2(t-1,i)))/(DX*DX);
end
if (abs(U22PosiNega)<abs(U22PosiPosi))
    if(i==1)
        Uposi=(0.5*Phi2(t-1,2)-0.5*Phi2(t-1,J-1))/DX;
    elseif(i==J)
        Uposi=(0.5*Phi2(t-1,2)-0.5*Phi2(t-1,J-1))/DX;
    else
        Uposi=(0.5*Phi2(t-1,i+1)-0.5*Phi2(t-1,i-1))/DX;
    end
else
    if(i==J)
        Uposi=(-0.5*Phi2(t-1,3)+2*Phi2(t-1,2)-1.5*Phi2(t-1,i))/DX;
    elseif(i==J-1)
        Uposi=(-0.5*Phi2(t-1,2)+2*Phi2(t-1,i+1)-1.5*Phi2(t-1,i))/DX;
    else
        Uposi=(-0.5*Phi2(t-1,i+2)+2*Phi2(t-1,i+1)-1.5*Phi2(t-1,i))/DX;
    end
```

```
    end
    H=abs(0.5*(Unega+Uposi))-0.5*(Uposi-Unega);
    PhiStar(i)=Phi2(t-1,i)-DT*H; end
% ---------------- Second Step Runge-Kutta -------------------
for i=1:J
    if(i==1)
        U22NegaPosi=((PhiStar(2)-PhiStar(1))-(PhiStar(1)-PhiStar(J-1)))/(DX*DX);
    elseif(i==J)
        U22NegaPosi=((PhiStar(2)-PhiStar(J))-(PhiStar(J)-PhiStar(J-1)))/(DX*DX);
    else
        U22NegaPosi=((PhiStar(i+1)-PhiStar(i))-(PhiStar(i)-PhiStar(i-1)))/(DX*DX);
    end
    if(i==1)
        U22NegaNega=((PhiStar(i)-PhiStar(J-1))-(PhiStar(J-1)-PhiStar(J-2)))/(DX*DX);
    elseif(i==2)
        U22NegaNega=((PhiStar(i)-PhiStar(i-1))-(PhiStar(i-1)-PhiStar(J-1)))/(DX*DX);
    else
        U22NegaNega=((PhiStar(i)-PhiStar(i-1))-(PhiStar(i-1)-PhiStar(i-2)))/(DX*DX);
    end
    if (abs(U22NegaNega)<abs(U22NegaPosi))
        if(i==1)
            Unega=(0.5*PhiStar(J-2)-2*PhiStar(J-1)+1.5*PhiStar(i))/DX;
        elseif(i==2)
            Unega=(0.5*PhiStar(J-1)-2*PhiStar(i-1)+1.5*PhiStar(i))/DX;
        else
            Unega=(0.5*PhiStar(i-2)-2*PhiStar(i-1)+1.5*PhiStar(i))/DX;
        end
    else
        if(i==1)
            Unega=(0.5*PhiStar(2)-0.5*PhiStar(J-1))/DX;
        elseif(i==J)
            Unega=(0.5*PhiStar(2)-0.5*PhiStar(J-1))/DX;
        else
            Unega=(0.5*PhiStar(i+1)-0.5*PhiStar(i-1))/DX;
        end
    end
    if(i==1)
        U22PosiNega=((PhiStar(2)-PhiStar(1))-(PhiStar(1)-PhiStar(J-1)))/(DX*DX);
    elseif(i==J)
        U22PosiNega=((PhiStar(2)-PhiStar(J))-(PhiStar(J)-PhiStar(J-1)))/(DX*DX);
    else
        U22PosiNega=((PhiStar(i+1)-PhiStar(i))-(PhiStar(i)-PhiStar(i-1)))/(DX*DX);
    end
    if(i==J)
        U22PosiPosi=((PhiStar(3)-PhiStar(2))-(PhiStar(2)-PhiStar(i)))/(DX*DX);
    elseif(i==J-1)
        U22PosiPosi=((PhiStar(2)-PhiStar(1))-(PhiStar(1)-PhiStar(J-1)))/(DX*DX);
    else
```

```
            U22PosiPosi=((PhiStar(i+2)-PhiStar(i+1))-(PhiStar(i+1)-PhiStar(i)))/(DX*DX);
        end
        if (abs(U22PosiNega)<abs(U22PosiPosi))
            if(i==1)
                Uposi=(0.5*PhiStar(2)-0.5*PhiStar(J-1))/DX;
            elseif(i==J)
                Uposi=(0.5*PhiStar(2)-0.5*PhiStar(J-1))/DX;
            else
                Uposi=(0.5*PhiStar(i+1)-0.5*PhiStar(i-1))/DX;
            end
        else
            if(i==J)
                Uposi=(-0.5*PhiStar(3)+2*PhiStar(2)-1.5*PhiStar(i))/DX;
            elseif(i==J-1)
                Uposi=(-0.5*PhiStar(2)+2*PhiStar(i+1)-1.5*PhiStar(i))/DX;
            else
                Uposi=(-0.5*PhiStar(i+2)+2*PhiStar(i+1)-1.5*PhiStar(i))/DX;
            end
        end
        H=abs(0.5*(Unega+Uposi))-0.5*(Uposi-Unega);
        % ---------------- Assemble the 2nd Order Runge-Kutta Scheme ----------------
        Phi2(t,i)=0.5*Phi2(t-1,i)+0.5*(PhiStar(i)-DT*H); end end
figure(2)
mesh(Taxis,Xaxis,Phi2),xlabel('T'),ylabel('X'),zlabel('U'),
set(get(gca, 'YLabel'), 'FontName', 'Times New Roman', 'FontSize', 16)
set(get(gca, 'XLabel'), 'FontName', 'Times New Roman', 'FontSize', 16)
set(get(gca, 'ZLabel'), 'FontName', 'Times New Roman', 'FontSize', 16)
% -----------Comparisons at Specific Time Sreps: t=0.10 ------------------
figure(3)
plot(X,Phi1(21,:),'-',X,Phi2(21,:),'r.-','Linewidth',2)
xlabel('X'),ylabel('U (T=0.10)'), legend('Lax-Friedrichs','Second Runge-Kutta / ENO')
% ---------------------- t=0.30 -------------------------------
figure(4)
plot(X,Phi1(61,:),'-',X,Phi2(61,:),'r.-','Linewidth',2)
xlabel('X'),ylabel('U (T=0.30)'), legend('Lax-Friedrichs','Second Runge-Kutta / ENO')
% ---------------------- t=0.50 -------------------------------
figure(5)
plot(X,Phi1(101,:),'-',X,Phi2(101,:),'r.-','Linewidth',2)
xlabel('X'),ylabel('U (T=0.50)'), legend('Lax-Friedrichs','Second Runge-Kutta / ENO')
% ---------------------- t=0.70 -------------------------------
figure(6)
plot(X,Phi1(141,:),'-',X,Phi2(141,:),'r.-','Linewidth',2)
xlabel('X'),ylabel('U (T=0.70)'), legend('Lax-Friedrichs','Second Runge-Kutta / ENO')
% ---------------------- t=0.90 -------------------------------
figure(7)
plot(X,Phi1(181,:),'-',X,Phi2(181,:),'r.-','Linewidth',2)
xlabel('X'),ylabel('U (T=0.90)'), legend('Lax-Friedrichs','Second Runge-Kutta / ENO')
```

Chapter 5 Static Equations

5.1 General Discussion

Are we sure that we would like to discuss static problem with FDM? Finite element method (FEM) could be the one that really fits such a goal since irregular spatial domains can be discretized easier with FEM comparing to FDM. So, if we need to work with static equations or handle irregular spatial domains, one option is to quit FDM here and refer to FEM for the spatial discretization. However, since we are in FDM, we make some very superficial discussion for the static problem.

An example 1D static equation can be written as

$$\begin{cases} -u_{xx} = f(x), & x \in [-1, 1] \\ u(-1) = u(1) = 0 \end{cases}$$

To solve the equation, we need to first perform the spatial discretization and then solve the discretized equation system.

Use the uniform mesh, and the central difference

$$u_{xx}(x_i) \approx \delta_x^2 U_j = \frac{U_{i+1} + U_{i-1} - 2U_i}{\Delta x^2}$$

The original problem can be discretized into

$$-\frac{U_{i+1} + U_{i-1} - 2U_i}{\Delta x^2} = F_i$$

It is simple to show that the truncation error is of the second order.

$$T_i = \frac{u(x_{i+1}) + u(x_{i-1}) - 2u(x_i)}{\Delta x^2} - f(x_i) \to O(\Delta x^2)$$

By listing all the equations, we obtain a linear system.

$$\begin{aligned} -U_0 + 2U_1 - U_2 &= \Delta x^2 F_1 \\ -U_1 + 2U_2 - U_3 &= \Delta x^2 F_2 \\ &\cdots\cdots \\ -U_{i-1} + 2U_i - U_{i+1} &= \Delta x^2 F_i \\ &\cdots\cdots \\ -U_{I-1} + 2U_I - U_{I+1} &= \Delta x^2 F_I \end{aligned}$$

Or we can write it into a matrix form

$$\underbrace{\begin{pmatrix} 2 & -1 & 0 & \dots & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ 0 & -1 & 2 & -1 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & -1 & 2 & -1 \\ 0 & \dots & 0 & 0 & -1 & 2 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ \dots \\ \dots \\ U_{I-1} \\ U_I \end{pmatrix}}_{\vec{U}} = \underbrace{\begin{pmatrix} \Delta x^2 F_1 + U_0 \\ \Delta x^2 F_2 \\ \Delta x^2 F_3 \\ \dots \\ \dots \\ \Delta x^2 F_{I-1} \\ \Delta x^2 F_I + U_{I+1} \end{pmatrix}}_{\vec{F}}$$

The BCs are put in the RHS of the first and last equations in the linear system.

Example 5.1 (Eikonal Equaiton)
Consider the following nonlinear HJ equation

$$\begin{cases} |\phi_x| = f(x),\ x \in [-1,1] \\ \phi(-1) = \phi(1) = 0 \end{cases}$$

where the Hamiltonian is $H(\phi_x) = |\phi_x|$.

In general, the scheme can be $\widehat{H}(u_i^-, u_i^+) = F_i$, where $u_i = \phi_x(x_i)$. The varnishing viscosity solution is defined as

$$H(\phi_x^\varepsilon) = |\phi_x^\varepsilon| = f(x) + \varepsilon\phi_{xx}^\varepsilon,\ \text{s.t. } \phi^\varepsilon \to \phi \text{ as } \varepsilon \to 0$$

Two example schemes can be written as

$$\text{Lax-Friedrichs Scheme:}\quad \widehat{H}^{LF}(u_i^-, u_i^+) = H\left(\frac{u_i^- + u_i^+}{2}\right) - \frac{\alpha}{2}\left(u_i^+ - u_i^-\right)$$

$$\text{Godunov Scheme:}\quad \widehat{H}^{G}(u_i^-, u_i^+) = \text{ext}_{x\in I(u_i^-,u_i^+)} H(u)\,,\ \text{ext}_{x\in I(u_i^-,u_i^+)} = \begin{cases} \min\limits_{a\le u\le b}, & \text{if } a \le b \\ \max\limits_{b\le u\le a}, & \text{if } b \le a \end{cases}$$

For Lax-Friedrichs Scheme, recall $u_i^- = (\phi_i - \phi_{i-1})/\Delta x$, $u_i^+ = (\phi_{i+1} - \phi_i)/\Delta x$, and we obtain,

$$H\left(\frac{\phi_{j+1} - \phi_{j-1}}{2\Delta x}\right) - \frac{\alpha\Delta x}{2}\left(\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2}\right) = F_i$$

$$\text{or}\qquad \frac{|\phi_{j+1} - \phi_{j-1}|}{2\Delta x} - \frac{\alpha\Delta x}{2}\left(\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2}\right) = F_i$$

We can further write is into an equation system

$$\begin{aligned} &G(\phi_0,\phi_1,\phi_2) = F_1 \\ &G(\phi_1,\phi_2,\phi_3) = F_2 \\ &\quad\cdots\cdots \\ &G(\phi_{i-1},\phi_i,\phi_{i+1}) = F_i \\ &\quad\cdots\cdots \\ &G(\phi_{I-2},\phi_{I-1},\phi_I) = F_{I-1} \\ &G(\phi_{I-1},\phi_I,\phi_{I+1}) = F_I \end{aligned} \quad \Longrightarrow \vec{\phi} = \vec{h}(\vec{\phi}) = \begin{cases} \phi_1 = h(\phi_0,\phi_2) \\ \phi_2 = h(\phi_1,\phi_3) \\ \quad\cdots\cdots \\ \phi_i = h(\phi_{i-1},\phi_{i+1}) \\ \quad\cdots\cdots \\ \phi_{I-1} = h(\phi_{I-2},\phi_I) \\ \phi_I = h(\phi_{I-1},\phi_{I+1}) \end{cases}$$

At step i, the Lax-Friedrichs scheme leads to the following formula to calculate ϕ_i

$$\phi_i = \frac{\Delta x}{\alpha}\left[F_i - \left[H\left(\frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x}\right) - \frac{\alpha}{2}\frac{\phi_{i+1} - \phi_{i-1}}{\Delta x}\right]\right]$$

Thus, given an initial guess, $\vec{\phi}_0$, using the equation system above to do the iterations. Then, the solution can be obtained when the iteration converges, i.e.,

$$|\vec{\phi}^{n+1} - \vec{\phi}^n| \le \delta$$

▲

Now we begin to discuss the algorithms for iterations. Give a linear system

$$\begin{aligned} a_{11}\phi_1 + a_{12}\phi_2 + ... + a_{1n}\phi_n &= F_1 \\ a_{21}\phi_1 + a_{22}\phi_2 + ... + a_{2n}\phi_n &= F_2 \\ &..... \\ a_{n1}\phi_1 + a_{n2}\phi_2 + ... + a_{nn}\phi_n &= F_n \end{aligned}$$

Write it into a matrix form, $A\vec{\phi} = \vec{F}$, and define the time step as t

1. **JACOBI:**

$$\begin{aligned} \phi_1^{t+1} &= \frac{1}{a_{11}}\left[F_1 - \sum_{j=2}^{n} a_{1j}\phi_j^t\right] \\ \phi_2^{t+1} &= \frac{1}{a_{22}}\left[F_2 - \sum_{j=1,j\neq 2}^{n} a_{2j}\phi_j^t\right] \\ &..... \\ \phi_n^{t+1} &= \frac{1}{a_{nn}}\left[F_n - \sum_{j=1,j\neq 2}^{n} a_{nj}\phi_j^t\right] \end{aligned}$$

2. **GAUSS-SEIDEL:**

$$\begin{aligned} \phi_1^{t+1} &= \frac{1}{a_{11}}\left[F_1 - \sum_{j=2}^{n} a_{1j}\phi_j^t\right] \\ \phi_2^{t+1} &= \frac{1}{a_{22}}\left[F_2 - \sum_{j>2}^{n} a_{2j}\phi_j^t - a_{11}\phi_1^{t+1}\right] \\ &..... \\ \phi_n^{t+1} &= \frac{1}{a_{nn}}\left[F_n - \sum_{j<n}^{n} a_{nj}\phi_j^{t+1}\right] \end{aligned}$$

Do the iterations until $|\vec{\phi}^{k+1} - \vec{\phi}^k| < \delta$.

Remark *For the Lax Friedrichs scheme, it returns a nonlinear system. Will the iteration converge? Since the Lax Friedrichs scheme is monotone, we can show that eventually, the solution will converge by using the monotone converge theorem. Monotone is a good property for Hamiltonian, which helps prove the convergence of the nonlinear system.*

5.2 Elliptic Equations

In this section, we just list some examples to show how to solve the elliptic equations with Dirichlet and Neumann BCs.

Example 5.2 Solve the elliptic problem with homogeneous BCs

$$\begin{cases} -\Delta u = 2\sin x \sin y, \quad G = (0,\pi)\times(0,\pi) \\ u|_{\partial G} = 0. \end{cases}$$

Solve: This is a problem with Dirichlet BCs, and we just need to plug the BC equations into the discrete equation system. And the unknowns in the discrete equation system are only the values for inner mesh points of the computational domain. The result is shown in Fig.5.1. ▲

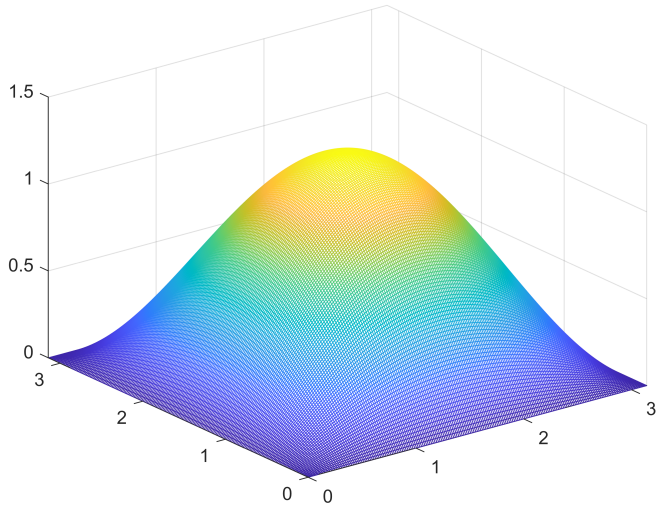

Figure 5.1: The Result of Ellipse Problem with Homogeneous BCs.

Appendix. The MATLAB code

```
% ------------ Math 517 Chap.5 Exercise 1 --------
clear all; close all; clc
f=@(x,y) 2*sin(x)*sin(y);
err=0.0001; check=1;n=200; X=pi;Y=pi;
dx=X/n; xaxis=0:dx:X; dy=Y/n; yaxis=0:dy:Y; [Xaxis,Yaxis]=meshgrid(xaxis,yaxis);
A=zeros((n+1)*(n+1),5); D=zeros((n+1)*(n+1),1);
U=ones((n+1)*(n+1),1); Um=ones((n+1)*(n+1),1);
for i=2:n
    A((i-1)*(n+1)+1,3)=1;D((i-1)*(n+1)+1,1)=0;
    for j=2:n
        % -------- Coefficients ------------------
        A((i-1)*(n+1)+j,1)=-1;A((i-1)*(n+1)+j,2)=-1;
        A((i-1)*(n+1)+j,3)=4;A((i-1)*(n+1)+j,4)=-1;
        A((i-1)*(n+1)+j,5)=-1;
        % -------- Integral for non-homogeneous term --------
        D((i-1)*(n+1)+j,1)=dx*dx*f((i-1)*dx,(j-1)*dy); end
    A(i*(n+1),3)=1;D(i*(n+1),1)=0; end
% --------- install the two bondary conditions --------------
for j=1:(n+1)
    A(j,3)=1;D(j,1)=0; end
for j=1:(n+1)
    A(n*(n+1)+j,3)=1;D(n*(n+1)+j,1)=0; end
% -------- Get U by Gauss-Seidel method ---------------
while(check)
    Um=U;
    for i=1:n+1
```

```
        U(i,1)=0; end
    for i=2:n
        U((i-1)*(n+1)+1,1)=0;
        for j=2:n
            U((i-1)*(n+1)+j,1)=(-A((i-1)*(n+1)+j,1)*U((i-2)*(n+1)+j,1)-A((i-1)*(n+1)+j,2)*U((
                i-1)*(n+1)+j-1,1)-A((i-1)*(n+1)+j,4)*U((i-1)*(n+1)+j+1,1)-A((i-1)*(n+1)+j,5)*U
                (i*(n+1)+j,1)+D((i-1)*(n+1)+j,1))/A((i-1)*(n+1)+j,3); end
        U(i*(n+1),1)=0; end
    for i=n*(n+1)+1:(n+1)*(n+1)
        U(i,1)=0; end
    error=0;
    for i=1:(n+1)*(n+1)
        if(abs(Um(i,1))>0)
            l=abs(U(i,1)-Um(i,1))/abs(Um(i,1));
            if(error<l)
                error=l;  end end end
    if(error<err)
        check=0;
    else
        check=1;
    end end
% -------- Rearrange The Solution ---------------
UU=zeros(n+1);
for i=1:n+1
    UU(:,i)=U((i-1)*(n+1)+1:i*(n+1)); end
figure(1)
mesh(Xaxis,Yaxis,UU');
```

Example 5.3 Solve the elliptic problem with non-homogeneous BCs

$$\begin{cases} -\left[\dfrac{\partial^2 u}{\partial x^2}+3\dfrac{\partial^2 u}{\partial y^2}\right]=16, \quad G=[0,1]\times[0,1] \\ u|_{x=1}=0,\ \partial_y u|_{y=1}=-u,\ \partial_x u|_{x=0}=0,\ \partial_y u|_{y=0}=0 \end{cases}$$

Solve: This is a problem with Neumann BCs. We need to make extrapolations using the BCs, i.e., the derivative terms. Then, we use the extrapolation data to form a similar discretized equation on those boundary points in the way similar to the inner points. The result is shown in Fig.5.2. ▲

Appendix. The MATLAB code

```
% ------------ Math 517 Chap.5 Exercise 2 --------
clear all; close all; clc
err=0.00005;check=1;n=100;X=1;Y=1;
dx=X/n; xaxis=0:dx:X;dy=Y/n; yaxis=0:dy:Y;
[Xaxis,Yaxis]=meshgrid(xaxis,yaxis);
A=zeros((n+1)*(n+1),5);D=zeros((n+1)*(n+1),1);
U=ones((n+1)*(n+1),1);Um=ones((n+1)*(n+1),1);
for i=2:n
    A((i-1)*(n+1)+1,1)=-1;A((i-1)*(n+1)+1,2)=0;A((i-1)*(n+1)+1,3)=8;
```

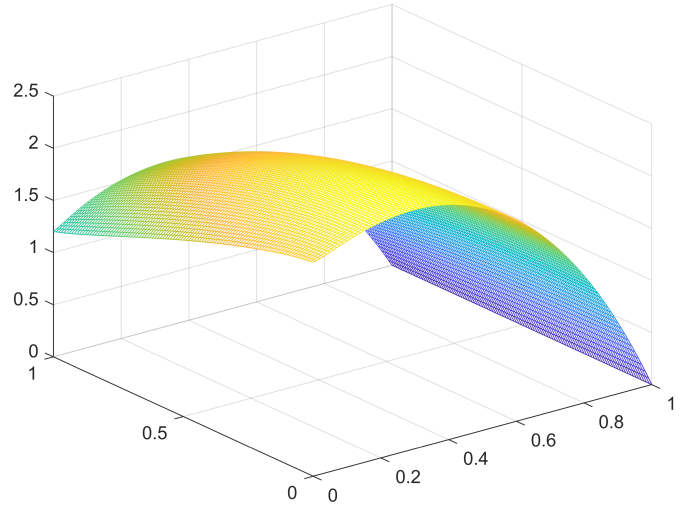

Figure 5.2: The Result of Ellipse Problem with Non-homogeneous BCs.

```
    A((i-1)*(n+1)+1,4)=-6;A((i-1)*(n+1)+1,5)=-1;D((i-1)*(n+1)+1,1)=16*dx*dx;
    for j=2:n
        % -------- Coefficients ------------------
        A((i-1)*(n+1)+j,1)=-1;A((i-1)*(n+1)+j,2)=-3;A((i-1)*(n+1)+j,3)=8;
        A((i-1)*(n+1)+j,4)=-3;A((i-1)*(n+1)+j,5)=-1;
        % -------- Integral for non-homogeneous term --------
        D((i-1)*(n+1)+j,1)=16*dx*dx; end
    A(i*(n+1),1)=-1;A(i*(n+1),2)=-6;
    A(i*(n+1),3)=8+6*dy;A(i*(n+1),4)=-0;
    A(i*(n+1),5)=-1;D(i*(n+1),1)=16*dx*dx; end
% --------- install the bondary conditions --------------
A(1,1)=0;A(1,2)=0;A(1,3)=8;A(1,4)=-6;A(1,5)=-2;D(1,1)=16*dx*dx;
for j=2:n
    A(j,1)=0;A(j,2)=-3;A(j,3)=8;A(j,4)=-3;A(j,5)=-2;D(j,1)=16*dx*dx; end
A(n+1,1)=0;A(n+1,2)=-6;A(n+1,3)=8+6*dy;A(n+1,4)=0;A(n+1,5)=-2;D(n+1,1)=16*dx*dx;
for j=1:(n+1)
    A(n*(n+1)+j,3)=1;D(n*(n+1)+j,1)=0; end
% -------- Get U by Gauss-Seidel method ---------------
while(check)
    Um=U;
    U(1,1)=-(A(1,4)*U(2,1)+A(1,5)*U(n+2,1)-D(1,1))/A(1,3);
    for i=2:n
        U(i,1)=-(A(1,2)*U(i-1,1)+A(1,4)*U(i+1,1)+A(1,5)*U(n+1+i,1)-D(i,1))/A(i,3); end
    U(n+1,1)=-(A(n+1,2)*U(n,1)+A(n+1,5)*U(2*(n+1),1)-D(n+1,1))/A(n+1,3);
    for i=2:n
        U((i-1)*(n+1)+1,1)=-(A((i-1)*(n+1)+1,1)*U((i-2)*(n+1)+1,1)+A((i-1)*(n+1)+1,4)*U((i-1)
            *(n+1)+2,1)+A((i-1)*(n+1)+1,5)*U((i)*(n+1)+1,1)-D((i-1)*(n+1)+1,1))/A((i-1)*(n+1)
            +1,3);
        for j=2:n
            k=0;
            k=k-A((i-1)*(n+1)+j,1)*U((i-2)*(n+1)+j,1)-A((i-1)*(n+1)+j,2)*U((i-1)*(n+1)+j-1,1)
```

```
            ;
        k=k-A((i-1)*(n+1)+j,4)*U((i-1)*(n+1)+j+1,1)-A((i-1)*(n+1)+j,5)*U((i)*(n+1)+j,1);
        U((i-1)*(n+1)+j,1)=(k+D((i-1)*(n+1)+j,1))/A((i-1)*(n+1)+j,3); end
    U(i*(n+1),1)=-(A(i*(n+1),1)*U((i-1)*(n+1),1)+A(i*(n+1),2)*U((i)*(n+1)-1,1)+A(i*(n+1)
        ,5)*U((i+1)*(n+1),1)-D(i*(n+1),1))/A(i*(n+1),3); end
for i=n*(n+1)+1:(n+1)*(n+1)
    U(i,1)=0; end
error=0;
for i=1:(n+1)*(n+1)
    if(abs(Um(i,1))>0)
        l=abs(U(i,1)-Um(i,1))/abs(Um(i,1));
        if(error<l)
            error=l;  end end end
if(error<err)
    check=0;
else
    check=1;
end end
% -------- Rearrange The Solution ---------------
UU=zeros(n+1);
for i=1:n+1
    UU(i,:)=U((i-1)*(n+1)+1:i*(n+1)); end
figure(1)
mesh(Xaxis,Yaxis,UU');
```