Numerical Partial Differential Equations

(Draft)

I-Liang Chern

National Taiwan University

May 13, 2024

Contents

Chapter 1

Numerical Ordinary Differential Equations

The goal of this course is to introduce theoretical analysis of finite difference methods for solving partial differential equations, with a focus on stability and convergence theory. The partial differential equations to be discussed include

- parabolic equations,
- elliptic equations,
- hyperbolic conservation laws.

In Chapter 1, we will discuss finite difference approximations for differential operators, basic ODE solvers, Runge-Kutta methods, multi-step methods, and basic stability analysis for ODEs.

1.1 Finite Difference Approximation

- To approximate the derivatives u_x, u_{xx}, and so on, using the grid-values of u.
- Method: Taylor expansion.

1.1.1 Simple examples

1. Finite difference: Given a smooth function u defined on $\mathbb{R}$, we want to approximate $u'(x)$ by $u(x)$, $u(x \pm h)$. Here are some examples:
   - Forward differencing: $D_+u(x) := \frac{u(x+h)-u(x)}{h}$,
   - Backward differencing: $D_-u(x) := \frac{u(x)-u(x-h)}{h}$,
   - Centered differencing: $D_0u(x) := \frac{u(x+h)-u(x-h)}{2h}$.

Here, h is called the mesh size. By Taylor expansion, we can obtain:

- $u'(x) = D_+u(x) + O(h)$,
- $u'(x) = D_-u(x) + O(h)$,
- $u'(x) = D_0u(x) + O(h^2)$.

The notation $O(h)$ denotes for a function of $h \sim 0$, say $e(h)$, such that

$$|e(h)| \leq Ch \text{ in a neighborhood of } 0,$$

for some constant C independent of h. *

2. These formulae can be derived by performing Taylor expansion of u at x. For instance, we expand

$$\begin{aligned} u(x+h) &= u(x) + u'(x)h + \frac{h^2}{2}u''(x) + \frac{h^3}{3!}u'''(x) + \cdots \\ u(x-h) &= u(x) - u'(x)h + \frac{h^2}{2}u''(x) - \frac{h^3}{3!}u'''(x) + \cdots . \end{aligned}$$

Subtracting these two equations yields

$$u(x+h) - u(x-h) = 2u'(x)h + \frac{2h^3}{3!}u'''(x) + \cdots .$$

This gives

$$u'(x) = D_0u(x) - \frac{h^2}{3!}u'''(x) + \cdots = D_0u(x) + O(h^2).$$

Thus, $u'(x)$ can be approximated by many difference operators with different orders of errors.

3. Example of third-order approximation:

$$u'(x) = D_3u(x) + O(h^3),$$

where

$$D_3u(x) = \frac{1}{6h}\left(2u(x+h) + 3u(x) - 6u(x-h) + u(x-2h)\right).$$

This formula can be derived by taking Taylor expansion of $u(x+h)$, $u(x-h)$, $u(x-2h)$ about x:

$$u(x+h) = u(x) + u'(x)h + \frac{h^2}{2}u''(x) + \frac{h^3}{3!}u'''(x) + \cdots$$

*Examples: the functions $3h + 5h^2$, $e^h - 1$, $\sin(h)$ are $O(h)$ functions. The function $1 - \cos h = O(h^2)$. Note that $O(h) + O(h) = O(h)$, $aO(h) = O(h)$, where a is a constant.

$$
\begin{aligned}
u(x-h) &= u(x) - u'(x)h + \frac{h^2}{2}u''(x) - \frac{h^3}{3!}u'''(x) + \cdots \\
u(x-2h) &= u(x) - 2u'(x)h + \frac{4h^2}{2}u''(x) - \frac{8h^3}{3!}u'''(x) + \cdots
\end{aligned}
$$

Taking the combination $2u(x+h) + 3u(x) - 6u(x-h) + u(x-2h)$, we can cancel the zeroth, second derivatives and obtain $u'(x) = D_3 u(x) + O(h^3)$.

1.1.2 General formula for finite difference approximation

1. Suppose $u(\cdot)$ is a smooth function. Let $\{x_i\}$ be discrete points, called the grid points. Let $\bar{x}$ be a specific point. Given $u_i = u(x_i)$, $i = 0, ..., n$, we want to approximate $u^{(k)}(\bar{x})$ by u_i, $i = 0, ..., n$. That is,

$$u^{(k)}(\bar{x}) = \sum_{j=0}^{n} c_j u(x_j) + O(h^{p-k+1}).$$

   Here, the mesh size h denotes $\max_{0\le i,j\le n}\{|x_i - x_j|\}$. The parameter $p \ge k$ is an approximate power to be determined.

2. This can also be done by finding a polynomial $P_n(x)$ of degree n which interpolates $u(\cdot)$ at $x_0, ..., x_n$. This polynomial P_n is unique. Then $u^{(k)}(\bar{x})$ can be approximated by $P_n^{(k)}(\bar{x})$ with error $O(h^{(n+1-k)}$.

3. To find the coefficients $c_j, j = 0, ..., n$, we take Taylor expansion of $u(x_j)$ about the point $\bar{x}$:

$$u(x_j) = \sum_{i=0}^{p} \frac{1}{i!}(x_j - \bar{x})^i u^{(i)}(\bar{x}) + O(h^{p+1}).$$

   We plug this expansion formula into the finite difference approximation formula for $u^{(k)}(\bar{x})$:

$$u^{(k)}(\bar{x}) = \sum_{j=0}^{n} c_j \sum_{i=0}^{p} \frac{1}{i!}(x_j - \bar{x})^i u^{(i)}(\bar{x}) + O(h^{p-k+1}).$$

   Comparing both sides, equating the coefficients of $u^{(i)}(\bar{x})$ for $i = 0, ..., p$, we obtain

$$\sum_{j=0}^{n} \frac{(x_j - \bar{x})^i}{i!} c_j = \left\{ \begin{array}{ll} 1 & \text{if } i = k \\ 0 & \text{otherwise} \end{array} \right\}, \text{ for } i = 0, ..., p.$$

   There are $p+1$ equations here, it is natural to choose $p = n$ to match the $n+1$ unknowns $(c_0, ..., c_n)$. This is an $(n+1)\times(n+1)$ Vandermonde system. It is nonsingular if $\{x_i, i = 0, ..., n\}$ are different. The matlab code fdcoeffV(k,xbar,x) can be used to compute these coefficients. Reference: Randy LeVeque's book and his Matlab code.

4. In the case of uniform grid, using central finite differencing, we can get high order approximation by using less grid points. For instance, let $x_j = jh$, where $j \in \mathbb{Z}$. Let $u_j = u(x_j)$. Then

$$\begin{aligned} u'(0) &= \frac{u_1 - u_{-1}}{2h} + O(h^2) \\ u''(0) &= \frac{u_1 - 2u_0 + u_{-1}}{h^2} + O(h^2) \\ u^{(3)}(0) &= \frac{1}{2h^3}(u_2 - 2u_1 + 2u_0 - 2u_{-1} + u_{-2}) + O(h^2). \end{aligned}$$

Homeworks 1.1. 1. Consider $x_i = ih$, $i = 0, ..., n$. Let $\bar{x} = x_m$. Find the coefficients c_i for $u^{(k)}(\bar{x})$ and the coefficient of the leading truncation error for the following cases:

- $k = 1$, $n = 2, 3$, $m = 0, 1, 2, 3$.
- $k = 2$, $n = 2$, $m = 0, 1, 2$.

1.2 Elementary Numerical Methods for Solving ODEs

1.2.1 Model equations and some target models

1. Some applications

   - Circuit simulations: this involves large algebraic-differential equations. Usually, they are linear. For instance, the Very-Large-Scale Integration (VLSI) circuits. There are also nonlinear circuit models for circuits with semiconductor devices.
   - Dynamical systems from celestial dynamics. This is important in space sciences.
   - Chemical kinetics, etc.
   - Dynamical systems derived from fluid systems
   - Molecular dynamics, N-body problems.
   - Dynamical systems from chemistry, biology, ...

2. Examples of some concrete models

   - For designing numerical solvers, we should have some simple model equations in mind to test, to justify the methods work. A simple model is the linear ODE system:

     $$\mathbf{y}' = A\mathbf{y}$$

     where $\mathbf{y}$ is an n-vector and A is an $n \times n$ matrix. The eigenvalues of A can be real, or complex. Thus, we consider the simplex case

     $$y' = ay$$

     The constant a can be positive, negative, or complex.

- Chemical kinetics.
- Synchronization model
- Van de Pole oscillator
- double pendulum

These can be used for testing your codes.

3. A basic assumption to design numerical algorithms for solving ODEs is the smoothness of the underlying solutions. This assumption is in general valid provided the coefficients are also smooth. Basic designing techniques include numerical interpolation, numerical integration, and finite difference approximation. Yet, there are problems whose solutions are discontinuous.[†] We shall not discuss those problems here.

1.2.2 Forward Euler method

1. **Forward finite differencing** The forward Euler method is the simplest numerical integrator for ODEs. The ODE
$$y' = f(t, y) \tag{1.1}$$
is discretized by
$$\boxed{y^{n+1} = y^n + kf(t^n, y^n).} \tag{1.2}$$
Here, $t^0, ..., t^n$ are the grid points of time t. The difference $k = t^{n+1} - t^n$ is called the time step size of the discretization. [‡] The forward Euler method simply replaces $dy/dt(t^n)$ by the forward finite difference $(y^{n+1} - y^n)/k$. The state y^{n+1} is expressed as an explicit formula in terms of y^n. Such a scheme is called an explicit scheme.

2. **Truncation error and Consistency** Let us plug a smooth solution $y(\cdot)$ of (1.1) into the discrete equation (1.2). The remaining term is called the *truncation error* τ^n:
$$y(t^{n+1}) = y(t^n) + kf(t^n, y(t^n)) + k\tau^n.$$
A numerical method is called *consistent* if $\tau^n(k) \to 0$ as $k \to 0$, uniformly in n. By the Taylor expansion:
$$y(t^{n+1}) - y(t^n) = y'(t^n)k + O(k^2).$$
Thus, the truncation error $\tau^n(k)$ of the forward Euler method satisfies
$$\tau^n(k) = O(k).$$
A numerical method is called of *order p* if $\tau^n(k) = O(k^p)$ uniformly in n. A numerical method with order $p \geq 1$ is always consistent. The forward Euler method is a first order method.

[†]Consider the ODE for a bumping ball. An ODE with constraint.
[‡]Let us assume fixed step size.

3. **True error** We are interested in the *true error*, which is defined to be $e^n := y^n - y(t^n)$. We have the following convergence theorem.

Theorem 1.1. *Assume $f \in C^1$ and suppose the solution $y' = f(t,y)$ with $y(0) = y_0$ exists on $[0,T]$. Then the forward Euler method converges at any $t \in [0,T]$. In fact, the true error e^n has the following estimate:*

$$|e^n| \le \frac{e^{\lambda t}}{\lambda} O(k) \to 0, \ \text{as } n \to \infty. \tag{1.3}$$

Here, $\lambda = \max |\partial f/\partial y|$, and $nk = t$ is fixed.

Proof. From the regularity of the solution, we have $y \in C^2[0,T]$ and

$$y(t^{n+1}) = y(t^n) + kf(t^n, y(t^n)) + k\tau^n. \tag{1.4}$$

Taking difference of (1.2) and (1.4), we obtain a finite difference inequality:

$$\begin{aligned} |e^{n+1}| &\le |e^n| + k|f(t^n, y^n) - f(t^n, y(t^n))| + k|\tau^n| \\ &\le (1 + k\lambda)|e^n| + k|\tau^n|, \end{aligned}$$

where

$$|f(t,x) - f(t,y)| \le \lambda |x - y|.$$

This finite difference inequality has a fundamental solution $G^n = (1+\lambda k)^n$, which is positive. Multiplying above equation by $(1+\lambda k)^{-n-1}$, we obtain

$$|e^{n+1}|G^{-n-1} \le |e^n|G^{-n} + kG^{-n-1}|\tau^n|.$$

Let us rename the index n by m, summing this equality in m from $m = 0$ to $n-1$, we get

$$|e^n|G^{-n} \le |e^0| + k \sum_{m=0}^{n-1} G^{-m-1}|\tau^m|$$

Note that $e^0 = 0$. Multiply both side by G^n, we get

$$\begin{aligned} |e^n| &\le \sum_{m=0}^{n-1} G^{n-m-1} k|\tau^m| \le \sum_{m=0}^{n-1} G^m O(k^2) \\ &= \frac{G^n - 1}{G - 1} O(k^2) \le \frac{G^n}{\lambda} O(k) \le \frac{e^{\lambda t}}{\lambda} O(k), \end{aligned}$$

where $t = nk$ and we have used $(1+\lambda k)^n = \left(1 + \frac{\lambda t}{n}\right)^n \le e^{\lambda t}$. □

4. **Remarks**

(a) The theorem states that the numerical method converges in $[0, T]$ as long as the solutions of the ODE exists. The proof above uses the existence and smoothness of the solution. One can also use this approach to prove the local existence theorem by showing the approximate solutions generated by the Euler method form a Cauchy sequence, and thus converge to a true solution. Here, however, we are interested in the error analysis of numerical method, not on the existence theory. We thus assume the existence and regularity theorems from ODE theory.

(b) The true error of the forward Euler method is $O(k)$, where $O(k)$ is of magnitude $e^{\lambda t}/\lambda$ with $\lambda \sim |f'(y)|$. This λ can be very large and the error constant $e^{\lambda t}$ can be very large. For accuracy purpose, we need to choose very small k. Thus, the selection of the step size k is based on both (i) stability and (ii) accuracy considerations.

5. **Absolute Stability and step-size restriction.**

(a) In many applications, we look for some stable equilibria. We want our numerical method can reproduce such stable process during computation. This leads to a restriction on the step size k. To derive a condition, we try a simple model:

$$y' = ay, \quad a < 0.$$

The solution $y(t) \to 0$ as $t \to \infty$. The forward Euler method gives an approximate solution

$$y^{n+1} = y^n + kay^n = (1 + ka)y^n.$$

The approximate solution is a geometric sequence:

$$y^n = (1 + ka)^n y^0,$$

where $y^0 = y(0)$ is the initial state. Because $a < 0$, the exact solution $y(t) \to 0$ as $t \to \infty$. (The state 0 is called a stable equilibrium.) However, the geometric sequence

$$|(1 + ka)^n| \to \begin{cases} 0 & \text{if } |1 + ka| < 1, \\ \infty & \text{if } |1 + ka| > 1. \end{cases}$$

Thus, to reproduce such stable process as we have for the continuous situation, we should choose the step size k such that $|1 + ka| < 1$.

(b) When $a < 0$ and $|a|$ is large, we need to choose very large k. Such ODE is called a stiff ODE. It is usually solved by a implicit method such as the backward Euler method below.

(c) In later applications in numerical PDEs, a can be a complex number. Let us allow complex value a. A numerical method for $y' = ay$ is called stable if its discrete solutions $\{y^n\}$ are bounded for all n. We combine $ka = z \in \mathbb{C}$. The region of z

so that $\{y^n\}$ is bounded is called an absolute stability region of the method. for the Forward Euler method, its absolute stability region is

$$\{z \in \mathbb{C} || 1 + z| \leq 1\}.$$

It is used to reproduce stable discrete solutions for stable equilibria.

1.2.3 Backward Euler method

1. **Backward Euler method** approximates (1.1) by

$$\boxed{y^{n+1} = y^n + kf(t^{n+1}, y^{n+1}).} \tag{1.5}$$

Note that y^{n+1} also appears inside $f(t^{n+1}, y^{n+1})$. We need to solve an equation for y^{n+1} in this scheme. A scheme with y^{n+1} appeared implicitly is called an implicit scheme. The backward Euler method is particular useful for stiff ODE, where $\partial f/\partial y << 0$.

2. **Truncation error** τ^n is defined by

$$y(t^{n+1}) = y(t^n) + kf(t^{n+1}, y(t^{n+1})) + k\tau^n(k),$$

where $y(\cdot)$ is a true solution. Comparing the Taylor expansion of the exact solution $y(\cdot)$ at t^{n+1}:

$$y(t^{n+1}) = y(t^n) + ky'(t^{n+1}) + k\tau^n(k),$$

we get that the truncation error

$$\tau^n(k) = O(k).$$

3. **Error analysis** The true error $e^n := y^n - y(t^n)$ satisfies

$$\begin{aligned} e^{n+1} &= e^n + k\left(f(t^{n+1}, y^{n+1}) - f(t^{n+1}, y(t^{n+1}))\right) + O(k^2) \\ &= e^n + k\left(\frac{\partial f}{\partial y}(t, \bar{y})\right) e^{n+1} + O(k^2). \end{aligned}$$

This implies

$$|e^{n+1}| \leq |e^n| + k\lambda|e^{n+1}| + O(k^2).$$

where

$$\lambda = \max \left|\frac{\partial f}{\partial y}(t, \bar{y})\right|.$$

We choose k small enough such that $1 - k\lambda > 0$. We then get the inequality:

$$|e^{n+1}| \leq (1 - k\lambda)^{-1}\left(|e^n| + O(k^2)\right).$$

Let $G = (1-k\lambda)^{-1}$. We get

$$\begin{aligned}
|e^n| &\le G\left(|e^{n-1}| + k\tau^{n-1}\right) \\
&\le G\left(G(|e^{n-2}| + k\tau^{n-2}) + k\tau^{n-1}\right) \\
&\vdots \\
&\le G^n|e^0| + k\left(G^{n-1}\tau^1 + \cdots G\tau^{n-1}\right) \\
&\le Ck^2\left(G^{n-1} + \cdots + G\right) \\
&\le Ck^2\frac{G^n - G}{G-1} \\
&\le Ck^2\frac{(1-k\lambda)^{-n+1} - 1}{k\lambda} \\
&\le Cke^{\lambda t}/\lambda
\end{aligned}$$

where $nk = t$ and we have used $(1-k\lambda)^{-n+1} \le e^{\lambda t}$. □

Remark. The backward Euler method is usually used for stiff ODE where $\partial f/\partial y < < 0$. In such cases, the error estimate is only applicable for very small t because the constant in the error analysis is too big. Below, we show a useful error estimate.

Suppose $\frac{\partial f}{\partial y}(t,\bar{y}) \le -\mu$ ($\mu > 0$) in the region we are interested. If $e^{n+1} \ge 0$, we have

$$\begin{aligned}
e^{n+1} &= e^n + k\left(\frac{\partial f}{\partial y}(t,\bar{y})\right)e^{n+1} + O(k^2) \\
&\le e^n - k\mu e^{n+1} + O(k^2).
\end{aligned}$$

If $e^{n+1} < 0$, we have

$$\begin{aligned}
-e^{n+1} &= -e^n + k\left(\frac{\partial f}{\partial y}(t,\bar{y})\right)(-e^{n+1}) + O(k^2) \\
&\le |e^n| - k\mu(-e^{n+1}) + O(k^2).
\end{aligned}$$

In both cases, we get

$$(1+\mu k)|e^{n+1}| \le |e^n| + O(k^2).$$

The error satisfies

$$\begin{aligned}
|e^n| &\le \sum_{m=0}^{n-1}(1+\mu k)^{-m}O(k^2) \\
&\le \frac{(1+\mu k)^{-n+1}}{\mu k}O(k^2) \\
&\le \frac{e^{-\mu t}}{\mu}O(k).
\end{aligned}$$

4. **Absolute stability region**
To find the absolute stability region, we consider the ODE $y' = ay$ with $a < 0$. The backward Euler scheme produces the approximate solution:

$$y^{n+1} = y^n + kay^{n+1}.$$

This gives

$$y^n = (1 - ka)^{-n} y^0.$$

We see that this solution $y^n \to 0$ as $n \to \infty$ for any $k > 0$. We thus call the backward Euler scheme *unconditional stable*. We combine $ka = z$. The region for z such that the scheme is absolutely stable is

$$\{z \in \mathbb{C} || 1 - z| \geq 1\}.$$

5. **Solving a nonlinear equation for implicit methods.** In the backward Euler method:

$$y^{n+1} = y^n + kf(t^{n+1}, y^{n+1}),$$

one needs to solve a nonlinear equation for y^{n+1}. Let us write this as a nonlinear equation in x:

$$F(x) = 0, \quad \text{where} \quad F(x) = x - y^n - kf(t^{n+1}, x)$$

There are two standard iterative methods to solve nonlinear equations:

- Newton's method The method generates a sequence of approximate solutions $\{x_n | n = 0, 1, \cdots\}$. We start by choosing an x_0 (here, we choose $x_0 = y^n$). Suppose we have obtain x_n, to find x_{n+1}, instead of solving $F(x) = 0$, we solve the linearized equation about x_n:

  $$F(x_n) + F'(x_n)(x - x_n) = 0.$$

  This gives

  $$x_{n+1} = x_n - (F'(x_n))^{-1} F(x_n).$$

  If x_0 is closed to the root x_*, then one can prove $x_n \to x_*$.
- Fixed-point method: this is also an iterative method which generates a sequence of approximate solutions $\{x_n\}$. We start by choosing an x_0 (say $x_0 = y^n$). The iteration scheme is

  $$x_{n+1} - x_n = \alpha F(x_n).$$

  The parameter α is chosen so that

  $$|1 + \alpha F'(x)| < 1,$$

  then one can show this scheme converges. [§] A particular choice of α is $-F'(x_0)^{-1}$. With this choice, $1 - \alpha F'(x) \sim 0$. If we choose different α at each step, say $\alpha_n = -F'(x_n)^{-1}$, then this is the Newton's method.

[§]contraction mapping The mapping $\Phi(x) := x + \alpha F(x)$ satisfies $|\Phi(x) - \Phi(y)| \leq \rho |x - y|$. Such a mapping has a fixed point: $\Phi(x) = x$.

The Newton's method may not converge. But if it converges, the convergent rate is second order, meaning

$$|x_{n+1} - x_*| = O(|x_n - x_*|^2).$$

The fixed point method is first order:

$$|x_{n+1} - x_*| \leq \rho |x_n - x_*| \text{ with } \rho < 1.$$

1.2.4 Trapezoidal method

1. **Design idea** We integrate $y' = f(t, y)$ from t^n to t^{n+1}. Then we approximate the integration $\int_{t^n}^{t^{n+1}} f \, dt$ by the trapezoidal method. We then get the trapezoidal method for ODE:

$$\boxed{y^{t^{n+1}} - y^n = \frac{k}{2} \left(f(t^n, y^n) + f(t^{n+1}, y^{n+1}) \right).} \tag{1.6}$$

2. **Truncation error** is

$$\tau = O(k^2).$$

3. **Absolutely stable region.** We consider the equation $y' = ay$ with $a < 0$. Using trapezoidal rule, we get

$$y^{n+1} - y^n = \frac{k}{2} \left(ay^n + ay^{n+1} \right).$$

Thus, we have

$$y^{n+1} = \frac{1 + ak/2}{1 - ak/2} y^n.$$

We call $z = ak$. The absolute stability region is

$$\left\{ z \in \mathbb{C} | \left| \frac{2 + z}{2 - z} \right| \leq 1 \right\}$$

You can show that this region is

$$\{ z = x + iy | x < 0 \},$$

the negative half complex plane.

1.2.5 Leap-frog method (midpoint method)

1. **Design idea** We integrate $y' = f(t, y)$ from t^{n-1} to t^{n+1}:

$$y(t^{n+1}) - y(t^{n-1}) = \int_{t^{n-1}}^{t^{n+1}} f(\tau, y(\tau)) \, d\tau.$$

We apply the midpoint rule for numerical integration, we then get

$$y(t^{n+1}) - y(t^{n-1}) = 2kf(t^n, y(t^n)) + O(k^3).$$

The midpoint method (or called leapfrog method) is

$$\boxed{y^{n+1} - y^{n-1} = 2kf(t^n, y^n).} \tag{1.7}$$

2. **Truncation error.** The corresponding truncation is

$$\tau^n(k) = O(k^2).$$

3. **Absolute stability region.** The absolute stability region for the midpoint method is

$$\{z = iy | -1 \le y \le 1\}.$$

I will postpone the discussion of this section to the end of this chapter.

We plot the absolute stability regions for the forward Euler method, the backward Euler method, the trapezoidal method, the midpoint method.

Homeworks 1.2. 1. Consider the ODE

$$y_1' = y_2, \quad y_2' = -y_1.$$

The solution is $y_1 = A\cos(t+\phi)$, $y_2(t) = -A\sin(t+\phi)$, where A is called its amplitude and $t+\phi$ is its phase. Try the forward Euler, backward Euler, leapfrog, trapezoidal (implicit) methods to see the changes in amplitude and phase. (see Durran, 2.2.2)

1.3 Runge-Kutta methods

The Runge-Kutta methods (RK) are designed by using numerical integration for $\int_{t^n}^{t^{n+1}} f\, d\tau$ by some *quadrature methods*. Below, RK2, RK4 are RK methods with different orders.

1.3.1 RK2 and RK4

1. **Design idea for RK2** A second-order RK, denoted by RK2, is based on the trapezoidal rule of numerical integration. First, we integrate the ODE $y' = f(t, y)$ to get

$$y(t^{n+1}) - y(t^n) = \int_{t^n}^{t^{n+1}} f(\tau, y(\tau))\, d\tau.$$

Next, this integration is approximated by

$$\int_{t^n}^{t^{n+1}} f(\tau, y(\tau))\, d\tau = \frac{k}{2}\left(f(t^n, y^n) + f(t^{n+1}, y^{n+1})\right) + O(k^3).$$

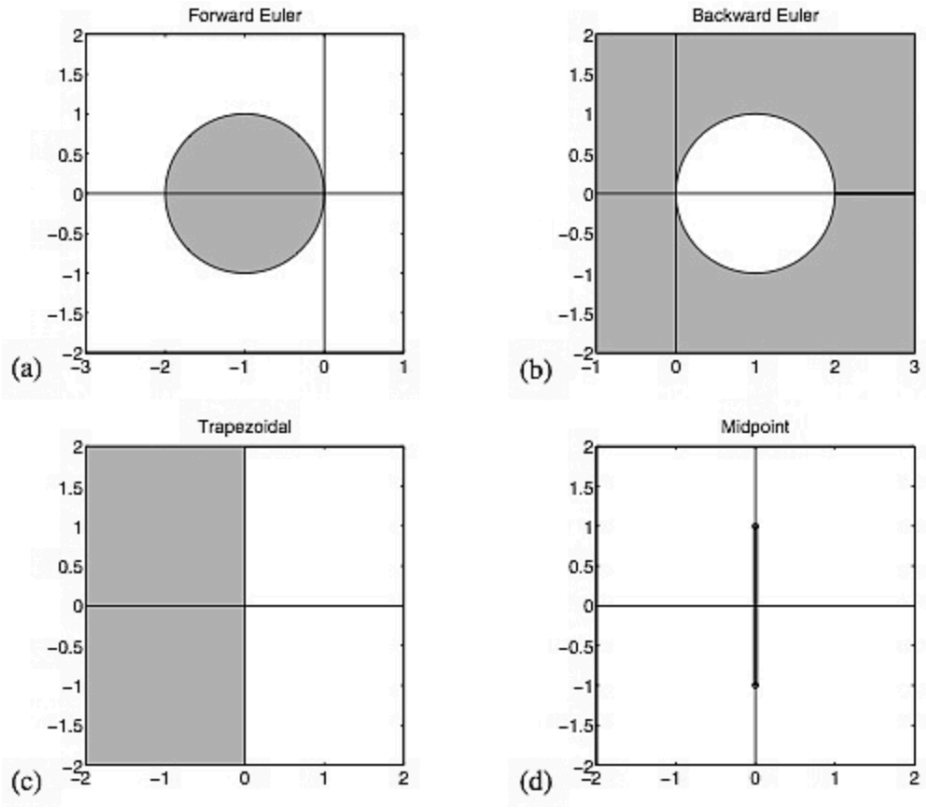


Figure 1.1: Absolute stability regions: (a) forward Euler method, (b) backward Euler method, (c) trapezoidal method, (d) leap frog method. The figure is quoted from a uci website

The second term involves y^{n+1}. An explicit Runge-Kutta method approximates y^{n+1} by $y^n + kf(t^n, y^n)$. Thus, the RK2 reads

$$\boxed{\begin{cases} \xi_1 = f(t^n, y^n) \\ y^{n+1} = y^n + \frac{k}{2}(f(t^n, y^n) + f(t^{n+1}, y^n + k\xi_1)). \end{cases}}$$

2. **Another RK2** Another kind of RK2 is based on the midpoint rule of integration. It reads

$$\begin{aligned} \xi_1 &= f(t^n, y^n) \\ y^{n+1} &= y^n + kf(t^{n+1/2}, y^n + \frac{k}{2}\xi_1) \end{aligned}$$

The truncation error τ^n of RK2 is defined as

$$y(t^{n+1}) = y(t^n) + kf(t^{n+1/2}, y^n + \frac{k}{2}\xi_1) + k\tau^n.$$

$$\begin{aligned}\tau(k) &:= \frac{y(t^{n+1}) - y(t^n)}{k} - \frac{y^{n+1} - y(t^n)}{k} \\ &= \frac{1}{k}\int_{t^n}^{t^{n+1}} f(\tau, y(\tau))\, d\tau - \frac{1}{2}\left(f(t^n, y^n) + f(t^{n+1}, y^n + k\xi_1)\right) + O(k^2) \\ &= \frac{1}{k}\int_{t^n}^{t^{n+1}} f(\tau, y(\tau))\, d\tau - \frac{1}{2}\left(f(t^n, y^n) + f(t^{n+1}, y(t^{n+1}))\right) + O(k^2) \\ &= O(k^2).\end{aligned}$$

In the last two line, we have used $y(t^{n+1}) = y(t^n) + ky'(t^n) + O(k^2)$.

3. **RK4** A 4th-order Runge-Kutta method uses Simpson's rule to approximate the integration:

$$\int_{t^n}^{t^{n+1}} f(t, y(t))\, dt = \frac{k}{6}\left(f(t^n, y(t^n)) + 4f(t^{n+1/2}, y(t^{n+1/2})) + f(t^{n+1}, y(t^{n+1}))\right) + O(k^5).$$

The quantity $y(t^{n+1/2})$ is approximated by the forward Euler method. It has the form

$$\begin{aligned} y^{n+1} &= y^n + \frac{k}{6}(\xi_1 + 2\xi_2 + 2\xi_3 + \xi_4) \\ \xi_1 &= f(t^n, y^n) \\ \xi_2 &= f(t^n + \frac{1}{2}k, y^n + \frac{k}{2}\xi_1) \\ \xi_3 &= f(t^n + \frac{1}{2}k, y^n + \frac{k}{2}\xi_2) \\ \xi_4 &= f(t^n + k, y^n + k\xi_3). \end{aligned}$$

The truncation error of RK4 is

$$\tau^n(k) := \frac{1}{k}\left(y(t^{n+1}) - y(t^n)\right) - \left(y^{n+1} - y(t^n)\right) = O(k^4).$$

Its proof is left for exercise.

1.3.2 General explicit Runge-Kutta methods

The method takes the following general form

$$\boxed{y^{n+1} = y^n + k\sum_{i=1}^{s} b_i\xi_i,} \tag{1.8}$$

where

$$\xi_1 \;=\; f(t^n, y^n),$$

$$
\begin{aligned}
\xi_2 &= f(t^n + c_2 k, y^n + k a_{21}\xi_1), \\
\xi_3 &= f(t^n + c_3 k, y^n + k a_{31}\xi_1 + k a_{32}\xi_2), \\
&\vdots \\
\xi_s &= f(t^n + c_s k, y^n + k(a_{s1}\xi_1 + \cdots + a_{s,s-1}\xi_{s-1})).
\end{aligned}
$$

We need to specify s (the number of stages), the coefficients $a_{ij}(1 \le j < i \le s)$, $b_i(i = 1, ..., s)$ and $c_i(i = 2, ..., s)$. We list them in the following Butcher table.

There are $s(s-1)/2 + s + (s-1)$ unknowns to be determined for a specific scheme. We

0					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
$\vdots$	$\vdots$		$\ddots$		
c_s	a_{s1}	a_{s2}	$\cdots$	$a_{s,s-1}$	
	b_1	b_2	$\cdots$	b_{s-1}	b_s

Table 1.1: Butcher's tableau for general Runge-Kutta methods. The weights $\sum b_i = 1$.

require the truncation error to be $O(k^{p+1})$. To find these coefficients, we need to expand the truncation error formula

$$y(t^{n+1}) - y^n = y^{n+1} - y^n + O(k^{p+1})$$

about (t^n, y^n) in terms of derivatives of $y(\cdot)$ at t^n. Then we can get p linear equations for the coefficients. You may see Runge-Kutta methods in wiki for the Butcher table. The table for RK4 is

0				
$1/2$	$1/2$			
$1/2$	0	$1/2$		
1	0	0	1	
	$1/6$	$1/3$	$1/3$	$1/6$

Table 1.2: Butcher's tableau for RK4.

1.3.3 Adaptive Runge-Kutta method (Runge-Kutta-Fehlberg method)

1. **The idea to get an approximate truncation error numerically** The adaptive Runge-Kutta method is designed to be able to estimate local truncation error in each time step. From which, we can adjust time step size to have roughly uniform truncation

error in each step. This is done by using two RK methods with the same sets of a_{ij} and c_i but different b_i, b_i^*. The set b_i produces RK method of order p. The auxiliary set b_i^* produces an RK method with order $p+1$. The corresponding solution $y^{n+1,*}$ is closer to the true solution. Thus, it can be used to estimate the local truncation for y^{n+1}:

$$y^{n+1} - y^{n+1,*} = k\sum_{i=1}^{s}(b_i - b_i^*)\xi_i = O(k^{p+1})$$

The step size k is then estimated so that the estimated truncation error is roughly the same in each time step.

2. **Runge-Kutta-Fehberg method: ode45** Below is the Butcher table for RK4 (b) and RK5 (b^*). Suppose ε is the "uniform" truncation error we prescribe. We define

0						
1/4	1/4					
3/8	3/32	9/32				
12/13	1932/2197	−7200/2197	7296/2197			
1	439/216	−8	3860/513	−845/4104		
1/2	−8/27	2	−3544/2565	1859/4104	−11/40	
b	25/216	0	1408/2565	2197/4104	−1/5	0
b^*	16/135	0	6656/12825	28561/56430	−9/50	2/55

$$TE = \left| k\sum_{i=1}^{6}(b_i - b_i^*)\xi_i \right|.$$

This TE is $|y^{n+1} - y^{n+1,*}|$ is an estimate of the truncation error. It is roughly the truncation error of the 4th order method, which is roughly

$$TE \sim Ck^5.$$

We use it to find a step size k so that the truncation error of the 4th order method is less than ε. Suppose $TE < \varepsilon$, then we use the present step size k and y^{n+1}, and then go to the next time step. If not, we look for k_{new} such that

$$\varepsilon \sim Ck_{new}^5.$$

Thus, we eliminating the constant C to get

$$\left(\frac{k_{new}}{k}\right)^5 \sim \frac{\varepsilon}{TE}.$$

Since TE is only an approximation of truncation error, we thus try

$$k_{new} = 0.9 \cdot k \cdot \left(\frac{\varepsilon}{TE}\right)^{1/5},$$

and repeat the above procedure until $TE < \varepsilon$. With the new step size k_{new}, the corresponding y^{n+1} has roughly the same truncation error ε. This method is an RK4 with roughly uniform truncation error. The corresponding matlab code is called the ode45. A natural question is why do we just use RK5 with a uniform step size? The answer is that the performance of such a uniform truncation error method is better than a higher-order method with uniform step size for stiff ODEs, because the magnitude the derivative $|f^{(p+1)}|$ can be very large in a stiff region.

1.3.4 *Convergence theory, an example

Let us see the proof of the convergence of the two stage Runge-Kutta method. The scheme can be expressed as

$$y^{n+1} = y^n + k\Psi(y^n, t^n, k) \tag{1.9}$$

where

$$\Psi(y^n, t^n, k) := f(y + \frac{1}{2}kf(y)). \tag{1.10}$$

Suppose $y(\cdot)$ is a true solution, the corresponding truncation error

$$\tau^n := \frac{y(t^{n+1}) - y(t^n)}{k} - \Psi(y(t^n), t^n, k) = O(k^2)$$

Thus, the true solution satisfies

$$y(t^{n+1}) - y(t^n) = k\Psi(y(t^n), t^n, k) + k\tau^n$$

The true error $e^n := y^n - y(t^n)$ satisfies

$$e^{n+1} = e^n + k(\Psi(y^n, t^n, k) - \Psi(y(t^n), t^n, k)) - k\tau^n.$$

This implies

$$|e^{n+1}| \le |e^n| + k\lambda'|e^n| + k|\tau^n|,$$

where λ' is the Lipschitz constant of $\Psi(y, t, k)$ with respect to y. Hence, we get

$$\begin{aligned} |e^n| &\le (1 + k\lambda')^n|e^0| + k\sum_{m=0}^{n-1}(1 + k\lambda')^{n-1-m}|\tau^m| \\ &\le e^{\lambda' t}|e^0| + \frac{e^{\lambda' t}}{\lambda'}\max_m |\tau^m|. \end{aligned}$$

We may assume $e^0 = 0$. The truncation error satisfies

$$\max_m |\tau^m| = O(k^2).$$

Thus, $|e^n| \to 0$ as $k \to 0$, with $nk = t$ fixed. □

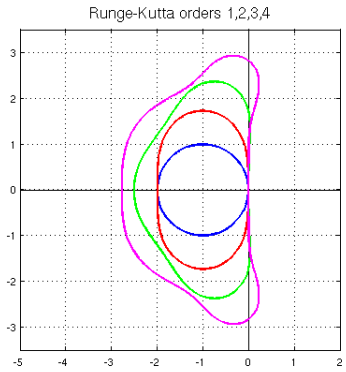


Figure 1.2: Absolutely stable regions of Runge-Kutta methods. The figure is quoted from a Stability Regions of ODE Formulas

1.3.5 Absolute stability region of RK methods

We apply RK2 to the equation $y' = ay$ to get

$$\begin{aligned} y^{n+1} &= y^n + \frac{k}{2}\left(f(y^n) + \xi_1\right) \\ &= y^n + \frac{k}{2}\left(ay^n + a\left(y^n + aky^n\right)\right) \\ &= \left(1 + ak + \frac{(ak)^2}{2}\right) y^n \end{aligned}$$

Let us write $z = ak$. The absolute stability region for RK2 is

$$\{z \in \mathbb{C} | \left|1 + z + \frac{z^2}{2}\right| \leq 1\}.$$

We refer a matlab code and plot for computing the absolute stability region for Runge-Kutta methods. Stability Regions of ODE Formulas

References:

- Lloyd N. Trefethen, Finite Difference and Spectral Methods for Ordinary and Partial Differential Equations,
- You may also google Runge-Kutta methods to get more references.

1.4 Multistep methods

1.4.1 Derivation of multistep methods

1. **Examples** The idea of multi-step methods is to derive a linear relation between, for

instance, y^{n+1}, y^n, y^{n-1}, y'^n and y'^{n-1} so that the corresponding truncation error is small. The simplest multistep method is the *midpoint method*.

- **Midpoint method** Suppose y^n and y^{n-1} is given. The new state y^{n+1} is defined by
$$y^{n+1} - y^{n-1} = 2ky'^n = 2kf(t^n, y^n).$$
The truncation error is
$$\tau^n := \frac{1}{k}\left(y(t^{n+1}) - y(t^{n-1}) - 2ky'(t^n)\right) = O(k^2).$$
Thus, the method is second order.
- **Adams-Bashforth 2-step method** We can also design a method which involves y^{n+1}, y^n, y^{n-1} and y'^n, y'^{n-1}. For instance,
$$y^{n+1} = y^n + \frac{k}{2}\left(3y'^n - y'^{n-1}\right) = y^n + \frac{k}{2}\left(3f(t^n, y^n) - f(t^{n-1}, y^{n-1})\right).$$
The truncation error τ^n satisfies
$$\begin{aligned} k\tau^n &:= y(t^{n+1}) - y(t^n) + \frac{k}{2}\left(3y'(t^n) - y'(t^{n-1})\right) \\ &= \left(y(t^n) + y'(t^n)k + \frac{y''(t^n)}{2}k^2\right) - y(t^n) + \frac{k}{2}\left(3y'(t^n) - (y'(t^n) - y''(t^n)k)\right) + O(k^3) \\ &= O(k^3). \end{aligned}$$
It can be derived by Taylor expansion of $y(\cdot)$ about t^n in the above formula.

2. **Derivation of general r-step methods** A general r-step multistep method involves $(y^{n+1}, y^n, ..., y^{n+1-r})$ and $(y'^{n+1}, y'^n, ..., y'^{n+1-r})$. It can be written as
$$\boxed{\sum_{m=0}^{r} a_m y^{n+1-r+m} = k\sum_{m=0}^{r} b_m y'^{n+1-r+m} = k\sum_{m=0}^{r} b_m f^{n+1-r+m}.} \tag{1.11}$$
We will normalize $a_r = 1$. Because it is the coefficient corresponding to y^{n+1}, which is what we want to find. When $b_r = 0$ the method is called *explicit*; otherwise it is *implicit*. For a smooth solution of (1.1), we define the truncation error τ^n to be
$$\tau^n := \frac{1}{k}\left(\sum_{m=0}^{r} a_m y(t^{n+1-r+m}) - k\sum_{m=0}^{r} b_m y'(t^{n+1-r+m})\right).$$

Definition 1.1. A multi-step method is called of order p if $\tau^n = O(k^p)$ uniformly in n. It is called consistent if $\tau^n(k) \to 0$ as $k \to 0$ uniformly in n.

Remark. When f is smooth, the solution of ODE $y' = f(t, y)$ is also smooth. Then the truncation is a smooth function of k. In this case, $\tau(k) \to 0$ is equivalent to $\tau(k) = O(k)$ as $k \to 0$.

3. **Determine the coefficients of multistep methods.** For notational convenience, let us extend a's and b's by setting $a_m = 0$, $b_m = 0$ for $m > r$. Taking Taylor expansion of $y(\cdot)$ about t^{n+1-r}, we get

$$\begin{aligned}
k\tau^n &= \sum_{m=0}^{r} a_m \sum_{j=0}^{\infty} \frac{1}{j!} y^{(j)} (mk)^j - k \sum_{m=0}^{r} b_m \sum_{j=1}^{\infty} \frac{1}{(j-1)!} y^{(j)} (mk)^{j-1} \\
&= \left(\sum_{m=0}^{r} a_m \right) y^{(0)} + \sum_{j=1}^{\infty} \frac{1}{j!} \sum_{m=0}^{r} \left(m^j a_m - j m^{j-1} b_m \right) k^j y^{(j)} \\
&= \left(\sum_{m=0}^{r} a_m \right) y^{(0)} + \sum_{j=1}^{\infty} \frac{1}{j!} \sum_{m=0}^{r} m^{j-1} \left(m a_m - j b_m \right) k^j y^{(j)} \\
&= \sum_{j=0}^{\infty} \frac{1}{j!} \sum_{m=0}^{r} m^{j-1} \left(m a_m - j b_m \right) k^j y^{(j)} \\
&= \sum_{j=0}^{\infty} C_j \frac{k^j}{j!} y^{(j)}.
\end{aligned}$$

Here, the derivatives of $y(\cdot)$ are evaluated at t^{n+1-r}. We list few equations for the coefficients a and b:

$$\begin{aligned}
C_0 &= a_0 + \cdots + a_r \\
C_1 &= (a_1 + 2a_2 + \cdots + r a_r) - (b_0 + \cdots + b_r) \\
C_2 &= (a_1 + 2^2 a_2 + \cdots + r^2 a_r) - 2(b_1 + \cdots + r b_r) \\
&\vdots \\
C_p &= \sum_{m=0}^{r} m^p a_m - \sum_{m=1}^{r} p m^{p-1} b_m.
\end{aligned}$$

To obtain a scheme of order p, we require

$$C_j = 0, \text{ for } j = 0, ..., p.$$

There are $2r + 1$ unknowns $(a_0, ..., a_r = 1, b_0, ..., b_r)$. In principle, we should choose $p = 2r + 1$ to have the same number of equations. Unfortunately, there are additional restrictions from the stability criterion for linear finite difference equations. The order of accuracy p should satisfy

$$p \leq \begin{cases} r + 2 \text{ if } r \text{ is even,} \\ r + 1 \text{ if } r \text{ is odd,} \\ r \text{ if it is an explicit scheme.} \end{cases}$$

This is called *the first Dahlquist stability barrier*. This stability criterion is related to the stability of linear finite difference equations. We will study them in the next section. You may also see Trefethen's book or Dahlquist's book.

4. **Initial setup** An r-step multi-step method needs $(y^0, y^1, ..., y^{r-1})^T$ to start. But only y^0 is given initially from ODE problems. We need to construct $y^1, ..., y^{r-1}$ by other methods. For instance, we can adopt RK methods. In order to maintain the order of accuracy, we should use a method of $p-1$ order. This will give initial error $y^i - y(t^i) = O(k^p)$ for $i = 0, ..., r-1$, which is consistent to the truncation error of the underlying multi-step method.

1.4.2 A formal algebra for multistep methods

1. Let us introduce the shift operator $Zy^n = y^{n+1}$, or in continuous sense, $Zy(t) = y(t+k)$. Let D be the differential operator $\frac{d}{dt}$. The Taylor expansion

$$y(t+k) = y(t) + ky'(t) + \frac{1}{2!}k^2D^2y(t) + \cdots$$

can be expressed formally as

$$Zy = \left(1 + (kD) + \frac{1}{2!}(kD)^2 + \cdots\right) y = e^{kD}y.$$

This means that the Taylor expansion gives the formula

$$\boxed{Z = e^{kD}.} \tag{1.12}$$

2. The multistep method can be expressed as

$$\mathcal{L}y := (a(Z) - kb(Z)D)\, y = \left(a(e^{kD}) - kDb(e^{kD})\right) y = (C_0 + C_1(kD) + \cdots)\, y.$$

Here,

$$a(Z) = \sum_{m=0}^{r} a_m Z^m, \quad b(Z) = \sum_{m=0}^{r} b_m Z^m$$

are the generating functions of $\{a_m\}$ and $\{b_m\}$. A multistep method is of order p means that

$$\left(a(e^{kD}) - kDb(kD)\right) y = O((kD)^{p+1})y.$$

We may abbreviate kD by a symbol κ. The above formula is equivalent to

$$a(e^{\kappa}) - \kappa b(e^{\kappa}) = O(\kappa^{p+1}).$$

Or equivalently,

$$\boxed{\frac{a(e^{\kappa})}{b(e^{\kappa})} = \kappa + O(\kappa^{p+1}) \ \text{ as } \kappa \to 0.} \tag{1.13}$$

3. We have the following theorem

Theorem 1.2. *A multistep method with $b(1) \neq 0$ is of order p if and only if*

$$\frac{a(z)}{b(z)} = \log z + O((z-1)^{p+1}) \text{ as } z \to 1. \tag{1.14}$$

It is consistent if and only if

$$a(1) = 0 \text{ and } a'(1) = b(1). \tag{1.15}$$

Proof. The first formula can be obtain from (1.13) by writing $e^{\kappa} = z$, and $\log(z) \sim z-1$. For the second formula, we expand $\log(z)$ about 1 to get

$$a(z) = b(z)\left((z-1) - \frac{(z-1)^2}{2} + \frac{(z-1)^3}{3} + \cdots\right) + O((z-1)^{p+1}).$$

We also expand $a(z)$ and $b(z)$ about $z = 1$, we can get

$$a(1) + (z-1)a'(1) = b(1)(z-1) + O((z-1)^2).$$

Note that $k \sim 0 \Leftrightarrow e^{kD} \sim 1 \Leftrightarrow z \sim 1$. The truncation error τ satisfies

$$(z-1)\tau = a(z) - b(z)\log(z)$$

for $z \sim 1$. Thus, the scheme is consistent if and only if $a(1) = 0$ and $a'(1) = b(1)$. □

Homeworks 1.3.

1. Consider the linear ODE $y' = \lambda y$, derive the finite difference equation using multistep method involving y^{n+1}, y^n, y^{n-1} and y'^n and y'^{n-1} for this linear ODE.

2. Solve the linear finite difference equations derived from the previous problem.

1.4.3 Examples of multistep methods

Let us see some concrete examples below.

- **Explicit Adams-Bashforth schemes** When $b_r = 0$, the method is explicit. Here are some examples of the explicit schemes called Adams-Bashforth schemes, where $a_r = 1$:
  - 1-step: $y^{n+1} = y^n + kf(y^n)$
  - 2-step: $y^{n+1} = y^n + \frac{k}{2}(3f(y^n) - f(y^{n-1}))$
  - 3 step: $y^{n+1} = y^n + \frac{k}{12}(23f(y^n) - 16f(y^{n-1}) + 5f(y^{n-2}))$

  The number of step is r and the order is $p = r$.

- **Implicit Adams-Moulton schemes** Another examples are the Adams-Moulton schemes, where $b_r \neq 0$ and the number of step is r and the order is $p = r + 1$.
  - 1-step: $y^{n+1} = y^n + \frac{k}{2}(f(y^{n+1}) + f(y^n))$
  - 2-step: $y^{n+1} = y^n + \frac{k}{12}(5f(y^{n+1}) + 8f(y^n) - f(y^{n-1}))$
  - 3 step: $y^{n+1} = y^n + \frac{k}{24}(9f(y^{n+1}) + 19f(y^n) - 5f(y^{n-1}) + f(y^{n-2}))$.
- **Explicit Adams-Bashforth-Moulton schemes** Sometimes, we can use an explicit scheme to guess y^{n+1} as a predictor in an implicit scheme. Such a method is called a *predictor-corrector method*. A standard one is the following Adams-Bashforth-Moulton schemes: Its predictor part is the Adams-Bashforth scheme:
$$\hat{y}^{n+1} = y^n + \frac{k}{12}(23f(y^n) - 16f(y^{n-1}) + 5f(y^{n-2}))$$
The corrector is the Adams-Moulton scheme:
$$y^{n+1} = y^n + \frac{k}{24}(9f(\hat{y}^{n+1}) + 19f(y^n) - 5f(y^{n-1}) + f(y^{n-2}))$$
The predictor-corrector is still an explicit scheme. However, for stiff problems, we should use implicit scheme instead.
- **Backward differentiation formula** The backward differentiation formula (BDF) has $b_0 = 1, b_1 = \cdots = b_r = 0$.
  - BDF1: $y^{n+1} - y^n = kf(t^{n+1}, y^{n+1})$
  - BDF2: $y^{n+1} - \frac{4}{3}y^n + \frac{1}{3}y^{n-1} = kf(t^{n+1}, y^{n+1})$
  - BDF3: $y^{n+1} - \frac{8}{11}y^n + \frac{9}{11}y^{n-1} - \frac{2}{11}y^{n-2} = \frac{6}{11}kf(t^{n+1}, y^{n+1})$
  - BDF4: $y^{n+1} - \frac{48}{25}y^n + \frac{36}{25}y^{n-1} - \frac{16}{25}y^{n-2} + \frac{3}{25}y^{n-3} = \frac{12}{25}kf(t^{n+1}, y^{n+1})$
  - BDF5: $y^{n+1} - \frac{300}{137}y^n + \frac{300}{137}y^{n-1} - \frac{200}{137}y^{n-2} + \frac{75}{137}y^{n-3} - \frac{12}{137}y^{n-4} = \frac{60}{137}kf(t^{n+1}, y^{n+1})$

  For $r \geq 7$, there is no stable BDF methods. The r-step BDF has order r, which can be obtained by taking Taylor expansion of each terms in the formula about t^{n+1}. The BDFs allow larger step size. It is particularly useful for stiff ODEs. The statement is below quoted from Chatgpt: "ode15s is actually a variable-order, variable-step-size solver that uses a combination of BDF methods of orders 1 through 5, along with other techniques such as extrapolation, interpolation, and local error control. The solver automatically adjusts the order and step size of the method to maintain a specified level of accuracy while minimizing computational cost." A reference is "Shampine, L. F., Reichelt, M. W. (1997). The MATLAB ODE Suite. SIAM Journal on Scientific Computing, 18(1), 1-22."
- **Matlab ODE solvers are available.** Matlab codes are available on Wikiversity with key words "Adams-Bashforth and Adams-Moulton methods." There are many matlab codes for non-stiff and stiff ODEs. You can consult with the website Matlab function reference: ode45, ode23, ode113, ode15s, ode23s, ode23t, ode23tb to choose a proper solver for your problem. The 's' in ODE15s stands for 'stiff'.

1.5 Linear difference equations

1.5.1 Second-order linear difference equation.

1. In linear ODE $y' = ay$, the above difference schemes result in a linear difference equations. For instance, the forward Euler method gives the first order difference equation: $y^{n+1} = (1 + ka)y^n$. A two-step method gives a second-order finite difference equation. For instance, the mid-point method is $y^{n+1} - y^{n-1} = 2kf(t^n, y^n)$. Apply this method to the equation $y' = ay$. We get

$$y^{n+1} - y^{n-1} = 2kay^n. \tag{1.16}$$

2. Below, let us consider the following second order linear difference equation with constant coefficients:

$$ay^{n+1} + by^n + cy^{n-1} = 0, \tag{1.17}$$

where $a \neq 0$. To find its general solutions, we try the ansatz $y^n = \rho^n$ for some number ρ. Here, the n in y^n is an index, whereas the n in ρ^n is a power. Plug this ansatz into the equation, we get

$$a\rho^{n+1} + b\rho^n + c\rho^{n-1} = 0.$$

This leads to

$$a\rho^2 + b\rho + c = 0. \tag{1.18}$$

This is called the characteristic equation for the difference equation (1.17). There are two solutions ρ_1 and ρ_2.

- Case 1: $\rho_1 \neq \rho_2$. The two solutions $\{\rho_1^n, \rho_2^n\}$ are independent. Since the equation is linear, any linear combination of these two solutions is again a solution. Moreover, the general solution can only depend on two free parameters, namely, once y^0 and y^{-1} are known, then $\{y^n\}_{n\in\mathbb{Z}}$ is uniquely determined. Thus, the general solution for equation (1.17) is

$$y^n = C_1\rho_1^n + C_2\rho_2^n,$$

where C_1, C_2 are constants.

- Case 2: $\rho_1 = \rho_2$. First ρ^n is a solution. Next, we can use perturbation method to find another set of solution. Consider a small perturbation of (1.17). The perturbation breaks the double roots $\rho_2 = \rho_1$ to two simple roots $\rho_2 \neq \rho_1$, but with $\rho_2 \sim \rho_1$. We can use the two solutions ρ_2^n and ρ_1^n with $\rho_2 \to \rho_1$ to produce another nontrivial solution:

$$\lim_{\rho_2\to\rho_1} \frac{\rho_2^n - \rho_1^n}{\rho_2 - \rho_1}.$$

This yields a second independent solution $n\rho_1^{n-1}$. Thus, the general solution is

$$C_1\rho_1^n + C_2 n\rho_1^{n-1}.$$

3. Let us analyze the solutions obtained from the midpoint method. The midpoint method for the equation $y' = ay$ gives the finite difference equation

$$y^{n+1} - y^{n-1} = 2aky^n.$$

Let us write $ak = z$. The two roots for the characteristic equation $\rho^2 - 1 - 2z\rho = 0$ are

$$\rho_1 = z + \sqrt{1+z^2}, \quad \rho_1 = z - \sqrt{1+z^2}.$$

The step size k is small. So as $z = ak$ is small, we approximate $\sqrt{1+z^2} \sim 1 + \frac{z^2}{2}$. Thus, the two roots are approximately

$$\rho_1 \sim 1 + z + \frac{z^2}{2}, \quad \rho_2 \sim -1 + z - \frac{z^2}{2}.$$

The solutions to the difference equation are

$$C_1\rho_1^n + C_2\rho_2^n.$$

We choose $kn = t$ fixed, $k \to 0$, we get

$$\rho_1^n \sim \left(1 + ka + \frac{(ka)^2}{2}\right)^n \sim e^{at}.$$

On the other hand,

$$\rho_2^n \sim \left(-1 + ka - \frac{(ka)^2}{2}\right)^n \sim (-1)^n \left(1 - ka + \frac{(ka)^2}{2}\right)^n \sim (-1)^n e^{-at}.$$

We have the following interpretation.

- Case 1: $a > 0$. ρ_1^n gives us the principal mode which converges to the true solution e^{at}. While ρ_2^n gives us an *exponential decay mode. It is oscillatory!*
- Case 2: $a < 0$. The ρ_2 gives an unstable mode.
- Case 3: $a = i\omega$ is pure imaginary. Both ρ_1^n and ρ_2^n are oscillatory, which is correct. The ρ_2^n is an extra mode. But its coefficient is small.
- Note that the absolute stability region is $a = i\omega$ and $-1 \le \omega \le 1$.

1.5.2 Linear finite difference equations of order r

We consider general linear finite difference equation of order r:

$$a_r y^{n+r} + \cdots + a_0 y^n = 0, \tag{1.19}$$

where $a_r \neq 0$. Since y^{n+r} can be solved in terms of $y^{n+r-1}, ..., y^n$ for all n, this equation together with initial data $y_0, ..., y_{-r+1}$ has a unique solution. The solution space is r dimensions.
To find fundamental solutions, we try the ansatz

$$y^n = \rho^n$$

for some number ρ. Plug this ansatz into equation, we get

$$a_r \rho^{n+r} + \cdots + a_0 \rho^n = 0,$$

for all n. This implies

$$a(\rho) := a_r \rho^r + \cdots + a_0 = 0. \tag{1.20}$$

The polynomial $a(\rho)$ is called the characteristic polynomial of (1.19) and its roots $\rho_1, ..., \rho_r$ are called the characteristic roots.

- Simple roots (i.e. $\rho_i \neq \rho_j$, for all $i \neq j$): The fundamental solutions are ρ_i^n, $i = 1, ..., r$.
- Multiple roots: if ρ_i is a multiple root with multiplicity m_i, then the corresponding independent solutions

$$\rho_i^n, n\rho_i^{n-1}, C_2^n \rho_i^{n-2} ..., C_{m_i-1}^n \rho_i^{n-m_i+1}.$$

Here, $C_k^n := n!/k!(n-k)!$. The solution $C_2^n \rho_i^{n-2}$ can be derived from differentiation $\frac{d}{d\rho} C_1^n \rho^{n-1}$ at ρ_i.

1.5.3 *System of linear difference equations

The above rth order linear difference equation is equivalent to a first order linear difference system:

$$\mathbf{A}_0 \mathbf{y}^{n+1} = \mathbf{A}\mathbf{y}^n, \tag{1.21}$$

where

$$\mathbf{y}^n = \begin{bmatrix} y_1^n \\ \vdots \\ y_r^n \end{bmatrix} = \begin{bmatrix} y^{n-r+1} \\ \vdots \\ y^n \end{bmatrix}$$

$$\mathbf{A}_0 = \begin{bmatrix} I_{(r-1)\times(r-1)} & 0 \\ 0 & a_r \end{bmatrix}, \ \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{r-1} \end{bmatrix}.$$

We may divide (1.21) by $\mathbf{A}_0$ and get

$$\mathbf{y}^{n+1} = \mathbf{G}\mathbf{y}^n.$$

We call $\mathbf{G}$ the fundamental matrix of (1.21). For this homogeneous equation, the solution is

$$\mathbf{y}^n = \mathbf{G}^n \mathbf{y}^0$$

Next, we compute $\mathbf{G}^n$ in terms of eigenvalues of $\mathbf{G}$.

In the case that all eigenvalues ρ_i, $i = 1, ..., r$ of $\mathbf{G}$ are distinct, then $\mathbf{G}$ can be expressed as

$$\mathbf{G} = \mathbf{T}\mathbf{D}\mathbf{T}^{-1}, \ \mathbf{D} = \ \text{diag}\ (\rho_1, \cdots, \rho_r),$$

and the column vectors of $\mathbf{T}$ are the corresponding eigenvectors.

When the eigenvalues of $\mathbf{G}$ have multiple roots, we can normalize it into Jordan blocks:

$$\mathbf{G} = \mathbf{T}\mathbf{J}\mathbf{T}^{-1}, \ \mathbf{J} = \ \text{diag}\ (\mathbf{J}_1, \cdots, \mathbf{J}_s),$$

where the Jordan block $\mathbf{J}_i$ corresponds to eigenvalue ρ_i with multiplicity m_i:

$$\mathbf{J}_i = \begin{bmatrix} \rho_i & 1 & 0 & \cdots & 0 \\ 0 & \rho_i & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & \rho_i \end{bmatrix}_{m_i \times m_i} .$$

and $\sum_{i=1}^{s} m_i = r$. Indeed, this form also covers the case of distinct eigenvalues.

In the stability analysis below, we are concerned with whether $\mathbf{G}^n$ is bounnded. It is easy to see that

$$\mathbf{G}^n = \mathbf{T}\mathbf{J}^n\mathbf{T}^{-1}, \mathbf{J}^n = \ \text{diag}\ (\mathbf{J}_1^n, \cdots, \mathbf{J}_s^n)$$

$$\mathbf{J}_i^n = \begin{bmatrix} \rho_i^n & n\rho_i^{n-1} & C_2^n \rho^{n-2} & \cdots & C_{m_i-1}^n \rho_i^{n-m_i+1} \\ 0 & \rho_i^n & n\rho_i^{n-1} & \cdots & C_{m_i-2}^n \rho_i^{n-m_i+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & n\rho_i^{n-1} \\ 0 & 0 & 0 & \cdots & \rho_i^n \end{bmatrix}_{m_i \times m_i} .$$

where $C_k^n := \frac{n!}{k!(n-k)!}$.

1.5.4 Stability of linear difference equations

Definition 1.2. The fundamental matrix $\mathbf{G}$ is called *stable* if $\mathbf{G}^n$ remains bounded under certain norm $\|\cdot\|$ for all n.

Theorem 1.3 (von Neumann)**.** *The fundamental matrix* $\mathbf{G}$ *is stable if and only if its eigenvalues satisfy the following condition:*

$$\begin{array}{l} \textit{either } |\rho| = 1 \textit{ and } \rho \textit{ is a simple root,} \\ \textit{or} \quad |\rho| < 1 \ . \end{array} \tag{1.22}$$

Proof. It is easy to see that the nth power of a Jordan form J_i^n is bounded if its eigenvalue $|\rho_i| < 1$ or if $|\rho_i| = 1$ but simple. On the other hand, if $|\rho_i| > 1$ then J_i^n is unbounded; or if $\rho_i| = 1$ but not simple, then the term $n\rho_i^{n-1}$ in J_i^n will be unbounded. □

Corollary 1.1. *There exists a norm in $\mathbb{R}^n$ such that the above root condition for $\mathbf{G}$ is equivalent to $\|\mathbf{G}\| \leq 1$ with this norm.*

Proof. 1. First, in $\mathbb{R}^n$ (or $\mathbb{C}^n$), we define $\|\mathbf{x}\|_\infty = \max_i |x_i|$. For a linear mapping $\mathbf{G} : \mathbb{R}^n \to \mathbb{R}^n$, we define its operator norm under the $\|\cdot\|_\infty$ by

$$\|\mathbf{G}\|_\infty := \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{G}\mathbf{x}\|_\infty}{\|\mathbf{x}\|_\infty}.$$

It is an easy exercise that for $\mathbf{G} = (a_{ij})_{n\times n}$, the operator norm

$$\|\mathbf{G}\|_\infty = \max_i \sum_j |a_{ij}|.$$

2. Second, a matrix $\mathbf{G}$ can be expressed as

$$\mathbf{G} = \mathbf{T}\mathbf{D}\mathbf{T}^{-1}, \quad \mathbf{D} = \text{ diag } (\mathbf{J}_1, \cdots, \mathbf{J}_s)$$

where $\mathbf{J}_i$ are Jordan blocks. For any $\varepsilon_i \neq 0$, we can further transform $\mathbf{J}_i$ into

$$\mathbf{J}_i = \boldsymbol{S}_i \boldsymbol{K}_i \boldsymbol{S}_i^{-1}$$

where

$$\boldsymbol{K_i} = \begin{bmatrix} \rho_i & \varepsilon & 0 & \cdots & 0 \\ 0 & \rho_i & \varepsilon & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \varepsilon \\ 0 & 0 & 0 & \cdots & \rho_i \end{bmatrix}_{m_i\times m_i}, \quad S_i = \text{ diag } (1, \varepsilon_i, ..., \varepsilon_i^{m_i-1}).$$

Let $\boldsymbol{S} = \text{ diag}(\boldsymbol{S}_1, ..., \boldsymbol{S}_s)$, $\boldsymbol{K} = \text{ diag}(\boldsymbol{K}_1, ..., \boldsymbol{K}_s)$. We can express $\mathbf{G}$ as

$$\mathbf{G} = \mathbf{T}\mathbf{S}\,\mathbf{K}(\mathbf{T}\mathbf{S})^{-1}$$

We now define the new norm of $\mathbf{G}$ as

$$\|\mathbf{G}\| := \|\mathbf{K}\|_\infty$$

This means that we define the new norm $\|\cdot\|$ in $\mathbb{R}^n$ by

$$\|\mathbf{x}\| := \|(\mathbf{T}\mathbf{S})^{-1}\mathbf{x}\|_\infty.$$

Since $\mathbf{T}\mathbf{S}$ is invertible, this does define a norm in $\mathbb{R}^n$. With this norm, the corresponding operator norm is $\|\mathbf{K}\|_\infty$.

3. For those $\mathbf{J}_i$ with $m_i > 1$, the stability condition requires that $|\rho_i| < 1$. We choose ε_i such that $|\rho_i| + \varepsilon_i \leq 1$. Then the corresponding $\|K_i\|_\infty \leq 1$. Thus, $\|\mathbf{G}\| \leq 1$ with the above operator norm.

□

1.5.5 *Non-homogeneous linear finite difference equations

In general, we consider the nonhomogeneous linear difference system:

$$\mathbf{y}^{n+1} = \mathbf{G}\mathbf{y}^n + \mathbf{f}^n \tag{1.23}$$

with initial data $\mathbf{y}^0$. Its solution can be expressed as

$$\begin{aligned} \mathbf{y}^n &= \mathbf{G}\mathbf{y}^{n-1} + \mathbf{f}^{n-1} \\ &= \mathbf{G}(\mathbf{G}\mathbf{y}^{n-2} + \mathbf{f}^{n-2}) + \mathbf{f}^{n-1} \\ &\vdots \\ &= \mathbf{G}^n\mathbf{y}^0 + \sum_{m=0}^{n-1} \mathbf{G}^{n-1-m}\mathbf{f}^m \end{aligned}$$

Homeworks 1.4. 1. Consider the linear ODE

$$y' = \lambda y$$

where λ considered here can be complex. Study the linear difference equation derived for this ODE by forward Euler method, backward Euler, midpoint. Find its general solutions.

2. Consider linear finite difference equation with source term

$$ay^{n+1} + by^n + cy^{n-1} = f^n$$

Given initial data $\bar{y}^0$ and $\bar{y}^1$, find its solution.

3. Find the characteristic roots for the Adams-Bashforth and Adams-Moulton schemes with steps 1-3 for the linear equation $y' = \lambda y$.

1.6 *Stability and convergence for multistep methods

There are two kinds of stability concepts in numerical ODE:

- Zero stability: Fix $t = nk$, the computed solution y^n remains bounded as $n \to \infty$ (or equivalently, $k \to 0$). The concept of zero stability is for convergence theory.

- Absolute stability: Fix $k > 0$, the computed solution y^n remains bounded as $n \to \infty$. The concept of absolute stability is for reproducing stable equilibria in numerical computations.

1.6.1 Zero Stability and Dahlquist equivalence theorem

Our goal is to develop a general convergence theory for multistep finite difference method for the ODE: $y' = f(t, y)$ with initial condition $y(0) = y_0$. An r-step multistep finite difference scheme can be expressed as

$$\mathcal{L}y^n = \sum_{m=0}^{r} a_m y^{n+1-r+m} - k \sum_{m=0}^{r} b_m f(t^{n+1-r+m}, y^{n+1-r+m}) = 0. \tag{1.24}$$

Definition 1.3. The truncation error $\tau^n(k)$ for the above multistep scheme is defined by

$$\tau^n(k) := \frac{1}{k} \left(\sum_{m=0}^{r} a_m y(t^{n+1-r+m}) - k \sum_{m=0}^{r} b_m f(t^{n+1-r+m}, y(t^{n+1-r+m})) \right),$$

where $y(\cdot)$ is a true solution of the ODE.

Definition 1.4. A difference scheme is called consistent if the corresponding truncation error $\tau^n(k) \to 0$ uniformly in n as the mesh size $k \to 0$. The scheme is of order p if $\tau^n(k) = O(k^p)$ uniform in n.

In multistep methods, the consistency is equivalent to $\tau(k) = O(k)$ because we assume $y(\cdot)$ is smooth and the truncation error is a smooth function in k. The consistency is $\tau(k) \to 0$ as $k \to 0$. Thus the smoothness of τ implies $\tau(k) = O(k)$.

Definition 1.5. A difference scheme is called zero stable if its solutions $\mathbf{y}^n$ at time step n remain bounded as the mesh size $k \to 0$ with $nk = t$ is fixed (accordingly $n \to \infty$).

The main theorem is the follows. We will postpone its proof at the end of this section.

Theorem 1.4 (Dahlquist equivalence theorem)**.** *For finite difference schemes for the ODE $y' = f(t, y)$, we have*

$$(\ \textit{consistency}\ +\ \textit{zero stability}\ ) \iff \textit{convergence}\ .$$

Stability criterion Let us investigate the condition on the coefficients a's and b's of an explicit multistep method for the stability

$$\mathcal{L}y^n = 0$$

to be bounded. We may assume $a_r = 1$ and $b_r = 0$. Let us write it in matrix form:

$$\mathbf{y}^{n+1} = \mathbf{A}\mathbf{y}^n + k\mathbf{B}\mathbf{f}^n$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & & & & \\ & 0 & 1 & & & \\ & & & \ddots & \ddots & \\ & & & & 0 & 1 \\ -a_0 & & \cdots & & -a_{r-2} & -a_{r-1} \end{bmatrix}, \quad \mathbf{y}^n = \begin{bmatrix} y^{n-r} \\ \cdots \\ y^n \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & & & & \\ & 0 & 0 & & & \\ & & & \ddots & \ddots & \\ & & & & 0 & 0 \\ b_0 & & \cdots & & b_{r-2} & b_{r-1} \end{bmatrix}, \quad \mathbf{f}^n = \begin{bmatrix} f^{n-r} \\ \cdots \\ f^n \end{bmatrix}.$$

In order to have solution to be bounded for a multistep scheme $\mathcal{L}y = 0$ for arbitrary f, it has at least to be valid when $f \equiv 0$. In this case, we need to invetigate the boundedness for the homogeneous equation:

$$\mathbf{y}^{n+1} = \mathbf{A}\mathbf{y}^n$$

We have seen in the last section that

Theorem 1.5. *The necessary and sufficient condition for $\|\mathbf{A}^n\|$ to be bounded is that the characteristic roots ρ_i of the characteristic equation $a(z) = 0$ satisfies:*

$$\begin{gathered} \textit{either } |\rho_i| < 1 \\ \textit{or } |\rho_i| = 1 \textit{ but simple.} \end{gathered}$$

Convergence ⇒ Stability

Proof. We only need to find an f such that the corresponding multistep is not stable implies that it does not converge. We choose $f \equiv 0$. ¶ Since $\mathbf{A}^n$ is unbounded, which means there is an eigenvalue ρ_i with eigenvector $\mathbf{y}^i$ such that $|\rho_i| > 1$ or $|\rho_i| = 1$ but not simple. We discuss the formal case. The latter case can also be prove easily. In the former case, let $\mathbf{y}_i$ be the eigenvector of $\mathbf{A}$ corresponding to the eigenvalue ρ_i which satisfies $|\rho_i| > 1$. Let us choose y^0 and generate $\mathbf{y}^0 = (y_0^{r-1}, \cdots, y_0)^T$ by some explicit scheme starting from y^0. We can choose $\mathbf{y}^0$ such that its component on $\mathbf{y}_i$ is nonzero. Then the corresponding $\mathbf{y}^n := \mathbf{A}^n\mathbf{y}^0$ will be unbounded. Hence it cannot converge to a constant, as $k \to 0$. On the other hand, $\mathbf{y}^0$ depends on the mesh size k and $\mathbf{y}^0(k) \to (y_0, \cdots, y_0)^T$ as $k \to 0$. Thus, the method does not converge for $f \equiv 0$. □

Convergence ⇒ Consistency

Proof. From Theorem 1.2, we need to show that $a(1) = 0$ and $a'(1) = b(1)$. To show the first, we consider the ODE: $y' = 0$ with $y(0) = 1$. For the second, we consider the ODE: $y' = 1$ and $y(0) = 0$.

- Show $a(1) = 0$: We choose $\mathbf{y}^0 = (1, \cdots, 1)^T$. From $\mathbf{y}^1 = A\mathbf{y}^0$, we get

$$y^r = -a_0 y^0 - \cdots - a_{r-1} y^{r-1} = -a_0 - \cdots - a_{r-1}.$$

¶Suppose a multistep method is convergence for every smooth f, then in particular, for $f \equiv 0$. In this case, if this multistep method is unstable, we want to show it does not converge. This is a contradiction.

Since y^r is independent of k, and we should have $y^r \to 1$ as $k \to 0$ (by convergence), we conclude that $y^r = 1$. Thus, we get $a(1) = a_0 + \cdots + a_{r-1} + 1 = 0$.

- Show $a'(1) = b(1)$. We choose $f \equiv 1$, $y(0) = 0$. The corresponding ODE solution is $y(t) = t$. The multistep method gives

$$a(Z)y^n - kb(Z)1 = 0. \tag{1.25}$$

We write

$$a(Z) = a'(1)(Z-1) + O((Z-1)^2), \quad b(Z)1 = b(1).$$

Then the principal part of the above finite difference is

$$(Z-1)y - k\frac{b(1)}{a'(1)} = 0.$$

This is an arithmetic series. Its solution is $y^n = nk\frac{b(1)}{a'(1)}$. Indeed, this sequence also satisfies (1.25) provided its initial data y^n also has the form $y^n = nk\frac{b(1)}{a'(1)}$ for $0 \le n < r$. Thus, arithmetic series $y^n = nk\frac{b(1)}{a'(1)}$ is a solution of the difference equation (1.25). Since $nk = t$, the convergence $y^n \to t$ as $n \to \infty$ enforces $\frac{b(1)}{a'(1)} = 1$.

□

Stability + Consistency ⇒ Convergence

Proof. We recall that the multistep scheme can be expressed as

$$\mathbf{y}^{n+1} = \mathbf{A}\mathbf{y}^n + k\mathbf{B}\mathbf{f}^n.$$

Let Y be an exact solution. We plug it into the above scheme to get the truncation error:

$$\mathbf{Y}^{n+1} = \mathbf{A}\mathbf{Y}^n + k\mathbf{B}\mathbf{F}^n + k\boldsymbol{\tau}^n,$$

where $\mathbf{Y}^n := (Y(t^{n-r}), ...Y(t^n))^T$. We subtract these two equations and call $\mathbf{e}^n := \mathbf{Y}^n - \mathbf{y}^n$ the true error. We get

$$\mathbf{e}^{n+1} = \mathbf{A}\mathbf{e}^n + k\mathbf{B}\left(\mathbf{F}^n - \mathbf{f}^n\right) + k\boldsymbol{\tau}^n.$$

The term $\mathbf{F}^n - \mathbf{f}^n$ can be expressed as

$$\begin{aligned} \mathbf{F}^n - \mathbf{f}^n &= (f(Y^{n-r}) - f(y^{n-r}), \cdots, f(Y^n) - f(y^n))^T \\ &= (L_{-r}e^{n-r}, \cdots, L_0 e^n)^T \\ &:= \mathbf{L}_n \mathbf{e}^n \end{aligned}$$

where

$$L_{-m} := \int_0^1 f'(y^{n-m} + te^{n-m})\, dt.$$

Thus, we get

$$\begin{aligned}\mathbf{e}^{n+1} &= (\mathbf{A} + k\mathbf{B}\mathbf{L}_n)\,\mathbf{e}^n + k\boldsymbol{\tau}^n \\ &:= \mathbf{G}_n(k)\mathbf{e}^n + k\boldsymbol{\tau}^n, \\ \mathbf{G}_n(k) &:= \mathbf{A} + k\mathbf{B}\mathbf{L}_n.\end{aligned}$$

Thus, we have

$$\begin{aligned}\mathbf{e}^n &\le \mathbf{G}_{n-1}\mathbf{e}^{n-1} + k\boldsymbol{\tau}^{n-1} \\ &\le \mathbf{G}_{n-1}\mathbf{G}_{n-2}\mathbf{e}^{n-2} + k\left(\mathbf{G}_{n-2}\boldsymbol{\tau}^{n-2} + \boldsymbol{\tau}^{n-1}\right) \\ &\le \mathbf{G}_{n-1}\mathbf{G}_{n-2}\cdots\mathbf{G}_0\mathbf{e}^0 \\ &\quad + k\left(\mathbf{G}_{n-2}\cdots\mathbf{G}_0\boldsymbol{\tau}^0 + \cdots + \mathbf{G}_{n-2}\boldsymbol{\tau}^{n-2} + \boldsymbol{\tau}^{n-1}\right).\end{aligned}$$

We note that the term $\|\mathbf{B}\mathbf{L}_k\|$ is uniformly bounded, because f is Lipschitz, thus the functions L_{-m} above are uniformly bounded (independent of n). We have the following lemma.

Lemma 1.1. *If $\|\mathbf{A}^n\|$ is bounded and $\|\mathbf{B}_n\|$ are uniformly bounded, then the product*

$$\left\|(\mathbf{A} + \frac{1}{n}\mathbf{B}_1)\cdots(\mathbf{A} + \frac{1}{n}\mathbf{B}_n)\right\|$$

is also uniformly bounded.

(continue the proof of the Theorem). From this lemma, we get

$$\|\mathbf{e}^n\| \le C\|\mathbf{e}^0\| + nkC\max_n\|\boldsymbol{\tau}^n\| \le C\|\mathbf{e}^0\| + O(k^p).$$

with C independent of n and k. Thus, $\|\mathbf{e}^n\| \to 0$ as $k \to 0$ with $nk = t$ fixed. □

Proof of Lemma 1.1

Proof. 1. We have seen that $\|\mathbf{A}^n\|$ is uniformly bounded under some norm is equivalent to $\|\mathbf{A}\| \le 1$ for some other operator norm. Thus, we may just assume $\|\mathbf{A}\| \le 1$.

2. Since all norms in finite dimension are equivalent, we may assume $\|\mathbf{B}_i\| \le b$ for all $i = 1, ..., n$.

3. We have

$$\left\|(\mathbf{A} + \frac{1}{n}\mathbf{B}_1)\cdots(\mathbf{A} + \frac{1}{n}\mathbf{B}_n)\right\| \le (\|\mathbf{A}\| + \frac{b}{n})^n \le (1 + \frac{b}{n})^n \le \exp(b).$$

□

Theorem 1.6 (First Dahlquist barrier)**.** *A zero-stable and linear r-step multistep method with p order of convergence should satisfies*

$$p \le \begin{cases} r+2 \text{ if } r \text{ is even,} \\ r+1 \text{ if } r \text{ is odd,} \\ r \text{ if it is an explicit scheme.} \end{cases}$$

For proof, see pp. 384-387, Hairer, Norsett, Wanner, Solving Ordinary Differential Equations.

1.6.2 Absolute stability regions of multistep methods

1. **Absolute stability** The concept of absolute stability is nothing to do with the concept of zero stability. In ODE theory, we encounter stable equilibria, which are those states $\bar{\mathbf{y}}$ which are stable under small perturbation. It means that the solutions stay in a small neighborhood of $\bar{\mathbf{y}}$ when their initial states do so. An equilibrium $\bar{\mathbf{y}}$ is called asymptotically stable if $\mathbf{y}(t) \to \bar{\mathbf{y}}$ with $\mathbf{y}(0) \sim \bar{\mathbf{y}}$. In this case, the linearized matrix $\partial \mathbf{f}/\partial \mathbf{y}\,(\bar{\mathbf{y}})$ has only negative eigenvalues $a < 0$. In numerical computation, we would like to reproduce such asymptotically stable solution. There is some restriction for k to achieve this goal. This is particularly important when some of the negative eigenvalues are large in magnitude. Such system is called stiff ODE. To derive such restriction, it is enough to consider the scalar equation

$$y' = ay, \quad a < 0.$$

Let us call $z = ak$. We have seen how we derive a region for z in which the discrete system can reproduce the stable process as that in the continuous case.

2. **Absolute stability region for the Midpoint method** The midpoint method for the equation $y' = ay$ reads

$$\begin{bmatrix} y^n \\ y^{n+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 2z \end{bmatrix} \begin{bmatrix} y^{n-1} \\ y^n \end{bmatrix}$$

The characteristic equation is

$$\lambda^2 - 2z\lambda - 1 = 0.$$

The two eigenvalues λ_k, $k = 1, 2$ satisfy

$$\lambda_1 \lambda_2 = -1, \quad \lambda_1 + \lambda_2 = 2z.$$

From the first equation, we have

$$|\lambda_1||\lambda_2| = 1.$$

From the stability restriction is

$$|\lambda_k| \le 1, k = 1, 2.$$

This gives

$$|\lambda_k| = 1 \text{ for } k = 1, 2.$$

Let us call

$$\lambda_k = e^{i\theta_k}, k = 1, 2.$$

From $\lambda_1 \lambda_2 = -1$, we get $\theta_1 + \theta_2 = \pi$. From $\lambda_1 + \lambda_2 = 2z$, we get

$$\cos\theta_1 + \cos\theta_2 + i(\sin\theta_1 + \sin\theta_2) = i2\sin\theta_1 = 2z.$$

This implies

$$z = iy, \quad -1 \leq y \leq 1.$$

Thus, the absolute stability region for the midpoint method is

$$\{z \in \mathbb{C} | z = iy, -1 \leq y \leq 1\}.$$

3. For absolute stability region for multistep methods and their plots on the complex plane by using matlab, we refer to Stability Regions of ODE Formulas

4. For more examples, we list some references for readers.

   - For absolute stability, see Chapter 7 of Randall LeVeque's book: Finite Difference Methods for Ordinary and Partial Differential Equations.
   - Stiffness and A-stability
   - Absolute Stability Region
   - A matlab code to plot the absolute stability region for Adams-Bashforth methods.

Homeworks 1.5. 1. Find the absolute stability regions of the explicit Adams-Bashforth schemes and the implicit Adams-Moulton schemes, respectively.

Comments

1. There are structure-preserving ODE solver, geometric integrator, symplectic algorithms for Hamiltonian flows.

2. For differential-algebraic equations (DAE), there are DAE packages for Electronic Design Automation (EDA).

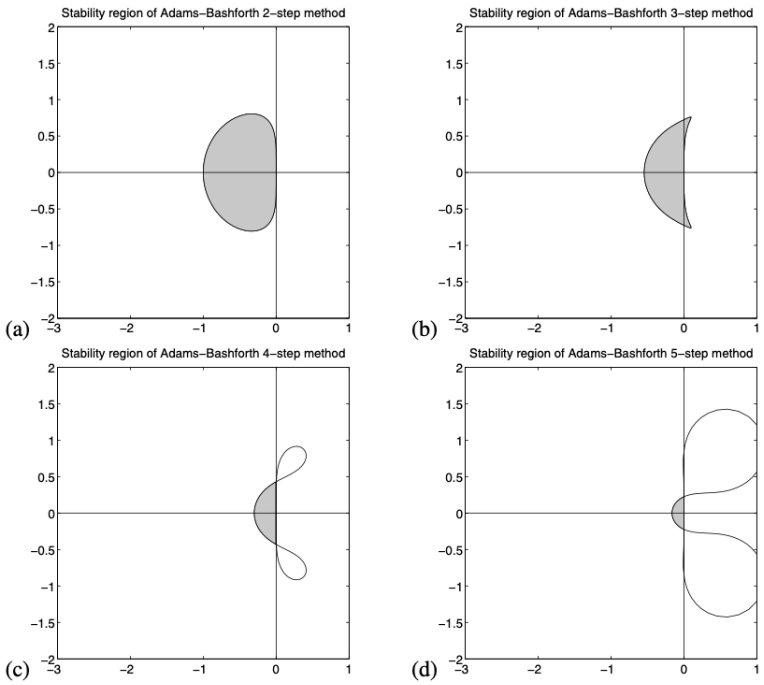


Figure 7.2. *Stability regions for some Adams–Bashforth methods. The shaded region just to the left of the origin is the region of absolute stability. See* Section 7.6.1 *for a discussion of the other loops seen in figures (c) and (d).*

Figure 1.3: The figure is copied from LeVeque's book: Finite Difference Methods for Ordinary and Partial Differential Equations

Project 1

1. Study the ozone photochemistry problem (Read Durran's book, 2.5.2., 2.5.4). This is a stiff ODE. The following chemical reaction is from atmospheric science. The reaction involves oxygen (O), nitrogen oxides (NO, NO_2), and ozone (O_3):

$$NO_2 + h\nu \xrightarrow{k_1} NO + O$$
$$O + O_2 \xrightarrow{k_2} O_3$$
$$NO + O_3 \overset{k_3}{ongrightarrow} O_2 + NO_2.$$

   Here, $h\nu$ denotes a photo of solar radiation. Let $c_1, ..., c_4$ be the concentrations of O, NO, NO_2 and O_3, respectively. The reaction system is

$$\dot{c}_1 = k_1 c_3 - k_2 c_1,$$
$$\dot{c}_2 = k_1 c_3 - k_3 c_2 c_4,$$
$$\dot{c}_3 = k_3 c_2 c_4 - k_1 c_3,$$

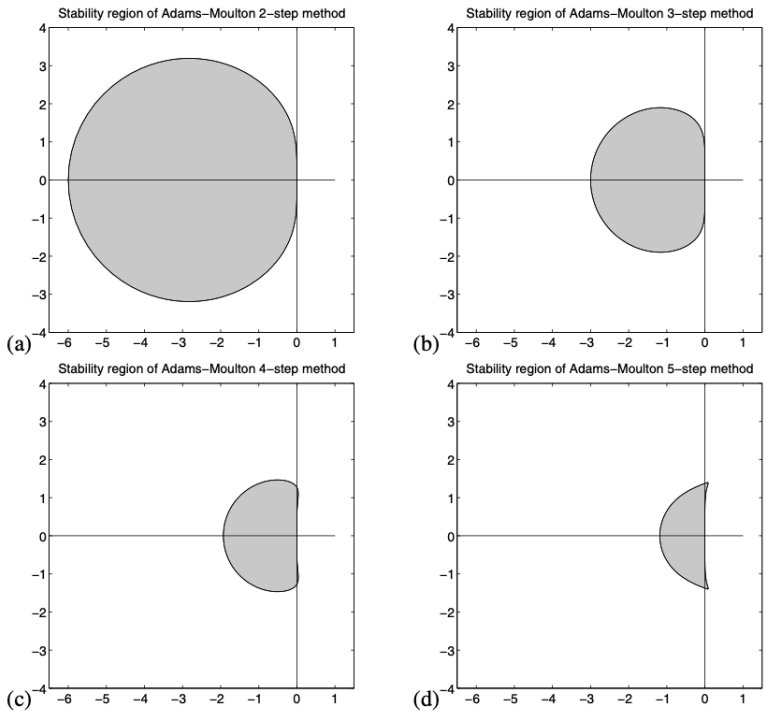


Figure 7.3. *Stability regions for some Adams–Moulton methods.*

Figure 1.4: The figure is copied from LeVeque's book: Finite Difference Methods for Ordinary and Partial Differential Equations

$$\dot{c}_4 = k_2 c_1 - k_3 c_2 c_4.$$

The numerical reaction rates are

$$k_1 = 10^{-2} \max[0, \sin(2\pi t/t_d)] s^{-1}, \quad t_d \text{ the length of 1 day},$$
$$k_2 = 10^5 s^{-1}, \quad k_3 = 10^{-16} cm^3 molecule^{-1} s^{-1}.$$

The initial concentration at sunrise is

$$(c_1, c_2, c_3, c_4) = (0, 0, 5 \times 10^{11}, 8 \times 10^{11}) \; molecules/cm^3$$

Find the concentration in the next two days. Plot your answers.

2. Consider the logistic equation

$$y' = ry(1-y), \quad 0 < y(0) < 1.$$

The state $y = 1$ is a stable equilibrium. You can test several methods to see the constraint on the step size k to be able to construct a good approximate solution. Think about how this is connected to the absolute-stability region. If k becomes large, what will you see?

Chapter 2

Finite Difference Methods for Heat Equation

2.1 Problem set-up

1. **The heat equation** We consider the heat equation in 3D:

$$u_t = K(u_{xx} + u_{yy} + u_{zz}). \tag{2.1}$$

This equation is derived from the law of conservation of energy

$$\frac{\partial}{\partial t}\int_\Omega c_v u \, d\mathbf{x} = \int_{\partial\Omega} \mathbf{f}\cdot(-\mathbf{n})\, dS.$$

Here, c_v is the specific heat constant, u the temperature, $\mathbf{f}$ the heat flux, $\mathbf{n}$ the outer normal of the boundary $\partial\Omega$. The domain $\Omega \subset \mathbb{R}^3$ is an arbitrary domain. The left-hand side is the rate-of-change of energy in Ω, where $c_v u$ is the energy density. The right-hand side is the heat flux flows into Ω from its surrounding. The heat should diffuse from high temperature to low temperature. Fourier proposes that heat flux has the form:

$$\mathbf{f} = -\kappa\nabla u,$$

where $\kappa > 0$ is called the heat conductivity. Plug this Fourier law into the above equation, apply divergence theorem, we obtain

$$\frac{\partial}{\partial t}\int_\Omega c_v u = \int_{\partial\Omega} (-\kappa\nabla u)\cdot(-\mathbf{n})\, dS = \int_\Omega \nabla\cdot(\kappa\nabla u)\, d\mathbf{x}.$$

This integral equation is valid for any arbitrary domain. As a result, we obtain

$$c_v u_t = \nabla\cdot(\kappa\nabla u).$$

If the material is homegeneous, then κ is a constant, we obtain

$$u_t = K\nabla^2 u,$$

where $K = \kappa/c_v > 0$ is called the dynamic heat conductivity. We can rescale $\mathbf{x}$ by $\mathbf{x}' = \mathbf{x}\sqrt{K}$, then

$$u_t = \nabla'^2 u.$$

We can also rescale t by $t' = tK$, then

$$u_{t'} = \nabla^2 u$$

Both rescaling leads us to the standard heat equation

$$\boxed{u_t = \nabla^2 u.}$$

The operator

$$\nabla^2 = \triangle = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}.$$

is called the Laplacian (operator).

2. **Geometry of domains** In applications, the domains can be an arbitrary domain in three dimensions. However, in developing numerical methods, we shall consider simpler domains. Then go to more and more complex domains. Simple domains include

   - rectangular domain in 1D, 2D and 3D with period, Dirichlet, Neumann boundary conditions;
   - sphere, spherical shell;
   - arbitrary geometry.

   Usually, rectangular mesh can be adopted for those nice rectangular domains. Then finite difference methods can be applied. For arbitrary domain, triangular meshing is always possible, and finite element methods can be applied.

3. **Initial condition**

$$u(x, 0) = u_0(x). \tag{2.2}$$

4. **Boundary conditions** There are various kinds of boundary conditions. The boundary of the domain Ω, denoted by Γ, is decomposed into

$$\Gamma = \Gamma_D \cup \Gamma_N \cup \Gamma_R$$

   On the boundary, we impose

   - Dirichlet: $u(x, t) = u_D(x), \quad x \in \Gamma_D$
   - Neumann: $\frac{\partial u}{\partial n} = g(x), \quad x \in \Gamma_N$.
   - Robin: $\frac{\partial u}{\partial n} - \alpha u(x) = h(x) \quad x \in \Gamma$

   Here, n is the outer normal of $\partial\Omega$. In the Rubin condition, $h(x)$ represents environmental temperature. We may assume it is zero. The term $\partial u/\partial n$ is the heat flux flowing outward. The term $-\alpha u$ is the radiation rate. When $u > 0$ (resp. $u < 0$), the heat flows outward (resp. inward) at rate $\alpha|u|$.

2.2 Finite Difference Method for 1-D heat equation

2.2.1 Discretization

1. **Spatial Discretization** Let us consider 1D case. First, we need to have a grid system. Let us consider the domain $\Omega = [0, 1]$. The grid is uniform grid

$$G = \{x_j = \frac{j}{N} | j = 0, ..., N\},$$

and N is the number of cells. Let us call $1/N = h$, the mesh size. On this uniform grid, a smooth function u can be approximated by grid value with error as

$$u_{xx}(x_j, t) = \frac{1}{h^2} \left(u(x_{j+1}, t) - 2u(x_j, t) + u(x_{j-1}, t) \right) + O(h^2).$$

You can check this by Taylor expansion. Thus, we consider a discrete function $U_j(t)$ to approcimate $u(x_j, t)$. The discrete function $U_j(t)$ satisfies

$$\boxed{\dot{U}_j(t) = \frac{1}{h^2} \left(U_{j+1}(t) - 2U_j(t) + U_{j-1}(t) \right).} \tag{2.3}$$

Here, $\dot{U}$ denotes for time derivative dU/dt. We can express this in matrix form as

$$\boxed{\dot{U} = \frac{1}{h^2} LU,} \tag{2.4}$$

where $U = [U_1, ..., U_{N-1}]^T$ is the unknowns. The operator L is called a *discrete Laplacian*. Note that U_0 and U_N as the prescribed boundary values, which are taken to be zeros:

$$U_0 = 0, \quad U_N = 0.$$

This is called the Dirichlet boundary condition. The matrix representation of the discrete Laplacian L is

$$L = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}_{(N-1)\times(N-1)} . \tag{2.5}$$

Equation (2.3) is called the spatial discretization equation for the heat equation. This is a system of ordinal differential equations. We can apply numerical ODE solver to solve this equation. Below, we introduce temporal discretization for solving this ODE.

2. **Temporal discretization** The goal here is to integrate (2.4) in time t. We shall introduce forward Euler method, backward Euler method and the Crank-Nicolson method for temporal integration. Let us also choose uniform mesh on the time interval $[0, T]$. The time step size is denoted by k. We use U_j^n to approximate $U_j(nk)$. Let us integrate (2.4) for one time step:

$$\int_{t^n}^{t^{n+1}} \dot{U}(t)\, dt = \int_{t^n}^{t^{n+1}} \frac{1}{h^2} LU(t)\, dt.$$

The LHS gives $U^{n+1} - U^n$. There are many ways to approximate the integration on the RHS. We lis few common used methods below, which use the rectangular rule (forward, backward) and trapezoidal rule for numerical integration.

- Forward Euler method:
$$U^{n+1} - U^n = \frac{k}{h^2} LU^n. \tag{2.6}$$
- Backward Euler method:
$$U^{n+1} - U^n = \frac{k}{h^2} LU^{n+1}. \tag{2.7}$$
This leads to
$$\left(I - \frac{k}{h^2} L\right) U^{n+1} = U^n$$
and gives
$$U^{n+1} = \left(I - \frac{k}{h^2} L\right)^{-1} U^n.$$
- Crank-Nicolson method. This is the trapezoidal method for numerical ODE.
$$U^{n+1} - U^n = \frac{k}{2h^2} \left(LU^n + LU^{n+1}\right). \tag{2.8}$$
This gives
$$U^{n+1} = \left(I - \frac{k}{2h^2} L\right)^{-1} \left(I + \frac{k}{2h^2} L\right) U^n.$$

Homeworks 2.1. 1. Find the eigenvalues of the discrete Laplacian (2.5).
A matlab code is available

LapEig.m

```
% Define the parameters
N = 101;          % Number of grid points
L = 1;              % Length of the domain
h = L/(N-1);     % Grid spacing
```

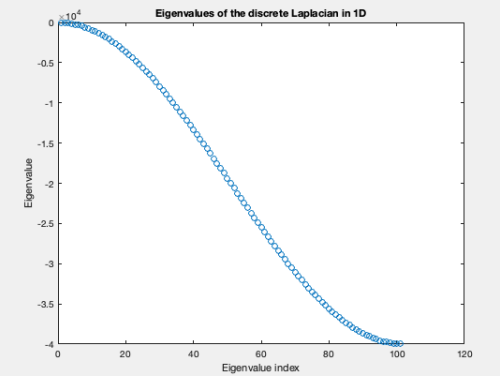


Figure 2.1: The eigenvalues of the discrete Laplacian in 1D with Dirichlet boundary condition. The number of grid points $N = 101$. Note that the eigenvalues ranging from $-4 \sim -N^2$.

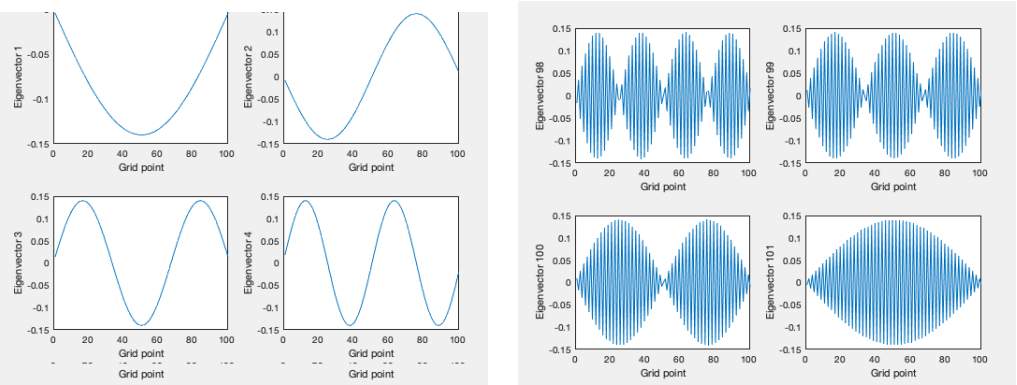


Figure 2.2: The first 4 (left) and the last 4 (right) eigenvectors of the discrete Laplacian in 1D with Dirichlet boundary condition. The number of grid points $N = 100$.

```
% Construct the discrete Laplacian matrix
A = -2*eye(N) + diag(ones(N-1,1),1) + diag(ones(N-1,1),-1);
A = (1/h^2) * A;

% Find the eigenvalues and eigenvectors
[V,D] = eig(A);

% Extract the eigenvalues
lambda = diag(D);

% Sort the eigenvalues in descending order
```

```
[lambda, idx] = sort(lambda, 'descend');

% Reorder the eigenvectors accordingly
V = V(:, idx);

% Plot the first and last four eigenvectors
figure
for i = 1:4 % plot first 4 eigenvectors
    subplot(2,2,i)
    plot(V(:,i))
    xlabel('Grid point')
    ylabel(['Eigenvector ',num2str(i)])
end

% Plot the first and last four eigenvectors
figure
for i = N-3:N % plot last 4 eigenvectors
    subplot(2,2,i-N+4)
    plot(V(:,i))
    xlabel('Grid point')
    ylabel(['Eigenvector ',num2str(i)])
end

% Plot the eigenvalues
figure
plot(lambda,'o')
xlabel('Eigenvalue index')
ylabel('Eigenvalue')
title('Eigenvalues of the discrete Laplacian in 1D')
```

2.2.2 Solving the discrete equations

Writing computer codes to solve the above linear equations. The following items are typical code structure.

1. Parameter set-up
    - Domain set up:$[X_L, X_R]$, N_X, Δx
    - Time domain set up: T_{stop}, N_T, or Δt
    - Ploting frequency:
    - Physical parameters: conductivity κ
2. Initialization and boundary conditions:

- $U^0[j]$, $j = 0, N_X$
- Boundary condition: U_L, U_R.

3. Select a method

4. Time stepping: solve the equation below until T_{stop}

   - Given U^n, solve for U^{n+1}.
   - Apply boundary condition.
   - Plot (or store) U^{n+1} while requested.

5. Diagnosis and analysis

   - Design benchmark problem to test
   - Plots, animations
   - Verification and Validation
   - Quantitative analysis, statistics, ...

2.2.3 Python codes

A python code for 1D heat equation is available in the open course Practical Numerical Methods with Python.

2.3 Stability analysis

The stability analysis here is mainly for the interior domain. So, we may either setup the domain to be $[0, 1]$, or S^1 (periodic domain). The computational grid is $\{x_j = jh | j \in \mathbb{Z}_n, h = 1/N\}$, or $\{x_j = jh | j \in \mathbb{Z}_N, h = 2\pi/N\}$. Below, we shall assume our domain is $\mathbb{R}$.

2.3.1 Maximum norm estimates

1. **Example of instability** In the above forward Euler scheme, you may observe that the solution can oscillate if the time step size is large. In such a situation, we say that the scheme is unstable. Let us analyze such instability of a scheme below. Let us denote

   $$\lambda := \frac{k}{h^2},$$

   as a natural *non-dimensional* parameter for a parabolic equation. We will fix it. Let us express the forward Euler method for just one step to analyze its behavior:

   $$U_j^{n+1} = U_j^n + \lambda \left(U_{j-1}^n - 2U_j^n + U_{j+1}^n \right) = \alpha U_{j-1}^n + \beta U_j^n + \gamma U_{j+1}^n. \tag{2.9}$$

Here, the coefficients

$$\alpha = \gamma = \lambda, \quad \beta = 1 - 2\lambda. \tag{2.10}$$

Note that

$$\alpha + \beta + \gamma = 1,$$

which implies that U_j^{n+1} is the "average" of U_{j-1}^n, U_j^n, and U_{j+1}^n if $\alpha, \beta, \gamma \geq 0$. If these coefficients are not all non-negative, say $\lambda = 1$, which gives $\beta = -1$, then

$$U_j^{n+1} = U_{j-1}^n - U_j^n + U_{j+1}^n.$$

Let us take $U_j^0 = \delta_j^0$ initially. *

- At $n = 1$ we get $U_0^1 = -1, \quad U_{\pm 1}^1 = 1.$
- At $n = 2$, we get $U_0^2 = 3, \quad U_{\pm 1}^2 = -2, \quad U_{\pm 2}^2 = 1.$
- At $n = 3$, we get $U_0^3 = -7, \quad U_{\pm 1}^3 = 6, \quad U_{\pm 2}^3 = -3, \quad U_{\pm 3}^3 = 1.$

You can continue this calculation and find a general formula by hand. Or you can put this scheme into computer to try few more steps. We observe that the solution U oscillates and grows. This violates the diffusion process. The heat should diffuse and decay, not oscillate.

2. **Example of stability** On the other hand, if we choose

$$\boxed{\lambda \leq 1/2,} \tag{2.11}$$

fixed, then from (2.10)

$$\alpha, \beta, \gamma \geq 0.$$

Equation (2.9) states that U_j^{n+1} is the *average* of U_{j-1}^n, U_j^n and U_{j+1}^n. This leads to

$$\begin{aligned} U_j^{n+1} &= \alpha U_{j-1}^n + \beta U_j^n + \gamma U_{j+1}^n \\ &\leq \alpha |U_{j-1}^n| + \beta |U_j^n| + \gamma |U_{j+1}^n| \\ &\leq (\alpha + \beta + \gamma) \max_i |U_i^n| \\ &= \|U^n\|_\infty. \end{aligned}$$

Here, $\|U\|_\infty := \max_i |U_i|$ is called the maximum norm of U, or the sup norm of U. Similarly,

$$\begin{aligned} -U_j^{n+1} &= -\left(\alpha U_{j-1}^n + \beta U_j^n + \gamma U_{j+1}^n\right) \\ &\leq \alpha |U_{j-1}^n| + \beta |U_j^n| + \gamma |U_{j+1}^n| \\ &\leq (\alpha + \beta + \gamma) \max_i |U_i^n| \end{aligned}$$

* $\delta_0^j := \begin{cases} 1 & \text{if } j = 0 \\ 0 & \text{otherwise.} \end{cases}$

$$= \|U^n\|_\infty.$$

These imply that

$$|U_j^{n+1}| \le \|U^n\|_\infty \text{ for all } j.$$

Hence.

$$\boxed{\|U^{n+1}\|_\infty \le \|U^n\|_\infty.} \tag{2.12}$$

The conclusion here is that $\|U^n\|_\infty$ remains bounded for all n.

To summarize, we have shown that the forward Euler method for the heat equation is stable (i.e., the solution does not grow unboundedly) if and only if $\lambda \le 1/2$. When $\lambda > 1/2$, the solution grows unboundedly and oscillates, violating the physical behavior of heat diffusion.

3. **Definition of Stability**

Definition 2.1. A scheme which produces approximate solutions $\{U^n\}$ is called stable if there exists a norm $\|\cdot\|$ [†] such that $\{\|U^n\|\}$ remain bounded uniformly in n.

Remark This condition is necessary for convergence. Let us fix a point (x,t). Consider a limiting process: we fix $\lambda = k/h^2$ and take

$$jh \to x, \quad nk \to t$$

while

$$n \to \infty, \quad j \to \infty.$$

If $U_j^n \to u(x,t)$, then $\{U_j^n\}$ has to be bounded. Thus, convergence implies stability. This is indeed a theorem, which will be discussed in detail later.

Homeworks 2.2. 1. For the forward Euler method (2.6) with $\lambda \le 1/2$, show that

$$\min_j U_j^{n+1} \ge \min_j U_j^n,$$

$$\max_j U_j^{n+1} \le \max_j U_j^n.$$

2.3.2 Energy method

1. The energy method studies the stability problem in L^2-norm:

$$\|U\|_2 := \left(\sum_j |U_j|^2 h \right)^{1/2}.$$

[†] A norm $\|\cdot\|$ in a vector space V is a mapping from V to $\mathbb{R}^+$ which satisfies (i) $\|\mathbf{v}\| \ge 0$ for all $\mathbf{v} \in V$, (ii) $\|\mathbf{v}\| = 0$ if and only if $\mathbf{v} = 0$, (iii) $\|\alpha\mathbf{v}\| = |\alpha|\|\mathbf{v}\|$, and (iv) $\|\mathbf{u}+\mathbf{v}\| \le \|\mathbf{u}\| + \|\mathbf{v}\|$. A vector space V endowed with a norm $\|\cdot\|$ is called a normed vector space. It is used to measure the length of a vector.

2. Let us write the finite difference scheme as

$$U_j^{n+1} = \alpha U_{j-1}^n + \beta U_j^n + \gamma U_{j+1}^n, \tag{2.13}$$

where

$$\alpha, \beta, \gamma \geq 0 \text{ and } \alpha + \beta + \gamma = 1.$$

We multiply (2.13) by U_j^{n+1} on both sides, apply Cauchy-Schwarz inequality,[‡] we get

$$\begin{aligned}(U_j^{n+1})^2 &= \alpha U_{j-1}^n U_j^{n+1} + \beta U_j^n U_j^{n+1} + \gamma U_{j+1}^n U_j^{n+1} \\ &\leq \frac{\alpha}{2}((U_{j-1}^n)^2 + (U_j^{n+1})^2) + \frac{\beta}{2}((U_j^n)^2 + (U_j^{n+1})^2) + \frac{\gamma}{2}((U_{j+1}^n)^2 + (U_j^{n+1})^2)\end{aligned}$$

Here, we have used $\alpha, \beta, \gamma \geq 0$. We multiply this inequality by h and sum it over $j \in \mathbb{Z}$ to get

$$\begin{aligned}\|U^{n+1}\|_2^2 &\leq \frac{\alpha}{2}(\|U^n\|_2^2 + \|U^{n+1}\|_2^2) + \frac{\beta}{2}(\|U^n\|_2^2 + \|U^{n+1}\|_2^2) + \frac{\gamma}{2}(\|U^n\|_2^2 + \|U^{n+1}\|_2^2) \\ &= \frac{1}{2}(\|U^n\|_2^2 + \|U^{n+1}\|_2^2).\end{aligned}$$

Here, $\alpha + \beta + \gamma = 1$ is used. Thus, we get the energy estimate

$$\boxed{\|U^{n+1}\|_2^2 \leq \|U^n\|_2^2.} \tag{2.14}$$

Homeworks 2.3. 1. Can the RK-2 method possess an energy estimate? (*Ans.* Yes.)

2.3.3 Entropy method

The property that U^{n+1} is a convex combination (average) of U^n is very important. Given any convex function $\eta(u)$, by Jenson's inequality, we have [§]

$$\eta(U_j^{n+1}) = \eta\left(\alpha U_{j-1}^n + \beta U_j^n + \gamma U_{j+1}^n\right) \leq \alpha\eta(U_{j-1}^n) + \beta\eta(U_j^n) + \gamma\eta(U_{j+1}^n). \tag{2.15}$$

Summing over all j and using $\alpha + \beta + \gamma = 1$, we get

$$\boxed{\sum_j \eta(U_j^{n+1}) \leq \sum_j \eta(U_j^n).} \tag{2.16}$$

The convex function η is called an *entropy* in this setting. The above inequality means that the "entropy" decreases in time. In particular, we choose

[‡] $|ab| \leq \frac{a^2}{2} + \frac{b^2}{2}$

[§] Jenson's inequality: for convex function η, we have

$$\eta(\alpha U + (1-\alpha)V) \leq \alpha\eta(U) + (1-\alpha)\eta(V).$$

- $\eta(u) = |u|^2$, this gives the L^2 stability estimate,
- $\eta(u) = |u|^p$, $1 \le p < \infty$, we get

$$\sum_j |U_j^{n+1}|^p \le \sum_j |U_j^n|^p.$$

This leads to

$$\left(\sum_j |U_j^{n+1}|^p h\right)^{1/p} \le \left(\sum_j |U_j^n|^p h\right)^{1/p},$$

the general L^p stability estimates. Taking $p \to \infty$, we recover L^∞ stability.[¶]

- $\eta(u) = |u - c|$ for any constant c, we obtain

$$\sum_j |U_j^{n+1} - c| \le \sum_j |U_j^n - c|.$$

This is called Kruzkov's entropy estimate. We will see this inequality in the theory of hyperbolic conservation laws in later chapter.

Homeworks 2.4. 1. Show that the solution of the difference equation derived from the RK2 satisfies the entropy estimate. What is the condition required on h and k for such entropy estimate?

2.3.4 Von Neumann's stability Analysis

1. The von Neumann analysis (via Fourier method) provides a necessary and sufficient condition for L^2-stability. It is applicable to general time-evolution linear systems (discrete or continuous parabolic equations, hyperbolic equations,...) with constant coefficients. For systems with variable coefficients, Kreiss' matrix theorem provides a good characterization of stability.

2. Let us consider an evolutionary finite difference system in a general form:

$$U_j^{n+1} = \sum_{k=-l}^{m} a_k U_{j+k}^n = (\boldsymbol{G}U^n)_j.$$

The operator $\boldsymbol{G} : U^n \mapsto U^{n+1}$.

3. Let us consider the following infinite dimensional vector space

$$\ell^2(\mathbb{Z}) := \{U : \mathbb{Z} \to \mathbb{C} | \sum_{j\in\mathbb{Z}} |U_j|^2 < \infty\}.$$

[¶] $\|U\|_p \to \|U\|_\infty$ as $p \to \infty$.

In $\ell^2(\mathbb{Z})$, we define the inner product

$$\langle U, V\rangle := \sum_{j\in\mathbb{Z}} U_j \bar{V}_j,$$

and the ℓ^2-norm

$$\|U\| := \sqrt{\langle U, U\rangle} = \left(\sum_{j\in\mathbb{Z}} |U_j|^2\right)^{1/2}.$$

The ℓ^2-space is a Hilbert space. [‖]

4. Let us consider another functional space $L^2(S^1)$:

$$L^2(S^1) := \{\phi : S^1 \to \mathbb{C} | \int_0^{2\pi} |\phi(\xi)|^2\, d\xi < \infty\}$$

In $L^2(S^1)$, we define the inner product

$$\langle \phi, \psi\rangle := \frac{1}{2\pi}\int_0^{2\pi} \phi(\xi)\overline{\psi(\xi)}\, d\xi,$$

and the L^2-norm

$$\|\phi\| := \sqrt{\langle \phi, \phi\rangle} = \left(\frac{1}{2\pi}\int_0^{2\pi} |\phi(\xi)|^2\, d\xi\right)^{1/2}.$$

This $L^2(S^1)$ space is also a Hilbert space.

5. **Fourier transform** Given $U = (U_j)_{j\in\mathbb{Z}} \in \ell^2(\mathbb{Z})$, we define its Fourier transform by

$$\hat{U}(\xi) = \frac{1}{2\pi}\sum_{j\in\mathbb{Z}} U_j e^{-ij\xi}, \xi \in [0, 2\pi).$$

The function $\hat{U} : S^1 \to \mathbb{C}$ is a 2π-periodic function. In fact, $\hat{U} \in L^2(S^1)$. The Fourier transform: $U \mapsto \hat{U}$ maps $\ell^2(\mathbb{Z})$ to $L^2(S^1)$.

6. There are two pros to analyze stability of a finite difference schemes using Fourier method.

   - The shift operator $T : \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z})$ is defined by $(TU)_j := U_{j+1}$. The shift operator is transformed to a multiplier:

$$\widehat{TU}(\xi) = e^{i\xi}\hat{U}(\xi).$$

[‖] A Hilbert space means it is a vector space endowed with an inner product structure. Moreover, it is complete, which means that every Cauchy sequence converges.

- The Parseval equality

$$\begin{aligned} \|U\|^2 &= \|\hat{U}\|^2 \\ &\equiv \frac{1}{2\pi}\int_0^{2\pi} |\hat{U}(\xi)|^2\, d\xi \end{aligned}$$

holds, thus one can control the ℓ^2-norm of U and $\boldsymbol{G}U$ in the Fourier space.

7. Now, let us consider a finite difference scheme of the form:

$$Q(T)U^{n+1} = P(T)U^n, \tag{2.17}$$

where Q, P are polynomials, say

$$(P(T)U)_j = \sum_{k=-l}^{m} a_k U_{j+k}, \quad (Q(T)U) = \sum_{k=-l}^{m} b_k U_{j+k}.$$

The Fourier transform of $P(T)U$ gives

$$\begin{aligned} \widehat{P(T)U}(\xi) &= \sum_{j\in\mathbb{Z}} \left(\sum_{k=-l}^{m} a_k U_{j+k} \right) e^{-ij\xi} \\ &= \sum_{k=-l}^{m} a_k \sum_{j\in\mathbb{Z}} U_{j+k} e^{-ij\xi} \\ &= \sum_{k=-l}^{m} a_k \sum_{j\in\mathbb{Z}} U_j e^{-i(j-k)\xi} \\ &= \sum_{k=-l}^{m} a_k e^{ik\xi} \sum_{j\in\mathbb{Z}} U_j e^{-ij\xi} \\ &= P(e^{i\xi})\widehat{U}(\xi). \end{aligned}$$

Thus, taking Fourier transform for the finite difference scheme (2.17), we get

$$Q(e^{i\xi})\widehat{U^{n+1}}(\xi) = P(e^{i\xi})\widehat{U^n}(\xi), \quad \widehat{U^{n+1}}(\xi) = \widehat{G}(\xi)\widehat{U^n}(\xi),$$

where

$$\widehat{G}(\xi) := \frac{P(e^{i\xi})}{Q(e^{i\xi})}.$$

From the Parseval equality,

$$\begin{aligned} \|U^{n+1}\|^2 &= \|\widehat{U^{n+1}}\|^2 \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |\widehat{G}(\xi)|^2\, |\widehat{U^n}(\xi)|^2\, d\xi \end{aligned}$$

$$\begin{aligned} &\leq \max_{\xi} |\widehat{G}(\xi)|^2 \frac{1}{2\pi} \int_{-\pi}^{\pi} |\widehat{U^n}(\xi)|^2 \, d\xi \\ &= |\widehat{G}|_\infty^2 \|U^n\|^2. \end{aligned}$$

Thus a sufficient condition for stability is

$$|\widehat{G}|_\infty \leq 1. \tag{2.18}$$

This is also a necessary condition. Indeed, suppose $|\widehat{G}(\xi_0)| > 1$ at some point $\xi_0 \in [0, 2\pi]$, from$\widehat{G}$ being a smooth function in ξ, we can find ε and δ such that

$$|\widehat{G}(\xi)| \geq 1 + \varepsilon \quad \text{for all} \quad |\xi - \xi_0| < \delta.$$

Let us choose an initial data $U^0 \in \ell^2(\mathbb{Z})$ such that $\widehat{U^0}(\xi) = 1$ for $|\xi - \xi_0| \leq \delta$. Then

$$\begin{aligned} \|\widehat{U^n}\|^2 &= \frac{1}{2\pi} \int |\widehat{G}|^{2n}(\xi) |\widehat{U^0}(\xi)|^2 \, d\xi \\ &\geq \frac{1}{2\pi} \int_{|\xi - \xi_0| \leq \delta} |\widehat{G}|^{2n}(\xi) |\widehat{U^0}|^2 \, d\xi \\ &\geq (1 + \varepsilon)^{2n} \frac{\delta}{2\pi} \to \infty \text{ as } n \to \infty \end{aligned}$$

Thus, the scheme is unstable. We conclude the above discussion by the following theorem.

Theorem 2.1. *A finite difference scheme*

$$Q(T)U^{n+1} = P(T)U^n$$

with constant coefficients is stable if and only if

$$\widehat{G}(\xi) := \frac{P\left(e^{i\xi}\right)}{Q\left(e^{i\xi}\right)}$$

satisfies

$$\max_{-\pi \leq \xi \leq \pi} |\widehat{G}(\xi)| \leq 1. \tag{2.19}$$

8. **Examples**

- **Forward Euler method**. The forward Euler method for the heat equation is

$$U_j^{n+1} = G(U_{j-1}, U_j, U_{j+1}) = \lambda U_{j-1} + (1 - 2\lambda) U_j + \lambda U_{j+1}, \quad \lambda = \frac{k}{h^2}.$$

The corresponding

$$\widehat{G}(\xi) = \lambda(e^{i\xi} + e^{-i\xi}) + (1 - 2\lambda) = 1 - 4\lambda \sin^2\left(\frac{\xi}{2}\right).$$

The condition (2.19) is equivalent to

$$\lambda \leq \frac{1}{2}.$$

That is,

$$\frac{k}{h^2} \leq \frac{1}{2}.$$

Or equivalently, U_j^{n+1} is the convex combination of U_{j-1}, U_j and U_{j+1}.

- **Backward Euler method**

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

Taking Fourier transform, we get

$$\widehat{U}^{n+1}(\xi) = \widehat{G}(\xi)\widehat{U}^n(\xi),$$

where

$$G(\xi) = \frac{1}{1 - \lambda \left(e^{i\xi} - 2 + e^{-i\xi} \right)} = \frac{1}{1 - \lambda \left(2\cos(\xi) - 2 \right)} = \frac{1}{1 + 4\lambda \sin^2 \left(\frac{\xi}{2} \right)}.$$

We see that $|G(\xi)| \leq 1$ for any $\xi \in [0, 2\pi]$. There is no restriction on $\lambda = k/h^2$. Thus, we say the backward Euler scheme is *unconditional stable*.

- **Crank-Nicolson method** The scheme is

$$\left(1 - \frac{\lambda}{2} L \right) U^{n+1} = \left(1 + \frac{\lambda}{2} L \right) U^n$$

where L is the discrete Laplacian:

$$LU = \left(T - 2 + T^{-1} \right) U$$

The Fourier transform of L is

$$\widehat{L} = e^{i\xi} - 2 + e^{-i\xi} = -4 \sin^2 \left(\frac{\xi}{2} \right).$$

Thus, the amplification $\widehat{G}(\xi)$ is

$$G(\xi) = \frac{1 - 2\lambda \sin^2 \left(\frac{\xi}{2} \right)}{1 + 2\lambda \sin^2 \left(\frac{\xi}{2} \right)}.$$

We see that

$$|\widehat{G}(\xi)| \leq 1$$

for all $\lambda > 0$ for all $\xi \in [0, 2\pi)$. Thus, the Crank-Nicolson method is *unconditional stable*.

Homeworks 2.5. 1. Compute the $\widehat{G}$ for the schemes: RK2, RK4.

2.4 * Relaxation of errors

In this section, we study the evolution of an error on a periodic domain $[0, 2\pi)$. We consider

$$u_t = u_{xx}, \quad x \in [0, 2\pi), \tag{2.20}$$

with initial data u_0. The grid points $x_j = 2\pi j/N$, $j = 0, ..., N$, and $h = 2\pi/N$.

Truncation error Let $u(\cdot, \cdot)$ be a smooth solution. We plug u into the difference equation (2.9). The remaining term is called the truncation error τ_j^n:

$$u(x_j, t^{n+1}) = u(x_j, t^n) + \frac{\Delta t}{h^2}\left(u(x_{j-1}, t^n) - 2u(x_j, t^n) + u(x_{j+1}, t^n)\right) + \Delta t \tau_j^n.$$

By Taylor expansion,

$$\tau_j^n = O(\Delta t) + O(h^2).$$

True error The true error $e_j^n := u(x_j, t^n) - U_j^n$ satisfies

$$e_j^{n+1} = e_j^n + \lambda(e_{j-1}^n - 2e_j^n + e_{j+1}^n) + \Delta t \tau_j^n. \tag{2.21}$$

We want to know how error e^n is relaxed to zero from an initial error e^0.

Homogeneous linear difference equation Let us study the homogeneous finite difference equation first. That is

$$e_j^{n+1} = e_j^n + \lambda(e_{j-1}^n - 2e_j^n + e_{j+1}^n). \tag{2.22}$$

or $e^{n+1} = G(u^n)$. The matrix is a tridiagonal matrix. It can be diagonalized by Fourier method. The eigenvalues and eigenfunctions are

$$\rho_k = 1 - 2\lambda + 2\lambda\cos(2\pi k/N) = 1 - 4\lambda\sin^2(\pi k/N),$$

$$v_{k,j} = \frac{1}{\sqrt{N}} e^{2\pi i j k/N}, k = 0, ..., N-1.$$

Note that $\{v_k\}_{k=0}^{N-1}$ are orthonormal. When $\lambda \le 1/2$, all eigenvalues are negative except ρ_0:

$$1 = \rho_0 > |\rho_1| > |\rho_2| > \cdots.$$

The eigenfunction corresponding to eigenvalue $\rho_0 = 1$ is

$$v_{0,j} = 1 \text{ for all } 0 \le j \le N-1.$$

Hence, the projection of any discrete function U onto this eigenfunction is the average: $\sum_j U_j$.

Now, we decompose the error into

$$e^n = \sum_{k=0}^{N-1} \langle e^n, v_k \rangle v_k, \quad n \geq 0$$

Then

$$e_k^{n+1} = \rho_k e_k^n.$$

Thus,

$$e_k^n = \rho_k^n e_k^0.$$

Since $\rho_0 = 1$, we see that $e_0^n = e_0^0$, which is the average of e^n, does not decay, unless $e_0^0 = 0$ initially. To guarantee the average of e^0 is zero, we may choose U_j^n to be the cell average of $u(x, t^n)$ in the jth cell:

$$U_j^n = \frac{1}{h} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t^n)\, dx.$$

instead of the grid data. This implies that the initial error has zero local averages.

For the decay behaviours of errors e_k^n for $k = 1, ..., N-1$, we notice that for $1 \leq k \leq N-1$,

$$\rho_k = 1 - 4\lambda \sin^2\left(\frac{\pi k}{N}\right) \approx 1 - 4\lambda \left(\frac{\pi k}{N}\right)^2, \text{ for } N >> 1.$$

The largest values of ρ's are ρ_1 and ρ_{N-1}:

$$\rho_1 = \rho_{N-1} \approx 1 - 4\lambda \left(\frac{\pi}{N}\right)^2 = 1 - 4\frac{\Delta t}{h^2}\frac{\pi^2}{N^2} = 1 - \Delta t.$$

They correspond to low frequency eigenmodes: $v_1 = (e^{2\pi ij/N})_{j=0}^{N-1}$ and $v_{N-1} = (e^{-2\pi ij/N})_{j=0}^{N-1}$. The corresponding decay rate is

$$\rho_1^n \approx (1 - \Delta t)^n = \left(1 - \frac{t}{n}\right)^n \approx e^{-t}.$$

Here, $t = n\Delta t$. This is the decay rate of e_1^n and e_{N-1}^n with $n\Delta t = t$. They are the slowest decay modes. For $k = N/2$, the corresponding eigenmode $v_{N/2} = ((-1)^j)_{j=0}^{N-1}$ is the highest frequency mode. The corresponding eigenvalue

$$\rho_{N/2} = 1 - 4\lambda = 1 - 4\frac{\Delta t}{h^2}.$$

The decay rate is

$$\rho_{N/2}^n = \left(1 - 4\frac{\Delta t}{h^2}\right)^n \approx e^{-\frac{4t}{h^2}}.$$

which decays very fast.

Inhomogeneous linear difference equation The contribution of the truncation error to the true error is given by the following inhomogeneous linear difference equation:

$$e_j^{n+1} = e_j^n + \lambda(e_{j-1}^n - 2e_j^n + e_{j+1}^n) + \Delta t \tau_j^n.$$

Let us expand e^n in eigenmode v_k: $e^n = \sum_{k=0}^{N-1} \langle e^n, v_k \rangle v_k$. The error $\langle e^n, v_k \rangle$ satisfies **

$$\langle e^{n+1}, v_k \rangle = \rho_k \langle e^n, v_k \rangle + \Delta t \langle \tau^n, v_k \rangle.$$

Its solution is

$$\langle e^n, v_k \rangle = \rho_k^n \langle e^0, v_k \rangle + \Delta t \sum_{m=0}^{n-1} \rho_k^{n-1-m} \langle \tau^m, v_k \rangle.$$

We see that the term $\langle e^n, v_0 \rangle$ does not tend to zero unless $\langle \tau^m, v_0 \rangle = 0$. This can be achieved if we choose U_j to be the cell averages instead the grid data. For $k \geq 1$, we use $\rho_1 \leq \rho_k$. We have

$$\begin{aligned} \|e^n\|^2 := \sum_k |\langle e^n, v_k \rangle|^2 &\leq \rho_1 \sum_{k=1}^{N} \langle e^0, v_k \rangle|^2 + \Delta t \sum_{m=0}^{n-1} \rho_1^{n-1-m} \sum_{k=1}^{N} |\langle \tau^m, v_k \rangle|^2 \\ &\leq \rho_1^n \|e^0\|^2 + \Delta t \sum_{m=0}^{n-1} \rho_1^{n-1-m} \|\tau^m\|^2. \end{aligned}$$

We have seen that the truncation error is second order. That is

$$\max_m \|\tau^m\| = O(h^2).$$

Then for $k \geq 1$,

$$\Delta t \sum_{m=0}^{n-1} |\rho_k|^{n-1-m} \leq \Delta t \sum_{m=0}^{n-1} |\rho_1|^{n-1-m} = \Delta t \frac{1-\rho_1^n}{1-\rho_1} \approx \Delta t \frac{1-e^{-t}}{1-(1-\Delta t)} = 1 - e^{-t}.$$

Thus, we obtain

$$|e^n|^2 \leq e^{-t} e^0 + (1 - e^{-t}) O(h^2)$$

with $n\Delta t = t$.
In summary, the kth mode decays at rate $e^{-k^2 t}$, $k = 1, ..., h^{-2}$. Thus, high frequency modes decay very fast.

2.5 Boundary conditions

2.5.1 Dirichlet boundary condition

Nonzero Dirichlet boundary condition Now, we consider the initial-boundary problem:

$$u_t = u_{xx}, \quad x \in [0, 1]$$

**The inner product $\langle e_j^n, v_k \rangle := \sum_{j=0}^{N-1} e_j^n \overline{v}_{k,j}$.

The Dirichlet boundary condition is

$$u(0) = a,\ u(1) = b. \tag{2.23}$$

The initial condition is

$$u(x, 0) = u_0(x).$$

We introduce uniform grids: $x_j = j/N,\ j = 0, ..., N$. The forward Euler method can be realized on $x_1, ..., x_{N-1}$ as

$$U_j^{n+1} - U_j^n = \frac{\Delta t}{h^2}\left(U_{j-1}^n - 2U_j^n + U_{j+1}^n\right), j = 1, ..., N-1.$$

Near the boundary point x_1, the finite difference approximation of u_{xx} at x_1 involves $u(x_0)$. We plug the boundary condition $u(x_0) = a$:

$$u_{xx}(x_1) = \frac{U_0 - 2U_1 + U_2}{h^2} + O(h^2) = \frac{a - 2U_1 + U_2}{h^2} + O(h^2) \tag{2.24}$$

Similarly,

$$u_{xx}(x_{N-1}) = \frac{U_{N-2} - 2U_{N-1} + U_N}{h^2} + O(h^2) = \frac{U_{N-2} - 2U_{N-1} + b}{h^2} + O(h^2)$$

The unknowns are $U_1^n, ..., U_{N-1}^n$ with $N-1$ finite difference equations at $x_1, ..., x_{N-1}$. Including boundary terms, we write the equation as

$$U^{n+1} = (I + \lambda L_D)U^n + \lambda B, \quad \lambda = \frac{\Delta t}{h^2}, \tag{2.25}$$

$$L_D = \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -2 \end{bmatrix}_{(N-1)\times(N-1)}, \quad B = \begin{bmatrix} a \\ 0 \\ \vdots \\ b \end{bmatrix}_{(N-1)\times 1}. \tag{2.26}$$

The matrix L_D is the discrete Laplacian with zero Dirichlet boundary condition. The term B comes from the Dirichlet boundary conditions.

We can have energy estimates, entropy estimates as the case of periodic boundary condition.

Solving heat equation with Dirichlet boundary condition In the implicit methods for solving heat equation, it involves an inversion of the discrete Laplacian. Since the eigenvalues of the Dirichlet BC Laplacian L_D(2.26) are all nonzeros, L_D is invertible. There is no problem in using implicit methods.

2.5.2 Neumann boundary condition

Discrete formulation The Neumann boundary condition is

$$u'(0) = -\sigma_0, \quad u'(1) = \sigma_1. \tag{2.27}$$

We may use the following discretization methods:

- First order:
$$\frac{U_1 - U_0}{h} = -\sigma_0.$$
- Second order: we use extrapolation
$$\frac{-3U_0 + 4U_1 - U_2}{2h} = -\sigma_0.$$

The unknowns are U_j^n with $j = 0, ..., N$. In the mean time, we add two more equations at the boundaries.

Discrete Laplacian with Neumann BC Let L_N be the discrete Laplacian with Neumann boundary condition using the first order approximation on the boundary. L_N is an $(N+1) \times (N+1)$ matrix.

$$L_N = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 \end{bmatrix}_{(N+1)\times(N+1)}. \tag{2.28}$$

This matrix has the following eigenvectors:

$$v_j^k = \cos(\pi jk/N), \quad k = 0, ..., N$$

with eigenvalue

$$\lambda^k = -2 + 2\cos(\pi k/N) = -4\sin^2\left(\frac{\pi k}{2N}\right), \quad k = 0, ..., N.$$

Note that $\lambda^0 = 0$. The corresponding eigenvector $v^0 = [1, ..., 1]^T$. The error corresponding this eigenmode does not decay.

```
% Define the parameters
N = 100;          % Number of grid points is N+1
L = 1;              % Length of the domain
h = L/N;     % Grid spacing
```

```
% Construct the discrete Laplacian matrix
A = -2*eye(N+1) + diag(ones(N,1),1) + diag(ones(N,1),-1);
 A(1,1) = -1; A(N+1,N+1) = -1;%For Neumann BC
A = (1/h^2) * A;

% Find the eigenvalues and eigenvectors
[V,D] = eig(A);

% Extract the eigenvalues
lambda = diag(D);

% Sort the eigenvalues in descending order
[lambda, idx] = sort(lambda, 'descend');

% Reorder the eigenvectors accordingly
V = V(:, idx);

% Plot the first and last four eigenvectors
figure
for i = 1:4 % plot first 4 eigenvectors
    subplot(2,2,i)
    plot(V(:,i))
    xlabel('Grid point')
    ylabel(['Eigenvector ',num2str(i)])
end

% Plot the first and last four eigenvectors
figure
for i = N-2:N+1 % plot last 4 eigenvectors
    subplot(2,2,i-N+3)
    plot(V(:,i))
    xlabel('Grid point')
    ylabel(['Eigenvector ',num2str(i)])
end

% Plot the eigenvalues
figure
plot(lambda,'o')
xlabel('Eigenvalue index')
ylabel('Eigenvalue')
title('Eigenvalues of the discrete Laplacian in 1D')
```

Alternatively, we can also use the Laplacian at x_0 and x_N. At, x_0, this gives

$$u_{xx}(x_0) = \frac{U_{-1} - 2U_0 + U_1}{h^2} + O(h^2).$$

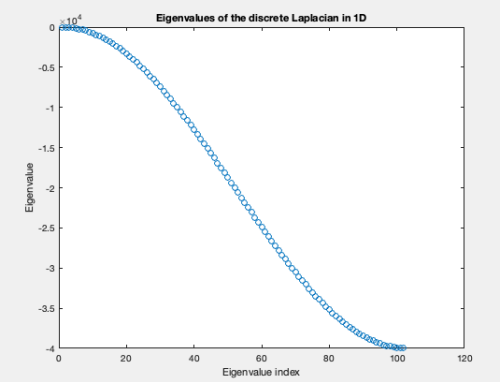


Figure 2.3: The eigenvalues of the discrete Laplacian in 1D with Neumann boundary condition. The number of grid points $N = 101$. Note that the eigenvalues ranging from $0 \sim -N^2$.

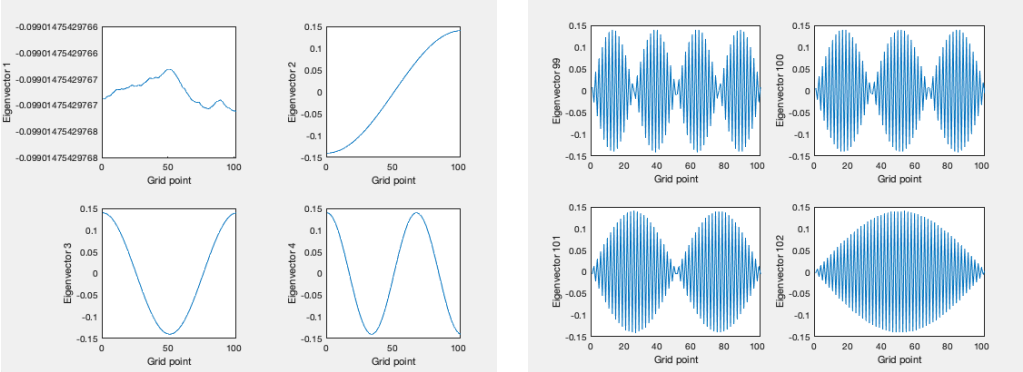


Figure 2.4: The first 4 (left) and the last 4 (right) eigenvectors of the discrete Laplacian in 1D with Neumann boundary condition. The number of grid points is $N = 101$. Note that the first eigenvector is a constant. The plot overwhelms the error.

The Neumann boundary condition $u_x(x_0) = -\sigma_0$ is approximated by

$$u_x(x_0) = \frac{U_1 - U_{-1}}{2h} + O(h^2).$$

This gives

$$u_{xx}(x_0) = \frac{-2U_0 + 2U_1}{h^2} + \frac{2\sigma_0}{h} + O(h).$$

The discretized equation becomes

$$\dot{U}_0 = \frac{-2U_0 + 2U_1}{h^2} + \frac{2}{h}\sigma_0.$$

This equation is equivalent to

$$\frac{1}{2}\dot{U}_0 = \frac{-U_0 + U_1}{h^2} + \frac{1}{h}\sigma_0.$$

The reason we divide this equation by 2 is to make the corresponding discrete Laplacian symmetric. Thus, the heat equation with Neumann boundary condition can be approximated by

$$\begin{bmatrix} \frac{1}{2}\dot{U}_0 \\ \dot{U}_1 \\ \vdots \\ \dot{U}_{N-1} \\ \frac{1}{2}\dot{U}_N \end{bmatrix} = \frac{1}{h^2}\begin{bmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -1 \end{bmatrix}\begin{bmatrix} U_0 \\ U_1 \\ \vdots \\ U_{N-1} \\ U_N \end{bmatrix} + \frac{1}{h}\begin{bmatrix} \sigma_0 \\ 0 \\ \vdots \\ 0 \\ \sigma_1 \end{bmatrix}$$

Homeworks 2.6.

1. Find the eigenfunctions and eigenvalues for the discrete Laplacian with the Neumann boundary condition (consider both first order and second order approximation at boundary). Note that there is a zero eigenvalue.

2. Consider

$$u_t = u_{xx} + f(x)$$

on $[0, 1]$ with Neumann boundary condition $u'(0) = u'(1) = 0$. If $\int f(x)\,dx \neq 0$. What will happen to u as $t \to \infty$?

Solving heat equation with Neumann BC by implicit methods Let us study the backward Euler method:

$$U^{n+1} = U^n + \Delta t L_N U^{n+1}.$$

We need to invert the matrix $I - \Delta t L_N$. The matrix is positive. So there is no problem.

2.6 Multidimensions and sources

2.6.1 2D heat equation

Consider the heat equation

$$u_t = \kappa\left(u_{xx} + u_{yy}\right) \text{ in a rectangle } [x_{min}, x_{max}] \times [y_{min}, y_{max}]$$

with zero Dirichlet boundary condition. The grid is

$$\{(x_i, y_j) | x_i = x_{min} + i\Delta x, i = 0, ..., n_x + 1, y_j = y_{min} + j\Delta y, j = 0, ..., n_y + 1\}$$

The Dirichlet boundary condition gives

$$U_{i,j} = 0 \text{ for } (i = 0) \text{ or } (i = n_x + 1) \text{ or } (j = 0) \text{ or } (j = n_y + 1).$$

The approximate solution is $U_{ij}, i = 1, ..., n_x, j = 1, ..., n_y$. The backward Euler scheme reads

$$\boxed{M\left(U^{n+1} - U^n\right) = \kappa \Delta t L U^{n+1}.} \tag{2.29}$$

where M is the mass operator

$$(MU)_{ij} = U_{ij}$$

and L is the 2D Laplacian:

$$(LU)_{ij} := \frac{1}{(\Delta x)^2}\left(U_{i+1,j} - 2U_{ij} + U_{i-1,j}\right) + \frac{1}{(\Delta y)^2}\left(U_{i,j+1} - 2U_{ij} + U_{i,j-1}\right).$$

The Crank-Nicolson method is

$$\boxed{M\left(U^{n+1} - U^n\right) = \frac{\kappa \Delta t}{2}\left(LU^{n+1} + LU^n\right).} \tag{2.30}$$

Below, we present a matlab code which uses sparse matrix solver. This code is much faster than those without using sparse matrix solver. There are special cares for index arithmetics.

2.6.2 Splitting method for reaction-diffusion equations

1. **Models** Let us consider the reaction-diffusion equation:

$$u_t = \triangle u + f(u) \tag{2.31}$$

The term $\triangle u$ is called the diffusion term, while $f(u)$ the reaction term. It can come from

- Phase field model: u is an order parameter, f is the derivative of a double-well free energy F. An example of F is $F(u) = (1-u^2)^2$, or $F(u) = u^2(1-u)^2$. For the later one, the corresponding $F'(u) = f(u) = u(1-2u)(1-u)$. The state 0 and 1 are stable equilibria, while $u = 1/2$ is an unstable equilibrium.
- Chemical reaction. Usually, u is a vector consisting of several chemical species involving reaction. An example is the following chemical reaction in atmosphere. The reaction involves oxygen (O), nitrogen oxides (NO, NO_2), and ozone (O_3):

$$NO_2 + h\nu \xrightarrow{k_1} NO + O$$
$$O + O_2 \xrightarrow{k_2} O_3$$
$$NO + O_3 \xrightarrow{k_3} O_2 + NO_2.$$

Here, $h\nu$ denotes a photo of solar radiation. Let $c_1, ..., c_4$ be the concentrations of O, NO, NO_2 and O_3, respectively. The reaction system is

$$\dot{c}_1 = k_1 c_3 - k_2 c_1,$$
$$\dot{c}_2 = k_1 c_3 - k_3 c_2 c_4,$$
$$\dot{c}_3 = k_3 c_2 c_4 - k_1 c_3,$$
$$\dot{c}_4 = k_2 c_1 - k_3 c_2 c_4.$$

2. Suppose we want to solve the reaction-diffusion equation in a small time step Δt. Let us start from a state called u_0. This equation can be viewed *roughly* as

$$u_t \approx \triangle(u - u_0) + f'(u_0)(u - u_0).$$

We view it as

$$v_t = Av + Bv,$$

where $v = u - u_0$, $Av = \triangle v$, $Bv = f(u_0)v$. We call these two operators as

- Diffusion operator: $e^{\Delta t \triangle}$, which solves the diffusion equation
- Reaction operator: $e^{\Delta t f'(u)}$, which solves the ODE: $u' = f(u)$ for a small time step.

Let us denote them by $e^{\Delta tA}$ and $e^{\Delta tB}$, respectively. Note that these two operators are not commutable.

3. **Splitting method** Formally, the solution of the equation (2.31) is

$$u = e^{\Delta t(A+B)} u_0.$$

We want to approximate it by

$$u \approx e^{\Delta tB} e^{\Delta tA} u_0.$$

We know how to find $e^{\Delta A}$ and $e^{\Delta tB}$. In other words, we will solve this reaction-diffusion equation by solving

$$u_t = \triangle u$$

and

$$u_t = f(u)$$

alternatively. Namely,

$$e^{t(A+B)} \approx \underbrace{e^{\Delta tB} e^{\Delta tA} \cdots e^{\Delta tB} e^{\Delta tA}}_{n \text{ times}} \; u_0, \quad n\Delta t = t.$$

Such a method is called a splitting method.

4. Let us study the error of $e^{\Delta t(A+B)} - e^{\Delta tB} e^{\Delta tA}$,

$$\begin{aligned} e^{\Delta t(A+B)} &= 1 + \Delta t(A+B) + \frac{\Delta t^2}{2}(A^2 + B^2 + AB + BA) + \cdots \\ e^{tB} \cdot e^{tA} &= (1 + \Delta tB + \frac{\Delta t^2}{2} B^2 + \cdots)(1 + \Delta tA + \frac{\Delta t^2}{2} A^2 + \cdots) \\ &= 1 + \Delta t(A+B) + \frac{\Delta t^2}{2}(A^2 + B^2) + \Delta t^2 BA + \cdots \end{aligned}$$

$$\therefore \quad e^{\Delta t(A+B)} - e^{\Delta tB} \cdot e^{\Delta tA} = \frac{\Delta t^2}{2}(AB - BA) + \mathcal{O}(\Delta t^3) = \frac{\Delta t^2}{2}[A, B] + \mathcal{O}(\Delta t^3).$$

Here, $[A, B] := AB - BA$ is called the *commutator* of A and B. The error is first order after time $n(\Delta t)^2 = O(\Delta t)$, where $n = t/\Delta t$ and t is the final time.

5. **High order splitting** To reach higher order time splitting, we may approximate $e^{t(A+B)}$ by polynomials $P(e^{tA}, e^{tB})$ or rational functions $R(e^{tA}, e^{tB})$ for small t. For example, the Strang splitting (or the Trotter product in physics literatures) is given by

$$\boxed{e^{t(A+B)} = e^{\frac{1}{2}tA}e^{tB}e^{\frac{1}{2}tA} + O(t^3).} \tag{2.32}$$

For $t = n\Delta t$,

$$\begin{aligned} e^{t(A+B)}u_0 &= (e^{\frac{1}{2}\Delta tA}e^{\Delta tB}e^{\frac{1}{2}\Delta tA})\cdots(e^{\frac{1}{2}\Delta tA}e^{\Delta tB}e^{\frac{1}{2}\Delta tA})(e^{\frac{1}{2}\Delta tA}e^{\Delta tB}e^{\frac{1}{2}\Delta tA})u_0 \\ &= e^{\frac{1}{2}\Delta tA}e^{\Delta tB}e^{\Delta tA}e^{\Delta tB}e^{\Delta tA}\cdots e^{\Delta tA}e^{\Delta tB}e^{\frac{1}{2}\Delta tA}u_0 \end{aligned}$$

The Strang splitting is second order.

Homeworks 2.7. 1. Design a third order splitting method for

$$u_t + Au + Bu.$$

2. Suppose we have three operators in our equation:

$$u_t = Au + Bu + Cu.$$

Design first order and second order splitting methods.

Project 2

You can do either one of the following two.

1. Study the ozone photochemistry process with diffusion. The reaction involves oxygen (O), nitrogen oxides (NO, NO_2), and ozone (O_3):

$$\begin{aligned} NO_2 + h\nu &\xrightarrow{k_1} NO + O \\ O + O_2 &\xrightarrow{k_2} O_3 \\ NO + O_3 &\xrightarrow{k_3} O_2 + NO_2. \end{aligned}$$

Here, $h\nu$ denotes a photo of solar radiation. Let $c_1, ..., c_4$ be the concentrations of O, NO, NO_2 and O_3, respectively. The reaction system is

$$\begin{aligned} \dot{c}_1 &= k_1c_3 - k_2c_1, \\ \dot{c}_2 &= k_1c_3 - k_3c_2c_4, \\ \dot{c}_3 &= k_3c_2c_4 - k_1c_3, \\ \dot{c}_4 &= k_2c_1 - k_3c_2c_4. \end{aligned}$$

The numerical reaction rates are

$$k_1 = 10^{-2} \max[0, \sin(2\pi t/t_d)] s^{-1}, \quad t_d \text{ the length of 1 day},$$

$$k_2 = 10^5 s^{-1}, \quad k_3 = 10^{-16} cm^3 molecule^{-1} s^{-1}.$$

The geometry is a rectangle. The initial data is a box function with

$$(c_1, c_2, c_3, c_4) = \begin{cases} (0, 0, 5 \times 10^{11}, 8 \times 10^{11}) & \text{inside box} \\ (0, 0, 0, 0) & \text{elsewhere.} \end{cases}$$

The boundary condition is Neumann BC. Use finite difference and spectral methods. Use Crank-Nicolson in time step. The final equation look like

$$\partial_t \mathbf{c} = \nu \triangle \mathbf{c} + f(\mathbf{c}).$$

For the source and Laplacian on the right-hand side, you can use Strang-splitting to treat theLaplacian and source separately. Or you combine them together as a discrete ODE system.

2. We will study the evolution process of a phase-field model in 2D. You can also search for Cahn-Hilliard model and Allan-Cahn model. We shall neglect the fluid part, only concentrate the evolution of the order parameter. The Allen-Cahn equation is

$$u_t = \triangle u + u - u^3.$$

Here, u represents the order parameter of some material which has two phases (-1 and 1). We choose rectangular domain with the Neumann boundary condition.

Chapter 3

Spectral methods and Fast Fourier Transform

References:

1. D.A. Kopriva, Implementing Spectral Methods for Partial Differential Equations, Scientific Computation, Springer (2009).

3.1 Fourier series expansion

3.1.1 Definition and basic properties

1. $L^2(\mathbb{T})$ **space** A 2π-periodic function can be identified as a function on a circle, which is expressed as $\mathbb{T} = \mathbb{R}/(2\pi\mathbb{Z})$. Let $L^2(\mathbb{T})$ denote for

$$L^2(\mathbb{T}) := \{f : \mathbb{T} \to \mathbb{C} | \int_{\mathbb{T}} |f(x)|^2 \, dx < \infty\}$$

It is a vector space. In $L^2(\mathbb{T})$, we define the inner product:

$$(f, g) := \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x)\overline{g(x)} \, dx.$$

The inner product $(\cdot, \cdot)$ satisfies

- $(f, f) \geq 0$,
- $(f, f) = 0 \Leftrightarrow f = 0$,
- $\overline{(f, g)} = (g, f)$,
- $(\cdot, g)$ is linear any $g \in L^2(\mathbb{T})$.

The space $L^2(\mathbb{T})$ endowed with the inner product structure is also *complete*.* A complete inner-product space is called a *Hilbert space*.

*An inner product space $\mathcal{H}$ is called complete if all its Cauchy sequences converge.

2. In $L^2(\mathbb{T})$, the set $\{e^{ikx} | k \in \mathbb{Z}\}$ is orthonormal:

$$\begin{aligned}(e^{ikx}, e^{imx}) &:= \frac{1}{2\pi}\int_0^{2\pi} e^{ikx}\overline{e^{imx}}\,dx \\ &= \frac{1}{2\pi}\int_0^{2\pi} e^{i(k-m)x}\,dx = \delta_{km}\end{aligned}$$

The set $\{e^{ikx} | k \in \mathbb{Z}\}$ constitutes a basis in $L^2(\mathbb{T})$. This will be justified in the Fourier convergence theory in the next section.

3. Wave number k Note that k is the wave number of the periodic function e^{ikx}. The wave length is $2\pi/k$.

4. **Definition of Fourier series expansion** A Fourier series expansion for a 2π-periodic function f has the following form:

$$\boxed{f(x) = \sum_{k=-\infty}^{\infty} a_k e^{ikx}.} \tag{3.1}$$

By taking the inner product of f with e^{imx}, and using the orthogonality of $\{e^{ikx}\}$, we obtain

$$\boxed{a_m = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-imx}\,dx.}$$

The coefficient a_m is called the Fourier modes, or Fourier coefficients, or Fourier multiples, or Fourier transform of f at wave number m, and is denoted by $\hat{f}_m$.

5. Examples

(a) $\text{sign}(x) := \begin{cases} 1 & \text{for } 0 < x < \pi \\ -1 & \text{for } -\pi < x < 0 \end{cases}$

$$\begin{aligned}a_k &= \frac{1}{2\pi}\left(\int_{-\pi}^{0} -e^{-ikx}\,dx + \int_0^{\pi} e^{-ikx}\,dx\right) \\ &= \frac{1}{2\pi}\left(\int_0^{\pi} -e^{ikx}\,dx + \int_0^{\pi} e^{-ikx}\,dx\right) \\ &= \frac{1}{ik\pi}\left(1 - (-1)^k\right).\end{aligned}$$

(b) $f(x) = \frac{1}{\pi}|x|$

```
% Define the sign function
N = 16; % Number of points
```

```
x = -pi : pi/N : pi-pi/N; % Grid points
f = sign(x); % Sign function

% Take the DFT
F = fft(f);

% Plot the magnitude and phase of the DFT
subplot(2,1,1)
stem(abs(F))
title('Magnitude of DFT')
xlabel('Frequency')
ylabel('Magnitude')

subplot(2,1,2)
stem(angle(F))
title('Phase of DFT')
xlabel('Frequency')
ylabel('Phase')
```

3.1.2 Basic properties of Fourier series

1. **Basic Properties:**
   - The Fourier transform is linear. $\widehat{f+g} = \hat{f} + \hat{g}$.
   - Translation property:
   $$\widehat{f(x+a)}_k = e^{-iak}\hat{f}_k$$
   - The differentiation becomes a multiplication under Fourier transform. It is also equivalent to say that the differential operator is diagonalized in Fourier basis.
   - The convolution becomes a multiplication under Fourier transform.
2. **Differentiation**

   Lemma 3.1. *If $f \in C^1[\mathbb{T}]$, then*
   $$\boxed{\widehat{f'}_k = ik\hat{f}_k.} \tag{3.2}$$

   Proof.
   $$\begin{aligned}\widehat{f'}_k &= \frac{1}{2\pi}\int_0^{2\pi} f'(x)e^{-ikx}\,dx \\ &= \frac{1}{2\pi}\, e^{-ikx}f(x)\Big|_{x=0}^{x=2\pi} - \frac{1}{2\pi}\int_0^{2\pi}(-ik)e^{-ikx}f(x)\,dx\end{aligned}$$

$$= \quad ik\hat{f}_k.$$

Here, we have used the periodicity of f in the last step. □

3. **Convolution** If f and g are in $L^2(\mathbb{T})$, we define the convolution of f and g by

$$(f * g)(x) = \frac{1}{2\pi}\int_{\mathbb{T}}\int_{\mathbb{T}} f(x-y)g(y)\,dy.$$

Many solutions of differential equations are expressed in convolution forms. For instance $-u'' = f$ in $\mathbb{T}$, its solution can be expressed as $u = g * f$, where g is the Green's function of $-d^2/dx^2$ in $\mathbb{T}$. Another example is that we can smoothen a function through convolution. Namely, consider a C^∞-function $\rho(x) > 0$ in $(-1/2, 1/2)$ and $\rho(x) = 0$ elsewhere, and $\int \rho(x)\,dx = 1$. We consider

$$\rho_\varepsilon(x) := \frac{1}{\varepsilon}\rho\left(\frac{x}{\varepsilon}\right),$$

and

$$f_\varepsilon = \rho_\varepsilon * f.$$

The functions $f_\varepsilon \in C^\infty$ and if $f \in L^1(\mathbb{T})$ and $f_\varepsilon \to f$ in L^1.

Lemma 3.2. *If $f, g \in C(\mathbb{T})$, then*

$$\boxed{\left(\widehat{f * g}\right)_k = \hat{f}_k\hat{g}_k.} \tag{3.3}$$

Proof.

$$\begin{aligned}
\left(\widehat{f * g}\right)_k &= \frac{1}{(2\pi)^2}\int_{\mathbb{T}} f * g(x)e^{-ikx}\,dx \\
&= \frac{1}{(2\pi)^2}\int_{\mathbb{T}}\int_{\mathbb{T}} f(x-y)g(y)\,dye^{-ikx}\,dx \\
&= \frac{1}{(2\pi)^2}\int_{\mathbb{T}}\int_{\mathbb{T}} f(x-y)e^{-ik(x-y)}g(y)\,dye^{-iky}\,dx \\
&= \frac{1}{2\pi}\int_{\mathbb{T}}\left(\frac{1}{2\pi}\int_{\mathbb{T}} f(x-y)e^{-ik(x-y)}\,dx\right)g(y)e^{-iky}\,dy \\
&= \frac{1}{2\pi}\int_{\mathbb{T}}\left(\frac{1}{2\pi}\int_{\mathbb{T}} f(x)e^{-ikx}\,dx\right)g(y)e^{-iky}\,dy \\
&= \hat{f}_k\hat{g}_k.
\end{aligned}$$

Here, we have used Fubini theorem. □

4. **Remark** The above two lemmas are valid for $f, g \in L^2(\mathbb{T})$. The proof is based on the L^2 convergence for nice functions and the density theorem in the next section.

3.1.3 Regularity and decay: Riemann-Lebesgue lemma

1. **Smooth functions** If f is smooth, then its Fourier coefficients decay very fast. Indeed, by taking integration-by-part n times, we have

$$\begin{aligned}
\hat{f}_k &= \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-ikx}\,dx \\
&= \frac{1}{-ik}\frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)de^{-ikx} \\
&= \frac{1}{-ik}\frac{1}{2\pi}\int_{-\pi}^{\pi} f'(x)e^{-ikx}\,dx \\
&\vdots \\
&= \frac{1}{(-ik)^n}\frac{1}{2\pi}\int_{-\pi}^{\pi} f^{(n)}(x)e^{-ikx}\,dx.
\end{aligned}$$

   Thus, if $f \in C^n(\mathbb{T})$, then $\hat{f}_k = O(|k|^{-n})$. In fact, we shall see later from the Riemann-Lebesgue lemma that $\hat{f}_k = o(|k|^{-n})$. [†] That is, *the regularity of f implies the decay of $\hat{f}_k$*. In other words, if the function f is very smooth, then its high frequency modes $\hat{f}_k$ is very small.

   The above property can also be observed by the following arguments. We note that

$$\hat{f}_k = -\frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-ik(x+\pi/k)}\,dx$$

   Hence,

$$\begin{aligned}
\hat{f}_k &= \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-ikx}\,dx \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{f(x)-f(x-\pi/k)}{2}e^{-ikx}\,dx \\
&:= \frac{1}{2\pi}\int_{-\pi}^{\pi} D_{\pi/k}f(x)e^{-ikx}\,dx \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} D^n_{\pi/k}f(x)e^{-ikx}\,dx
\end{aligned}$$

   Here, $D_{\pi/k}$ is a backward finite difference operator. We see that if f is smooth, then $D^n_{\pi/k}f = O(|k|^{-n})$. Thus, $\hat{f}_k$ measures the oscillation property of f at scale π/k.

2. **Integrable functions** When f is not so smooth, say in L^1, [‡] we still have $\hat{f}_k \to 0$ as $|k| \to \infty$. This is the following Riemann-Lebesgue lemma.

[†] The notation $o(h)$ means that it is a function $f(h)$ defined in a neighborhood of $h = 0$ and it satisfies $f(h)/h \to 0$ as $h \to 0$.

[‡] A function $f \in L^1(a,b)$ means that $\int_a^b |f(x)|\,dx < \infty$.

Lemma 3.3 (Riemann-Lebesgue)**.** *If f is in $L^1(a,b)$, then*

$$\hat{f}_A := \int_a^b f(x)\sin(Ax)\,dx \to 0, \ \text{as } A \to \infty.$$

Proof. (i) First, we show that the lemma holds for uniformly continuous functions. From $\sin(A(x-\pi/A)) = -\sin(Ax)$, we have

$$\begin{aligned} 2\hat{f}_A &= \int_a^b f(x)\sin(Ax)\,dx - \int_a^b f(x)\sin(A(x-\pi/A))\,dx \\ &= -\int_{a-A/\pi}^{a} f(x+\pi/A)\sin(Ax)\,dx + \int_{b-\pi/A}^{b} f(x)\sin(Ax)\,dx \\ &\quad + \int_a^{b-\pi/A} (f(x)-f(x+\pi/A))\sin(Ax)\,dx. \end{aligned}$$

From the uniform continuity and integrability of f, we have $|\hat{f}_A| \to 0$ as $A \to \infty$.

(ii) When $f \in L^1(a,b)$, we use density theorem, which states that every L^1 function can be approximated by smooth functions in L^1-norm, that is, for any ε, there exists a smooth function g such that $\|f-g\|_{L^1} < \varepsilon$.

(iii) It holds for any A

$$|\widehat{(f-g)}_A| \le \int_a^b |f(x)-g(x)|\,dx := \|f-g\|_{L^1} < \varepsilon.$$

From (i), there exists M such that for $A > M$, $|\hat{g}_A| < \varepsilon$.

(iv) Given $f \in L^1(a,b)$, and given any $\varepsilon > 0$, from (ii), we can find a smooth function g such that $\|f-g\|_{L^1} < \varepsilon$. From (i), there exists an $M > 0$ such that for any $A > M$ we have $|\hat{g}_A| < \varepsilon$. From (iii), we have $|\widehat{(f-g)}_A| \le \|f-g\|_{L^1} < \varepsilon$. Combining all these together, we get

$$|\hat{f}_A| \le |\hat{g}_A| + |\widehat{(f-g)}_A| \le 2\varepsilon.$$

□

We conclude with the following lemma.

Lemma 3.4. *If $f \in C^n(\mathbb{T})$, then $\hat{f}_k = o(|k|^{-n})$.*

Proof. We have seen that

$$\hat{f}_k = \frac{1}{(-ik)^n}\int_{\mathbb{T}} f^{(n)} e^{-ikx}\,dx \tag{3.4}$$

Since $f \in C^n(\mathbb{T})$, $f^{(n)} \in C(\mathbb{T})$. By Riemann-Lebesgue lemma, we get that $\int_{\mathbb{T}} f^{(n)} e^{-ikx}\,dx = o(1)$. This together with (3.4), we get $\hat{f}_k = o(|k|^{-n})$. □

3. **Remarks.**

   - If f is a Dirac delta function, we can also define its Fourier transform

   $$\hat{f}_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} \delta(x)e^{-ikx}\,dx = \frac{1}{2\pi}.$$

   In this case, $\delta \notin L^1$ and $\hat{\delta}_k = 1/2\pi$ does not converge to 0 as $|k| \to \infty$.

   - If f is a piecewise smooth function with finite many jumps, then it holds that $\hat{f}_k = O(1/k)$. To prove such a result, one can first show the result when f has only one jump. Next, consider the case that f is the sum of a step function g with finite many jumps and an absolutely continuous function h. We have seen that $\hat{h}_k$ decays as $o(1/k)$. For the step function g, you can show that $\hat{g}_k = O(1/k)$.

3.2 Convergence Theory

Let us denote the partial sum of the Fourier expansion by f_N:

$$f_N(x) := \sum_{k=-N}^{N} \hat{f}_k e^{ikx}.$$

We shall show that under proper condition, f_N will converge to f. The convergence is in the sense of uniform convergence for smooth functions, in L^2 sense for L^2 functions, and in pointwise sense for BV functions.

3.2.1 Convergence theory for Smooth function

Theorem 3.1. *If f is a 2π-periodic, C^∞-function, then for any $s > 0$, there exists a constant C_s such that*

$$|f_N(x) - f(x)| \le C_s N^{-s}. \tag{3.5}$$

Proof.

$$\begin{aligned} f_N(x) &:= \sum_{|k|\le N} \hat{f}_k e^{ikx} \\ &= \sum_{|k|\le N} \frac{1}{2\pi}\int_{-\pi}^{\pi} f(y)e^{ik(x-y)}\,dy \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \left(\sum_{|k|\le N} e^{ik(x-y)}\right) f(y)\,dy \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\sin(N+\frac{1}{2})(x-y)}{\sin(\frac{1}{2}(x-y))} f(y)\,dy \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\sin(N+\frac{1}{2})t}{\sin\frac{t}{2}} f(x+t)\,dt \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} D_N(t) f(x+t)\,dt
\end{aligned}
$$

Here, we have used $D_N(x) := \sum_{|k|\le N} e^{ikx} = \frac{\sin(N+1/2)x}{\sin(x/2)}$. [§] Using $\int_0^\pi D_N(x)dx = \pi$, we have

$$
\begin{aligned}
f_N(x) - f(x) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\sin(N+\frac{1}{2})t}{\sin\frac{t}{2}}(f(x+t)-f(x))\,dt \\
&:= \frac{1}{2\pi}\int_{-\pi}^{\pi} \sin((N+\frac{1}{2})t) g(t)\,dt.
\end{aligned}
$$

The function $g(t) := (f(x+t)-f(x))/\sin(t/2) = \int_0^1 f'(x+\alpha t)\,d\alpha \cdot t/\sin(t/2)$ is 2π periodic and in C^∞. We can apply integration-by-part s times to arrive

$$
f_N(x) - f(x) = (N+\frac{1}{2})^{-s}\frac{(-1)^{s/2}}{2\pi}\int_{-\pi}^{\pi} g^{(s)}(t)\sin((N+\frac{1}{2})t)\,dt
$$

for even s. Similar formula for odd s. This completes the proof. □

Remark. The constant C_s, which depends on $\int |g^{(s)}|\,dt$, is in general not big, as compared with the term N^{-s}. Hence, the approximation (3.5) is highly efficient for smooth functions. For example, $N = 20$ is sufficient in many applications. The accuracy property (3.5) is called the *spectral accuracy*.

3.2.2 L^2-Convergence Theory

1. The Fourier transform maps a 2π-periodic function f into its Fourier coefficients $(\hat{f}_k)_{k=-\infty}^{\infty}$. We may view the Fourier transform maps $L^2(\mathbb{T})$ space into ℓ^2 space:

   $$L^2(\mathbb{T}) \to \ell^2(\mathbb{Z}).$$

   The function spaces L^2 and ℓ^2 are defined below.

2. The space $L^2(\mathbb{T})$:

   $$L^2(\mathbb{T}) := \{f \mid f \text{ is } 2\pi\text{-periodic and } \int_{-\pi}^{\pi} |f(x)|^2\,dx < \infty\}$$

   with the inner product

   $$(f,g) := \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)\overline{g(x)}\,dx$$

[§] D_N is called a Dirichlet function.

and L^2-norm: $\|f\| = \sqrt{(f,f)}$.

An important fact is that all L^2-function can be approximated by smooth functions. Such a smooth function can be obtained by convoling f with a smooth function, called mollifier. Let $\rho \in C^\infty(\mathbb{T})$, which is positive in a neighborhood of 0 and is zero elsewhere, and $\int_{\mathbb{T}} \rho(x)\,dx = 1$. Given a function $f \in L^p(\mathbb{T})$, $p \geq 1$, define

$$f_\varepsilon(x) := \frac{1}{\varepsilon}\int \rho\left(\frac{x-y}{\varepsilon}\right) f(y)\,dy$$

Then f_ε is a C^∞ function and $f_\varepsilon \to f$ in L^p. This is called the density theorem. We shall not prove here.

3. The space $\ell^2(\mathbb{Z})$ is defined as

$$\ell^2(\mathbb{Z}) := \{(a_k)_{k=-\infty}^{\infty} \mid \sum_{k=-\infty}^{\infty} |a_k|^2 < \infty\}.$$

with inner product $(a,b) := \sum_k a_k \overline{b_k}$.

4. **Bessel's ineqlaity** It is easy to check that e^{ikx} are orthogonal in L^2. From this, we have for any N,

$$0 \leq (f - f_N, f - f_N) = \|f\|^2 - \sum_{|k|\leq N} |\hat{f}_k|^2.$$

Or equivalently,

$$\boxed{\sum_{|k|\leq N} |\hat{f}_k|^2 \leq \|f\|^2.} \tag{3.6}$$

This is called the Bessel inequality. It says that the Fourier transform maps continuously from $L^2(\mathbb{T})$ to $\ell^2(\mathbb{Z})$.

5. **Isometry property**

Theorem 3.2 (Isometry property)**.** *The Fourier transform is an isometry from $L^2(\mathbb{T})$ to $\ell^2(\mathbb{Z})$:*

$$\boxed{(f,g) = \sum_k \hat{f}_k \overline{\hat{g}_k}.} \tag{3.7}$$

Proof. To show this, we first assume that f is a smooth function. We can apply the convergence theorem for f. This yields

$$\begin{aligned}(f,g) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)\overline{g(x)}\,dx \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \sum_k \hat{f}_k e^{ikx}\overline{g(x)}\,dx\end{aligned}$$

$$= \sum_k \hat{f}_k \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{ikx} \overline{g(x)}\, dx$$
$$= \sum_k \hat{f}_k \overline{\hat{g}_k} = (\bar{f}, \bar{g}).$$

To show this formula is valid for all $f, g \in L^2(\mathbb{T})$, we notice that any function in L^2 can be approximated by smooth functions f_ε, namely, $f_\varepsilon \to f$ in L^2.

Now, the isometry property is valid for f_ε and g:

$$(f_\varepsilon, g) = (\widehat{f}_\varepsilon, \hat{g}). \tag{3.8}$$

As $\varepsilon \to 0$,

$$|(f_\varepsilon - f, g)| \le \|f_\varepsilon - f\| \|g\| \to 0,$$

and

$$|(\widehat{f}_\varepsilon - \hat{f}, \hat{g})| \le \|\widehat{f}_\varepsilon - \hat{f}\| \|\hat{g}\| \le \|f_\varepsilon - f\| \|g\| \to 0.$$

The last inequality is from the Bessel inequality. □

The isometry property says that the Fourier transformation preserves the inner product structure. When $g = f$ in the above isometry property, we obtain the following Parseval identity.

6. **Parseval equality**

Corollary 3.1 (Parseval identity)**.** *For $f \in L^2$, we have*

$$\|f\|^2 = \sum_k |\hat{f}_k|^2.$$

7. **Convergence theorem**

Theorem 3.3 (L^2-convergence theorem)**.** *If $f \in L^2$, then*

$$f_N = \sum_{k=-N}^{N} \hat{f}_k e^{ikx} \to f \text{ in } L^2.$$

Proof. First, the sequence $\{f_N\}$ is a Cauchy sequence in L^2. This follows from $\|f_N - f_M\| = \sum_{N \le |k| < M} |\hat{f}_k|^2$ and the Bessel inequality. Suppose f_N converges to g. We see that

$$(\widehat{f - f_N})_k = \frac{1}{2\pi} \int_{\mathbb{T}} (f - f_N)(x) e^{-ikx}\, dx = 0 \text{ if } |k| \le N.$$

Thus, for each fixed k, taking $N \to \infty$, we get

$$(\widehat{f - g})_k = 0.$$

This holds for any $k \in \mathbb{Z}$. Thus, the Fourier coefficients of $f - g$ are all zeros. From the Parseval identity, we have $f = g$. □

3.2.3 *BV-Convergence Theory

1. **BV functions** A function is called a BV function (function of bounded variation) on an interval (a, b), or function of finite total variation, if for any partition $\pi = \{a = x_0 < x_1 < \cdots < x_n = b\}$,
$$\|f\|_{BV} := \sup_{\pi} \sum_i |f(x_i) - f(x_{i-1})| < \infty.$$

2. **BV-norm** An important property of BV functions is that its singularity can only have jump discontinuities, i.e. at a discontinuity, say x_0, f has both left limit $f(x_0-)$ and right limit $f(x_0+)$.

   Further, any BV function f can be decomposed into $f = f_0 + f_1$, where f_0 is a piecewise constant function andf_1 is absolutely continuous (i.e. f_1 is differentiable and f_1' is integrable). The jump points of f_0 are countable. The BV-norm of f is exactly equal to
$$\|f\|_{BV} = \sum_i |[f(x_i)]| + \int |f_1'(x)|\, dx.$$
   where x_i are the jump points of f (also f_0) and $[f(x_i)] := f(x_i+) - f(x_i-)$ is the jump of f at x_i.

3. **Fourier inversion formula for BV functions**

   Theorem 3.4 (Fourier inversion theorem for BV functions)**.** *If f is in BV, then*
$$\boxed{f_N(x) := \sum_{k=-N}^{N} \hat{f}_k e^{ikx} \to \frac{1}{2}(f(x+) + f(x-)).} \tag{3.9}$$

   Proof. Recall that
$$\begin{aligned} f_N(x) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} D_N(x-y) f(y)\, dy \\ &= \frac{1}{2\pi}\left(\int_{-\pi}^{0} + \int_{0}^{\pi}\right) D_N(t) f(x+t)\, dt \\ &= f_N^-(x) + f_N^+(x). \end{aligned}$$
   Here, $D_N(x) = \sum_{|k|\le N} e^{ikx} = \frac{\sin(N+1/2)x}{\sin(x/2)}$. Using $\int_0^\pi \frac{\sin(N+1/2)x}{\sin(x/2)}\, dx = \pi$, we have
$$\begin{aligned} f_N^+(x) - \frac{1}{2}f(x+) &= \frac{1}{2\pi}\int_0^\pi \frac{\sin(N+\frac{1}{2})t}{\sin\frac{t}{2}}(f(x+t) - f(x))\, dt \\ &:= \frac{1}{2\pi}\int_0^\pi \sin((N+\frac{1}{2})t) g(t)\, dt \end{aligned}$$
   From f being in BV, the function $g(t)$ is in $L^1(0,\pi)$. By the Riemann-Lebesgue lemma, $f_N^+(x) - \frac{1}{2}f(x+) \to 0$ as $N \to \infty$. Similarly, we have $f_N^-(x) - \frac{1}{2}f(x-) \to 0$ as $N \to \infty$. □

3.2.4 *Pointwise estimate and Gibbs phenomenon

In applications, we encounter piecewise smooth functions frequently. In this case, the approximation is not uniform. An overshoot and undershoot always appear across discontinuities. Such a phenomenon is called the *Gibbs phenomenon*. Since a BV function can be decomposed into a piecewise constant function and a smooth function, we concentrate to the case when there is only one discontinuity. A typical example is the function

$$f(x) = \begin{cases} 1 & \text{for } 0 < x < \pi \\ -1 & \text{for } -\pi < x < 0 \end{cases}$$

The corresponding f_N is

$$\begin{aligned} f_N(x) &= \sum_{k=-N}^{N} \hat{f}_k e^{ikx} \\ &= \frac{1}{2\pi} \int_{x-\pi}^{x} \frac{\sin((N+\frac{1}{2})t)}{\sin(t/2)}\, dt - \frac{1}{2\pi} \int_{x}^{x+\pi} \frac{\sin((N+\frac{1}{2})t)}{\sin(t/2)}\, dt \end{aligned}$$

First, we show that we may replace $\frac{1}{2\sin(t/2)}$ by $\frac{1}{t}$ with possible error $o(1/N)$. This is because the function $\frac{1}{t} - \frac{1}{2\sin(t/2)}$ is in C^1 on $[-\pi, \pi]$ and the Riemann-Lebesgue lemma. Thus, we have

$$\begin{aligned} f_N(x) &= \frac{1}{\pi} \int_{x-\pi}^{x} \frac{\sin((N+\frac{1}{2})t)}{t}\, dt - \frac{1}{\pi} \int_{x}^{x+\pi} \frac{\sin((N+\frac{1}{2})t)}{t}\, dt + o(1/N) \\ &= \frac{1}{\pi} \int_{(x-\pi)(N+1/2)}^{x(N+1/2)} \operatorname{sinc}(t)\, dt - \frac{1}{\pi} \int_{x(N+1/2)}^{(x+\pi)(N+1/2)} \operatorname{sinc}(t)\, dt + o(1/N). \end{aligned}$$

Here, the function $\operatorname{sinc}(t) := \sin(t)/t$. It has the following properties:

$$\int_0^\infty \operatorname{sinc}(t)\, dt = \pi/2.$$

For any $z > 0$,

$$\int_z^\infty \operatorname{sinc}(t)\, dt = O\left(\frac{1}{z}\right)$$

To see the latter inequality, we rewrite

$$\int_z^\infty \operatorname{sinc}(t)\, dt = \left(\int_z^{n\pi} + \sum_{k \geq n} \int_{n\pi}^{(n+1)\pi} \right) \operatorname{sinc}(t)\, dt$$

where $n = [z/\pi] + 1$. Notice that the series is an alternating series. Thus, the series is bounded by its leading term, which is of $O(1/z)$. Let us denote the integral $\int_0^z \operatorname{sinc}(t)\, dt$ by $\operatorname{Si}(z)$.

To show that the sequence f_N does not converge uniformly, we pick up $x = z/(N+1/2)$ with $z > 0$. After changing variable, we arrive

$$\begin{aligned} f_N\left(\frac{z}{N+1/2}\right) &= \frac{1}{\pi}\int_{z-(N+1/2)\pi}^{z} \operatorname{sinc}(t)\,dt - \frac{1}{\pi}\int_{z}^{z+(N+1/2)\pi} \operatorname{sinc}(t)\,dt + o(1/N) \\ &= \frac{1}{\pi}\int_{-\infty}^{z} \operatorname{sinc}(t)\,dt - \frac{1}{\pi}\int_{z}^{\infty} \operatorname{sinc}(t)\,dt + O(1/(z+N)) + O(1/(z-N)) \\ &= \frac{2}{\pi}\int_{0}^{z} \operatorname{sinc}(t)\,dt + (1/(z+N)) + O(1/(z-N)) \\ &= 1 - \frac{2}{\pi}\int_{z}^{\infty} \operatorname{sinc}(t)\,dt + (1/(z+N)) + O(1/(z-N)) \end{aligned}$$

In general, for function f with arbitrary jump at 0, we have

$$\begin{aligned} f_N(\frac{z}{(N+1/2)}) &= f(0+) - \frac{[f]}{\pi}\int_{z}^{\infty} \operatorname{sinc}(t)\,dt + (1/(z+N)) + O(1/(z-N)) \\ &= f(0+) + O(1/z) + O(1/(z-N)). \end{aligned}$$

where, the jump $[f] := f(0+) - f(0-)$.

We see that the rate of convergence is slow if $z = N^\alpha$ with $0 < \alpha < 1$. This means that if the distance of x and the nearest discontinuity is $N^{-1+\alpha}$, then the convergent rate at x is only $O(N^{-\alpha})$. If the distance is $O(1)$, then the convergent rate is $O(N^{-1})$. This shows that the convergence is not uniform.

The maximum of $\mathrm{Si}(z)$ indeed occurs at $z = \pi$ where

$$\frac{1}{\pi}\mathrm{Si}(\pi) \approx 0.58949$$

This yields

$$f_N(\frac{\pi}{N+1/2}) = f(0+) + 0.08949\,(f(0+) - f(0-)).$$

Hence, there is about 9% overshoot. This is called the Gibbs phenomenon.

```
% Define the step function
N = 100;                % Number of points
x = linspace(-pi,pi,N); % Domain
f = sign(x);            % Step function

% Compute the FFT
F = fft(f);

% Add more terms to the FFT to simulate the Gibbs phenomenon
Nterms = 10;            % Number of additional terms
F(N/2+1-Nterms:N/2+1+Nterms) = F(N/2+1-Nterms:N/2+1+Nterms) .* linspace
   (0,0,Nterms*2+1);
```

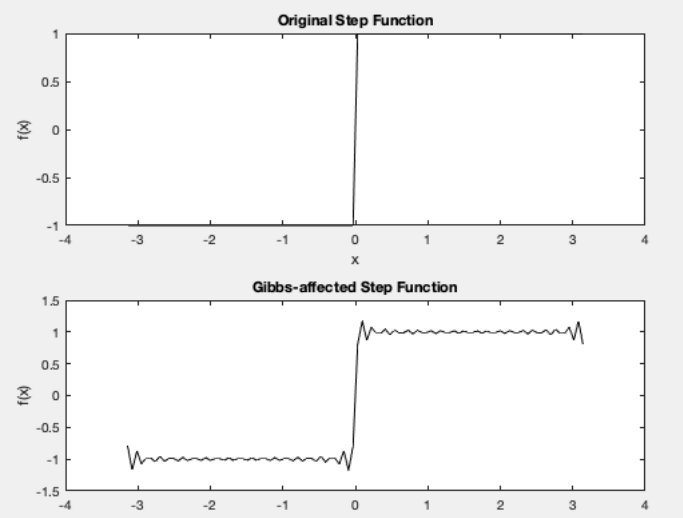


Figure 3.1: Gibbs phenomenon. The original Fourier modes are $\hat{f}_k$, $k = -49, ..., 50$. We reset the last 20 Fourier modes to be 0's. That is, we approximate f by f_{40}.

```
% Compute the inverse FFT
f_gibbs = real(ifft(F));

% Plot the original and Gibbs-affected functions
figure
subplot(2,1,1)
plot(x,f,'k-')
title('Original Step Function')
xlabel('x')
ylabel('f(x)')

subplot(2,1,2)
plot(x,f_gibbs,'k-')
title('Gibbs-affected Step Function')
xlabel('x')
ylabel('f(x)')
```

Homeworks

1. Derive the Fourier expansion formula for periodic functions with period L.

2. What is the limit of the above Fourier expansion formula as $L \to \infty$.

3. Derive the Fourier expansion for the following functions: $f(x) = |x| - 1/2$ for $|x| \leq 1$ and f is a periodic function with period 2.

4. What is the convergence rate of the above function in L^2 and pointwise convergence rate at $x = 0$?

3.2.5 Fourier expansion of real-valued functions

1. **Fourier expansion for real-valued functions**

   Proposition 3.1. *When f is real-valued, f can be expressed as*

   $$f(x) \sim \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx)$$

   with

   $$a_n = \frac{1}{2\pi} \int_0^{2\pi} f(x) \cos nx \, dx, \;\; b_n = \frac{1}{2\pi} \int_0^{2\pi} f(x) \sin nx \, dx.$$

   Proof. We have

   $$\hat{f}_n = \frac{1}{2\pi} \int_{\mathbb{T}} f(x) e^{-inx} \, dx, \quad \hat{f}_{-n} = \frac{1}{2\pi} \int f(x) e^{inx} \, dx.$$

   Thus, when f is a real-valued function,

   $$\hat{f}_n = \overline{\hat{f}_{-n}}.$$

   If we express $\hat{f}_n = \frac{1}{2}(a_n - ib_n)$, where $a_n, b_n \in \mathbb{R}$, then $\hat{f}_{-n} = \frac{1}{2}(a_n + ib_n)$ and

   $$\begin{aligned} f(x) &= \sum_{n \in \mathbb{Z}} \hat{f}_n e^{inx} \\ &= \frac{a_0}{2} + \frac{1}{2} \sum_{n=1}^{\infty} (a_n - ib_n) e^{inx} + \frac{1}{2} \sum_{n=1}^{\infty} (a_n + ib_n) e^{-inx} \\ &= \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx) \end{aligned}$$

   Here,

   $$\begin{aligned} \frac{1}{2}(a_n - ib_n) &= \frac{1}{2\pi} \int_{\mathbb{T}} f(x) e^{-inx} \, dx \\ &= \frac{1}{2\pi} \int_{\mathbb{T}} f(x) \left(\cos nx - i \sin nx \right) \, dx. \end{aligned}$$

   Thus,

   $$a_n = \frac{1}{2\pi} \int_0^{2\pi} f(x) \cos nx \, dx, \;\; b_n = \frac{1}{2\pi} \int_0^{2\pi} f(x) \sin nx \, dx.$$

   □

2. **Orthogonality** The functions $\{\cos nx, \sin nx\}$ are orthogonal to each other. But

$$\frac{1}{2\pi}\int_0^{2\pi} \cos^2 nx\, dx = \frac{1}{2\pi}\int_0^{2\pi} \sin^2 nx\, dx = \frac{1}{2} \text{ for all } n.$$

3. **Parseval equility** The Parseval equality reads

$$\frac{1}{2\pi}\int_{\mathbb{T}} |f(x)|^2\, dx = 2\sum_n \left(a_n^2 + b_n^2\right).$$

3.3 Discrete Fourier Transform

3.3.1 Definition and the inversion formula

1. **Definition of the discrete Fourier transform** Given a 2π-periodic function f, we define

$$\tilde{f}_k := \frac{1}{N}\sum_{j=0}^{N-1} f(x_j)e^{-ikx_j} \tag{3.10}$$

to be the discrete Fourier transform of f. This definition is the numerical integral formula for the Fourier transform

$$\frac{1}{2\pi}\int_{\mathbb{T}} f(x)e^{-ikx}\, dx \approx \sum_{j=0}^{n} f(x_j)w_j$$

with quadrature points $x_j = 2\pi j/N$ and weights $w_j := 2\pi/N$.

2. This is exactly the trapezoidal approximation for numerical integration of the Fourier modes:

$$\begin{aligned}\frac{1}{2\pi}\int_0^{2\pi} f(x)e^{-ikx}\, dx &= \frac{1}{2\pi}\sum_{j=0}^{N-1}\int_{x_j}^{x_{j+1}} f(x)e^{-ikx}\, dx \\ &\approx \frac{1}{2\pi}\sum_{j=0}^{N-1}\frac{1}{2}\left(f_j e^{-ikx_j} + f_{j+1}e^{-ikx_{j+1}}\right)\frac{2\pi}{N} \\ &= \frac{1}{N}\sum_{j=0}^{N-1} f_j e^{-ikx_j}.\end{aligned}$$

Here, $f_j := f(x_j)$. When $f \in C^\infty$, according to the Euler-MacLaurin summation formula for periodic functions,

$$\left|\frac{1}{2\pi}\int_0^{2\pi} f(x)e^{-ikx}\, dx - \frac{1}{N}\sum_{j=0}^{N-1} f_j e^{-ikx_j}\right| = O(N^{-s})$$

for any $s > 0$. Thus, the discrete Fourier modes can approximate Fourier modes with *spectral accuracy*, provided the underlying function is C^∞.

3. N-periodicity of $\tilde{f}_k$:
$$\tilde{f}_{k+N} = \tilde{f}_k$$

4. **Inversion formula** . We shall always choose N to be even. From the periodicity of $\tilde{f}_k$, we define
$$I_N f(x) := \sum_{k=-N/2}^{N/2-1} \tilde{f}_k e^{ikx}.$$
We claim that
$$I_N f(x_j) = f(x_j) \text{ for } j = 0, ..., N-1.$$
In other words, $I_N f$ is a trigonometric interpolant of f at $\{x_j\}_{j=0}^{N-1}$. To see this, we plug the formula for $\tilde{f}_k$ into the formula for f_N:
$$\begin{aligned} I_N f(x) &= \sum_{k=-N/2}^{N/2-1} \frac{1}{N} \sum_{k=0}^{N-1} f_j e^{ik(x-x_j)} \\ &= \sum_{j=0}^{N-1} D_N(x-x_j) f_j \end{aligned}$$
where
$$D_N(x) = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} e^{ikx} = e^{-ix/2} \frac{\sin(Nx/2)}{N \sin(x/2)}.$$
This function is a 2π-periodic function. For $x \sim 0$,
$$\lim_{x \to 0} D_N(x) = 1.$$
For $x_j = 2\pi j/N$, $j \neq 0$, $\sin(Nx_j/2) = 0$. Thus, we get
$$D_N(x_j) = \begin{cases} 1 & \text{for } j \equiv 0 \ (\text{ mod } N) \\ 0 & \text{for } j \not\equiv 0 \ (\text{ mod } N) \end{cases}$$
Hence, $I_N f(x_j) = f_j$.

5. **Problem of this discrete Dirichlet kernel.** First, it is not real-valued. Second, its derivative produces an oscillatory term due to e^{ix}. The problem comes from the asymmetric summation in $\sum_{N/2}^{N/2-1} e^{ikx}$. To fix this problem, we can just take its real part, namely,
$$\boxed{I_N f(x) := \sum_{j=0}^{N-1} f(x_j) \varphi_N(x - x_j)} \tag{3.11}$$
where
$$\boxed{\varphi_N(x) = Re\,(D_N(x)) = \cot(x/2) \frac{\sin(Nx/2)}{N}} \tag{3.12}$$

is a real-valued 2π-periodic function. It is called the *periodic sinc function*. It satisfies

$$\varphi_N(x_j) = \delta_{0,j} \tag{3.13}$$

for $j = 0, 1, ..., N-1$. It is a trigonometric polynomial interpolating $\{z_j = e^{ix_j}\}$. Its derivative is

$$\varphi_N'(x_j) = \delta_{0,j} \frac{1}{2} (-1)^j \cot(jh/2)$$

where $h = 2\pi/N$ is the mesh size. Alternatively, we can use the property $\tilde{f}_{N/2} = \tilde{f}_{-N/2}$, define

$$I_N f(x) = \sum_{k=-N.2}^{N/2} \frac{\tilde{f}_k}{c_k} e^{ikx}$$

where

$$c_k = \begin{cases} 2 & \text{if } k = \pm N/2 \\ 1 & \text{otherwise} \end{cases}$$

This gives

$$I_N f(x) = \sum_{j=0}^{N-1} f(x_j) \varphi_N(x - x_j).$$

In other words, $I_N f$ is the *trigonometric polynomial interpolating f at $\{x_j\}$*.

6. **Isometry property** Let S_N be the space of the trigonometric polynomial of degree $N/2$:

$$S_N := \text{span}\{E_k(x) = e^{ikx} \ \mid \ -N/2 \le k < N/2\}.$$

In this space, the inner product defined by

$$(f, g) := \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) \overline{g(x)}\, dx,$$

is equivalent to the discrete inner product

$$(f, g)_N := \frac{1}{N} \sum_{j=0}^{N-1} f_j \bar{g}_j.$$

It is easy to check that $\{E_k(x)\}_{-N/2 \le k < N/2}$ are orthonormal in both inner products. Hence, these two inner products are identical any $f, g \in S_N$.

Again, from orthonormality of $\{E_k(x)\}$, we have the isometry property:

$$\boxed{(f, g)_N = \sum_{-N/2 \le k < N/2} \tilde{f}_k \overline{\tilde{g}_k},}$$

and the Parseval identity:

$$\frac{1}{N} \sum_{j=0}^{N} |f_j|^2 = \sum_{-N/2 \le k < N/2} |\tilde{f}_k|^2.$$

3.3.2 Approximation issues

1. **Modal projection vs. Nodal projection** Given a 2π-periodic function f, the mapping
$$P_N f(x) := \sum_{-N/2 \le k < N/2} \hat{f}_k e^{ikx}$$
is an orthogonal projection from $L^2(-\pi, \pi)$ to S_N. On the other hand, the interpolation operator $I_N f$:
$$I_N f(x) := \sum_{j=0}^{N-1} f(x_j)\varphi_N(x - x_j) = \sum_{-N/2 \le k < N/2} \tilde{f}_k e^{ikx}$$
is a projection onto S_N, and is characterized by $I_N f(x_j) = f(x_j)$, $j = 0, \cdots, N-1$. P_N is expressed in terms of Fourier mode $\hat{f}_k$, and is called the *modal projection*. While I_N, as expressed in terms of node data $f(x_j)$, and is called the *nodal projection*.

2. **Aliasing error** The difference between P_N and I_N is called "aliasing error." It can be characteristized as the follows. First,
$$\begin{aligned}
\tilde{f}_k &= \frac{1}{N}\sum_{j=1}^{N-1} f(x_j)e^{-ikx_j} = \frac{1}{N}\sum_{j=1}^{N-1} \sum_{-\infty<\ell<\infty} \hat{f}_\ell e^{i(\ell-k)x_j} \\
&= \sum_{-\infty<\ell<\infty} \hat{f}_\ell \frac{1}{N}\sum_{j=0}^{N-1} e^{i(\ell-k)x_j} = \sum_{-\infty<\ell<\infty} \hat{f}_\ell D_N(x_\ell - x_k) \\
&= \sum_{-\infty<m<\infty} \hat{f}_{k+mN} = \hat{f}_k + \sum_{\substack{-\infty < m < \infty \\ m \neq 0}} \hat{f}_{k+mN}
\end{aligned}$$
From the orthogonality of E_k in L^2, we see that
$$R_N f := I_N f - P_N f = \sum_{-N/2 \le k < N/2} \left[\sum_{\substack{-\infty < m < \infty \\ m \neq 0}} \hat{f}_{k+mN} \right] E_k$$
and
$$\begin{aligned}
\|R_N f\|^2 &= \sum_{-N/2 \le k < N/2} |\tilde{f}_k - \hat{f}_k|^2 \\
&\le \sum_{-N/2 \le k < N/2} \sum_{\substack{-\infty < m < \infty \\ m \neq 0}} |\hat{f}_{k+mN}|^2 \\
&= \sum_{k \ge N/2, k < -N/2} |\hat{f}_k|^2.
\end{aligned}$$

3. Since P_N is an orthogonal projection, we have

$$\|f - I_N f\|^2 = \|f - P_N f\|^2 + \|R_N f\|^2.$$

4. **Sobolev Spaces** It is not difficult to find the approximation error for P_N. Indeed, let H^s denote the Sobolev space of order s:

$$H^s := \{f \text{ is } 2\pi\text{-periodic, and } f, \cdots, f^{(s)} \in L^2\}$$

with the norm $\|f\|_{H^s}^2 := \sum_{m=0}^{s} \|f^{(m)}\|^2$. From the Parseval identity, this norm is equivalent to $\sum_k (1 + |k|^2)^s |\hat{f}_k|^2$. **Projection error** We have the following approximation theorem.

Theorem 3.5. *If $f \in H^s$, then*

$$\|f - P_N f\| \leq C N^{-s} \|f^{(s)}\|$$

Proof. We use the facts that $\{E_k\}_{k \in Z}$ is a basis in L^2 and the Parseval identity:

$$\begin{aligned} \|f - P_N f\|^2 &= \sum_{|k| \geq N/2} |\hat{f}_k|^2 \\ &= \sum_{|k| \geq N/2} |k|^{-2s} |k|^{2s} |\hat{f}_k|^2 \\ &\leq O(N^{-2s}) \|f^{(s)}\|^2. \end{aligned}$$

□

5. **Interpolation error** For the interpolation operator, we have similar result. In other words, the aliasing error has the same spectral error as that of the truncated Fourier polynomial for smooth functions. This follows from

$$\|R_N f\|^2 \leq \sum_{k \geq N/2, k \leq -N/2} |\hat{f}_k|^2.$$

Thus, we have proved the following theorem (Kreiss and Oliger). We refer its detailed proof to (p.280**??**).

Theorem 3.6. *If $f \in H^s$, $s \geq 1$, then there is a constant C_s such that*

$$\|f - I_N f\| \leq C_s N^{-s} \|f^{(s)}\|.$$

Corollary 3.2. *If $f \in C^\infty$, then for any $s > 0$, there exists constant C_s and C_s' such that*

$$\|f - P_N f\| \leq C_s N^{-s}, \quad \|f - I_N f\| \leq C_s' N^{-s}$$

This is called *spectral accuracy.*

3.4 Fast Fourier Transform

Spectral methods become practical due to the birth of fast Furier transform which reduces the operation counts from $O(N^2)$ to $N \ln N$. We explain Cooley-Tukey's fast algorithm below.

3.4.1 The FFT algorithm

1. **Fourier transform matrix** Recall that both f and $\tilde{f}$ are periodic, and the transform can be rewritten as

$$\begin{aligned} \tilde{f}_k &= \frac{1}{N}\sum_{j=0}^{N-1} f_j e^{-ikx_j}, k = 0, \cdots, N-1 \\ f_j &= \sum_{k=0}^{N-1} \tilde{f}_k e^{ikx_j}, j = 0, \cdots, N-1. \end{aligned}$$

   The transformation matrix $\mathcal{F}_N$ is

$$\mathcal{F}_N = \left(\omega_N^{ij}\right)_{\substack{0 \le i < N \\ 0 \le j < N}} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \omega & \omega^2 & \cdots & \omega^{N-1} \\ \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^{(N-1)} & \omega^{(N-1)2} & \cdots & \omega^{(N-1)^2} \end{bmatrix}_{N\times N}. \tag{3.14}$$

$$\omega = \omega_N = e^{-2\pi\sqrt{-1}/N}.$$

   Note that

$$\bar{\mathcal{F}}_N \mathcal{F}_N = N I_{N\times N}$$

   For simplicity, below, let us call $\tilde{f} = \mathcal{F}_N f$ instead of $\tilde{f} = \frac{1}{N}\mathcal{F}_N f$.

2. **Splitting of the transformation matrix** Let us suppose N is even, say $N = 2M$. Then we have

$$\begin{aligned} \tilde{f}_k &= \sum_{j=0}^{N-1} \omega_N^{kj} f_j \\ &= \sum_{j=0}^{M-1} \omega_N^{k2j} f_{2j} + \sum_{j=0}^{M-1} \omega_N^{k(2j+1)} f_{2j+1} \end{aligned}$$

   We define $f' = (f_0, f_2, \cdots, f_{2M-2})$, $f'' = (f_1, f_3, \cdots, f_{2M-1})$.

- For $0 \le k < M := N/2$, we have

$$\begin{aligned} \tilde{f}_k &= \sum_{j=0}^{M-1} \omega_M^{kj} f_{2j} + \omega_N^k \sum_{j=0}^{M-1} \omega_M^{kj} f_{2j+1} \\ &= (\mathcal{F}_M f')_k + \omega_N^k (\mathcal{F}_M f'')_k \end{aligned}$$

  Here, we have used

$$\omega_N^2 := \omega_{2M}^2 = \omega_M.$$

- For $\tilde{f}_{M+k}$, $k = 0, ..., M-1$, we have

$$\begin{aligned} \tilde{f}_{k+M} &= \sum_{j=0}^{M-1} \omega_{2M}^{(M+k)2j} f_{2j} + \sum_{j=0}^{M-1} \omega_{2M}^{(M+k)(2j+1)} f_{2j+1} \\ &= \sum_{j=0}^{M-1} \omega_M^{kj} f_{2j} + \sum_{j=0}^{M-1} \omega_{2M}^{M(2j+1)} \omega_{2M}^k \omega_M^{kj} f_{2j+1} \\ &= \sum_{j=0}^{M-1} \omega_M^{kj} f_{2j} - \omega_{2M}^k \sum_{j=0}^{M-1} \omega_M^{kj} f_{2j+1} \\ &= (\mathcal{F}_M f')_k - \omega_{2M}^k (\mathcal{F}_M f'')_k \end{aligned}$$

  Here, we have used

$$\omega_{2M}^{2M} = 1, \quad \omega_{2M}^M = -1.$$

Thus, the discrete Fourier transform can be calculated as the follows.

(a) Split $f = (f_0, \cdots, f_{N-1})$ into even and old parts:

$$f' = (f_0, f_2, \cdot, f_{N-2}),\ f'' = (f_1, f_3, \cdots, f_{N-1})$$

(b) Let $M = N/2$, perform

$$\tilde{f}' = \mathcal{F}_M f',\ \tilde{f}'' = \mathcal{F}_M f''$$

(c) For $0 \le k < M$, compute

$$\begin{aligned} \tilde{f}_k &= \tilde{f}'_k + \omega_N^k \tilde{f}''_k \\ \tilde{f}_{k+M} &= \tilde{f}'_k - \omega_N^k \tilde{f}''_k \end{aligned}$$

3. **Splitting in matrix form** In matrix form, $\mathcal{F}_N$ can be splitted into

$$\mathcal{F}_N = Q_N \begin{bmatrix} \mathcal{F}_{N/2} & 0 \\ 0 & \mathcal{F}_{N/2} \end{bmatrix} P_N \tag{3.15}$$

Here, $\mathcal{P}_N$ is a permutation matrix which maps

$$(f_0, f_1, \cdots, f_{N-1})^t \mapsto (f_0, f_2, \cdots, f_{N-2}, f_1, f_3, \cdots, f_{N-1})^t;$$

the matrix Q_N is defined as

$$Q_N = \begin{bmatrix} I_{N/2} & D_{N/2} \\ I_{N/2} & -D_{N/2} \end{bmatrix}, \ I: \text{ identity matrix}, D_{N/2} = \text{diag}(1, \omega, \cdots, \omega^{N/2-1})$$

4. **Computation complexity** Notice that both P_N and Q_N are sparse matrices. The amount of work to perform P_N and Q_N is $O(N)$. Let the operation count to perform P_N and Q_N be CN. Suppose $N = 2^L$. Let C_{2^L} be the operation counts to perform $\mathcal{F}_{2^L}$. Then we have

$$C_{2^L} = C2^L + 2\, C_{2^{L-1}}.$$

This yields

$$\begin{aligned} C_N = C_{2^L} &= C2^L + 2C_{2^{L-1}} = C2^L + 2\left(C2^{L-1} + 2C_{2^{L-2}}\right) \\ &= C\left(2^L + 2\cdot 2^{L-1} + 2^2 2^{L-2} + \cdots + 2^{L-1}2\right) = CL2^L = CN\ln N. \end{aligned}$$

Thus, the computational complexity of FFT is $N \ln N$. Since $\ln N$ is relatively small, we call such complexity is *essentially linear.*

3.4.2 Variations of FFT

Trigonometric representation

When all $f_j \in \mathbb{R}$, then $\tilde{f}_k = \tilde{f}_{-k} = \tilde{f}_{N-k}$, for $k = 1, N/2$. Let

$$M = \begin{cases} N/2 & \text{for even } N \\ (N+1)/2 & \text{for odd } N \end{cases}$$

$\tilde{f}_k = c_{2k-1} - ic_{2k}$, $k = 1, \cdots, M-1$, and $c_0 = \tilde{f}_0$, and $c_{N-1} = f_{N/2}$. Then

$$\begin{aligned} f_j &= \tilde{f}_0 + (-1)^j f_{N/2} + \sum_{k=1}^{N/2-1} (\tilde{f}_k e^{ikx_j} + \overline{\tilde{f}_k} e^{-ikx_j}) \\ &= c_0 + (-1)^j c_{N-1} + 2\sum_{k=1}^{M} c_{2k-1}\cos(kx_j) + c_{2k}\sin(kx_j) \end{aligned}$$

and

$$\begin{aligned} c_0 &= \frac{1}{N}\sum_{j=0}^{N-1} f_j \\ c_{2k-1} &= \frac{1}{N}\sum_{j=0}^{N-1} f_j \cos(kx_j), k = 1, \cdots, N/2-1 \\ c_{2k} &= \frac{1}{N}\sum_{j=0}^{N-1} f_j \sin(kx_j), k = 1, \cdots, N/2 \end{aligned}$$

$$c_{N-1} = \frac{1}{N}\sum_{j=0}^{N-1}(-1)^j f_j$$

Fourier Cosine Transform

When f_j is an even sequence, i.e. $f_{N-j} = f_j$, $j = 1, \cdots, N/2$, then for $k = 0, \cdots N/2-1$,

$$\begin{aligned}\tilde{f}_k &= \frac{1}{N}\sum_{j=-N/2}^{N/2-1} f_j e^{-ikx_j} \\ &= \frac{1}{N}\left[f_0 + (-1)^k f_{N/2} + \sum_{j=1}^{N/2-1} 2f_j \cos(kx_j)\right]\end{aligned}$$

Its inverse transform is

$$\begin{aligned}f_j &= \sum_{k=-N/2}^{N/2-1} \tilde{f}_k e^{ikx_j} \\ &= f_0 + (-1)^j \tilde{f}_{N/2} + \sum_{k=1}^{N/2-1} 2f_k \cos(kx_j)\end{aligned}$$

Fourier Sine Transform

When f_j is an odd sequence, i.e. $f_{N-j} = -f_j$, $j = 0, \cdots, N/2$, then for $k = 1, \cdots N/2-1$,

$$\begin{aligned}\tilde{f}_k &= \frac{1}{N}\sum_{j=-N/2}^{N/2-1} f_j e^{-ikx_j} \\ &= \frac{1}{N}\sum_{j=1}^{N/2-1} 2f_j \sin(kx_j)\end{aligned}$$

Its inverse transform is, for $j = 1, \cdots, N/2-1$,

$$\begin{aligned}f_j &= \sum_{k=-N/2}^{N/2-1} \tilde{f}_k e^{ikx_j} \\ &= \sum_{k=1}^{N/2-1} 2f_k \sin(kx_j)\end{aligned}$$

3.4.3 List of matlab commands regarding FFT

You can access the malab functions.

fft	Fast Fourier transform
fft2	2-D fast Fourier transform
fftn	N-D fast Fourier transform
nufft	Nonuniform fast Fourier transform
nufftn	N-D nonuniform fast Fourier transform
fftshift	Shift zero-frequency component to center of spectrum
fftw	Define method for determining FFT algorithm
ifft	Inverse fast Fourier transform
ifft2	2-D inverse fast Fourier transform
ifftn	Multidimensional inverse fast Fourier transform
ifftshift	Inverse zero-frequency shift
nextpow2	Exponent of next higher power of 2
interpft	1-D interpolation (FFT method)

3.4.4 Solving the heat equation on torus

1. Representation of d/dx in modal representation.
   The matrix representation of d/dx under the basis $\{E_k\}_{-N/2}^{N/2-1}$ is

   $$D = \text{diag}(i(-N/2), ..., i(N/2-1)).$$

2. Representation of d^2/dx^2 in modal representation.

   $$D^2 = \text{diag}(-(-N/2)^2, ..., -(N/2-1)^2).$$

3. Representation of d/dx in nodal representation.

4. Representation of d^2/dx^2 in nodal representation.

5. Solving heat equation: $u_t = \triangle u$ on $\mathbb{T}$. Suppose the modal representation of the an approximate solution is $u_N(x,t) = \sum_{k=-N/2}^{N/2-1} \tilde{U}_k(t) E_k(x)$. Plug this into the heat equation. Using the representation of d^2/dx^2 under the basis $\{E_k\}$. We get that

   $$\dot{\tilde{U}}_k = -k^2 \tilde{U}_k.$$

   This gives

   $$\tilde{U}_k(t) = e^{-k^2 t} \tilde{U}_k(0) = e^{-k^2 t} \langle u(0), E_k \rangle$$

   The approximate solution u_N is given by

   $$u_N(x,t) = \sum_{k=-N/2}^{N/2-1} e^{-k^2 t} \tilde{U}_k(0)$$

6. For reaction-diffusion equation:

$$u_t = \triangle u + f(u) \quad \text{on } \mathbb{T},$$

we can use the splitting method. The part $u_t = \triangle u$ can be solved exactly by modal approximation. The part $u_t = f(u)$ can be solved using nodal representation. Thus, there are two representation in the calculation:

$$U_N(x,t) = \sum_{k=-N/2}^{N/2-1} \tilde{U}_k(t) E_k(x) = \sum_{j=0}^{N-1} U_j \phi(x - x_j).$$

The transformation $(U_0, ..., U_{N-1}) \mapsto (\tilde{U}_{-N/2}, ..., \tilde{U}_{N/2-1})$ is the discrete Fourier transform.

References

- Spectral method Codes
- Parallel Codes in python and matlab

3.5 Orthogonal Polynomials and the corresponding discrete transform

3.5.1 Orthogonal polynomials

The Fourier series is one example of orthogonal polynomials, which is for the periodic domain $\mathbb{T}$. There is a general theory for orthogonal polynomial on more general domain Ω with weight w. The weight w defines the inner product

$$\langle f, g \rangle = \int_\Omega f(x) \overline{g(x)} w(x)\, dx.$$

Here is a list of standard orthogonal polynomials:

- $w(x) = 1$ on $[-1, 1]$: the Legendre polynomials P_k
- $w(x) = 1/\sqrt{1-x^2}$ on $[-1, 1]$: the Chebyshev polynomials T_k
- $w(x) = x^\alpha e^{-x}$ on $[0, \infty)$: the Laguerre polynomials L_k^α
- $w(x) = e^{-x^2}$ on $(-\infty, \infty)$: the Hermite polynomials H_k
- $w(x) = (1-x)^\alpha (1+x)^\beta$ on $(-1, 1)$ with $\alpha, \beta > -1$: the Jacobi polynomials $P_k^{(\alpha,\beta)}$

We shall only study the case of Legendre polynomials. Ref. Numerical Recipes, Sec. 4.5. Gaussian quadrature and orthogonal polynomials.

3.5.2 Gaussian quadrature for numerical integration

To find mode presentation of a function f, we need to perform numerical integration. For instance

$$\hat{f}_k = \int_0^{2\pi} f(x)e^{-ikx}\,dx \approx \tilde{f}_k := \frac{1}{N}\sum_{j=0}^{N} f(x_j)e^{-ikx_j}.$$

Similarly, to find the Legendre mode $\hat{f}_k$, we approximate

$$\hat{f}_k := \frac{2k+1}{2}\int_{-1}^{1} f(x)P_k(x)\,dx$$

by numerical integration:

$$\tilde{f}_k := \frac{2k+1}{2}\sum_{j=0}^{N} f(x_j)P_k(x_j)w_j.$$

In this subsection, we study the numerical integration methods for the weighted integral

$$\int_a^b f(x)w(x)\,dx,$$

where $w > 0$ is the weight. We want to approximate it by:

$$\int_a^b f(x)w(x)\,dx \approx \sum_{i=1}^{n} f(x_i)w_i, \tag{3.16}$$

where $\{x_i | i = 1, ..., n\}$ are called the quadrature points, $\{w_i | i = 1, ..., n\}$ the corresponding weights.

1. **Determine the weights**. Let

   $$\Pi_n := \{p \text{ is a polynomial with } \deg(p) \le n\}.$$

   We want (3.16) to be exact for $f \in \Pi_{n-1}$. This means that this formula is exact for $f(x) = 1, x, ..., x^{n-1}$. Suppose $\{x_i\}_{i=1}^n$ are given, then the above "exact" condition involves n equations for n knowns $w_1, ..., w_n$. In particular, the integration should be exact for the Lagrange interpolants

   $$\ell_i(x) := \frac{\prod_{j\neq i}(x - x_j)}{\prod_{j\neq i}(x_i - x_j)} \in \Pi_{n-1}.$$

   That is,

   $$\int_a^b \ell_i(x)w(x)\,dx = \sum_{k=1}^{n} \ell_i(x_j)w_j \text{ for } i = 1, ..., N.$$

   From

   $$\ell_i(x_j) = \delta_{ij}, \quad 1 \le i, j \le N,$$

we get a formula for w_i:

$$w_i = \sum_{k=1}^{n} \ell_i(x_j) w_j = \int_a^b \ell_i(x) w(x)\, dx. \tag{3.17}$$

2. **Determine the quadrature points** Next, we can choose the quadrature points so that (3.16) is exact for $f \in \Pi_{2n-1}$. This involves $2n$ conditions for $2n$ knowns $\{w_1, ..., w_n\}$ and $\{x_1, ..., x_n\}$. We introduce a theory of orthogonal polynomials to determine the quadrature points. The orthogonal polynomials are w-orthogonal under the weighted inner product

$$\langle f, g \rangle := \int_a^b f(x) g(x) w(x)\, dx.$$

We have the following theorem. See Kincaid and Chaney, pp. 311, pp. 457.

Theorem 3.7. *Let $w > 0$ in $[a, b]$ be a positive weight.*

(i) Suppose $q \in \Pi_n$ is w-orthogonal to Π_{n-1}. If $\{x_1, ..., x_n\}$ are the zeros of q, then the numerical integration formulae (3.16) (3.17) are exact for $f \in \Pi_{2n-1}$.

(ii) For general $f \in C^{2n}[a, b]$, the error of the numerical integration formula (3.16) is

$$\frac{f^{(2n)}(\xi)}{(2n)!} \int_a^b q^2(x) w(x)\, dx, \text{ for some } \xi \in [a, b].$$

Proof. (i) If $f \in \Pi_{2n-1}$, from $q \in \Pi_n$, we can write

$$f = qp + r$$

with $p, r \in \Pi_{n-1}$. This gives

$$f(x_i) = r(x_i) \text{ for } i = 1, ..., n, \quad \because q(x_i) = 0.$$

Note that $\int_a^b p(x) q(x) w(x)\, dx = 0$ for any $p \in \Pi_{n-1}$ because q is w-orthogonal to Π_{n-1} by our assumption. Using these two properties, we have

$$\begin{aligned} \int_a^b f(x) w(x)\, dx &= \int_a^b \left(q(x)p(x) + r(x)\right) w(x)\, dx = \int_a^b r(x) w(x)\, dx \\ &= \sum_{i=0}^{n} r(x_i) w_i = \sum_{i=0}^{n} f(x_i) w_i \end{aligned}$$

(ii) Given $f \in C^{2n}[a, b]$. Let $p \in \Pi_{2n-1}$ such that ¶

$$p(x_i) = f(x_i), \quad p'(x_i) = f'(x_i), \quad i = 1, ..., n,$$

¶Such p is called the Hermite interpolant of f at $x_1, ..., x_N$. See Kincaid and Chaney, Sec. 6.3.

$$q(t) := \prod_{j=1}^{n}(t - x_j), \quad \phi = f - p - \lambda q^2.$$

With a prescribed x, the parameter λ is chosen so that $\phi(x) = 0$. That is,

$$\lambda := \frac{f(x) - p(x)}{q(x)^2}.$$

Then ϕ has $n + 1$ zeros at $x, x_1, ..., x_n$ in $[a, b]$. By Roll's Theorem, ϕ' has at least n zeros different from $\{x, x_1, ..., x_n\}$. On the other hand, $\phi'(x_i) = 0$, $i = 1, ..., n$ from our construction. Thus, ϕ' has at least $2n$ distinct zeros in (a, b). Using the same argument, we get ϕ'' has at least $2n-1$ distinct zeros in (a, b), and so on. By repeating this argument, $\phi^{(2n)}$ has a zero $\xi \in (a, b)$. We have

$$0 = \phi^{(2n)}(\xi) = f^{(2n)}(\xi) - p^{(2n)}(\xi) - \lambda(q^2)^{(2n)}(\xi).$$

Since $p \in \Pi_{2n-1}$, we get $p^{(2n)} = 0$. The leading term of $q^2(t)$ is t^{2n}. We get $(q^2)^{(2n)}(\xi) = (2n)!$. Thus, we have

$$0 = \phi^{(2n)}(\xi) = f^{(2n)}(\xi) - \lambda(q^2)^{(2n)}(\xi)) = f^{(2n)}(\xi) - \frac{f(x) - p(x)}{q^2(x)}(2n)!.$$

That is,

$$f(x) - p(x) = \frac{f^{(2n)}(\xi)}{(2n)!}q^2(x).$$

Integrating this from a to b with weight w, using the mean value theorem, we get the final result. □

Remarks.

- Quadrature rule: The quadrature nodes are chosen to be the zeros of $q \in \Pi_n$ which is w-orthogonal to Π_{n-1}.
- The above roots of q are in (a, b) and distinct. (See Kincaid and Channey, pp. 457.)

3.6 Legendre polynomials and Legendre Transform

3.6.1 Legendre polynomials

Legendre polynomials are polynomials which are orthogonal in $L^2(-1, 1)$. While trigonometric polynomials are useful basis for functions on periodic domains, Legendre polynomials are the useful orthogonal polynomials on bounded domains. Reference: Legendre Polynomials, wiki

1. The Legendre polynomials P_n, $n = 0, 1, ...$ are defined as:

   - P_n is a polynomial of degree n;
   - P_n is normalized with $P_n(1) = 1$.
   - $\{P_n\}$ are orthogonal in $L^2([-1, 1], dx)$:

$$\int_{-1}^{1} P_n(x)P_m(x)\,dx = 0 \text{ if } m \neq n. \tag{3.18}$$

2. **Recursion formula** The polynomials P_n can be obtained from the Gram-Schmidt orthogonalization of the monomials $\{1, x, x^2, \cdots\}$. By definition, $P_{n+1} \perp \Pi_n$. We can use this to derive a 3-step recursion formula. Suppose we want to go from P_n to P_{n+1}. We consider $xP_n \in \Pi_{n+1}$. Note that $xP_n \perp \Pi_{n-2}$ because for any $p \in \Pi_{n-2}$,

$$\int_{-1}^{1} xP_n(x)p(x)\,dx = \int_{-1}^{1} P_n(x)(xp(x))\,dx = 0 \quad \because xp \in \Pi_{n-1}.$$

   Thus, xP_n can be expressed as a linear combination of P_{n+1}, P_n and P_{n-1}. This gives

$$xP_n = a_nP_{n+1} + b_nP_n + c_nP_{n-1},$$

   where

$$a_n = \frac{\langle xP_n, P_{n+1}\rangle}{\|\mathbb{P}_{n+1}\|^2}, \quad \frac{\langle xP_n, P_n\rangle}{\|\mathbb{P}_n\|^2}, \quad c_n = \frac{\langle xP_n, P_{n-1}\rangle}{\|P_{n-1}\|^2}.$$

   After some computation, we obtain the following recursion formula

$$\boxed{(n+1)P_{n+1} = (2n+1)xP_n - nP_{n-1}.} \tag{3.19}$$

   We take

$$P_0 = 1, \quad P_1 = x.$$

   A matlab code provided by ChatGPT is shown below.

```
function P = legendre_polynomials(N, x)
% Compute Legendre polynomials up to degree N at the points x
% Input: N = degree of highest polynomial to compute
%        x = vector of evaluation points
% Output: P = (N+1) x length(x) matrix of Legendre polynomials

% Initialize matrix of Legendre polynomials
P = zeros(N+1, length(x));

% Compute first two polynomials
P(1,:) = 1;
P(2,:) = x;
```

```
% Recursively compute higher order polynomials using recurrence
   relation
for k = 2:N
    P(k+1,:) = (2*k-1)/k * x .* P(k,:) - (k-1)/k * P(k-1,:);
end
```

This code initializes a matrix P of size $(N+1)\times\text{length}(x)$ to store the Legendre polynomials, and then computes the first two Legendre polynomials using the known expressions $P_0(x) = 1$ and $P_1(x) = x$. The higher order polynomials are then computed recursively using the recurrence relation for Legendre polynomials. The resulting matrix P contains the Legendre polynomials up to degree N evaluated at the points x.

3. The Legendre polynomials have the following properties:

   - P_n has the expression (Rodrigues' formula)

   $$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}\left(x^2-1\right)^n. \tag{3.20}$$

   - P_n is the eigenfunction of the differential operator:

   $$\frac{d}{dx}\left((1-x^2)\frac{d}{dx}\right)P_n = -n(n+1)P_n. \tag{3.21}$$

   - $\{P_k | k = 0, 1, \cdots\}$ is complete in $L^2([-1,1], dx)$. This means that any $f \in L^2([-1,1], dx)$ can be expanded as

   $$f(x) = \sum_{k=0}^{\infty} \hat{f}_k P_k(x)$$

   with coefficient (Legendre mode)

   $$\hat{f}_k = \frac{2k+1}{2}\int_{-1}^{1} f(x)P_k(x)\,dx.$$

Note that $\|P_k\|^2 = \frac{2k+1}{2}$. A matlab code to plot Legendre polynomials is shown below.

```
x = linspace(-1, 1, 1000); % Define range of x values

figure % Create new figure window

for n = 0:5
    y = legendreP(n, x); % Calculate Legendre polynomial for current n
```

```
    plot(x, y, 'DisplayName', ['P_' num2str(n) '(x)']) % Plot
        polynomial
    hold on % Hold plot for next iteration
end

xlabel('x') % Label x-axis
ylabel('y') % Label y-axis
title('Legendre Polynomials') % Add title
legend('show') % Show legend
```

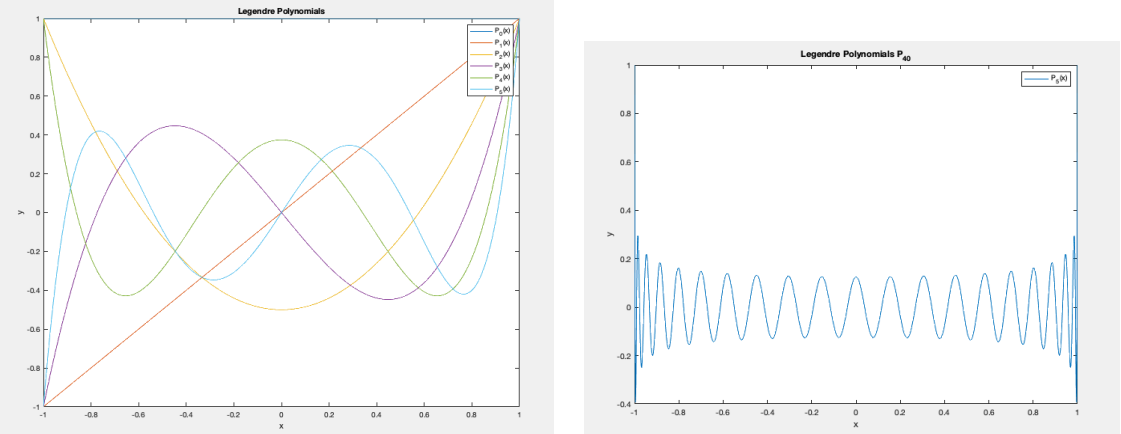


Figure 3.2: The left figure is the graph of Legendre polynomials P_0 - P_5. You can see the zeros of P_i and P_{i+1} are interlaced each other. The right figure is P_{40}. You can see the zeros are clustered more on the boundary.

4. **Examples** (see Kopriva, Implementing Spectral Methods for Partial Differential Equations (2009)

   - The Legendre modes of $f(x) = sign(x)$ is

   $$\hat{f}_k = \frac{2k+1}{2}\langle f, P_k\rangle = (-1)^k \frac{(4k+3)(2k)!}{2^{2k+1}(k+1)!k!} = O(k^{-1/2})$$

   Note that $\|P_k\|^2 = \frac{2k+1}{2}$. If we normalize it by $\bar{P}_k = \sqrt{\frac{2}{2k+1}}P_k$. Then $\bar{P}_k$ are orthonormal, and

   $$f = \sum_k \hat{f}_k P_k = \sum_k \sqrt{\frac{2}{2k+1}}\bar{P}_k = \sum_k \langle f, \bar{P}_k\rangle \bar{P}_k.$$

   The Legendre coefficient $\langle f, \bar{P}_k\rangle = O(1/k)$, the same decay rate as that in the Fourier expansion.

- $f(x) = |x|$: for odd k, $\hat{f}_k = 0$, for even k,

$$\hat{f}_k = (-1)^{k/2+1} \frac{(2k+1)\,k!}{2^{k+1}\,(k/2!)^2\,(k-1)(k/2-1)} = O(k^{-3/2}).$$

5. **Spectral accuracy**: The finite Legendre expansion of a smooth f converges exponentially fast. Namely, for any $s > 0$, there exist a constant C_s such that if $f \in H^s(-1,1)$, then

$$\|f - \sum_{k=0}^{N} \hat{f}_k P_k\| \le C_s N^{-s}.$$

Canuto,C.,Hussaini,M.,Quarteroni,A.,Zang,T.: Spectral Methods: Fundamentals in Single Domains. Springer, Berlin (2006).

3.7 Discrete Legendre transform

3.7.1 Gauss-Legendre quadrature method

1. In this subsection, we will introduce numerical integration method to compute the Legendre modes:

$$\hat{f}_k = \frac{2k+1}{2} \int_{-1}^{1} f(x) P_k(x) dx \tag{3.22}$$

We will introduce the Gauss-Legendre quadrature method and the Gauss-Lobetto quadrature method. The Legendre modes will be approximated by the quadrature method:

$$\tilde{f}_k = \frac{2k+1}{2} \sum_{j=1}^{n} f(x_j) P_k(x_j) w_j. \tag{3.23}$$

where $\{x_j\}$ are the quadrature points, $\{w_j\}$ the weights.

2. **Gauss-Legendre quadrature** Let us consider the integration on $[-1,1]$ with weight $w \equiv 1$. From the quadrature rule, the quadrature nodes are chosen to be the zeros of $q \in \Pi_n$ and w-orthogonal to Π_{n-1}, which is the Legendre polynomial $P_n(x)$. That is,

$$P_n(x_j) = 0, \quad j = 1, ..., N.$$

These quadrature points are called the Gauss-Legendre quadrature points. The corresponding weights are

$$w_j := \int_{-1}^{1} \ell_j(x)\, dx,$$

where $\ell_i(x_j) = \delta_{ij}$ are the Lagrange polynomials determined by the nodes $\{x_j\}_{j=1}^N$. The weights have an explicit expression (Abramowitz & Stegun 1972)

$$w_j = \frac{2}{(1-x_j^2)P_n'(x_j)^2}, \quad j = 1, ..., N. \tag{3.24}$$

3. Numerical algorithms to compute the Gauss-Legendre quadrature nodes and weights:

   - Newton-Raphson method: This is to solve the equation $P_n(x) = 0$ by Newton's method.
   - Golub-Welsch method.
   - Fast algorithm is also available in the paper Fast and accurate computation of Gauss-Legendre quadratures.

4. **Newton-Raphson method for Gauss-Legendre quadrature rule**. The roots of P_N can be obtained by Newton's method:

$$x_j^{k+1} = x_j^k - \frac{P_N(x_j^k)}{P_N'(x_j^k)}, \quad j = 1, ..., N.$$

   The coefficient of the polynomial P_N and P_N' can be obtained from the recursion formula. The initial guess of the Newton's iteration is

$$x_j^0 = -\cos\left(\frac{2j-1}{2N}\pi\right), \quad j = 1, ..., N.$$

   A matlab code provided by ChatGPT is shown below.

```
function [x, w] = legendre_zeros(N)
% Compute zeros and weights of Legendre polynomial of degree N
% Input: N = degree of Legendre polynomial
% Output: x = vector of N zeros of Legendre polynomial
%         w = vector of N weights for Gaussian quadrature

% Initial guess for zeros
x0 = cos(pi*(0:N-1)/(N-0.5));

% Tolerance for convergence
tol = eps;

% Maximum number of iterations
maxiter = 100;

% Compute zeros using Newton's method
x = x0;
for k = 1:N
    fk = legendre_polynomial(N, x(k));
    dfk = legendre_derivative(N, x(k));
    iter = 0;
    while abs(fk) > tol && iter < maxiter
        x(k) = x(k) - fk/dfk;
```

```
        fk = legendre_polynomial(N, x(k));
        dfk = legendre_derivative(N, x(k));
        iter = iter + 1;
    end
end

% Compute weights for Gaussian quadrature
w = 2./(1-x.^2)./legendre_derivative(N-1, x).^2;
```

5. **Golub-Welsch method** We rewrite the recursion formula for the Legendre polynomials as

$$P_{n+1} = a_{n+1} x P_n - c_{n+1} P_{n-1}$$

where

$$a_{n+1} = \frac{2n+1}{n+1}, \quad c_{n+1} = \frac{n}{n+1}.$$

In matrix form, it reads

$$\begin{bmatrix} 0 & 1/a_1 & & & \\ c_2/a_2 & 0 & 1/a_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-1}/a_{n-1} & 0 & 1/a_{n-1} \\ & & & c_n/a_n & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_{n-2} \\ P_{n-1} \end{bmatrix} = x \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_{n-2} \\ P_{n-1} \end{bmatrix} - \frac{1}{a_n} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ P_n \end{bmatrix}$$

This tridiagonal matrix can be symmetrized by a diagonal matrix D to

$$\begin{bmatrix} 0 & \beta_1 & & & \\ \beta_1 & 0 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-2} & 0 & \beta_{n-1} \\ & & & \beta_{n-1} & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_{n-2} \\ P_{n-1} \end{bmatrix} = x \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_{n-2} \\ P_{n-1} \end{bmatrix} - \frac{n}{2n-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ P_n \end{bmatrix}$$

where

$$\beta_i = \sqrt{\frac{c_{i+1}}{a_i a_{i+1}}} = \sqrt{\frac{\frac{i}{i+1}}{\frac{2i-1}{i}\frac{2i+1}{i+1}}} = \frac{1}{2\sqrt{1-\left(\frac{1}{2i}\right)^2}} \tag{3.25}$$

We note that $P_n(x_j) = 0$ if and only if x_j is the eigenvalue of the above matrix. Moreover, if $\mathbf{v}_j = (v_{j,1}, ..., v_{j,n})$ is the corresponding unit eigenvector, then the weight is given by

$$w_j = \left(\int_{-1}^{1} w(x)\, dx \right) v_{j,1}^2 = 2 v_{j,1}^2.$$

A matlab code generated by ChatGPT for Gauss-Legendre quadratures using Golub-Welsch method is available.

```
function [x, w] = gauss_legendre(n,a,b)
% Computes the Gauss-Legendre quadrature points and weights on the
   interval [-1, 1].
% n: number of quadrature points and weights (degree of polynomial
   accuracy)
% x: array of quadrature points
% w: array of quadrature weights

beta = 0.5./sqrt(1-(2*(1:n)).^(-2)); % beta coefficients
T = diag(beta,1) + diag(beta,-1); % Jacobi matrix
[V, D] = eig(T); % eigenvectors and eigenvalues of Jacobi matrix
x = diag(D); % quadrature points are the eigenvalues
w = 2*V(1,:).^2; % quadrature weights
x = (b-a)/2*x + (b+a)/2; % rescale and shift nodes
w = (b-a)/2*w; % rescale weights
end
```

6. A MATLAB code that computes the Legendre coefficients of a function $f(x)$ using Gauss-Legendre quadrature. Now we perform the numerical integration for the Legendre coefficients.

```
% Define the function f(x)
f = @(x) x.^2 + sin(x);

% Define the maximum order of the Legendre polynomial
nmax = 5;
 % Compute the integral using Gaussian-Legendre quadrature
[xq,wq] = gauss_legendre(nmax,-1,1); % gauss_legendre() is a function
   that returns the quadrature points xq and weights wq

% Compute the Legendre coefficients using Gauss-Legendre quadrature
ck = zeros(nmax+1,1);
for k = 0:nmax
    % Define the Legendre polynomial Pk(x)
    Pk = legendreP(k,x); % x is a variable that takes values in the
       interval [-1,1]
    integrand = f(xq).*Pk(xq);
    ck(k+1) = sum(integrand.*wq);
end
```

In this example, we define the function $f(x) = x^2 + sin(x)$ and the maximum order of the Legendre polynomial as $nmax = 5$. We then loop over all orders of the Legendre polynomial from 0 to $nmax$ and compute the corresponding Legendre coefficient c_n

using the formula above.

To compute the integral using Gauss-Legendre quadrature, we use the gauss_legendre function to generate $nmax$ quadrature points xq and weights wq in the interval $[-1, 1]$. We then evaluate the integrand at these points and sum the weighted function values to obtain the Legendre coefficient.

3.7.2 Legendre-Lobetto quadrature method

1. The Legendre-Lobetto quadrature method is to approximate

$$\hat{f}_k = \frac{2k+1}{2} \int_{-1}^{1} f(x) P_k(x)\, dx$$

by

$$\tilde{f}_k = \frac{2k+1}{2} \sum_{j=0}^{N} f(x_j) P_k(x_j) w_j.$$

It is almost the same as the Gauss-Legendre quadrature except it uses the end points as the quadrature points. It is useful for boundary-value problems.

2. **The Legendre-Lobetto quadrature rule** :

   - The nodes $\{x_0, ..., x_N\}$ are roots of $(1-x^2)P_N'(x) = 0$.
   - The corresponding weights are

$$w_j = \frac{2}{(N+1)NP_N(x_j)^2},\ j = 0, ..., N. \tag{3.26}$$

They are determined so that quadrature formula

$$\int_{-1}^{1} f(x)\, dx \approx \sum_{j=0}^{N} f(x_j) w_j$$

is exact for $f \in \Pi_{2N-1}$. Note that $x_0 = -1, x_N = 1$ are already prescribed. Thus, there are only $2N$ unknowns: $x_1, ..., x_{N-1}$ and $w_0, w_1, ..., w_N$. We can only have $2N$ conditions to have the above numerical integration formula to be exact. That is, the formula is exact for $f = 1, x, ..., x^{2N-1}$. These give $2N$ conditions.

3. A function f defined on $[-1, 1]$ can be represented as

$$f(x) \approx \sum_j f(x_j) \ell_j(x)$$

where $\ell_j(x)$ is the Lagrange interpolant

$$\ell_j(x) = \frac{\prod_{i \neq j}(x - x_i)}{\prod_{i \neq j}(x_j - x_i)}, \quad j = 0, ..., N.$$

The Lagrange interpolants satisfy

$$\ell_i(x_j) = \delta_{ij}.$$

There is another expression for ℓ_j:

$$\ell_j(x) = \frac{-(1-x^2)P_N'(x)}{(N+1)NP_N(x_j)(x-x_j)}, \quad j = 0, ..., N.$$

4. Here is a matlab code to find the nodes and weights.

```
function [x,w] = legendre_lobatto(N)
% Calculates the Legendre-Lobatto nodes and weights for numerical
   integration on [-1,1].
% n: degree of the Legendre polynomial
% x: array of nodes
% w: array of weights

% Define the Legendre polynomial of degree n
syms x
Pn = legendreP(N,x);

% Find the roots of Pn'
Pn_prime = diff(Pn);
x = solve(Pn_prime == 0, x);

% Add the endpoints -1 and 1 to the list of roots
x = [x; -1; 1];
x = sort(x);

% Calculate the weights
w = zeros(N+1,1);
for i = 1:N+1
    w(i) = 2/(N*(N+1)*(legendreP(N,x(i)))^2);
end

end
```

5. Example:

3.8 Spectral methods

3.8.1 Introduction

A nice book for implementing the spectral method is Kopriva, Implementing Spectral Methods for Partial Differential Equations (2009)

1. In this section, we introduce spectral method for PDEs on

   - the periodic domain $[0, 2\pi]$ using Fourier series,
   - on $[-1, 1]$ using Legendre polynomials.

   The PDEs include

   - Reaction-diffusion equation and Phase-field models
     - Allan-Cahn equation
       $$u_t = \triangle u + u - u^3.$$
     - Cahn-Hilliard equatio:
       $$u_t = - \triangle \left(\triangle u + u - u^3 \right).$$

       A recent review article for spectral method for phase-field model can be found in Fourier-Spectral Method for the Phase-Field Equations.
   - Nonlinear Schrödinger equation
     $$i\psi_t = - \triangle \psi - |\psi|^2 \psi \quad \text{in } [0.2\pi]$$

     A nice survey article is:
     Weizhu Bao, Shi Jin, Peter Markowich, Numerical study of time-splitting spectral discretization of nonlinear Schrödinger equations (2003).

2. The unknown u can be a modal representation, or a nodal representation:
   $$u_N(x) = \sum_{k=1}^{N} u_k \phi_k(x)$$

   The functions $\{\phi_k\}$ are called the *trial functions*. Usually, the trial functions are required to satisfy the boundary conditions. The trial functions are orthogonal polynomials:

   - Fourier
   - Legendre
   - Chebeshev
   - Jacobi, Laguarre, Hermite

3. We plug an approximate unknown u_N into the equation and get a residual R_N. For instance
   $$R_N = \partial_t u_N - \triangle u_N$$
   for the heat equation. The equations are realized (projected) on *test functions* $\{\psi_k\}$:
   $$\langle R_N, \psi_k \rangle = 0, \quad k = 1, ..., N.$$

- **Galerkin**: the test functions are the same as the trial functions.
- **Petrov-Galerkin**: the test functions are different from the trial functions.
- **Collocation**: The collocation points $\{x_1, ..., x_N\}$ are pre-assigned. The residual R_N satisfies
$$R_N(x_j) = 0, \quad j = 1, ..., N.$$

The test functions should also satisfy the boundary conditions. The spectral-collocation method is also called the pseudo-spectral method.

4. The general strategies are:

- **Operator splitting**: for problem like
$$u_t = \triangle u + f(u) = Au + Bu.$$
A first order splitting method is to approximate $e^{(A+B)\Delta t}$ by $e^{\Delta t B} e^{\Delta t A}$. That is, we solve the two problems:
$$v_t = \triangle v, \quad v(0) = u_0$$
and
$$w_t = f(w), \quad w(0) = v(\Delta t).$$
Then
$$u(\Delta t) \approx w(\Delta t).$$
- **Spectral method for the linear part** Linear equation with constant coefficients can be solved exactly by spectral method.
$$u_t = u_{xx}$$
is solved by
$$\tilde{U}_k = e^{-k^2 t} U_k(0).$$
- **Collocation method for the nonlinear part.** For nonlinear equations, one should use collocation method. That is, we solve
$$\dot{U}(x_j, t) = f(U(x_j, t)), \quad j = 1, ..., N.$$
Implicit methods may be needed. For instance, the backward Euler method for this ODE:
$$U^{n+1} - U^n = \Delta t f(U^{n+1}). \tag{3.27}$$
Let us denote this equation by
$$F(y) = 0, \quad \text{where } y = U^{n+1}, \quad F(y) = y - \Delta t f(y) - U^n.$$

We solve this equation by Newton's iteration:

$$y^{i+1} = y^i - F'(y^i)^{-1} F(y),$$

We can start from $y^0 = U^n$ since we expect the final solution $y = U^{n+1}$ is not too far from U^n. The derivative $F'(y^i) = I - \Delta f(y^i)$. To avoid many derivative calculations, we can use just one derivative, namely $F'(y^0)$. We put a coefficient α to stablize the scheme. The final iteration is

$$y^{i+1} = y^i - \alpha F'(y^0)^{-1} F(y).$$

In terms of $U^{n+1,i}$, it reads

$$U^{n+1,i+1} = U^{n+1,i} - \alpha \left(I - \Delta t f'(U^n)\right)^{-1} \left(U^{n+1,i} - \Delta t f(U^{n+1,i}) - U^n\right).$$

The stablizing parameter α is chosen so that

$$\alpha \left| \left(I - \Delta t f'(U)\right)^{-1} \right| < 1, \text{ for } U \text{ under consideration.}$$

If we do not put the term $\left(I - \Delta t f'(U)\right)^{-1}$ in the iteration, the resulting iteration is

$$U^{n+1,i+1} = U^{n+1,i} - \alpha \left(U^{n+1,i} - \Delta t f(U^{n+1,i}) - U^n\right).$$

This is a fixed point iteration for the nonlinear equation (3.27).

3.8.2 Legendre-Galerkin method

Let us solve the heat equation

$$u_t = u_{xx}, \quad \text{on } [-1,1], \quad u(-1) = 0, \quad u(1) = 0.$$

by the Legendre-Galerkin method.

1. The trial functions ϕ_k are chosen to be

$$\phi_k := \frac{1}{\sqrt{4k+6}} \left(P_k - P_{k+2}\right), \quad k = 0, ..., N-2. \tag{3.28}$$

The Legendre polynomial P_k satisfies $P_k(1) = 1$, $P_k(-1) = (-1)^k$. Thus, the trial functions ϕ_k satisfy the boundary conditions:

$$\phi_k(-1) = \phi_k(1) = 0.$$

The unknown u is represented as

$$u \approx \sum_{l=0}^{N-2} \hat{u}_k(t) \phi_k(x).$$

2. The Galerkin method for the heat equation is

$$\langle u_t, \phi_k \rangle = \langle u_{xx}, \phi_k \rangle, \quad k = 0, ..., N-2.$$

With $u = \sum \hat{u}_l \phi_l$, we get

$$\sum_l \langle \phi_k, \phi_l \rangle \dot{\hat{u}}_l = -\sum_l \langle \phi_k', \phi_l' \rangle \hat{u}_l.$$

In matrix form:

$$M\dot{\hat{U}} = -S\hat{U}.$$

where $\hat{U} = (\hat{u}_0, ..., \hat{u}_{N-2})^T$. Here, M is called the mass matrix, while S, the stiff matrix.

3. The stiff matrix:

$$S = \left(\langle \phi_i', \phi_j' \rangle \right)_{0 \le i,j \le N-2} = (\delta_{ij}).$$

4. The mass matrix

$$M = \left(\langle \phi_i, \phi_j \rangle \right)_{0 \le i,j \le N-2} = \alpha_i \alpha_j \left(\beta_i \delta_{ij} + \gamma_j \delta_{i+2,j} + \mu_j \delta_{i,j+2} \right),$$

$$\alpha_i = \frac{1}{4i+6}, \quad \gamma_i = \frac{2}{2i+1}, \quad \mu_i = -\frac{2}{2i+5}, \quad \beta_i = -(\gamma_i + \mu_i).$$

5. The equation $M\dot{\hat{U}} = -S\hat{U}$ can be solved by the Crank-Nilcolson method

$$M\left(\hat{U}^{n+1} - \hat{U}^n \right) = -\frac{\Delta t}{2} \left(\hat{U}^n + \hat{U}^{n+1} \right),$$

with initial state $\hat{U}^0 = \left(\hat{u}_0^0, ..., \hat{u}_{N-2}^0 \right)^T$.

6. The initial state $\hat{U}^0$ is obtained from the initial data $u^0 = \sum_l \hat{u}_{0,l} \phi_l$. The coefficients $\hat{u}_k^0$ satisfy

$$\langle u_0, \phi_k \rangle = \langle \sum_l \hat{u}_l^0 \phi_l, \phi_k \rangle, \quad k = 0, ..., N-2.$$

In matrix form, it is

$$M\hat{U}^0 = \hat{u}^0, \quad \hat{u}^0 := \left(\langle u^0, \phi_0 \rangle, \cdots, \langle u^0, \phi_{N-2} \rangle \right)^T.$$

The inner product

$$\langle u^0, \phi_k \rangle = \frac{1}{\sqrt{4k+6}} \langle u^0, P_k - P_{k+2} \rangle, \quad k = 0, ..., N-2.$$

Finally, one can use Legendre quadrature method to find the Legendre coefficients $\langle u^0, P_k \rangle$.

Below are matlab codes generated by ChatGPT for the matrix representation of the derivative operator and the Laplacian.

1. Q: Write a matlab code to generate the matrix representation for the d/dx under the Legendre polynomials.
   ChatGPT:
   To generate the matrix representation for the derivative operator $\frac{d}{dx}$ under the Legendre polynomials, we can use the fact that the Legendre polynomials satisfy the following recurrence relation:

   $(k+1)P_{k+1}(x) = (2k+1)xP_k(x) - kP_{k-1}(x)$

   Using this recurrence relation, we can derive a matrix representation for the derivative operator in terms of the Legendre polynomials. Here's a MATLAB code to generate the matrix:

```
function D = legendre_derivative_matrix(n)
% Generate matrix representation for d/dx under the Legendre
   polynomials
% Input: n = degree of Legendre polynomials
% Output: D = n x n matrix representing the derivative operator

% Compute Legendre polynomial coefficients
P = zeros(n+1);
P(1,1) = 1;
P(2,2) = 1;
for k = 2:n
    P(k+1,:) = ((2*k-1)*[0,P(k,1:n)] - (k-1)*[0,0,P(k-1,1:n-1)]) / k;
end

% Compute derivative matrix
D = zeros(n);
for i = 1:n
    for j = 1:n
        if i == j
            if i == 1
                D(i,j) = n*(n+1)/4;
            else
                D(i,j) = i*P(i,j-1);
            end
        elseif j == i+1
            D(i,j) = (2*i+1)/2;
        elseif j == i-1
            D(i,j) = -(i-1)/2;
        end
    end
```

```
end
```

This code first computes the coefficients of the Legendre polynomials up to degree n, using the recurrence relation given above. It then constructs the derivative matrix D using the coefficients of the Legendre polynomials. The diagonal entries of D are computed using the formula $D_{ii} = iP_{i-1}(x_i)$, where x_i are the roots of the n-th Legendre polynomial, and $P_{i-1}(x_i)$ are the Legendre polynomials evaluated at those roots. The off-diagonal entries of D are computed using the recurrence relation for the Legendre polynomials.

2. Q: Generate matrix representation for d^2/dx^2 on $[-1, 1]$ using the matrix representation for the derivative operator d/dx.
ChatGPT: To generate the matrix representation for the second derivative operator $\frac{d^2}{dx^2}$ on $[-1, 1]$ using the matrix representation for the derivative operator $\frac{d}{dx}$, we can use the fact that the second derivative can be written as the composition of two first derivative operators:

$$\frac{d^2}{dx^2} = \frac{d}{dx} \circ \frac{d}{dx}$$

Using this fact, we can compute the matrix representation for the second derivative by taking the product of the matrix representation for the derivative operator with itself.

Here's a MATLAB code to generate the matrix representation for $\frac{d^2}{dx^2}$ on $[-1, 1]$:

```
function D2 = legendre_second_derivative_matrix(n)
% Generate matrix representation for d^2/dx^2 under the Legendre
   polynomials
% Input: n = degree of Legendre polynomials
% Output: D2 = n x n matrix representing the second derivative operator

% Compute derivative matrix
D = legendre_derivative_matrix(n);

% Compute second derivative matrix
D2 = D * D;
```

This code first computes the matrix representation for the derivative operator using the legendre_derivative_matrix function from the previous answer. It then computes the matrix representation for the second derivative operator by taking the product of the derivative matrix with itself. The resulting matrix D2 is the desired matrix representation for $\frac{d^2}{dx^2}$ on $[-1, 1]$ in terms of the Legendre polynomials.

3.8.3 Legendre collocation method

We solve the heat equation

$$u_t = u_{xx} \text{ on } [-1,1], \text{ with } u(-1,t) = u(1,t) = 0, \quad u(\cdot,0) = u^0(\cdot).$$

1. We represent the unknown

$$u_N(x,t) = \sum_{j=0}^{N} U(x_j,t)\ell_j(x),$$

where $x_0 = -1, ..., x_N = 1$ are the Lengendre-Lobetto quadrature nodes with weights $\{w_j\}_{j=0}^N$. The function ℓ_j is the Lagrange interplant satisfying $\ell_j(x_i) = \delta_{ij}$ and $deg(\ell_j) = N$.

2. Let us denote $U(x_j,t)$ by $U_j(t)$. At boundaries, $U_0(t) = U_N(t) = 0$. The unknowns are $U = (U_1, ..., U_{N-1})^T$.

3. The equation is realized at x_i, $i = 1, ..., N-1$:

$$\langle u_{N,t}, \ell_i\rangle = \langle u_{N,xx}, \ell_i\rangle, \quad i = 1, ..., N-1.$$

$$\langle \sum_{j=1}^{N-1} \dot{U}_j\ell_j, \ell_i\rangle = -\langle \sum_{j=1}^{N-1} U_j\ell_j', \ell_i'\rangle.$$

In matrix form, it reads

$$M\dot{U} = -SU,$$

where the mass matrix M and the stiff matrix S are given by

$$M = (\langle \ell_i, \ell_j\rangle)_{(N-1)\times(N-1)} = \sum_{m=0}^{N} \ell_i(x_m)\,\ell_j(x_m)w_m = \delta_{ij}w_i,$$

$$S = \left(\langle \ell_i', \ell_j'\rangle\right)_{(N-1)\times(N-1)} = \sum_{m=0}^{N} \ell_i'(x_m)\,\ell_j'(x_m)w_m.$$

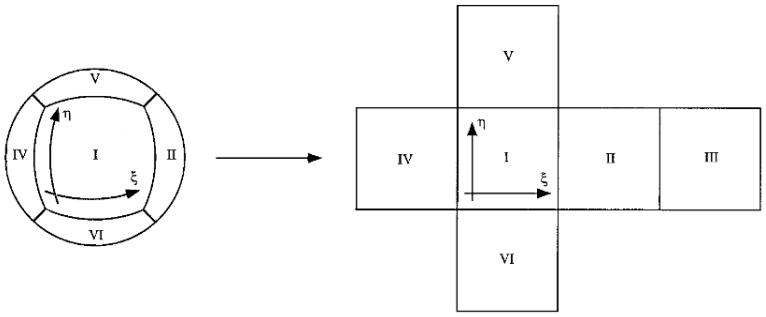
The matrix $\ell_i'(x_m)$ is the polynomial derivative matrix. It has an explicit form.

4. The discrete equation $M\dot{U} = -SU$ can be solved by the Crank-Nicolson method.

3.9 Spectral element methods

Project 3

- Solve the Poisson equation on the interval $[-1,1]$ by using the Legendre-Lobetto spectral method.



- Solve the Poisson equation on the box $[-1, 1] \times [-1, 1]$ by using Legendre-Lobetto spectral method.
- Solve the Poisson equation on the sphere using cubic sphere grid. (The ‘‘Cubed Sphere’’: A New Method for the Solution of Partial Differential Equations in Spherical Geometry, JCP 1996). The sphere is parametrized by an inscribed cube. There are 6 surfaces on the cube. Adopt the Legendre-Lobetto spectral method on each surface.
- Implement the (geometric or algebraic) multigrid method for solving the Poisson equation on a rectangle. See Algebraic multigrid method.

Chapter 4

Finite Difference Methods for the Poisson Equations

4.1 Discrete Laplacian in two dimensions

In this chapter, we will solve the Poisson equation

$$\triangle u = f \text{ in } \Omega \subset \mathbb{R}^2, \tag{4.1}$$

with Dirichlet boundary condition

$$u = g \text{ on } \partial\Omega.$$

Such a problem is a core problem in many applications. We may assume $g = 0$ by subtracting a suitable function from u. Thus, we limit our discussion to the case of zero boundary condition. Let h be the spatial mesh size. For simplicity, let us assume $\Omega = [0,1] \times [0,1]$. Yet, many discussion below can be extended to general smooth bounded domain.

4.1.1 Discretization methods

Centered finite differencing The Laplacian is approximated by

$$A = \frac{1}{h^2}\left(U_{i-1,j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1} - 4U_{i,j}\right).$$

For the square domain, the indeces run from $1 \le i, j \le N-1$ and

$$U_{0,j} = U_{N,j} = U_{i,0} = U_{i,N} = 0$$

from the boundary condition.

Let us order the unknowns U by $i + j * (N-1)$ with j being outer loop index and i the

inner loop index, then the matrix form of the discrete Laplacian is

$$A = \frac{1}{h^2}\begin{bmatrix} T & I & & & \\ I & T & I & & \\ & I & T & I & \\ & & \ddots & \ddots & \ddots \\ & & & I & T \end{bmatrix}$$

This is an $(N-1)\times(N-1)$ block tridiagonal matrix. The block T is an $(N-1)\times(N-1)$ matrix

$$T = \begin{bmatrix} -4 & 1 & & & \\ 1 & -4 & -1 & & \\ & 1 & -4 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -4 \end{bmatrix}$$

Since this discrete Laplacian is derived by centered finite differencing over uniform grid, it is second order accurate, the truncation error

$$\begin{aligned} \tau_{i,j} &:= \frac{1}{h^2}\left(u(x_{i-1},y_j)+u(x_{i+1},y_j)+u(x_i,y_{j-1})+u(x_i,y_{j+1})-4u(x_i,y_j)\right) \\ &= O(h^2). \end{aligned}$$

4.1.2 The 9-point discrete Laplacian

The Laplacian is approximated by

$$\triangle_9 = \frac{1}{6h^2}\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$

One can show by Taylor expansion that

$$\triangle_9 u = \triangle u + \frac{1}{12}h^2 \triangle^2 u + O(h^4).$$

If u is a solution of $\triangle u = f$, then

$$\triangle_9 u = f + \frac{1}{12}h^2 \triangle f + O(h^4).$$

Thus, we get a 4th order method:

$$\triangle_9 U_{ij} = f_{ij} + \frac{h^2}{12}\triangle f_{ij}$$

4.2 Stability of the discrete Laplacian

We have seen that the true solution of $\triangle u = f$ with Dirichlet boundary condition satisfies

$$Au = f + \tau,$$

where A is the discrete Laplacian and τ is the truncation error and satisfies $\tau = O(h^2)$ in maximum norm. The numerical solution U satisfies $AU = f$. Thus, the true error satisfies

$$Ae = \tau,$$

where $e = u - U$. Thus, e satisfies the same equation with right-hand side τ and with the Dirichlet boundary condition. To get the convergence result, we need an estimate of e in terms of τ. This is the stability criterion of A. We say that A is stable if there exists some norm $\|\cdot\|$ and a constant C such that

$$\|e\| \le C\|Ae\|.$$

4.2.1 Fourier method

Since our domain $\Omega = [0,1] \times [0,1]$ and the coefficients are constant, we can apply Fourier transform. Let us see one dimensional case first. Consider the Laplacian d^2/dx^2 on domain $[0,1]$ with Dirichlet boundary condition. The discrete Laplacian is $A = \frac{1}{h^2}\text{diag}\,(1,-2,1)$, where $h = 1/N$. We can check below that the eigenvectors of A are $v_k = (\sin(\pi jkh))_{j=1}^{N-1}$, $k = 1, ..., N-1$. The corresponding eigenvalues are $-\frac{4}{h^2}\sin^2(\pi hk/2)$.

$$\begin{aligned}[Av_k]_j = [A\sin(j\pi kh)]_j &= \frac{1}{h^2}\left(\sin((j+1)\pi hk) + \sin((j-1)\pi hk) - 2\sin(j\pi hk)\right) \\ &= \left[\frac{2}{h^2}(\cos(\pi hk) - 1)\right]\sin(j\pi hk) = -\frac{4}{h^2}\sin^2(\pi hk/2)[v_k]_j.\end{aligned}$$

For two dimensional case, the eigenfunctions of the discrete Laplacian are $U^{k,\ell}$, $1 \le, k, \ell \le N-1$,

$$(U^{k,\ell})_{i,j} = \sin(i\pi kh)\sin(j\pi \ell h), \quad 1 \le i, j \le N-1.$$

The corresponding eigenvalues are

$$\begin{aligned}\lambda^{k,\ell} &= \frac{2}{h^2}(\cos(k\pi h) + \cos(\ell\pi h) - 2) \\ &= -\frac{4}{h^2}(\sin^2(k\pi h/2) + \sin^2(\ell\pi h/2)), \quad 1 \le, k, \ell \le N-1.\end{aligned}$$

The smallest eigenvalue (in magnitude) is

$$\lambda^{1,1} = -\frac{8}{h^2}\sin^2(\pi h/2) \approx -2\pi^2 \quad \text{for small } h.$$

To show the stability, we take Fourier transform of U and A. We then have

$$\|\widehat{A}\hat{U}\|\|\hat{U}\| \geq \left|\langle \widehat{A}\hat{U}, \hat{U}\rangle\right| \geq |\lambda^{1,1}|\|\hat{U}\|^2 \approx 2\pi^2\|\hat{U}\|^2.$$

Hence, the L^2 norm of $\widehat{A}$ has the following estimate:

$$\|\widehat{A}\hat{U}\| \geq 2\pi^2\|\hat{U}\|.$$

Thus, we get

$$\|\hat{U}\| \leq \frac{1}{2\pi^2}\|\widehat{A}\hat{U}\|.$$

From Parseval equality, we have

$$\|U\| \leq \frac{1}{2\pi^2}\|AU\|$$

Applying this stability to the formula: $Ae = \tau$, we get

$$\|e\| \leq \frac{1}{2\pi^2}\|\tau\| = O(h^2).$$

Homeworks 4.1. 1. Compute th eigenvalues and eigenfunctions of the 9-point discrete Laplacian on the domain $[0,1] \times [0,1]$ with zero boundary condition.

4.2.2 Energy method

Below, we use energy method to prove the stability result for discrete Laplacian. We shall prove it for rectangular domain. However, it can be extended to more general domain. To perform energy estimate, we rewrite the discrete Laplacian as

$$AU_{i,j} = \frac{1}{h^2}\left(U_{i-1,j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1} - 4U_{i,j}\right) = ((D_{x+}D_{x-} + D_{y+}D_{y-})U)_{i,j}$$

where

$$(D_{x+}U)_{i,j} = \frac{U_{i+1,j} - U_{i,j}}{h}$$

the forward differencing. We multiply the discrete Laplacian by $U_{i,j}$, then sum over all i, j. By applying the summation by part, we get

$$\begin{aligned}\langle AU, U\rangle &= \langle (D_{x+}D_{x-} + D_{y+}D_{y-})U, U\rangle \\ &= -\langle D_{x-}U, D_{x-}U\rangle - \langle D_{y-}U, D_{y-}U\rangle \\ &= -\|\nabla_h U\|_h^2\end{aligned}$$

Here, the discrete L^2 norm is defined by

$$\|U\|_h^2 = \sum_{i,j} |U_{i,j}|^2 h^2.$$

The boundary term does not show up because we consider the zero Dirichlet boundary problem. Thus, the discrete Poisson equation has the estimate

$$\|\nabla_h U\|_h^2 = |\langle f, U\rangle| \leq \|f\|_h \|U\|_h. \tag{4.2}$$

Next, for the zero Dirichlet boundary condition, we have the Poincaré inequality, which will be shown below. Before stating the Poincare inequality, we need to clarify the meaning of zero boundary condition in the discrete sense. We define the Sobolev space $H^1_{h,0}$ to be the completion of the restriction of all C^1_0 functions to the grid points under the discrete H^1 norm. Here, C^1_0 function is a C^1 function that is zero on the boundary; the discrete H^1 norm is

$$\|U\|_{h,1} := \|U\|_h + \|\nabla_h U\|_h.$$

Lemma 4.1 (Poincaré inequality)**.** *Let Ω be a bounded domain in $\mathbb{R}^2$, then there exist a constant d_Ω, which is the diameter of the domain Ω, such that for any $U \in H^1_{h,0}$,*

$$\|U\|_h \leq d_\Omega \|\nabla_h U\|_h. \tag{4.3}$$

Proof. Let us take $\Omega = [0, X] \times [0, Y]$ as an example for the proof. We assume $X = Mh, Y = Nh$. From zero boundary condition, we have

$$\begin{aligned} U_{i,j}^2 &= (\sum_{i'=1}^{i} D_{x-} U_{i',j} h)^2 \\ &\leq (\sum_{i'=1}^{i} 1^2) \cdot (\sum_{i'=1}^{i} (D_{x-} U_{i',j})^2) h^2 \text{ (Hölder's inequality)} \\ &\leq i(\sum_{i'=1}^{M} (D_{x-} U_{i',j})^2) h^2 \end{aligned}$$

multiply both sides by h^2 then sum over all i, j, we get

$$\begin{aligned} \|U\|_h^2 &= \sum_{i,j} U_{i,j}^2 h^2 \\ &\leq (\sum_{i=1}^{M} i) h^2 \sum_{i',j} (D_{x-} U_{i',j})^2 h^2 \\ &\leq \frac{M^2}{2} h^2 \sum_{i',j} (D_{x-} U_{i',j})^2 h^2 \\ &= \frac{M^2}{2} h^2 \|D_{x-} U\|_h^2 \end{aligned}$$

Similarly, we have

$$\|U\|_h^2 \leq \frac{N^2}{2} h^2 \|D_{y-} U\|_h^2$$

Thus,

$$\begin{aligned} \|U\|_h^2 &\leq h^2 \frac{1}{2} \max\{M^2, N^2\} \|\nabla_h U\|^2 \\ &\leq d_\Omega^2 \|\nabla_h U\|_h^2. \end{aligned}$$

□

With the Poincaré inequality, we can obtain two estimates for U.

Proposition 4.1. *Consider the discrete Laplacian with zero boundary condition. We have*

$$\|U\|_h \leq d_\Omega^2 \|f\|_h, \tag{4.4}$$

$$\|\nabla_h U\| \leq d_\Omega \|f\|_h. \tag{4.5}$$

Proof. From

$$\|\nabla_h U\|_h^2 \leq \|f\|_h \cdot \|U\|_h$$

We apply the Poincaré inequality to the left-hand side, we obtain

$$\|U\|_h^2 \leq d_\Omega^2 \|\nabla U\|_h^2 \leq d_\Omega^2 \|f\|_h \|U\|_h$$

This yields

$$\|U\|_h \leq d_\Omega^2 \|f\|_h$$

If we apply the Poincare inequality to the right-hand side, we get

$$\|\nabla_h U\|_h^2 \leq \|f\|_h \cdot \|U\|_h \leq \|f\|_h \cdot d_\Omega \|\nabla_h U\|_h$$

Thus, we obtain

$$\|\nabla_h U\| \leq d_\Omega \|f\|_h$$

When we apply this result to $Ae = \tau$, we get

$$\begin{aligned} \|e\| &\leq d_\Omega^2 \|\tau\| = O(h^2) \\ \|\nabla_h e\| &\leq d_\Omega \|\tau\| = O(h^2). \end{aligned}$$

□

Remark The discrete Laplacian has many good properties as those of continuous Laplacian. For continuous Laplacian, we can have $\|u\|_{H^{s+2}}$ estimated by some $\|f\|_{H^s}$. In the case of discrete Laplacian, we have similar result. As the truncated error is of $\|\tau\|_{H^s} = O(h^2)$ in terms of the discrete norm, then we have $\|e\|_{H^{s+2}} = O(h^2)$. Using Sobolev inequality, wecan get $|e|_\infty = O(h^2)$.

4.3 Solving the Poisson equation

4.4 Multigrid method

Project 3

1. Implement the (geometric or algebraic) multigrid method for solving the Poisson equation on a rectangle. See Algebraic multigrid method.

Chapter 5

Finite Difference Methods For Linear Hyperbolic Equations

5.1 Linear hyperbolic equations

Hyperbolic PDEs have the property that information propagates at a finite speed and is governed by the characteristics of the system. This is in contrast to parabolic PDEs where information is diffused and elliptic PDEs where information is transmitted instantaneously.

For systems of linear PDEs, hyperbolicity can be characterized in terms of the eigenvalues of the coefficient matrix. Specifically, if all eigenvalues are real, then the system is hyperbolic. If some eigenvalues are complex, the system may be parabolic or elliptic.

5.1.1 Linear advection equation

1. **Constant speed**
   We start with the Cauchy problem of the linear advection equation with constant speed:

$$\begin{aligned} u_t + a u_x &= 0, \quad x \in \mathbb{R}, && (5.1) \\ u(x,0) &= u^0(x). && (5.2) \end{aligned}$$

   Its solution is simply a translation of u^0 with speed a, namely,

$$u(x,t) = u^0(x - at).$$

2. **Variable speed**
   More generally, we can solve the linear advection equation with variable coefficients by the *method of characteristics*. Consider

$$u_t + a(x,t) u_x = 0.$$

Let us interpret this equation as the direction derivative of u:

$$D_V u := \nabla u \cdot V = \begin{bmatrix} u_x \\ u_t \end{bmatrix} \cdot \begin{bmatrix} a \\ 1 \end{bmatrix} = 0.$$

Here, the direction of the derivative is

$$V = \begin{bmatrix} a \\ 1 \end{bmatrix}.$$

Let us consider the integral curves of the vector field V:

$$\begin{bmatrix} dx \\ dt \end{bmatrix} \parallel \begin{bmatrix} a \\ 1 \end{bmatrix},$$

which is governed by the ODE

$$\frac{dx}{dt} = a(x,t).$$

Let $x(t,\xi)$ be its solution with initial data $x(0,\xi) = \xi$. Then

$$\begin{aligned} \left.\frac{d}{dt}\right|_\xi u(x(t,\xi),t) &= \partial_t u + \partial_x u \frac{dx}{dt} \\ &= u_t + a u_x = 0. \end{aligned}$$

In other words, u is unchanged along the curve: $dx/dt = a$. Such curves are called the characteristic curves of equation (5.1). Suppose the mapping

$$\xi \mapsto x(t,\xi)$$

is invertible, say $\xi(t,x)$ for any $t > 0$ and $x \in \mathbb{R}$. Then the solution to the Cauchy problem (5.1),(5.2) is given by $u(x,t) = u^0(\xi(t,x))$. Note that the signal propagates along characteristic curves.

3. **With source term**
Lastly, we study the linear advection equation with a source term:

$$u_t + a(x,t) u_x = f(x,t).$$

Along the forementioned characteristic curves $x(t,\xi)$, the equation becomes

$$\left.\frac{d}{dt}\right|_\xi u(x(t,\xi),t) = u_t + a u_x = f(x(t,\xi),t).$$

We integrate this equation in t with fixed ξ to obtain

$$u(x(t,\xi),t) = u^0(\xi) + \int_0^t f(x(s,\xi),s)\, ds.$$

This is a function in (ξ,t). The final solution is obtained by replacing ξ by $\xi(t,x)$.

4. **Example:** Let us consider
$$u_t - |x| u_x = 0.$$
The characteristics are
$$\frac{dx}{dt} = -|x|$$
Its solution is
$$x(t, \xi) = e^{-t}\xi.$$
Thus, the solution is
$$u(x, t) = u^0(\xi) = u^0(e^t x).$$

Homeworks

1. Find the solution of
$$u_t - (\tanh x) u_x = 0$$
with initial data u^0. Also show that $u(x, t) \to 0$ as $t \to \infty$, provided $u^0(x) \to 0$ as $|x| \to \infty$.

2. Show that the initial value problem for
$$u_t + (1 + x^2) u_x = 0$$
is not well-defined. (Show the characteristics issued from x-axis do not cover the entire domain: $x \in R, t \geq 0$.)

5.1.2 Linear hyperbolic systems of equations

1. **Hyperbolicity** Let us consider the following linear system of equations
$$u_t + A(x, t) u_x = B(x, t) u + f(x, t), \quad x \in \mathbb{R}, t > 0, \tag{5.3}$$
with initial data
$$u(x, 0) = u^0(x), \quad x \in \mathbb{R}.$$
Here, u is an n-vector and A, B are $n \times n$ matrices. The system is called *hyperbolic if A is diagonalizable with real eigenvalues*. That is, A has real eigenvalues
$$\lambda_1 \leq \cdots \leq \lambda_n$$
with left/right eigenvectors l_i / r_i:
$$l_i A = \lambda_i l_i, \quad A r_i = \lambda_i r_i,$$

respectively. Let us normalize $\|r_i\| = 1$, for $i = 1, ..., n$, then normalize l_i such that $l_i r_j = \delta_{ij}$. * Let

$$L := \begin{bmatrix} l_1 \\ \vdots \\ l_n \end{bmatrix}, \quad R := [r_1, \cdots, r_n], \quad \Lambda := \text{diag}(\lambda_1, \cdots, \lambda_n).$$

Then

$$LA = \Lambda L, \quad AR = R\Lambda, \quad LR = I.$$

The eigenvalues λ_i are wave speeds, the eigenvectors r_i are the wave modes.

2. **Method of characteristics**
We use L to diagonalize system (5.3). We multiply equation (5.3) by L from the left:

$$Lu_t + LAu_x = LBu + Lf.$$

By introducing $v = Lu$ and $u = Rv$†, we can rewrite the equation as follows:

$$v_t + \Lambda v_x = Cv + g,$$

where $C := LBR + L_t R + \Lambda L_x R$ and $g := Lf$. The i-th equation can be written as:

$$v_{i,t} + \lambda_i v_{i,x} = \sum_j c_{ij} v_j + g_i,$$

which represents an ODE in the direction of $dx/dt = \lambda_i(x,t)$. This is a linear advection equation with a characteristic speed λ_i. The vector field $(\lambda_i, 1)$ in x-t plane is known as the ith characteristic field. Its integral curves are referred to as the ith characteristic curves. For a given (x,t) with $t \geq 0$, let $y_i(s,t,x)$, $0 \leq s \leq t$ be the ith characteristic curve defined by:

$$\frac{dy}{ds} = \lambda_i(y,s), \quad 0 \leq s \leq t, \quad y(t,t,x) = x.$$

We then integrate the i-th equation along this curve to obtain

$$v_i(x,t) = v_i^0(y_i(0,t,x)) + \int_0^t \left[\sum_j c_{ij} v_j + g_i \right] (y_i(s,t,x), s)\, ds,$$

which represents a system of integral equations for v. Here, $v^0 = Lu^0$ and u^0 is the initial data of v. From these integral equations, we can draw some conclusions:

*From $\lambda_i l_i r_j = l_i A r_j = \lambda_j l_i r_j$. Thus, $l_i r_j = 0$ if $\lambda_i \neq \lambda_j$. We normalize l_j such that $l_j r_j = 1$. For multiple eigenvalue λ, we first select a set of basis (unit right eigenvectors) in the right invariant space of λ, then choose a et of left eigenvectors such that $l_i r_j = \delta_{ij}$.

†Note that $LR = I$ implies that both L and R are 1-1 and onto (a consequence of the rank-nullity theorem). This implies $RL = I$ as well.

- *The domain of dependence* of (x,t), denote by $D(x,t)$, is $[y_n(0,t,x), y_1(0,t,x)]$, which is a finite interval. This implies that if the initial data u^0 is zero on $D(x,t)$, then the solution $u(x,t)$ is identically zero.
- Another conclusion we can draw from these integral equations is the local existence theorem, which can be obtained if v^0 and v^0_x are bounded. The proof is similar to that of the local existence of ODEs.

  We define a function space $C_b(\mathbb{R})$, the bounded continuous functions on $\mathbb{R}$, using the sup norm: $\|v\|_\infty := \sup_x |v(x)|$. We then define the map $T : C_b(\mathbb{R}) \to C_b(\mathbb{R})$ as follows:

$$Tv = v_{0,i}(\xi_i(0,t,x)) + \int_0^t \left[\sum_j c_{ij} v_j + g_i\right](y_i(s,t,x), s)\, ds.$$

  The integral equation can then be written as finding a fixed point of T in the space $C_b(\mathbb{R})$, i.e., solving $v = Tv$.

  If the time is sufficiently short, then T is a contraction map in $C_b(\mathbb{R})$, and therefore the contraction mapping theorem yields a unique fixed point, which is the solution to the integral equation. This implies the existence of a local solution to the original PDE.
- The global existence theorem for the PDE (5.3) can be obtained using a priori estimates, such as C^1-estimates, based on the integral equations derived earlier. A sufficient condition for such a priori estimates is that $A(x,t)$ is bounded in the upper half plane in the x-t space.
- A necessary condition for global existence is that all characteristics emanating from any point (x,t), where $x \in \mathbb{R}$ and $t > 0$, should be traced back to the initial time.
- A nice reference for the method of characteristics for systems of hyperbolic equations in one dimension is Fritz John's book on PDEs, Sec. 5, Chapter 2.

3. **Example** Consider the wave equation

$$u_{tt} - c^2 u_{xx} = 0.$$

   This is a second order PDE. It can be rewritten as the following first order system:

$$\begin{bmatrix} u_x \\ u_t \end{bmatrix}_t - \begin{bmatrix} 0 & 1 \\ c^2 & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_t \end{bmatrix}_x = 0. \tag{5.4}$$

   Here, the first equation is the consistency equation (i.e. $u_{xt} = u_{tx}$), while the second equation is the original wave equation. The eigenvalues and the left/right eigenvectors of the matrix are

$$\lambda_1 = -c, \quad \ell_1 = (-c, 1), \quad r_1 = \frac{1}{\sqrt{1+c^2}} \begin{bmatrix} 1 \\ -c \end{bmatrix},$$

$$\lambda_2 = c, \quad \ell_2 = (c, 1), \quad r_2 = \frac{1}{\sqrt{1+c^2}} \begin{bmatrix} 1 \\ c \end{bmatrix}.$$

We multiply the system by ℓ_1 from the left to obtain

$$(u_t - cu_x)_t + c(u_t - cu_x)_x = 0$$

and by ℓ_2 to obtain

$$(u_t + cu_x)_t - c(u_t + cu_x)_x = 0.$$

Let $v_1 = u_t - cu_x$, and $v_2 = u_t + cu_x$. Then v_1 and v_2 satisfy the following linear advection equations:

$$v_{1,t} + cv_{1,x} = 0, \quad v_{2,t} - cv_{2,x} = 0.$$

They can be solved by previous characteristic method for the linear advection equation:

$$v_1(t, x) = v_1^0(x - ct), \quad v_2(t, x) = v_2^0(x + ct).$$

We thus obtain

$$u_t = \frac{v_1(t, x) + v_2(t, x)}{2}, \quad u_x = \frac{v_2(t, x) - v_1(t, x)}{2c}.$$

Once u_t and u_x are obtained, noting that $u_x dx + u_t dt$ is integrable from the consistency equation, we can obtain u by integrating the 1-form $u_x dx + u_t dt$.

5.1.3 *Linear symmetric hyperbolic systems in multi-dimensions

1. **Example of symmetric hyperbolic systems** For system (5.4), we can convert it to so-called symmetric hyperbolic system:

$$\begin{bmatrix} 1 & 0 \\ 0 & c^{-2} \end{bmatrix} \begin{bmatrix} u_x \\ u_t \end{bmatrix}_t - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} u_x \\ u_t \end{bmatrix}_x = 0.$$

We denote it as

$$A_0 u_t + A u_x = 0,$$

where A_0 and A are symmetric and A_0 is positive definite. Such a system is called a symmetric hyperbolic system.

2. **Symmetric hyperbolic system**[‡]
A linear symmetric hyperbolic system in d dimensions has the form:

$$A_0 u_t + \sum_{i=1}^{d} A_i(x, t) u_{x_i} = B(x, t)u + f, \tag{5.5}$$

[‡]The notion of Symmetric hyperbolic systems were proposed by K.O. Friedrichs. He notices most of physical systems can be expressed in this form. Such systems have nice mathematical structure and naturally meet physical requirements.

where u is an n-vector, A_0, A_i are $n \times n$ symmetric matrices and A_0 is positive definite. Given a direction $\xi \in \mathbb{R}^d$, the matrix $A(\xi) := \sum_{i=1}^d A_i \xi_i$ is symmetric. Its the eigenvalues $\lambda_k(\xi)$ and right eigenvector r_k are

$$\left(\sum_{i=1}^d A_i \xi_i \right) r_k = \lambda_k A_0 r_k, \quad k = 1, ..., n.$$

Below, we introduce another method, the energy method, for the existence and well-posedness. It is useful for stability analysis.

3. **Energy estimates**

   (a) We take inner product of this equation with u, later we integrate in x over the whole space. For simplicity, we assume A_0 and A are constant matrices temporarily. We get

   $$\frac{\partial}{\partial t} \frac{1}{2} A_0 u \cdot u + \sum_{i=1}^d \frac{\partial}{\partial x_i} \left(\frac{1}{2} A_i u \cdot u \right) = Bu \cdot u + f \cdot u.$$

   Here, we have used the symmetric properties of A_0 and A_i:

   $$\frac{\partial}{\partial x_i} (A_i u \cdot u) = A_i u_{x_i} \cdot u + A_i u \cdot u_{x_i} = 2 A_i u_{x_i} \cdot u.$$

   As we integrate in x over the whole space, we get

   $$\frac{d}{dt} \frac{1}{2} \langle A_0 u, u \rangle = \langle Bu, u \rangle + \langle f, u \rangle. \tag{5.6}$$

   The boundary term is gone because we assume $u \sim 0$ at ∞.

   (b) The positivity of A_0 yields that $\langle A_0 u, u \rangle$ is a norm

   $$|||u|||^2 := \langle A_0 u, u \rangle$$

   It is called the energy norm. It is equivalent to $\|u\|_2^2$, namely, there are two constants C_1 and C_2 such that for any $u \in L^2(\mathbb{R})$,

   $$C_1 \int |u|^2 \, dx \le \langle A_0 u, u \rangle \le C_2 \int |u|^2 \, dx.$$

   From (5.6), we get

   $$\frac{d}{dt} \frac{1}{2} |||u(t)|||^2 \le C |||u|||^2 + C' |||u||| \cdot \|f\|$$

   Here, we have used the boundedness of B. Eliminating $|||u|||$, we get

   $$\frac{d}{dt} |||u(t)||| \le C |||u||| + C' \|f\|$$

This yields (by the Gronwall inequality)

$$\boxed{|||u(t)||| \le e^{Ct}|||u(0)||| + C' \int_0^t e^{C(t-s)}\|f(s)\|\, ds}$$

Thus, $|||u(t)|||$ is bounded for any finite time if $\|u(0)\|$ is bounded. This is called the energy estimate for system (5.5).

(c) By differentiating the system in x, applying the same method to u_x, we can obtain energy estimate for u_x. This way can give us the boundedness of all derivatives of u, from which we can get compactness for approximate solutions and existence theorem. This is a standard theory for general symmetric hyperbolic systems in arbitrary dimensions. We refer the readers to Chapter 6 of John's book.

5.2 Finite difference methods for linear advection equation

5.2.1 Design procedure

We shall explain some design procedure for the linear advection equation:

$$u_t + au_x = 0. \tag{5.7}$$

We shall assume that $a > 0$ is a constant. Despite its simplicity, the linear advection equation is a prototype equation for designing numerical methods for nonlinear hyperbolic equations in multiple dimensions.

1. **The grids** We choose $h = \Delta x$ and $k = \Delta t$ to be the spatial and temporal mesh sizes, respectively. We discretize the x-t plane using the grid points (x_j, t^n), where $x_j = j\Delta x$ and $t^n = n\Delta t$. We use U_j^n to approximate $u(x_j, t^n)$. We also abbreviate $u(x_j, t^n)$ by u_j^n. To derive finite difference schemes, we use finite differences to approximate derivatives. We demonstrate spatial discretization first, followed by temporal discretization.

2. **Spatial discretization.** There are two important design principles in spatial discretization: interpolation and upwinding.

   (a) Derivatives are approximated by finite differences. For examples, $u_x(x_j)$ can be replaced by

   $$\frac{U_j - U_{j-1}}{h},\ \frac{U_{j+1} - U_{j-1}}{2h},\ \text{or}\ \frac{3U_j - 4U_{j-1} + U_{j-2}}{2h}.$$

   The first one is a one-sided, first-order finite difference. The second one is the central differencing, which is second order. The third one is a one-sided, second-order finite difference. These formulas can be obtained by making a Taylor expansion of u about x_j.

(b) Upwinding. We assume $a > 0$, which means that the information comes from the left. Therefore, it is reasonable to approximate $u_x(x_j)$ by the left-sided finite differencing:

$$\frac{U_j - U_{j-1}}{h} \quad \text{or} \quad \frac{3U_j - 4U_{j-1} + U_{j-2}}{2h}.$$

3. **Temporal discretization.**

   (a) Forward Euler: We replace $u_t(x_j, t^n)$ with $(U_j^{n+1} - U_j^n)/k$. By combining this with spatial upwinding finite differencing, we obtain the upwinding scheme.

   (b) Backward Euler: We replace $u_t(x_j, t^n)$ with $(U_j^{n+1} - U_j^n)/k$, but replace u_x with $(D_x)_j^{n+1}$, where D_x is a spatial finite difference.

   (c) Leap frog: We replace $u_t(x_j, t^n)$ with $(U_j^{n+1} - U_j^{n-1})/2k$.

4. **List of some finite difference schemes**
   Assuming $a > 0$, let $\sigma = ak/h$.

$$\begin{aligned}
\text{Upwind} &: U_j^{n+1} = U_j^n - \sigma(U_j^n - U_{j-1}^n) \\
\text{Lax-Friedrichs} &: U_j^{n+1} = \frac{U_{j+1}^n + U_{j-1}^n}{2} - \frac{\sigma}{2}(U_{j+1}^n - U_{j-1}^n) \\
\text{Backward Euler} &: U_j^{n+1} - U_j^n = \frac{\sigma}{2}(U_{j-1}^{n+1} - U_{j+1}^{n+1}) \\
\text{Lax-Wendroff} &: U_j^{n+1} = U_j^n - \frac{\sigma}{2}(U_{j+1}^n - U_{j-1}^n) + \frac{\sigma^2}{2}(U_{j+1}^n - 2U_j^n + U_{j-1}^n) \\
\text{Beam-Warming} &: U_j^{n+1} = U_j^n - \frac{\sigma}{2}(3U_j^n - 4U_{j-1}^n + U_{j-2}^n) + \frac{\sigma^2}{2}(U_j^n - 2U_{j-1}^n + U_{j-2}^n) \\
\text{MacCormack} &: U_j^{n+1} = U_j^n - \sigma(U_j^n - U_{j-1}^n) + \frac{\sigma(1-\sigma)}{2}(U_{j+1}^n - 2U_j^n + U_{j-1}^n).
\end{aligned}$$

5. In general, a (explicit) finite difference scheme for the linear advection equation can be written as

$$U_j^{n+1} = G(U_{j-l}^n, U_{j-l+1}^n, \cdots, U_{j+m}^n) = \sum_{k=-l}^{m} a_k U_{j+k}^n \tag{5.8}$$

   where a_k are coefficients and l and m are integers that define the stencil, or the set of grid points used to compute the numerical approximation at (x_j, t^{n+1}).

5.2.2 Courant-Friedrichs-Levy condition

1. **Numerical domain of dependence**
   For a finite difference scheme:

$$U_j^{n+1} = G(U_{j-\ell}^n, \cdots, U_{j+m}^n),$$

we can define numerical domain of dependence of (x_j, t^n) to be $[x_{j-n\ell}, x_{j+nm}]$ (denoted by $D(j,n)$). For instance, the numerical domain of upwind method is $[x_{j-n}, x_j]$. If $U_k^0 = 0$ on $D(j,n)$, then $U_j^n = 0$.

2. **CFL condition**
In order to have our finite difference schemes physically meaningful, a natural condition is

physical domain of dependence $\subset$ numerical domain of dependence.

This gives a constraint on the ratio of h and k. Such a condition is called the Courant-Friedrichs-Levy (C-F-L) condition. For the linear advection equation with $a > 0$, the condition is

$$\{x_j - ak\} \subset [x_{j-\ell}, x_{j+m}].$$

This leads to

$$-\ell h \leq -ak \quad \Rightarrow \quad \frac{ak}{\ell h} \leq 1$$

3. **Violating the CFL condition**

- If the CFL condition is violated, we can easily construct an initial datum which is zero in numerical domain of dependence of (x,t), but $u(x,t) \neq 0$. The finite difference scheme will produce 0 at (x,t). Thus, its limit is also 0. But the true solution $u(x,t)$ is not zero.
- We will see below that a scheme violating the CFL condition is *unstable*.

Below, we shall fix the ratio h/k during the analysis and take $h \to 0$ in the approximation procedure.

5.2.3 Consistency and Truncation Errors

1. **Truncation error**
Let us express our finite difference scheme as:

$$U^{n+1} = GU^n.$$

Given a smooth solution $u(x,t)$ to the PDE (5.7), we plug it into this finite difference equation, then make Taylor expansion of u about (jh, nk). For instance, we plug a smooth function u into the upwind scheme. Let

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

This error is called the *truncation error*. In general, the truncation error is defined as

$$\tau^n(h,k) = \frac{u^{n+1} - Gu^n}{k}.$$

2. **Consistency**
   A finite difference scheme is called consistent if $\tau(h,k) \to 0$ as $h,k \to 0$. Naturally, this is a minimal requirement for a finite difference scheme. If the scheme is expressed as

$$U_j^{n+1} = \sum_{k=-l}^{m} a_k U_{j+k}^n,$$

   then a necessary and sufficient condition for consistency is

$$\sum_{k=-l}^{m} a_k = 1.$$

   This is easy to see because the constant is a trivial solution.

3. **Order of a scheme**
   If $\tau = O(k^r)$, then the scheme is called order r. We can easily check that $\tau = O(k)$ for the upwind method by Taylor expansion about (x_j, t^n):

$$\begin{aligned}
\tau &= \frac{1}{k}\left(u_j^{n+1} - u_j^n - \sigma(u_j^n - u_{j-1}^n)\right) \\
&= \frac{1}{k}\left(u_t k + \frac{1}{2}u_{tt}k^2 - \frac{ak}{h}(-u_x h + \frac{1}{2}u_{xx}h^2)\right) + HOT \\
&= (u_t + au_x) + \frac{k}{2}\left(u_{tt} - \frac{ah}{k}u_{xx}\right) + HOT \\
&= (u_t + au_x) + \frac{k}{2}\left(a^2 u_{xx} - \frac{ah}{k}u_{xx}\right) + HOT \quad \because u_t = -au_x \\
&= (u_t + au_x) - \frac{h^2}{2k}\sigma(1-\sigma)u_{xx} + HOT
\end{aligned}$$

   The term $\frac{h^2}{2k}\sigma(1-\sigma)u_{xx}$ is $O(h)$ because we keep $\sigma = ak/h$ fixed. Thus, the upwind scheme is first order.

Homework Find the truncation errors of the schemes listed above.

5.2.4 Upwinding and numerical diffusion

1. **Upwinding** From the characteristic method, we have $u(x_j, t^{n+1}) = u(x_j - ak, t^n)$. We can approximate it by interpolating using neighboring grid points. For example, a linear interpolation at x_{j-1} and x_j gives

$$u_j^{n+1} \approx \frac{ak}{h}u_{j-1}^n + (1 - \frac{ak}{h})u_j^n.$$

The corresponding finite difference scheme is defined as

$$U_j^{n+1} = \frac{ak}{h} U_{j-1}^n + (1 - \frac{ak}{h}) U_j^n,$$

which is the well-known upwind scheme. Here, the spatial discretization is exactly the above one-sided, first-order finite differencing.

2. **Forward Euler introduces an anti-diffusion** The term $(u_j^{n+1} - u_j^n)/k$ in the forward Euler method introduces an *anti-diffusion term* $-\frac{a^2k}{2} u_{xx}$, namely,

$$\frac{u_j^{n+1} - u_j^n}{k} = u_t + \frac{k}{2} u_{tt} + O(k^2) = u_t + \frac{a^2 k}{2} u_{xx} + O(k^2).$$

Here, we have replaced u_{tt} with

$$u_{tt} = -a u_{xt} = a^2 u_{xx},$$

then approximate u_{xx} with central finite difference.

If we use

$$u_t \leftarrow \frac{u_j^{n+1} - u_j^n}{k}, \quad u_x \leftarrow \frac{u_{j+1}^n - u_{j-1}^n}{2h}.$$

Then we have

$$u_t + a u_x = -\frac{a^2 k}{2} u_{xx} + O(k^2) + O(h^2).$$

The term $-\frac{a^2k}{2} u_{xx}$ comes from the forward Euler method, is an anti-diffusion term.

We can obtain a second-order approximation of u_t by removing this anti-diffusion term:

$$u_t = \frac{U_j^{n+1} - U_j^n}{k} - \frac{k}{2h^2}(U_{j+1}^n - 2U_j^n + U_{j-1}^n) + O(k^2)$$

The last term is to cancel the anti-diffusion term $-\frac{a^2k}{2} u_{xx}$. It is called a *numerical diffusion*. For the u_x term, we also need to have second order approximation. We consider two cases.

(a) Lax-Wendroff scheme: we approximate u_x by the central finite difference:

$$u_x = \frac{U_{j+1}^n - U_{j-1}^n}{2h} + O(h^2).$$

The resulting scheme is

$$U_j^{n+1} - U_j^n = -\frac{ak}{2h}\left(U_{j+1}^n - U_{j-1}\right) + \frac{a^2k^2}{2h^2}(U_{j+1}^n - 2U_j^n + U_{j-1}^n).$$

This is a second-order scheme in both space and time. The scheme is the Lax-Wendroff scheme.

(b) Beam-Warming scheme: the term u_x is approximated using second-order upwinding finite differencing:

$$u_x = \frac{1}{2h}(3U_j^n - 4U_{j-1}^n + U_{j-2}^n) + O(h^2).$$

Here, the upwinding means that: for $a > 0$, the information comes from the left, and we use U_{j-2}, U_{j-1}, and U_j as our stencil. The resulting scheme is

$$U_j^{n+1} = U_j^n - \frac{\sigma}{2}(3U_j^n - 4U_{j-1}^n + U_{j-2}^n) + \frac{\sigma^2}{2}(U_j^n - 2U_{j-1}^n + U_{j-2}^n).$$

3. **Backward Euler introduces a diffusion** If the spatial finite differencing is high-order, say $\frac{\sigma}{2}(3U_j^n - 4U_{j-1}^n + U_{j-2}^n)$ for u_x, such scheme is unstable because the only second-order derivatives come from the above anti-diffusion term which is unstable. On the other hand, temporal backward Euler scheme introduces a diffusion term $\frac{a^2k}{2}u_{xx}$:

$$\frac{u_j^{n+1} - u_j^n}{k} = {u_t}_j^{n+1} - \frac{k}{2}{u_{tt}}_j^{n+1} + O(k^2) = u_t - \frac{a^2k}{2}{u_{xx}}_j^{n+1} + O(k^2).$$

4. **MacCormack scheme is an upwind scheme followed by an anti-diffusion step.** The MacCormack scheme can be described as a two-step process. In the first step, an upwind scheme is used to calculate the intermediate value U_j^*. The scheme is given by:

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

This step introduces numerical viscosity, which can be expressed as

$$\frac{1}{2}\frac{a\Delta t}{\Delta x}(1 - \frac{a\Delta t}{\Delta x})u_{xx}.$$

In the second step, a correction is applied to the intermediate value to obtain the final value U_j^{n+1}. The correction term can be written as:

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

This correction step removes the diffusion term introduced by the upwind scheme and is called the anti-diffusion term.

Alternatively, the MacCormack scheme can be expressed as a two-step process. In the first step, a forward upwind step is used to produce an intermediate value U^*. The forward upwind scheme contains advection and numerical viscosity terms, which introduce some error in the solution. In the second step, a backward upwind step is used to produce a corrected intermediate value U^{**}:

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

Both the forward and backward upwind schemes contain advection and numerical viscosity terms, but the advection terms cancel each other out. The difference between U_j^{**} and U_j^n is twice the numerical viscosity. The final value U_j^{n+1} is then calculated as:

$$U_j^{n+1} = U_j^* - \frac{1}{2}\left(U_j^{**} - U_j^n\right).$$

This correction step removes the numerical viscosity introduced by the upwind schemes, resulting in a more accurate solution.

In summary, the prediction step of the MacCormack scheme uses an upwind scheme to calculate an intermediate value, while the correction step uses a backward upwind scheme to remove the numerical viscosity introduced in the prediction step.

Homeworks

1. Derive the Lax-Wendroff scheme using the trick $u_{tt} = a^2 u_{xx}$ and the central finite difference.

2. Derive a finite difference scheme using method of characteristics and a quadratic interpolation at x_{j-2}, x_{j-1} and x_j. Is this scheme identical to the Beam-Warming scheme?

3. Do the same thing with cubic interpolation at $x_{j-2}, \cdots, x_{j+1}$?

4. Write a computer program using the above listed schemes to the linear advection equation. Use periodic boundary condition. The initial condition are

   (a) square wave,

   (b) hat function

   (c) Gaussian

   (d) $e^{-x^2/D} \sin mx$

   Refine the mesh by a factor of 2 to check the convergence rates.

5.2.5 Modified equations

We shall study the performance of a finite difference scheme for the linear advection equation. Consider the upwind scheme for the linear advection equation:

$$u_t + a u_x = 0. \tag{5.9}$$

1. **Modified equation for the upwind scheme** Let $u(x,t)$ be a smooth function of (5.9). Taking Taylor expansion of u, we obtain

$$u_j^{n+1} - G(u^n)_j = (u_t + a u_x)\Delta t - \frac{(\Delta x)^2}{2}(\sigma - \sigma^2) u_{xx} + O((\Delta t)^3).$$

The truncation error for the upwind method is $O(\Delta t)$ if u satisfies the linear advection scheme. However, if we fix Δx and Δt, then the error is $O(\Delta t^3)$, provided u satisfies

$$u_t + au_x - \nu u_{xx} = 0,$$

where

$$\nu = \frac{(\Delta x)^2}{2\Delta t}(\sigma - \sigma^2).$$

This equation is called the modified equation of the upwind scheme. *The solution of the finite difference equation is closer to the solution of this modified equation than the original equation.* The role of νu_{xx} is a diffusion term in the scheme.

- The term $-(\Delta x)^2/(\Delta t)\sigma^2 u_{xx}$ comes from the forward Euler approximation to u_t. It is an anti-diffusion term.
- The term $(\Delta x)^2/(\Delta t)\sigma u_{xx}$ comes from the upwind discretization for au_x. It is a diffusion term.
- The effective diffusion is νu_{xx}, where $\nu > 0$ is called numerical viscosity.

We observe that $\nu \geq 0$ if and only if $0 \leq \sigma \leq 1$, which is exactly the CFL condition for stability. This is consistent to the well-postedness of diffusion equations (i.e. $\nu \geq 0$).

2. **Effect of numerical viscosity**

   (a) The numerical viscosity will cause solution smoother, and will smear out discontinuities.

   (b) To see this, let us solve the Cauchy problem:

$$\begin{aligned} u_t + au_x &= \nu u_{xx} \\ u(x,0) = H(x) &:= \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases} \end{aligned}$$

   The function H is called the Heaviside function. The corresponding solution is given by

$$\begin{aligned} u(x,t) &= \frac{1}{\sqrt{4\pi\nu t}} \int_{-\infty}^{\infty} e^{-\frac{(x-at-y)^2}{4\nu t}} u(y,0)\, dy \\ &= \frac{1}{\sqrt{4\pi\nu t}} \int_{0}^{\infty} e^{-\frac{(x-at-y)^2}{4\nu t}} dy \\ &= \text{erf}((x-at)/\sqrt{4\nu t}), \end{aligned}$$

   where

$$\text{erf}(x) := \frac{2}{\sqrt{\pi}} \int_{-\infty}^{x} e^{-z^2}\, dz.$$

The error function has the property:

$$\lim_{x\to-\infty}\operatorname{erf}(x)=0,\quad \lim_{x\to\infty}\operatorname{erf}(x)=1,\quad \operatorname{erf}(x)-\frac{1}{2}\text{ is an odd function.}$$

Let $u_e(x,t)$ be the exact solution of $u_t+au_x=0$ with $u(x,0)=H(x)$. Then

$$|u_e(y+at,t)-u(y+at,t)|=\operatorname{erf}(-|y|/\sqrt{4\nu t}).$$

Hence,

$$\begin{aligned}\|u_e(\cdot,t)-u(\cdot,t)\|_{L^1} &= 2\int_{-\infty}^{0}\operatorname{erf}(\frac{y}{\sqrt{4\nu t}})\,dy\\ &= C\sqrt{\nu t}\end{aligned}$$

Since $\nu=O(\Delta t)$, we see that

$$\|u_e^n-u^n\|_{L^1}=O(\sqrt{\Delta t}).$$

On the other hand, if U is the solution of the finite difference equation, then we expect that $\|U^n-u^n\|_{L^1}=O(\Delta t)$, because it is first order. Indeed, it is only $O(\sqrt{\Delta t})$ and

$$\boxed{\|U^n-u_e^n\|_{L^1}=O(\sqrt{\Delta t}).}$$

Thus, *a first-order scheme is only half-order for "linear discontinuities."*

(c) One can also observe the smearing (averaging) of discontinuities from the finite difference scheme directly. In the upwind scheme, U_j^{n+1} may be viewed as weighted averages of U_j^n and U_{j-1}^n:

$$U_j^{n+1}=(1-\sigma)U_j^n+\sigma U_{j-1}^n.$$

If $U_{j-1}^n=0$ and $U_j^n=1$, then U_j^{n+1} is a value between 0 and 1. This is a smearing process (averaging process). The smearing process will spread out. The width of spread-out after n time steps is $(\sqrt{n}\Delta x)=O(\sqrt{\Delta t})$ from the estimate of binomial distribution.

(d) The numerical viscosity of the upwind and the Lax-Friedrichs schemes are:

- Upwind: $\mu=\Delta x\sigma(1-\sigma)$
- Lax-Friedrichs: $\mu=\Delta x\,(1-\sigma^2)$.

Note that the magnitude of the numerical viscosity of the upwind method is smaller than that of the Lax-Friedrichs method, because CFL condition $0\le\sigma\le 1$. The upwind method uses the information of characteristic speed whereas the Lax-Friedrichs does not use this.

3. **Modified equations for second-order schemes**
The modified equations for second-order schemes (Beam-Warming, Lax-Wendroff) have the following form

$$u_t + au_x = \mu u_{xxx}, \tag{5.10}$$

where

$$\mu \begin{cases} > 0 & \text{for Beam-Warming} \\ < 0 & \text{for Lax-Wendroff.} \end{cases}$$

The term μu_{xxx} is called the dispersion term. Below, we study the effect of this dispersion term for Heaviside initial data.

(a) By taking Fourier transform on equation (5.10) in x:

$$\hat{u}(\xi, t) := \int u(x,t) e^{-ix\xi}\, dx,$$

we get

$$\hat{u}_t = (-ia\xi - i\mu\xi^3)\hat{u} = -i\omega(\xi)\hat{u}.$$

Hence

$$u(x,t) = \frac{1}{2\pi} \int e^{i(x\xi - \omega(\xi)t)} \hat{u}(\xi, 0)\, d\xi. \tag{5.11}$$

(b) Let us consider the Heaviside function $H(x)$ as the initial data. It can be expanded as superposition of waves at different wave numbers. As these waves propagate according to (5.11), they form a wave package: a high frequency wave modulated by a low frequency wave (modulated wave). By the method of stationary phase, we see that the major contribution of the integral (5.11) is on the set when

$$\frac{d}{d\xi}(x\xi - \omega(\xi)t) = 0.$$

The correspond wave $e^{i(x-\omega'(\xi)t)}$ is the modulated wave. Its speed $\omega'(\xi)$ is called the group velocity v_p.

(c) The group velocities are

$$v_p = a + 3\mu\xi^2 \begin{cases} > a & \text{for Beam-Warming} \\ < a & \text{for Lax-Wendroff.} \end{cases}$$

Since $\mu < 0$ for the Lax-Wendroff, while $\mu > 0$ for the Beam-Warming, we observe that the wave package leaves behind (resp. ahead) the discontinuity in the Lax-Wendroff (resp. Beam-Warming).

(d) One can also observe this oscillation phenomena directly from the schemes. In Beam-Warming, we know that U_j^{n+1} is a quadratic interpolation of U_{j-2}^n, U_{j-1}^n and U_j^n. If $U_{j-2}^n = 0$, and $U_{j-1}^n = U_j^n = 1$, then the quadratic interpolation gives an overshoot at U_j^{n+1} (that is, $U_j^{n+1} > 1$). Similarly, in the Lax-Wendroff scheme, U_j^{n+1} is a quadratic interpolation of U_{j-1}^n, U_j^n and U_{j+1}^n. If $U_{j-1}^n = U_j^n = 0$, and $U_{j+1}^n = 1$, then $U_j^{n+1} < 0$ (an undershoot).

Homeworks 5.1. 1. Find the modified equations for the following schemes:

$$\begin{aligned}
\text{Lax-Friedrichs} &: u_t + au_x = \frac{(\Delta x)^2}{2\Delta t}(1-\sigma^2)u_{xx} \\
\text{Lax-Wendroff} &: u_t + au_x = \frac{(\Delta x)^2}{6}a(\sigma^2-1)u_{xxx} \\
\text{MacCormack} &: u_t + au_x = ??u_{xxx} \\
\text{Beam-Warming} &: u_t + au_x = \frac{(\Delta x)^2}{6}a(2-3\sigma+\sigma^2)u_{xxx}
\end{aligned}$$

2. Expand u up to u_{xxxx}, find the modified equation with the term u_{xxxx} for the Lax-Wendroff scheme and Beam-Warming. That is

$$u_t + au_x = \mu u_{xxx} + \kappa u_{xxxx}.$$

Show that the coefficient $\kappa < 0$ for both scheme if and only if the C-F-L stability condition.

3. Find the solution U_j^n of the upwind scheme with initial data $U_j^0 = \delta_{j0}$. (Hint: a binomial distribution.) Now, condider the Heaviside function as our initial data. Using the above solution formula, superposition principle and the Stirling formula, show that $\sum_j |u_j^n - U_j^n|\Delta x = O(\sqrt{n}\Delta x) = O(\sqrt{\Delta t})$.

4. Measure the width of the oscillation as a function of number of time steps n.

5.2.6 Lax's equivalence theorem

1. **Definition of stability**
Suppose U^n is generated from a finite difference scheme: $U^{n+1} = G(U^n)$, we wish the solution remain bounded under certain norm as the mesh size $\Delta t \to 0$, or equivalently, the time steps $n \to \infty$. Thus, we have the following definition. A scheme is called stable if $\|U^n\|$ remains bounded under certain norm $\|\cdot\|$ for all n.

2. **True error and convergence** Let u be an exact solution of some linear hyperbolic P.D.E. and U be the solution of a corresponding finite difference equation, We want to estimate the true error $e_j^n = u_j^n - U_j^n$.

First we estimate how much errors accumulate in one time step.

$$e^{n+1} := u^{n+1} - U^{n+1} = ke^n + Gu^n - GU^n = ke^n + Ge^n.$$

If we can have an estimate (called stability condition) like

$$\|GU\| \le \|U\| \tag{5.12}$$

under certain norm $\|\cdot\|$, then we obtain

$$\|u^n - U^n\| \le \|u^0 - U^0\| + k(\tau^{n-1} + \cdots + \tau^1).$$

From the consistency, we obtain $\|e^n\| \to 0$ as $k \to 0$. If the scheme is of order r, then we obtain

$$\|e^n\| \leq \|u^0 - U^0\| + O(k^r).$$

Thus, we have the following theorems.

Theorem 5.1 (Lax equivalence theorem)**.** *Given a linear hyperbolic partial differential equation. Then a consistent finite difference scheme is stable if and only if is is convergent.*

We have proven stability $\Rightarrow$ convergence. We shall prove the other part in the next section.

Theorem 5.2. *For smooth solutions, the associated true error computed by a finite difference scheme of order r is $O(k^r)$.*

5.2.7 Stability analysis

1. Since we only deal with smooth solutions in this section, the L^2-norm is a proper norm to our stability analysis. For linear hyperbolic systems with constant coefficients, the von Neumann analysis (via Fourier method) provides a necessary and sufficient condition for stability. For systems with variable coefficients, the Kreiss' matrix theorem provides characterizations of stability condition.

2. **The von Neumann stability analysis.**
   Given $\{U_j\}_{j\in\mathbb{Z}}$, we define

   $$\|U\|^2 = \sum_j |U_j|^2$$

   and its Fourier transform

   $$\hat{U}(\xi) = \frac{1}{2\pi} \sum U_j e^{-ij\xi}.$$

   The advantages of Fourier method for analyzing finite difference scheme are

   - the shift operator is transformed to a multiplier:

     $$\widehat{TU}(\xi) = e^{i\xi}\hat{U}(\xi),$$

     where $(TU)_j := U_{j+1}$;
   - the Parseval equility

     $$\|U\|^2 = \|\hat{U}\|^2 := \frac{1}{2\pi} \int_{-\pi}^{\pi} |\hat{U}(\xi)|^2 \, d\xi.$$

If a finite difference scheme is expressed as

$$U_j^{n+1} = (GU^n)_j = \sum_{i=-l}^{m} a_i (T^i U^n)_j,$$

then

$$\widehat{U^{n+1}} = \widehat{G}(\xi)\widehat{U^n}(\xi).$$

From the Parseval equality,

$$\begin{aligned} \|U^{n+1}\|^2 &= \|\widehat{U^{n+1}}\|^2 \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |\widehat{G}(\xi)|^2 \, |\widehat{U^n}(\xi)|^2 \, d\xi \\ &\leq \max_{\xi} |\widehat{G}(\xi)|^2 \frac{1}{2\pi}\int_{-\pi}^{\pi} |\widehat{U^n}(\xi)|^2 \, d\xi \\ &= |\widehat{G}|_\infty^2 \|U\|^2 \end{aligned}$$

Thus a necessary condition for stability is

$$|\widehat{G}|_\infty \leq 1. \tag{5.13}$$

Conversely, Suppose $|\widehat{G}(\xi_0)| > 1$, from$\widehat{G}$ being a smooth function in ξ, we can find ε and δ such that

$$|\widehat{G}(\xi)| \geq 1 + \varepsilon \text{ for all } |\xi - \xi_0| < \delta.$$

Let us choose an initial data U_0 in ℓ^2 such that $\widehat{U^0}(\xi) = 1$ for $|\xi - \xi_0| \leq \delta$. Then

$$\begin{aligned} \|\widehat{U^n}\|^2 &= \frac{1}{2\pi}\int_{\mathbb{T}} |\widehat{G}|^{2n}(\xi)|\widehat{U^0}|^2 \\ &\geq \int_{|\xi-\xi_0|\leq\delta} |\widehat{G}|^{2n}(\xi)|\widehat{U^0}|^2 \\ &\geq (1+\varepsilon)^{2n}\delta \to \infty \text{ as } n \to \infty \end{aligned}$$

The operator G^n is unbounded in $\|\cdot\|_2$ operator norm. It is a fact that it will not be bounded by any equivalent norm, which involves more analysis and will be omit here. Thus, the scheme can not be stable. We conclude the above discussion by the following theorem.

Theorem 5.3. *A finite difference scheme*

$$U_j^{n+1} = \sum_{k=-l}^{m} a_k U_{j+k}^n$$

with constant coefficients is stable if and only if

$$\widehat{G}(\xi) := \sum_{k=-l}^{m} a_k e^{-ik\xi}$$

satisfies

$$\max_{-\pi\le\xi\le\pi} |\widehat{G}(\xi)| \le 1. \tag{5.14}$$

3. **Example** As a simple example, we show that the scheme:

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

is unstable. The operator $G = 1 + \frac{\sigma}{2}(T - T^{-1})$. The corresponding $\widehat{G}(\xi) = 1 + i\sigma \sin\xi$, which cannot be bounded by 1 in magnitude. One the other hand, the Lax-Friedrichs scheme replaces U_j^n in the above scheme by the average $(U_{j-1}^n + U_{j+1}^n)/2$. The corresponding $\widehat{G}(\xi) = \cos\xi + i\sigma \sin\xi$, which is bounded by 1 in magnitude provided $|\sigma| \le 1$. The above replacement is equivalent to add a term $(U_{j-1}^n - 2U_j^n + U_{j+1}^n)/2$ to the right hand side of the above unstable finite difference. It then stabilizes the scheme. This quantity is called a numerical viscosity. We see the discussion in the next section.

Homeworks 5.2. 1. Compute the $\widehat{G}$ for the schemes: backward Euler, Lax-Friedrichs, Lax-Wendroff, MacCormack, and leap-frog.

5.3 Finite difference schemes for linear hyperbolic systems with constant coefficients

5.3.1 Some design techniques

We consider the linear hyperbolic system

$$u_t + Au_x = 0$$

with A being a constant $n \times n$ matrix.

1. The **Lax-Friedrichs scheme** is

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

Note that he last term is a dissipation term.

2. A modified L-F scheme as

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

where D is a positive constant matrix. D is chosen so that the scheme is stable by the von-Neumann analysis.

3. **Upwind scheme**

   (a) Using the left/right eigenvectors: $AR = R\Lambda$, $LA = \Lambda L$, $LR = I$, we can express $A = L^{-1}\Lambda L = R\Lambda L$. We decompose A into two parts:

$$\begin{aligned} A &= R\Lambda L \\ &= R(\Lambda^+ - \Lambda^-)L \\ &= A^+ - A^-. \end{aligned}$$

   Here, $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$ and $\Lambda^\pm$ are the positive/negative parts of Λ.

   (b) With this decomposition, we define the upwind scheme as

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

4. The **Lax-Wendroff scheme** is given by

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

5. The C-F-L condition for upwind, Lax-Friedrichs, Lax-Wendroff is

$$\boxed{\max_i |\lambda_i| \frac{\Delta t}{\Delta x} \le 1.}$$

Homeworks 5.3. 1. Find the modified equation for the above schemes.

2. What is the stability condition on D for the modified L-F scheme.

3. Write a computer code to compute the solution of the wave equation:

$$\begin{aligned} u_t &= v_x \\ v_t &= c^2 u_x \end{aligned}$$

   using upwind, modified L-F, L-W schemes. The initial data is chosen as those for the linear advection equation. Use the periodic boundary condition.

5.3.2 *Stability analysis

The definition of L^2-stability is that the L^2-norm of the solution of finite difference scheme

$$\sum_j |U_j^n|^2 \Delta x$$

is uniformly bounded.

This L^2-theory for smooth solutions was well developed in the 60s. First, Lax's equivalence theorem was originally proved for well-posed linear systems in multi-dimension. Thus, the essential issue for finite difference scheme is still the stability problem.

Let us suppose the system is expressed as

$$u_t = \sum_i A_i u_{x_i} + Bu + f$$

Here, A_i, B are constant matrices. We assume that the system is hyperbolic. This means that $\sum_i \xi A_i$ is diagonal with real eigenvalues. Suppose the corresponding finite difference scheme is expressed as

$$U^{n+1} = GU^n = \sum a_\alpha T^\alpha U^n.$$

Here, $\alpha = (\alpha_1, \cdots, \alpha_n)$ is multi-index, a_α are matrices. Consider the Fourier transform of G:

$$\widehat{G}(k) = \sum_\alpha a_\alpha e^{i \sum_m \alpha_m k_m \Delta x_m}$$

If we take Δx_m as a function of Δt, then $\widehat{G}$ is a function of $(k, \Delta t)$. Using $\widehat{G}$, we have

$$\widehat{U^n} = \widehat{G}^n \widehat{U^0}.$$

From the Parseval equality: $\|U\|^2 = \int |\hat{U}|^2$, we obtain that the stability of a scheme $U^{n+1} = GU^n$ is equivalent to $\|\widehat{G}^n\|$ is uniformly bounded. Von Neumann gave a necessary condition for stability for system case.

Theorem 5.4. *A necessary condition for stability is that all eigenvalues of $\widehat{G}(k, \Delta t)$ satisfies*

$$|\lambda_i(k, \Delta t)| \le 1 + O(\Delta t), \quad \textit{for all } k, \quad \textit{for all } \Delta t \le \tau.$$

Proof. The spectral radius of $\widehat{G}(k, \Delta t)$ is the maximum value of the absolute values of the its eigenvalues. That is,

$$\rho(\widehat{G}) := \max_i |\lambda_i|$$

Since there is an eigenvector v such that $|\widehat{G}v| = \rho|v|$, we have that

$$\rho \le \|\widehat{G}\| := \max_u \frac{|\widehat{G}u|}{|u|}.$$

Also, the eigenvalues of $\widehat{G}^n$ are λ_i^n. Hence we have

$$\rho(\widehat{G}^n) = \rho(\widehat{G})^n.$$

Combine the above two, we obtain

$$\rho(\widehat{G})^n \le \|\widehat{G}^n\|.$$

Now, if $\|\widehat{G}^n\|$ is uniformly bounded, say by a constant C depends on $t := n\Delta t$, then

$$\begin{aligned} \rho &\leq C^{1/n} \\ &\leq 1 + O(\Delta t). \end{aligned}$$

□

For single equation, we have seen that von Neumann condition is also a sufficient condition for stability.

In general, Kreiss provided characterization of matrices which are stable.

Definition 5.1. A family of matrices {A} is stable if there exists a constant C such that for all $A \in \{A\}$ and all positive integer n,

$$\|A^n\| \leq C.$$

Theorem 5.5 (Kreiss matrix theorem)**.** *The stability of $\{A\}$ is equivalent to each of the following statements:*

(i) There exists a constant C such that for all $A \in \{A\}$ and $z \in C, |z| > 1$, $(A - zI)^{-1}$ exists and satisfies

$$\|(A - zI)^{-1}\| \leq \frac{C}{|z| - 1}.$$

(ii) There exist constants C_1 and C_2 such that for all $A \in \{A\}$, there exists nonsingular matrix S such that (1) $\|S\|, \|S^{-1}\| \leq C_1$, and (2) $B = SAS^{-1}$ is upper triangular and its off-diagonal elements satisfy

$$|B_{ij}| \leq C_2 \min\{1 - |\kappa_i|, 1 - |\kappa_j|\}$$

where κ_i are the diagonal elements of B.

(iii) There exists a constant $C > 0$ such that for all $A \in \{A\}$, there exists a positive definite matrix H such that

$$C^{-1}I \leq H \leq CI$$

$$A^* H A \leq H$$

Remarks.

1. In the first statement, the spectral radius of A is bounded by 1.
2. In the second statement, it is necessary that all $|\kappa_i| \leq 1$.
3. The meaning of the last statement means that we should use the norm $\sum |U_j|^2 = \sum_j (HU_j, U_j)$ instead of the Euclidean norm. Then A^n is nonincreasing under this norm.

5.4 *Finite difference methods for linear hyperbolic systems with variable coefficients

1. **Necessary condition for stability**
   The essential issue is stability because Lax's equivalence theorem. Kreiss showed by an example that the local stability (i.e. the stability for the frozen coefficients) is neither necessary nor sufficient for overall stability of linear variable systems. However, if the system $u_t = Au$ with A being first order, Strang showed that the overall stability does imply the local stability. Therefore, *for linear first-order systems with variable coefficients, the von Neumann condition is also a necessary condition for the overall stability.*

2. **Monotone schemes are stable in L^2.** For sufficient condition, we need some numerical dissipation to damp the high frequency component from spatial inhomogeneity. To illustrate this, let us consider the following scalar equation:

   $$u_t + a(x)u_x = 0,$$

   and a finite difference scheme

   $$U^{n+1}(x) = A(x)U^n(x - \Delta x) + B(x)U^n(x) + C(x)U^n(x + \Delta x).$$

   For consistency, we need to require

   $$\begin{aligned} A(x) + B(x) + C(x) &= 1 \\ A(x) - C(x) &= a(x) \end{aligned}$$

   Now, we impose another condition for local stability:

   $$0 \le A(x), B(x), C(x) \le 1.$$

   We show stability result. Multiply the difference equation by $U^{n+1}(x)$, use Cauchy-Schwartz inequality, we obtain

   $$\begin{aligned} (U^{n+1}(x))^2 &= A(x)U^n(x - \Delta x)U^{n+1}(x) + B(x)U^n(x)U^{n+1}(x) + C(x)U^n(x + \Delta x)U^{n+1}(x) \\ &\le \frac{A(x)}{2}((U^n(x - \Delta x))^2 + (U^{n+1}(x))^2) + \frac{B(x)}{2}((U^n(x))^2 + (U^{n+1}(x))^2) \\ &\quad + \frac{C(x)}{2}((U^n(x + \Delta x))^2 + (U^{n+1}(x))^2) \\ &= \frac{A(x)}{2}(U^n(x - \Delta x))^2 + \frac{B(x)}{2}(U^n(x))^2 + \frac{C(x)}{2}(U^n(x + \Delta x))^2 + \frac{1}{2}(U^{n+1}(x))^2 \end{aligned}$$

   This implies

   $$(U^{n+1}(x))^2 \le A(x)(U^n(x - \Delta x))^2 + B(x)(U^n(x))^2 + C(x)(U^n(x + \Delta x))^2$$

$$= A(x-\Delta x)(U^n(x-\Delta x))^2 + B(x)(U^n(x))^2 + C(x+\Delta x)(U^n(x+\Delta x))^2$$
$$+(A(x)-A(x-\Delta x))(U^n(x-\Delta x))^2 + (C(x)-C(x+\Delta x))(U^n(x+\Delta x))^2$$

Now, we sum over $x = x_j$ for $j \in Z$. This yields

$$\|U^{n+1}\|^2 \le \|U^n\|^2 + O(\Delta t)\|U^n\|^2$$

Hence,

$$\|U^n\|^2 \le (1+O(\Delta t))^n \|U^0\|^2 \le e^{Kt}\|U^0\|^2.$$

The above analysis show that monotone schemes are stable in L^2. Indeed, the scheme has some dissipation to damp the errors from the variation of coefficient (i.e. the term like $(A(x)-A(x-\Delta x))$).

3. **High-order dissipation implies stability** For higher-order scheme, we need to estimate higher order finite difference ΔU, this will involves $|\Delta a|\|\Delta U\|$, or their higher order finite differences. We need some dissipation to damp the growth of this high frequency modes. That is, the *eigenvalues of the amplification matrix should satisfies*

$$|\lambda_i| \le 1 - \delta|k\Delta x|^{2r}, \ \textit{when } |k\Delta x| \le \pi$$

for some $\delta > 0$.

To be more precisely, we consider first-order hyperbolic system in high-space dimension:

$$u_t + \sum_{i=1}^{d} a_i(x) u_{x_i} = 0, \tag{5.15}$$

where $u \in \mathbb{R}^N$, a_i, $i = 1, ..., d$, are $N \times N$ matrices. Consider a finite difference approximation:

$$U^{n+1}(x) = \sum_{\alpha} A_\alpha(x) T^\alpha U^n(x). \tag{5.16}$$

Here, $\alpha = (\alpha_1, \cdots, \alpha_d)$ is a multi-index.

Let $\widehat{G}(x, \Delta t, \xi) = \sum_\alpha A_\alpha e^{i\alpha\cdot\xi}$ be the Fourier transform of the frozen finite difference operator.

Definition 5.2. A finite difference scheme with amplification matrix $\widehat{G}(x, \Delta t, \xi)$ is called dissipative of order $2r$ if there exists a constant $\delta > 0$ such that all eigenvalues of $\widehat{G}$ satisfy

$$|\lambda_i(x, \Delta t, \xi)| \le 1 - \delta|\xi|^{2r}$$

for all $\max_i |\xi_i| \le \pi$, all x, and all $\Delta t < \tau$ for some constant τ.

An important theorem due to Kreiss is the following stability theorem.

Theorem 5.6. *Suppose that system (5.15) is symmetric hyperbolic, meaning that the matrices a_i are symmetric, and that the coefficient matrices A_α of the finite difference scheme (5.16) are also symmetric. Assume that all coefficients are uniformly bounded. If the scheme is of order $2r-1$ and dissipative of order r, then the scheme is stable.*

Project 4

1. Implement the following methods for the linear advection equation, the inviscid Burgers equation and the gas dynamics to solve the Riemann problems in 1D. The methods are (1) Lax-Friedrichs, (2) Upwind, (3) Lax-Wendroff, (4) MacCormack, (5) Second-order Godunov, (6) WENO3. See Sec. 5.1, example 1 of Shu's note (pp. 54) for the Riemann data.

   (a) For linear advection equation and the Burgers equation, the domain is $[-2, 10]$, the initial datum is

   $$u(x,0) = \begin{cases} 1 & 0 < x < 1 \\ 0 & \text{elsewhere.} \end{cases}$$

   You can set periodic boundary condition.

   (b) For the gas dynamic equations, the domain is $[-5, 5]$. The initial data are the Riemann data. There are two test cases:

   - Sod test case

   $$(\rho_L, q_L, P_L) = (1, 0, 1), \quad (\rho_R, q_R, P_R) = (0.125, 0, 0.1)$$

   - Lax test case

   $$(\rho_L, q_L, P_L) = (0.445, 0.698, 3.528), \quad (\rho_R, q_R, P_R) = (0.5, 0, 0.571).$$

   You can set the Dirichlet boundary condition.

Chapter 6

Hyperbolic Conservation Laws

6.1 Scalar conservation laws

6.1.1 Physical models

Many partial differential equations are derived from physical conservation laws such as conservation of mass, momentum, energy, charges, etc. This class of PDEs is called conservation laws. The scalar conservation law is a conservation law with single equation. Below, we give three examples.

1. **Traffic flow model** An interesting model is the following traffic flow model on a high way. We use macroscopic model, which means that $\Delta x \approx 100$ m. Let ρ be the car density, u be the average car velocity. The car flux at a point x is the number of cars passing through x per unit time. In a time period Δt, the car which can pass x must be in the region $u(x,t)\Delta t$. Thus, the flux at x is $(\rho(x,t)u(x,t)\Delta t)/(\Delta t) = \rho(x,t)u(x,t)$. Now, consider an arbitrary region (a,b), we have

   the change of number of cars in $(a,b) =$ [the car flux at a] $-$ [the car flux at b].

   In mathematical formula, it reads

$$\begin{aligned} \frac{d}{dt}\int_a^b \rho(x,t)\,dx &= \rho(a,t)u(a,t) - \rho(b,t)u(b,t) \\ &= -\int_a^b (\rho u)_x\,dx. \end{aligned}$$

   This holds for any (a,b). Hence, we have

$$\rho_t + (\rho u)_x = 0. \tag{6.1}$$

   This equation is usually called the continuity equation in continuum mechanics. It is not closed because it involves two knowns ρ and u. Empirically, u can be teated as a

function of ρ which satisfies $u \to 0$ as $\rho \to \rho_{\max}$. For instance,

$$u(\rho) = u_{\max}(1 - \frac{\rho}{\rho_{\max}}),$$

if there is a upper velocity limit, or

$$u(\rho) = a \log(\rho_{\max}/\rho),$$

if there is no restriction of velocity. Furthermore, we can model u to depend on ρ_x as well. For instance,

$$u = u(\rho) - \nu \frac{\rho_x}{\rho}.$$

The quantity $\rho_x/\rho = -V_x/V$ is the negative expansion rate, where V is called the specific length, the space occupied by a car including the front and rear spaces (i.e. $V = 1/\rho$). If the expansion rate is positive, then the car train is rarefied. Thus, if the car number becomes denser (resp. rarefied), then the speed is reduced (resp. increased). Here, ν is the diffusion coefficient (viscosity) which is a positive number. Thus, the final equation is

$$\rho_t + f(\rho)_x = 0, \tag{6.2}$$

or

$$\rho_t + f(\rho)_x = \nu \rho_{xx}, \tag{6.3}$$

where $f(\rho) = \rho u(\rho)$.

2. **Burgers' equation** The Burgers equation is given by

$$u_t + \frac{1}{2}(u^2)_x = \varepsilon u_{xx}, \tag{6.4}$$

where ε is a positive constant. When $\varepsilon = 0$, this equation is called the inviscid Burgers equation. This equation is a prototype equation used to study conservation laws.

The following Hopf-Cole transformation linearizes this nonlinear equation. Let

$$\phi(x,t) := \int_{-\infty}^{x} u(y,t)\, dy, \quad v = e^{-\frac{1}{\varepsilon}\phi}.$$

Then ϕ satisfies the Hamilton-Jacobi equation

$$\phi_t + \frac{\phi_x^2}{2} = \frac{\varepsilon}{2}\phi_{xx},$$

and v satisfies heat equation:

$$v_t = -\frac{1}{\varepsilon}v, \quad v_x = -\frac{1}{\varepsilon}\phi_x v,$$

$$v_{xx} = -\frac{1}{\varepsilon}\phi_{xx} v + \left(\frac{1}{\varepsilon}\phi_x\right)^2 v.$$

Thus,

$$v_t = \frac{\varepsilon}{2} v_{xx} \quad \Leftrightarrow \quad \phi_t + \frac{\phi_x^2}{2} = \frac{\varepsilon}{2}\phi_{xx}.$$

The solution to the heat equation can be expressed as

$$v(x,t) = \frac{1}{\sqrt{2\pi\varepsilon t}} \int_{-\infty}^{\infty} e^{-\frac{(x-y)^2}{2\varepsilon t}} v(y,0)\, dy.$$

From

$$\phi = -\varepsilon \ln v, \quad u = \phi_x = -\varepsilon \frac{v_x}{v},$$

we can obtain an explicit form of the solution u as

$$\begin{aligned} u(x,t) &= -\varepsilon \frac{1}{v(x,t)} \frac{1}{\sqrt{2\pi\varepsilon t}} \int_{-\infty}^{\infty} \left(-\frac{x-y}{t\varepsilon}\right) e^{-\frac{(x-y)^2}{2t\varepsilon}} e^{-\frac{1}{\varepsilon}\phi(y,0)}\, dy \\ &= \int_{-\infty}^{\infty} \left(\frac{x-y}{t}\right) p_\varepsilon(x,y,t)\, dy, \end{aligned}$$

where

$$p_\varepsilon(x,y,t) = \frac{e^{-\frac{1}{\varepsilon} I(x,y,t)}}{\int_{-\infty}^{\infty} e^{-\frac{1}{\varepsilon} I(x,y,t)}\, dy}, \quad I(x,y,t) = \frac{(x-y)^2}{2t} + \phi(y,0).$$

Taking $\varepsilon \to 0+$, we obtain

$$u(x,t) = \frac{x - y(x,t)}{t},$$

where

$$y(x,t) := \arg\min_x I(x,y,t).$$

3. **Two-phase flow model** The Buckley-Leverett equation models how oil-water flow in a reservoir (a porous media). The unknown u is the saturation of water, $0 \le u \le 1$. The equation is given by

$$u_t + f(u)_x = 0, \quad f(u) = \frac{u^2}{u^2 + a(1-u^2)^2}.$$

where $a > 0$ is a constant. Unlike previous examples, the flux f here is a non-convex function.

6.1.2 Basic theory

1. **Blow-up and weak solutions**

(a) **Method of characteristics for smooth solutions** Let us consider scalar conservation law

$$u_t + f(u)_x = 0. \tag{6.5}$$

The equation can be viewed as a directional derivative $\partial_t + f'(u)\partial_x$ of u is zero. That implies u is constant along the characteristic curve

$$\frac{dx}{dt} = f'(u(x,t)).$$

This yields that the characteristic curve is indeed a straight line. Using this we can solve the Cauchy problem of (6.5) with initial data u_0 implicitly:

$$u = u_0(x - ut).$$

For instance, for inviscid Burgers' equation with $u_0(x) = x$, the solution u is given by $u = x - ut$, or $u = x/(1+t)$.

(b) **Weak solutions** The solution may blow up (i.e. $|u_x| \to \infty$) in finite time due to the intersection of characteristic curves. A shock wave (discontinuity) is formed. We have to extend our solution class to to include these discontinuous solutions. We can view (6.5) in "weak sense." That is, for every smooth test function ϕ with compact support in $R \times [0, \infty)$,

$$\int_0^\infty \int_{-\infty}^\infty \phi[u_t + f(u)_x]\, dx\, dt = 0.$$

We take integrate-by-part to obtain

$$\int_0^\infty \int_{-\infty}^\infty [\phi_t u + \phi_x f(u)]\, dx\, dt + \int_{-\infty}^\infty \phi(x,0)u(x,0)\, dx = 0, \tag{6.6}$$

In this formulation, it allows u to be discontinuous.

Definition 6.1. A function u is called a weak solution of (6.5) if it satisfies (6.6) for all smooth test function ϕ with compact support in $\mathbb{R} \times [0, \infty)$.

(c) **Jump conditions**

Lemma 6.1. *Suppose u is a weak solution with discontinuity across a curve $x(t)$. Suppose u is smooth on the two sides of $x(t)$. Then u satisfies the following jump condition across $x(t)$:*

$$\boxed{\frac{dx}{dt}[u] = [f(u)],} \tag{6.7}$$

where $[u] := u(x(t)+, t) - u(x(t)-, t)$.

Proof. Let us consider a fixed shock position $(t_0, x(t_0))$. Suppose $\dot{x}(t_0) = 0$. The shock is a standing shock at t_0. The fluxes on its two sides must be equal due to conservation law. That is

$$f(u(x(t_0)-, t_0) = f(u(x(t_0)+, t_0).$$

Suppose $\dot{x}(t_0) = \sigma \neq 0$. We make a change-of-variable: $x' = x - \sigma t$, $t' = t$. Then the resulting equation is

$$u_{t'} + (f(u) - \sigma u)_{x'} = 0.$$

For this equation, the shock is standing at $t' = t_0$. This gives

$$[f(u) - \sigma u] = 0,$$

which is

$$[f(u)] - \sigma[u] = 0,$$

the jump condition at $(t_0, x(t_0))$ for $\sigma \neq 0$. □

2. **Riemann problems** The Riemann problem is a Cauchy problem for equation (6.5) with the following initial data

$$u(x, 0) = \begin{cases} u_\ell & \text{for } x < 0 \\ u_r & \text{for } x > 0. \end{cases} \tag{6.8}$$

Here u_ℓ, u_r are two constant states. The importance of Riemann problems lies in the following reasons:

(i) Both equation (6.5) and the Riemann data (6.8) are invariant under the Galilean transform: $x \to \lambda x, t \to \lambda t$ for all $\lambda > 0$. If uniqueness holds, the solution to the Riemann problem is self-similar, that is, $u = u(x/t)$. This reduces the PDE problem to an ODE problem.

(ii) Discontinuities are common for nonlinear conservation laws. Hence, near a discontinuity, the Riemann problem is also generic locally.

(iii) For physical problems, the far-field states are typically two constant states. Due to hyperbolicity, we expect the solution to be a perturbation of the solution to the Riemann problem at large time. Therefore, Riemann problem is also generic globally.

When $f'' \neq 0$, say, $f'' > 0$, here are two important classes of solutions.

(a) Shock wave: $u_\ell \geq u_r$

$$u(x, t) = \begin{cases} u_\ell & \text{for } x < \sigma t \\ u_r & \text{for } x > \sigma t \end{cases} \tag{6.9}$$

where $\sigma = (f(u_r) - f(u_\ell))/(u_r - u_\ell)$.

(b) Rarefaction wave: $u_\ell < u_r$

$$u(x,t) = \begin{cases} u_\ell & \text{for } x < \lambda_\ell t \\ u & \text{for } \lambda_\ell < \lambda(u) = \frac{x}{t} < \lambda_r \\ u_r & \text{for } x > \lambda_r t \end{cases} \tag{6.10}$$

where $\lambda(u) = f'(u)$ is an increasing function.

These two solution are of fundamental importance. We shall denote them by (u_ℓ, u_r).

3. **Non-uniqueness problem** The weak solution is not unique. For instance, in the case of $u_\ell < u_r$, both (6.10) and (6.9) are weak solutions. Indeed, there are infinite many weak solutions to such a Riemann problem. Therefore, additional condition is needed to guarantee uniqueness. Such a condition is called an entropy condition.

Homeworks 6.1. 1. If f is convex and u_0 is increasing, then the Cauchy problem for equation (6.5) has global solution.

2. If f is convex and $u_0' < 0$ at some point, then $u_x \to -\infty$ at finite time.

3. Prove Lemma 6.1.

6.1.3 Entropy conditions

To find a suitable entropy condition for general hyperbolic conservation laws, let us go back to study the gas dynamic problems. The hyperbolic conservation laws are simplified equations. The original physical equations usually contain a viscous term νu_{xx}, as that in the Navier-Stokes equation. We assume the viscous equation has uniqueness property. Therefore let us make the following definition.

Definition 6.2. A weak solution is called admissible if it is the limit of

$$u_t^\varepsilon + f(u^\varepsilon)_x = \varepsilon u_{xx}^\varepsilon, \tag{6.11}$$

as $\varepsilon \to 0+$.

We shall label this condition by (A). In gas dynamics, the viscosity causes the physical entropy increases as gas particles passing through a shock front. One can show that such a condition is equivalent to the admissibility condition. Notice that this entropy increasing condition does not involve viscosity explicitly. Rather, it is a limiting condition as $\varepsilon \to 0+$. This kind of conditions is what we are looking for. For general hyperbolic conservation laws, there are many of them. We list some of them below.

(L) Lax's entropy condition: across a shock (u_ℓ, u_r) with speed σ, the Lax's entropy condition is

$$\boxed{\lambda_\ell > \sigma > \lambda_r.} \tag{6.12}$$

where λ_ℓ (resp. λ_r) is the left (resp. right) characteristic speed of the shock.
The meaning of this condition is that the information can only enter a shock and then disappear. Information cannot come out of a shock. Thus, if we draw a characteristic curve from any point (x,t) backward in time, we will always meet the initial axis. It cannot stop at a shock in the middle of time because that would violate the entropy condition. In other words, all information can be traced back to the initial time. This is a *causality property* and is also *time irreversible*, consistent with the second law of thermodynamics. However, Lax's entropy is only suitable for flux f with $f'' \neq 0$.

(OL) **Oleinik-Liu's entropy condition**: Let

$$\sigma(u,v) := \frac{f(u)-f(v)}{u-v}.$$

The Oleinik-Liu's entropy condition is that, across a shock

$$\sigma(u_\ell, v) \geq \sigma(u_\ell, u_r) \tag{6.13}$$

for all v between u_ℓ and u_r. This condition is applicable to nonconvex fluxes.

(GL) The above two conditions are conditions across a shock. Lax proposed another global entropy condition. First, he define entropy-entropy flux: a pair of function $(\eta(u), q(u))$ is called an entropy-entropy flux for equation (6.5) if (i) η is convex, and (ii) $q' = \eta' f'$. A weak solution $u(x,t)$ is said to satisfy entropy condition if for any entropy-entropy flux pair (η, q), $u(x,t)$ satisfies

$$\eta(u(x,t))_t + q(u(x,t))_x \leq 0 \tag{6.14}$$

in weak sense.

(K) Another global entropy proposed by Kruzkov is for any constant c,

$$\int_0^\infty \int_{-\infty}^\infty [|u-c|\phi_t + \text{sign}(u-c)(f(u)-f(c))\phi_x]\, dx \geq 0 \tag{6.15}$$

for all positive smooth ϕ with compact support in $\mathbb{R} \times (0,\infty)$. **(GL) $\Rightarrow$ (K):**
For any c, we choose $\eta(u) = |u-c|$, which is a convex function. One can check the corresponding $q(u) = \text{sign}(u-c)(f(u)-f(c))$. Thus, (K) is a special case of (GL). We may remark here that we can choose even simplier entropy-entropy flux:

$$\eta(u) = u \vee c,\ q(u) = f(u \vee c),$$

where $u \vee c := \max\{u, c\}$.

When the flux is convex, each of the above conditions is equivalent to the admissibility condition. When f is not convex, each but the Lax's entropy condition is equivalent to the admissibility condition.

We shall not provide general proof here. Rather, we study special case: the weak solution is only a single shock (u_ℓ, u_r) with speed σ.

Theorem 6.1. *Consider the scalar conservation law (6.5) with convex flux f. Let (u_ℓ, u_r) be its shock with speed σ. Then the above entropy conditions are all equivalent.*

Proof. **(L) $\Leftrightarrow$ (OL);**
We need to assume f to be convex. This part is easy. It follows from the convexity of f. We leave the proof to the reader.
(A) $\Leftrightarrow$ (OL):
We also need to assume f to be convex. Suppose (u_ℓ, u_r) is a shock. Its speed

$$\sigma = \frac{f(u_r) - f(u_\ell)}{u_r - u_\ell}.$$

We shall find a solution of (6.11) such that its zero viscosity limit is (u_ℓ, u_r). Consider a solution haing the form $\phi((x - \sigma t)/\varepsilon)$. In order to have $\phi \to (u_\ell, u_r)$, we need to require far field condition:

$$\phi(\xi) \to \begin{cases} u_\ell & \xi \to -\infty \\ u_r & \xi \to \infty \end{cases} \tag{6.16}$$

Plug $\phi((x - \sigma t)/\varepsilon)$ into (6.11), integrate in ξ once, we obtain

$$\phi' = F(\phi). \tag{6.17}$$

where $F(u) = f(u) - f(u_\ell) - \sigma(u - u_\ell)$. We find $F(u_\ell) = F(u_r) = 0$. This equation with far-field condition (6.16) is solvable if and only if, for all u between u_ℓ and u_r, (i) $F'(u) > 0$ when $u_\ell < u_r$, or (ii) $F'(u) < 0$ when $u_\ell > u_r$. One can check that (i) or (ii) is equivalent to (OL).

Next, we study global entropy conditions.
(A) $\Rightarrow$ (GL)
If u is an admissible solution. This means that it is the limit of u^ε which satisfy the viscous conservation law (6.11). Let (η, q) be a pair of entropy-entropy flux. Multiply (6.11) by $\eta'(u^\varepsilon)$, we obtain

$$\begin{aligned} \eta(u^\varepsilon)_t + q(u^\varepsilon)_x &= \varepsilon\eta'(u^\varepsilon)u^\varepsilon_{xx} \\ &= \varepsilon\eta(u^\varepsilon)_{xx} - \varepsilon\eta''(u^\varepsilon_x)^2 \\ &\le \varepsilon\eta(u^\varepsilon)_{xx} \end{aligned}$$

We multiply this equation by any positive smooth test function ϕ with compact support in $R \times (0, \infty)$, then integrate by part, and take $\varepsilon \to 0$, we obtain

$$\int_0^\infty \int_{-\infty}^\infty [\eta(u)\phi_t + q(u)\phi_x]\, dx\, dt \ge 0$$

This means that $\eta(u)_t + q(u)_x \le 0$ in weak sense.
(K) $\Rightarrow$ (OL) for single shock:
Suppose (u_ℓ, u_r) is a shock. Suppose it satisfies (K). We want to show it satisfies (OL). The condition (GL), as applied to a single shock (u_ℓ, u_r), is read as

$$-\sigma[\eta] + [q] \le 0.$$

Here, we choose $\eta = |u - c|$. The condition becomes

$$-\sigma(|u_r - c| - |u_\ell - c|) + \text{sign}(u_r - c)(f(u_r) - f(c)) - \text{sign}(u_\ell - c)(f(u_\ell) - f(c)) \le 0$$

Or

$$-\sigma(u_\ell, u_r)(|u_r - c| - |u_\ell - c|) + |u_r - c|\sigma(u_r, c) - |u_\ell - c|\sigma(u_\ell, c) \le 0 \tag{6.18}$$

We claim that this condition is equivalent to (OL). First, if c lies outside of u_ℓ and u_r, then the left-hand side of (6.18) is zero. So (6.18) is always true in this case. Next, if c lies betrween u_ℓ and u_r, one can easily check it is equivalent to (OL). □

6.1.4 *Riemann problems for non-convex fluxes

The Oleinik-Liu's entropy condition can be interpreted as the follows graphically. Suppose (u_ℓ, u_r) is a shock, then the condition (OL) is equivalent to one of the follows. Either $u_\ell > u_r$ and the graph of f between u_ℓ, u_r lies below the secant $(u_r, f(u_r)), (u_\ell, f(u_\ell))$. Or $u_\ell < u_r$ and the graph of f between u_ℓ, u_r lies above the secant $((u_\ell, f(u_\ell)), (u_r, f(u_r)))$. With this, we can construct the solution to the Riemann problem for non-convex flux as the follows.

- Case $u_\ell < u_r$: We connect $(u_\ell, f(u_\ell))$ and $(u_r, f(u_r))$ by a convex envelope of f (i.e. the largest convex function below f). The straight line of this envelope corresponds to an entropy shock. In the curved part, where $f'(u)$ increases, corresponds to a centered rarefaction wave. Note that $f'(u^*) = \sigma$ at the intersection of the straight line and the curved portion, meaning that the shock speed is the same as the characteristic speed of the rarefaction wave. We call such a shock a *contact sock*. The solution is a composition of rarefaction waves and contact shocks, and is called a composite wave.

- Case $u_\ell \ge u_r$: We simply replace convex envelope by a concave envelope. The portions of straight lines of the concave envelop correspond to shocks, while the concave curved portions are the center rarefaction waves. The solution is a composition wave.

Example. Consider the cubic flux: $f(u) = \frac{1}{3}u^3$.

- Case $u_\ell < 0$, $u_r > 0$: From u_ℓ, we can draw a line tangent to the graph of f at $u_\ell^* = -u_\ell/2$.
  - If $u_r > u_\ell^*$: The wave structure is a contact shock (u_ℓ, u_ℓ^*) followed by a rarefaction wave (u_ℓ^*, u_r).
  - If $u_r \le u_\ell^*$, then the wave is a single shock.

- Case $u_\ell > 0$, $u_r < 0$:

Homeworks 6.2. 1. For the flux $f(u) = u^3/3$, construct the general solution to the Riemann problem for general left/right states u_ℓ andu_r.

6.1.5 *Uniqueness and Existence

Theorem 6.2 (Kruzkov)**.** *Assume f is Lipschitz continuous and the initial data u_0 is in $L^1 \cap BV$. Then there exists a global entropy solution (satisfying condition (K)) to the Cauchy problem for (6.5). Furthermore, the solution operator is contractive in L^1, that is, if u, v are two entropy solutions, then*

$$\|u(t) - v(t)\|_{L^1} \leq \|u(0) - v(0)\|_{L^1} \tag{6.19}$$

As a consequence, we have uniqueness theorem and the total variation diminishing property:

$$T.V.u(\cdot, t) \leq T.V.u(\cdot, 0) \tag{6.20}$$

Proof. The part of total variation diminishing is easy. We prove it here. The total variation of u is defined by

$$T.V.u(\cdot, t) = \mathrm{Sup}_{h>0} \int \frac{|u(x+h,t) - u(x,t)|}{h}\, dx$$

We notice that if $u(x,t)$ is an entropy solution, so is $u(x+h,t)$. Apply the contraction estimate for $u(\cdot,t)$ and $v = u(\cdot + h, t)$. We obtain the total variation diminishing property.

To prove the L^1-contraction property, we claim that the constant c in the Kruzhkov entropy condition **(K)** can be replaced by any other entropy solution $v(t,x)$. That is

$$\iint [|u(t,x) - v(t,x)|\psi_t + \mathrm{sign}(u(t,x) - v(t,x))(f(u(t,x)) - f(v(t,x)))\psi_x]\, dx\, dt \geq 0$$

for all positive smooth ψ with compact support in $\mathbb{R} \times [0, \infty)$. To see this, we choose a test function $\phi(s,x,t,y)$, the entropy conditions for u and v are

$$\iint [|u(s,x) - k|\phi_s(s,x,t,y) + \mathrm{sign}(u(s,x) - k)(f(u(s,x)) - f(k))\phi_x(s,x,t,y)]\, dx\, ds \geq 0$$

$$\iint [|v(t,y) - k'|\phi_t(s,x,t,y) + \mathrm{sign}(v(t,y) - k')(f(v(t,y)) - f(k'))\phi_y(s,x,t,y)]\, dx\, ds \geq 0$$

Set $k = v(t,y)$ in the first equation and $k' = u(s,x)$ in the second equation. Integrate the rest variables and add them together. We get

$$\iiiint \{|u(s,x) - v(t,y)|(\phi_s + \phi_t) + \mathrm{sign}(u(s,x) - v(t,y)) \cdot [f(u(s,x)) - f(v(t,y))] \cdot (\phi_x + \phi_y)\}\, dx\, ds\, dy\, dt \geq$$

Now we choose $\phi(s,x,t,y)$ such that it concentrates at the diagonal $s = t$ and $x = y$. To do so, let $\rho_h(x) = h^{-1}\rho(x/h)$ be an approximation of the Dirac mass measure. Let $\psi(T,X)$ be a non-negative test function on $(0,\infty) \times \mathbb{R}$. Choosing

$$\phi(s,x,t,y) = \psi\left(\frac{s+t}{2}, \frac{x+y}{2}\right) \rho_h\left(\frac{s-t}{2}\right) \rho_h\left(\frac{x-y}{2}\right),$$

we get

$$\iiint \rho_h\left(\frac{s-t}{2}\right)\rho_h\left(\frac{x-y}{2}\right)\left\{|u(s,x)-v(t,y)|\psi_T\left(\frac{s+t}{2},\frac{x+y}{2}\right)\right.$$
$$\left.+\text{sign}(u(s,x)-v(t,y))\cdot[f(u(s,x))-f(u(v(t,y)))]\cdot\psi_X\left(\frac{s+t}{2},\frac{x+y}{2}\right)\right\}dx\,dy\,ds\,dt\geq 0.$$

Now taking limit $h\to 0$, we can get the desired inequality.

Next, we choose

$$\psi(t,x)=[\alpha_h(t)-\alpha_h(t-\tau)]\cdot[1-\alpha_h(|x|-R+L(\tau-t))],$$

where $\alpha_h(z)=\int_{-\infty}^{z}\rho_h(s)\,ds$. We can get the desired L^1 contraction estimate. □

The existence theorem mainly based on the same proof of the uniqueness theorem. Suppose the initial data is in $L^1\cap L^\infty\cap BV$, we can construct a sequence of approximate solutions which satisfy entropy conditions. They can be construncted by finite difference methods (see the next section), or by viscosity methods, or by wave tracking methods (by approximate the flux function by piecewise linear functions). Let us suppose the approximate solutions are constructed via viscosity method, namely, u^ε are solutions of

$$u_t^\varepsilon+f(u^\varepsilon)_x=\varepsilon u_{xx}^\varepsilon.$$

Following the same proof for $(GL)\Rightarrow(K)$, we can get that the total variation norms of the approximate solutions u^ε are bounded by $T.V.u_0$. This gives the compactness in L^1 and a convergent subsequence leads to an entropy solution.

Remark. The general existence theorem can allow only initial data $u_0\in L^1\cap L^\infty$. Even the initial data is not in BV, the solution immediately has finite total variation at any $t>0$.

6.2 Systems of Hyperbolic Conservation Laws

6.2.1 Hyperbolicity

We consider the following system of PDEs in one space dimension:

$$u_t+f(u)_x=0,\quad u=\begin{bmatrix}u_1\\u_2\\\vdots\\u_n\end{bmatrix},\quad f:\mathbb{R}^n\to\mathbb{R}^n\ \text{(called flux)}\tag{6.21}$$

System (6.21) is called *hyperbolic* if $\forall u$, the $n\times n$ matrix $f'(u)$ is diagonalizable with real eigenvalues

$$\lambda_1(u)\leq\lambda_2(u)\leq\cdots\leq\lambda_n(u).$$

Let us denote the corresponding left/right eigenvectors by $\ell_i(u)/r_i(u)$, $i=1,...,n$, respectively. We can normalize them with $\|r_i\|=1$, $i=1,...,n$ and $\ell_i r_j=\delta_{ij}$.

6.2.2 Elementary waves and Riemann problems

1. **Self-similar solutions**

   (a) Note that the system is Galilean invariant, namely, the equation is unchanged under the transform:

   $$t \to \lambda t, \qquad x \to \lambda x, \qquad \forall \lambda > 0.$$

   This suggests that we can look for special solutions of the form $u(\frac{x}{t})$.

   (b) Let us plug $u(\frac{x}{t})$ into (6.21). It gives

   $$u' \cdot (-\frac{x}{t^2}) + f'(u)u' \cdot \frac{1}{t} = 0$$
   $$\Longrightarrow \qquad f'(u)u' = \frac{x}{t}u'.$$

   Note that $f'(u)$ is an $n \times n$ matrix, while u' is an n-vector. The above formula states that u' is an eigenvector of the matrix $f'(u)$ with eigenvalue x/t. Thus, there exists i such that $\frac{x}{t} = \lambda_i(u(\frac{x}{t}))$, an eigenvalue of $f'(u)$, and $u' = r_i(u)$ is the corresponding eigenvector.

   (c) To find such a solution, we first construct the integral curve of $r_i(u)$: $u' = r_i(u)$. Let $\mathfrak{R}_i(u_0, s)$ be the integral curve of $r_i(u)$ passing through u_0, parameterized by its arc-length. Along $\mathfrak{R}_i$, the variation of the speed λ_i is:

   $$\frac{d}{ds}\lambda_i(\mathfrak{R}_i(u_0, s)) = \nabla\lambda_i \cdot \mathfrak{R}_i' = \nabla\lambda_i \cdot r_i.$$

   (d) We have the following definition.

   Definition 6.3. The i-th characteristic field is called

   - genuinely nonlinear if $\nabla\lambda_i(u) \cdot r_i(u) \neq 0 \quad \forall u$.
   - linearly degenerate if $\nabla\lambda_i(u) \cdot r_i(u) \equiv 0 \quad \forall u$.
   - non-genuinely nonlinear if $\nabla\lambda_i(u) \cdot r_i(u) = 0$ on isolated hypersurfaces in $\mathbb{R}^n$.

   For scalar equation, the genuine nonlinearity is equivalent to the convexity (or concavity) of the flux f, linear degeneracy corresponds to $f(u) = au$, and the non-genuine nonlinearity is existence of inflection points of f.

2. **Rarefaction Waves** When the i-th field is genuinely nonlinear, we define

   $$\mathfrak{R}_i^+(u_0) = \{u \in \mathfrak{R}_i(u_0) | \lambda_i(u) \geq \lambda_i(u_0)\}.$$

   For any $u_1 \in \mathfrak{R}_i^+(u_0)$, note that $\lambda_i(u_0) < \lambda_i(u_1)$, we construct the following centered rarefaction wave, denoted it by (u_0, u_1), as:

   $$(u_0, u_1)\left(\frac{x}{t}\right) = \begin{cases} u_0 & \text{if } \frac{x}{t} \leq \lambda_i(u_0) \\ u_1 & \text{if } \frac{x}{t} \geq \lambda_i(u_1) \\ u & \text{if } \lambda_i(u_0) \leq \frac{x}{t} \leq \lambda_i(u_1) \text{ and } \lambda_i(u) = \frac{x}{t}. \end{cases}$$

   It is easy to check this is a solution. We call (u_0, u_1) an *i-rarefaction wave*.

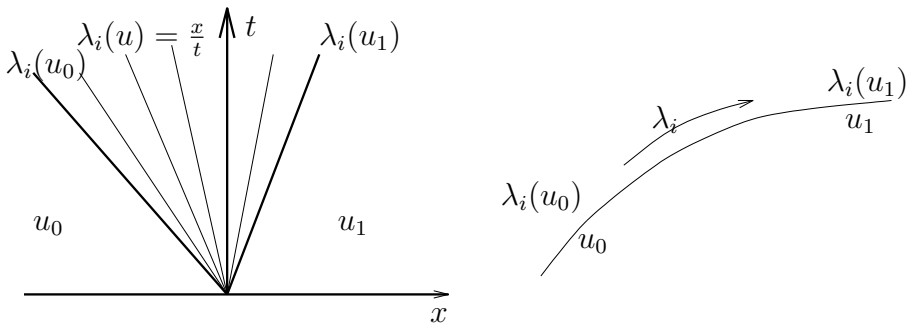


Figure 6.1: The integral curve of $u' = r_i(u)$ and the rarefaction wave.

3. **Shock Waves** A shock wave depends on three parameters: (u_0, u_1, σ), and is expressed as:

$$u\left(\frac{x}{t}\right) = \begin{cases} u_0 & \text{for } \frac{x}{t} < \sigma \\ u_1 & \text{for } \frac{x}{t} > \sigma. \end{cases}$$

Here, σ is a constant, representing the shock speed. The three quantities (u_0, u_1, σ) should satisfy the jump condition:

$$f(u_1) - f(u_0) = \sigma(u_1 - u_0). \tag{6.22}$$

Lemma 6.2. *(Local structure of shock waves)*

1. *The solution of (6.22) for (u, σ) consists of n algebraic curves passing through u_0 locally, named them by $\mathfrak{S}_i(u_0), i = 1, \cdots, n$.*
2. *$\mathfrak{S}_i(u_0)$ is tangent to $\mathfrak{R}_i(u_0)$ up to second order, i.e., $\mathfrak{S}_i^{(k)}(u_0) = \mathfrak{R}_i^{(k)}(u_0), k = 0, 1, 2$. Here, the derivatives are arclength derivatives.*
3. *$\sigma_i(u_0, u) \to \lambda_i(u_0)$ as $u \to u_0$, and $\sigma_i'(u_0, u_0) = \frac{1}{2}\lambda_i'(u_0)$*

Proof. (a) Let $\mathfrak{S}(u_0) = \{u | f(u) - f(u_0) = \sigma(u - u_0) \text{ for some } \sigma \in \mathbb{R}\}$. We claim that $\mathfrak{S}(u_0) = \bigcup_{i=1}^{n} \mathfrak{S}_i(u_0)$, where $\mathfrak{S}_i(u_0)$ is a smooth curve passing through u_0 with tangent $r_i(u_0)$ at u_0. To show this, first we express

$$\begin{aligned} f(u) - f(u_0) &= \left[\int_0^1 f'(u_0 + t(u - u_0)\, dt\right](u - u_0) \\ &= \tilde{A}(u_0, u)(u - u_0). \end{aligned}$$

When u is on $\mathfrak{S}(u_0)$, we can rewrite the jump condition as

$$\tilde{A}(u_0, u)(u - u_0) = \sigma(u - u_0).$$

$$\therefore \quad u \in \mathfrak{S}(u_0) \Longleftrightarrow (u - u_0) \text{ is an eigenvector of } \tilde{A}(u_0, u). \tag{6.23}$$

Assume $A(u_0) = f'(u_0)$ has real and distinct eigenvalues $\lambda_1(u_0) < \cdots < \lambda_n(u_0)$. The matrix $\tilde{A}(u_0, u)$ is a perturbation of $A(u)$. From the perturbation theory, for $u \sim u_0$, $\tilde{A}(u_0, u)$ also has real and distinct eigenvalues. Let us denote them by $\tilde{\lambda}_1(u_0, u) < \cdots < \tilde{\lambda}_n(u_0, u)$, with left/right eigenvectors $\tilde{\ell}_i(u_0, u)$ and $\tilde{r}_i(u_0, u)$, respectively. And they converge to $\lambda_i(u_0), \ell_i(u_0), r_i(u_0)$ as $u \to u_0$, respectively.

(b) Let us normalize the eigenvectors by $\|\tilde{r}_i\| = 1$, $\tilde{\ell}_i \tilde{r}_j = \delta_{ij}$. The eigenvector $\tilde{r}_i$, which is parallel to r_i at $u = u_0$, can be determined by

$$\tilde{\ell}_k(u_0, u)(u - u_0) = 0 \text{ for } k \neq i, k = 1, \cdots, n. \tag{6.24}$$

Thus, we can define

$$\mathfrak{S}_i(u_0) = \{u | \tilde{\ell}_k(u_0, u)(u - u_0) = 0, k \neq i, k = 1, \cdots, n\}.$$

We claim this is a smooth curve passing through u_0. Let us choose coordinate system $r_1(u_0), \cdots, r_n(u_0)$. Differentiate this equation $\tilde{\ell}_k(u_0, u)(u - u_0) = 0$ at $u = u_0$ in $r_j(u_0)$ direction:

$$\frac{\partial}{\partial r_j}\bigg|_{u=u_0} (\tilde{\ell}_k(u_0, u)(u - u_0)) = \tilde{\ell}_k.(u_0, u_0) \cdot r_j(u_0) = \delta_{jk},$$

Thus, this is the Jacobian matrix of the map: $u \mapsto (\tilde{\ell}_k(u_0, u)(u - u_0), k \neq i)$ at u_0, which is an $(n-1) \times n$ full rank matrix. By the implicit function theorem, the set $\mathfrak{S}_i(u_0)$ is a smooth curve passing through u_0.

(c) Note that $\mathfrak{R}_i(u_0) = u_0 = \mathfrak{S}_i(u_0)$.

$$f(u) - f(u_0) = \sigma_i(u_0, u)(u - u_0) \quad \forall u \in \mathfrak{S}_i(u_0)$$

Take arclength derivative along $\mathfrak{S}_i(u_0)$

$$f'(u)u' = \sigma_i'(u - u_0) + \sigma_i u' \text{ and } u' = S_i'.$$

As $u \to u_0$

$$f'(u_0)S_i'(u_0) = \sigma_i(u_0, u_0)S_i'(u_0)$$

$$\implies \qquad S_i'(u_0) = r_i(u_0) \text{ and } \sigma_i(u_0, u_0) = \lambda_i(u_0).$$

Consider the second derivative.

$$(f''(u)u', u') + f'(u)u'' = \sigma_i''(u - u_0) + 2\sigma_i' \cdot u' + \sigma_i u''$$

At $u = u_0$, $u' = \mathfrak{S}_i'(u_0) = \mathfrak{R}_i'(u_0) = r_i(u_0)$ and $u'' = \mathfrak{S}_i''(u_0)$,

$$\implies \qquad (f'' r_i, r_i) + f' \mathfrak{S}_i'' = 2\sigma_i' r_i + \sigma_i \mathfrak{S}_i''$$

On the other hand, we take derivative of $f'(u)r_i(u) = \lambda_i(u)r_i(u)$ along $\mathfrak{R}_i(u_0)$, then evaluate at $u = u_0$.

$$(f''r_i, r_i) + f'(\nabla r_i \cdot r_i) = \lambda_i' r_i + \lambda_i \nabla r_i \cdot r_i,$$

where $\nabla r_i \cdot r_i = \mathfrak{R}_i''$.

$$\Longrightarrow \qquad (f' - \lambda_i)(\mathfrak{S}_i'' - \mathfrak{R}_i'') = (2\sigma_i' - \lambda_i')r_i$$

Taking inner product with ℓ_i leads to

$$2\sigma_i' = \lambda_i'.$$

Let $\mathfrak{S}_i'' - \mathfrak{R}_i'' = \sum_k \alpha_k r_k(u_0)$. Taking inner product with ℓ_k leads to

$$\sum_{k \neq i} (\lambda_k - \lambda_i)\alpha_k r_k = 0 \Longrightarrow \alpha_k = 0 \ \forall k \neq i$$

On the other hand, from $(\mathfrak{R}_i', \mathfrak{R}_i') = 1$ and $(\mathfrak{S}_i', \mathfrak{S}_i') = 1$, we get $(\mathfrak{R}_i'', \mathfrak{R}_i') = 0$ and $(\mathfrak{S}_i'', \mathfrak{S}_i') = 0$. Since $\mathfrak{R}_i' = \mathfrak{S}_i' = r_i$, we then get

$$(\mathfrak{S}_i'' - \mathfrak{R}_i'', r_i) = 0.$$

Hence $\mathfrak{S}_i'' = \mathfrak{R}_i''$ at u_0.

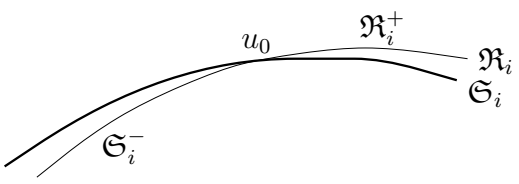
$$\begin{aligned} \because \quad & (\mathfrak{R}_i', \mathfrak{R}_i') = 1 \quad (\mathfrak{S}_i', \mathfrak{S}_i') = 1 \\ \text{and} \quad & (\mathfrak{R}_i'', \mathfrak{R}_i') = 0 \quad (\mathfrak{S}_i'', \mathfrak{S}_i') = 0 \\ \therefore \quad & (\mathfrak{R}_i'' - \mathfrak{S}_i'') \perp r_i \\ \therefore \quad & \alpha_i = 0 \end{aligned}$$

Hence $\mathfrak{R}_i'' = \mathfrak{S}_i''$ at u_0.

□

4. Let $\mathfrak{S}_i^-(u_0) = \{u \in \mathfrak{S}_i(u_0) | \lambda_i(u) \leq \lambda_i(u_0)\}$.
   If $u_1 \in \mathfrak{S}_i^-(u_0)$, define

$$(u_0, u_1) = \begin{cases} u_0 & \text{for } \frac{x}{t} < \sigma_i(u_0, u_1) \\ u_1 & \text{for } \frac{x}{t} > \sigma_i(u_0, u_1) \end{cases}$$



   (u_0, u_1) is a weak solution.

Suppose the i-th characteristic field is genuinely nonlinear. The Lax entropy condition reads

$$\lambda_i(u_0) > \sigma_i(u_0, u_1) > \lambda_i(u_1) \tag{6.25}$$

Let us define $\mathfrak{S}_i^-(u_0)$ to be the branch of $\mathfrak{S}_i(u_0)$ which satisfies entropy condition:

$$\mathfrak{S}_i(u_0) := \{u \in \mathfrak{S}_i(u_0) | \lambda_i(u) < \lambda_i(u_0)\}$$

Then for $u_1 \in \mathfrak{S}_i^-(u_0)$, and $u_1 \sim u_0$, (6.25) is always valid. This follows easily from $\lambda_i = 2\sigma_i'$ and $\sigma_i(u_0, u_0) = \lambda_i(u_0)$. For $u_1 \in \mathfrak{S}_i^-(u_0)$, we call the solution (u_0, u_1) an *i-shock* or *Lax-shock*.

5. **Contact Discontinuity (Linear Wave)** When $\nabla\lambda_i(u) \cdot r_i(u) \equiv 0$, we call the i-th characteristic field *linearly degenerate*. In the case of scalar equation, this corresponds to $f'' = 0$. We claim that, along a linearly degenerate field r_i, we have

$$\mathfrak{R}_i(u_0) = \mathfrak{S}_i(u_0) \text{ and } \sigma_i(u_0, u) = \lambda_i(u_0) \text{ for all } u \in \mathfrak{S}_i(u_0) \text{ or } \mathfrak{R}_i(u_0). \tag{6.26}$$

Indeed, along $\mathfrak{R}_i(u_0)$, we have

$$f'(u)u' = \lambda_i(u)u'.$$

and $\lambda_i(u)$ is a constant $\lambda_i(u_0)$ from the linear degeneracy. We integrate the above equation from u_0 to u along $\mathfrak{R}_i(u_0)$, we get

$$f(u) - f(u_0) = \lambda_i(u_0)(u - u_0).$$

This gives the shock condition. Thus, $\mathfrak{S}_i(u_0) \equiv \mathfrak{R}_i(u_0)$ and $\sigma(u, u_0) \equiv \lambda_i(u_0)$.

6. **Wave curve** Define $T_i(u_0) = \mathfrak{R}_i^+(u_0) \cup \mathfrak{S}_i^-(u_0)$. It is called the *$i$-th wave curve*. For $u_1 \in T_i(u_0)$, the wave (u_0, u_1) is either a rarefaction wave, an entropy shock, or a contact discontinuity.

7. **Solutions to the Riemann problems**

Theorem 6.3. *(Lax) For strictly hyperbolic system (6.21), if each field is either genuinely nonlinear or linear degenerate, then for $u_L \sim u_R$, the Riemann problem with two end states (u_L, u_R) has a unique self-similar solution which consists of n elementary waves. Namely, there exist $u_0 = u_L, \cdots, u_n = u_R$ such that (u_{i-1}, u_i) is an i-wave.*

Proof. Given $(\alpha_1, \cdots, \alpha_n) \in \mathbb{R}^n$, we define u_i successively as the follows. First we define $u_0 = u_L$. Then we follow T_1 curve from u_0 with length α_1. This gives $u_1 \in T_1(u_0)$ and (u_0, u_1) forms a 1-wave with strength α_1 (measured by the arc length α_1 on $T_1(u_0)$. From u_1, we follow $T_2(u_1)$ with length α_2 to u_2. This gives (u_1, u_2) a 2-wave with strength α_2. We continue this process until $u_n := f(u_L, \alpha_1, ..., \alpha_n)$. This gives a map from strengths $(\alpha_1, ..., \alpha_n)$ to the final state u_n with $f(u_L, 0, ..., 0) = u_L$. The mapping

is C^2 because the curves $T_i \in C^2$. Now, we are given the final state u_R. We solve the inverse problem

$$u_R = f(u_L, \alpha_1, ..., \alpha_n).$$

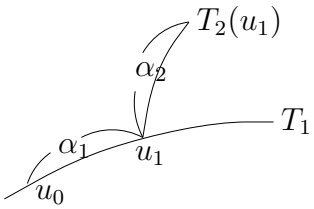
This mapping is locally invertible because the Jacobian

$$\frac{\partial f}{\partial \alpha_k}(u_L, 0, ..., 0) = r_k(u_L), \quad k = 1, ..., n$$

is invertible at u_L. By the inverse function theorem, when $u_R \sim u_L$, there exists a unique $(\alpha_1, ..., \alpha_n)$ such that $u_R = f(u_L, \alpha_1, ..., \alpha_n)$. □

$$u_i = f(u_0, \alpha_1, \cdots, \alpha_i)$$

We want to find $\alpha_1, \cdots, \alpha_n$ such that



When $u_R \sim u_L$ and $\{r_i(u_0)\}$

$$u_R = f(u_L, \alpha_1, \cdots, \alpha_n).$$

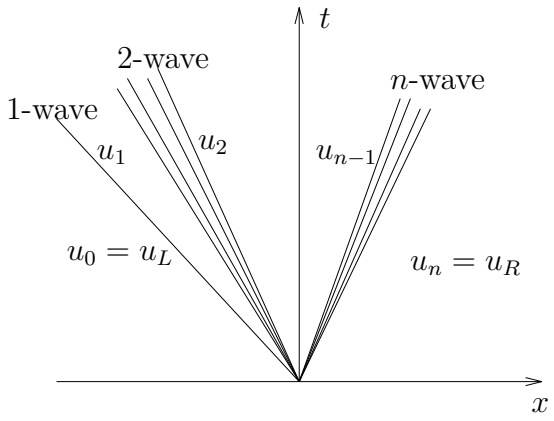
are independent,

$$\left.\frac{\partial}{\partial \alpha_i}\right|_{\alpha=0} f(u_L, 0, \cdots, 0) = r_i(u_0) \text{ and } f \in C^2$$

By Inverse function theorem, for $u_R \sim u_L$, there exists unique α such that $u_R = f(u_L, \alpha)$. Uniqueness leaves as an exercise.

6.2.3 Gas dynamics

The equations of gas dynamics can be derived based on conservation of mass, momentum and energy. Before we derive these equations, let us review some thermodynamics.



1. **Thermo relations** First, the basic thermo variables are pressure (p), specific volume (V), called state variables. The internal energy (e) is a function of p and V. Such a relation is called a constitutive equation. The basic assumption are

$$\left.\frac{\partial e}{\partial p}\right|_V > 0, \quad \left.\frac{\partial e}{\partial V}\right|_p > 0$$

Sometimes, it is convenient to express p as a function of (V, e).

In an adiabetic process (no heat enters or losses), the first law of thermodynamics (conservation of energy) reads

$$de + pdV = 0. \tag{6.27}$$

This is called a Pfaffian equation mathematically. A function $\sigma(e, V)$ is called an integral of (6.27) if there exists a function $\mu(e, V)$ such that

$$d\sigma = \mu \cdot (de + pdV).$$

Thus, $\sigma =$ constant represents a specific adiabetic process. For Pfaffian equation with only two independent variables, one can always find its integral. First, one can derive equation for μ: from

$$\sigma_e = \mu \text{ and } \sigma_V = \mu p$$

and using $\sigma_{eV} = \sigma_{Ve}$, we obtain the equation for μ:

$$\mu_V = (\mu p)_e.$$

This is a linear first-order equation for μ. It can be solved by the method of characteristics in the region $V > 0$ and $e > 0$. The solutions of μ and σ are not unique. If σ is a solution, so does $\bar{\sigma}$ with $d\bar{\sigma} = \nu(\sigma)d\sigma$ for any function $\nu(\sigma)$. We can choose μ such that if two systems are in thermo-equilibrium, then they have the same value μ. In other words, μ is only a function of emperical temperature. We shall denote it by $1/T$. Such T is called the absolute temperature. The corresponding σ is called the physical entropy S. The relation $d\sigma = \mu(de + pdV)$ is re-expressed as

$$de = TdS - pdV. \tag{6.28}$$

For ideal gas, which satisfies the laws of Boyle and Gay-Lussac:

$$pV = RT, \tag{6.29}$$

where R is the universal gas constant. From this and (6.28), treating S and V as independent variables, one obtains

$$Re_S(S, V) + Ve_V(S, V) = 0.$$

We can solve this linear first-order equation by the method of characteristics. We rewrite this equation as a directional differentiation:

$$\left(R\frac{\partial}{\partial S}+V\frac{\partial}{\partial V}\right)e=0.$$

This means that e is constant along the characteristic curves

$$R\frac{dV}{dS}=V.$$

These characteristics can be integrated as

$$Ve^{-S/R}=\phi.$$

Here ϕ is a positive constant. The energy $e(V,S)$ is constant when $Ve^{-S/R}$ is a constant. That is, $e=h(\phi)$ for some function h. We notice that $h'<0$ because $p=-(\frac{\partial e}{\partial V})_S=-e^{-S/R}h'(VH)>0$. From $T=(\frac{\partial e}{\partial S})_V=-\frac{1}{R}h'(\phi)\cdot\phi$, we see that T is a function of ϕ. In most cases, T is a decreasing function of ϕ. We shall make this as an assumption. With this, we can invert the relation between T and ϕ and treat ϕ as a decreasing function of T. Thus, we can also view e as a function of T, say $e(T)$, and $e(T)$ is now an increasing function. Now, we have five thermo variables p,V,e,S,T, and three relations:

$$\begin{aligned} pV &= RT \\ e &= e(T) \\ de &= TdS-pdV \end{aligned}$$

Hence, we can choose two of as independent thermo variables and treat the rest three as dependent variables.

For instance, e is a linear function of T, i.e. $e=c_vT$, where c_v is a constant called specfic heat at constant volume. Such a gas is called polytropic gas. We can obtain

$$pV=RT \text{ and } e=c_vT=\frac{pV}{\gamma-1} \tag{6.30}$$

or in terms of entropy,

$$\begin{aligned} p &= A(S)V^{-\gamma} \\ T &= \frac{A(S)}{R}V^{-\gamma+1} \\ e &= \frac{c_vA(S)}{R}V^{-\gamma+1} \end{aligned}$$

where

$$A(S) = (\gamma-1)\exp((S-S_0)/c_v)$$

$$\gamma \;=\; 1 + R/c_v$$

If we define $dQ = TdS$, it is easy to see that c_v and c_p are the specific heat at constant volume and constant pressure, respectively.

$$\begin{aligned} c_v &= \left(\frac{\partial Q}{\partial T}\right)_V = \left(\frac{\partial e}{\partial T}\right)_V, \\ c_p &:= \left(\frac{\partial Q}{\partial T}\right)_p = ((\frac{\partial e}{\partial V})_p + p)/(\frac{\partial T}{\partial V})_p \\ &= \left(\frac{\partial e}{\partial T}\right)_p + p\left(\frac{\partial V}{\partial T}\right)_p \end{aligned}$$

In general, $c_p > c_v$. Because c_p is the amount of heat added to a system per unit mass at constant pressure. In order to maintain constant pressure, the volume has to expand (otherwise, pressure will increase), the extra amount of work due to expansion is supplied by the extra amount of heat $c_p - c_v$.

2. **Equations for Gas dynamics** Next, we derive the equation of gas dynamics. Let us consider an arbitrary domain $\Omega \subset R^3$. The mass flux from outside to inside per unit time per unit area dS is $-\rho v\cdot$, where n is the outer normal of $\partial\Omega$. Thus, the conservation of mass can be read as

$$\begin{aligned} \frac{d}{dt}\int_\Omega \rho\, dx &= \int_{\partial\Omega}[-\rho v\cdot n]dS \\ &= -\int_\Omega \text{div}\ (\rho\, v)\, dx. \end{aligned}$$

This holds for arbitrary Ω, hence we have

$$\rho_t + \ \text{div}(\rho\, v) = 0. \tag{6.31}$$

This is called the continuity equation.

Now, we derive momentum equation. Let us suppose the only surface force is from pressure (no viscous force). Then the momentum change in Ω is due to (i) the momentum carried in through boundary, (ii) the pressure force exerted on the surface, (iii) the body force. The first term is $-\rho vv\cdot n$, the second term is $-pn$. Thus, we have

$$\begin{aligned} \frac{d}{dt}\int_\Omega \rho v\, dx &= \int_{\partial\Omega} -[\rho vv\cdot n + pn]\, dS + \int F\, dx \\ &= \int_\Omega \text{div}[-\rho v\otimes v - pI] + F\, dx \end{aligned}$$

This yields

$$(\rho v)_t + \ \text{div}(\rho\, v\otimes v) + \nabla p = F. \tag{6.32}$$

Here, the notation $\nabla \cdot \rho v \otimes v$ stands for a vector whoes ith component is $\sum_j \partial_j(\rho v^i v^j)$.

The energy per unit volume is $E = \frac{1}{2}\rho\, v^2 + \rho e$. The energy change in Ω per unit time is due to (i) the energy carried in through boundary (ii) the work done by the pressure from boundary, and (iii) the work done by the body force. The first term is $-Ev \cdot n$. The second term is $-pv \cdot n$. The third term is $F \cdot v$. The conservation of energy can be read as

$$\frac{d}{dt}\int_\Omega E\, dx = \int_{\partial\Omega} [-Ev \cdot n - pv \cdot n]\, dS + \int_\Omega F \cdot v\, dx$$

By applying divergence theorem, we obtain the energy equation:

$$E_t + \mathrm{div}[(E+p)v] = \rho F \cdot v. \tag{6.33}$$

In one dimension, the equations are (without body force)

$$\begin{aligned} \rho_t + (\rho u)_x &= 0 \\ (\rho u)_t + (\rho u^2 + p)_x &= 0 \\ (\frac{1}{2}\rho u^2 + \rho e)_t + [(\frac{1}{2}\rho u^2 + \rho e + p)u]_x &= 0. \end{aligned}$$

Here, the unknowns are two thermo variable ρ and e, and one kinetic variable u. Another thermal variable p is given by the constitutive equation $p(\rho, e)$.

6.2.4 Riemann Problems for Gas Dynamics

This subsection is mainly comes from Courant and Friedrichs' book: Supersonic Flow and Shock Waves.

Hyperbolicity of the equations of gas dynamics We use (ρ, u, S) as our unknown variables. The equations of gas dynamics can be expressed as

$$\begin{bmatrix} \rho \\ u \\ S \end{bmatrix}_t + \begin{bmatrix} u & \rho & 0 \\ \frac{c^2}{\rho} & u & \frac{P_S}{\rho} \\ 0 & 0 & u \end{bmatrix} \begin{bmatrix} \rho \\ u \\ S \end{bmatrix}_x = 0$$

Here, $P(\rho, S) = A(S)\rho^\gamma$, $\gamma > 1$, and $c^2 = \left.\frac{\partial P}{\partial \rho}\right|_S$. This system is hyperbolic. The eigenvalues and eigenvectors are

$$\begin{aligned} &\lambda_1 = u - c, \quad \lambda_2 = u, \quad \lambda_3 = u + c, \\ &r_1 = \begin{bmatrix} \rho \\ -c \\ 0 \end{bmatrix}, \quad r_2 = \begin{bmatrix} -P_S \\ 0 \\ c^2 \end{bmatrix}, \quad r_3 = \begin{bmatrix} \rho \\ c \\ 0 \end{bmatrix}, \\ &\ell_1 = [c, -\rho, \frac{P_S}{c}], \quad \ell_2 = [0, 0, 1], \quad \ell_3 = [c, \rho, \frac{P_S}{c}]. \end{aligned}$$

Note that

$$\begin{aligned}
\nabla\lambda_1 \cdot r_1 &= \frac{1}{c}(\frac{1}{2}\rho P_{\rho\rho} + c^2) > 0,\\
\nabla\lambda_3 \cdot r_3 &= \frac{1}{c}(\frac{1}{2}\rho P_{\rho\rho} + c^2) > 0,\\
\nabla\lambda_2 \cdot r_2 &\equiv 0.
\end{aligned}$$

These show that the 1st and 3rd characteristic fields are genuinely nonlinear, while the 2nd is linearly degenerate.

Rarefaction curves The rarefaction curve $\mathfrak{R}_1$ is the integral curve of the vector field r_1, that is, $(d\rho, du, dS)^T \parallel r_1$. Note that $\ell_2 r_1 = 0$, $\ell_3 r_1 = 0$. Thus, the differential equations for $\mathfrak{R}_1$ are govern by

$$\begin{aligned}
&\begin{cases} (d\rho, du, dS)\cdot(0,0,1) = 0\\ (d\rho, du, dS)\cdot(c, \rho, \frac{P_S}{c}) = 0. \end{cases}\\
\Longrightarrow\quad &\begin{cases} dS = 0\\ cd\rho + \rho du + \frac{P_S}{c} dS = 0 \end{cases}
\end{aligned}$$

Thus, $\mathfrak{R}_1$ can be expressed as

$$\begin{cases} dS = 0\\ \frac{c}{\rho} d\rho + du = 0 \end{cases}$$

Similarly, $\mathfrak{R}_3$ is expressed as

$$\begin{cases} dS = 0\\ \frac{c}{\rho} d\rho - du = 0. \end{cases}$$

Since $S = S_0$, a constant, on $\mathfrak{R}_1$ and $\mathfrak{R}_3$, it is convenient to project the rarefaction curves $\mathfrak{R}_1$ and $\mathfrak{R}_3$ onto the u-P plane. The rarefaction curves $\mathfrak{R}_1$ and $\mathfrak{R}_3$ are given by

$$\begin{cases} \mathfrak{R}_1 : u - u_0 = -\ell + \ell_0\\ \mathfrak{R}_3 : u - u_0 = \ell - \ell_0. \end{cases}$$

where

$$\ell(P, S) := \int \frac{c(\rho, S)}{\rho}\, d\rho.$$

Below, we express ℓ in terms of (P, S). From $P = A(S)\rho^\gamma$, $c = \sqrt{P_\rho} = \sqrt{A(S)\gamma\rho^{\gamma-1}}$, we obtain

$$\ell := \int \frac{c}{\rho}\, d\rho = \sqrt{\gamma A(S)}\frac{2}{\gamma - 1}\rho^{\frac{\gamma-1}{2}} = \frac{2}{\gamma - 1}\sqrt{\frac{\gamma P}{\rho}}.$$

Note that

$$P\rho^{-\gamma} = A(S) = A(S_0) = P_0\rho_0^{-\gamma}.$$

We can express ρ in terms of P, P_0, ρ_0:

$$\rho^{-1} = \rho_0^{-1} \left(\frac{P_0}{P}\right)^{1/\gamma}.$$

Hence,

$$\begin{aligned} \ell - \ell_0 &= \frac{2}{\gamma - 1}\left(\sqrt{\gamma P \left(\frac{P_0}{P}\right)^{1/\gamma} \rho_0^{-1}} - \sqrt{\frac{\gamma P_0}{\rho_0}}\right) \\ &= \frac{2\sqrt{\gamma}}{\gamma - 1}\rho_0^{-\frac{1}{2}} P_0^{\frac{1}{2\gamma}} (P^{\frac{\gamma-1}{2\gamma}} - P_0^{\frac{\gamma-1}{2\gamma}}) := \psi_0(P). \end{aligned}$$

$$\begin{aligned} \therefore \quad \mathfrak{R}_1 : &\quad u = u_0 - \psi_0(P) \\ \mathfrak{R}_3 : &\quad u = u_0 + \psi_0(P). \end{aligned}$$

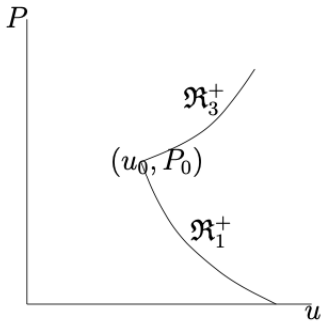


Figure 6.2: The integral curve of the rarefaction curves $\mathfrak{R}_1$ and $\mathfrak{R}_3$ on the u-P plane. Here (u_0, P_0) is a left state. For any point (u_1, P_1) on $\mathfrak{R}_1^+$, $((u_0, P_0), (u_1, P_1))$ forms a 1-rarefaction wave. Note that the entropy $S = S_0$ along a rarefaction curve.

The contact discontinuity On $\mathfrak{R}_2$, $(d\rho, du, dS) \perp \ell_1, \ell_3$, which gives

$$\begin{aligned} \Longrightarrow \quad &\begin{cases} c^2 d\rho + c\rho du + P_S dS = 0 \\ c^2 d\rho - c\rho du + P_S dS = 0 \end{cases} \\ \Longrightarrow \quad &\begin{cases} dP + c\rho du = 0 \\ dP - c\rho du = 0 \end{cases} \end{aligned}$$

Thus, $\mathfrak{R}_2$ is given by

$$\begin{cases} dP = 0 \\ du = 0 \end{cases}$$

For any $(u_1, P_1) \in \mathfrak{R}_2$, we have $u_1 = u_0$, $P_1 = P_0$, and $((u_0, P_0, S_0), (u_0, P_0, S_1))$ constitutes a contact discontinuity.

Note that $\mathfrak{R}_2 = \mathfrak{S}_2$ because the 2-characteristic field is linearly degenerate. You can check the jump conditions for the 2-characteristic field. Which gives $\rho_0(u_0 - \sigma) = \rho(u - \sigma) = 0$ because $u_0 = u = \sigma$, and $P = P_*$. (see the paragraph of shock curves below.)

Shock curves Let us consider a 1-shock (resp. 3-shock) with left state (resp. right state) $(0) := (\rho_0, u_0, P_0)$ and shock speed σ. We want to find the shock curves $\mathfrak{S}_1$ (resp. $\mathfrak{S}_3$) passing through the state (0). Indeed, we want to have expressions of $\mathfrak{S}_1$ (resp. $\mathfrak{S}_3$) on the u-P plane.

Let $v := u - \sigma$. The jump conditions give

$$\begin{cases} [\rho v] = 0 \\ [\rho v^2 + P] = 0 \\ [(\frac{1}{2}\rho v^2 + \rho e + P)v] = 0. \end{cases}$$

Let

$$m := \rho v.$$

From the first jump condition, we have

$$m = m_0.$$

The second jump condition is

$$\rho_0 v_0^2 + P_0 = \rho v^2 + P \quad \Longrightarrow \quad m v_0 + P_0 = m v + P.$$

This gives

$$m = -\frac{P - P_0}{v - v_0} = -\frac{P - P_0}{mV - mV_0},$$

where $V = \frac{1}{\rho}$ is the specific volume. Note that $m \neq 0$. *

$$\therefore \qquad m^2 = -\frac{P - P_0}{V - V_0}, \quad v - v_0 = -\frac{P - P_0}{m}$$

These give

$$(u - u_0)^2 = (v - v_0)^2 = -(P - P_0)(V - V_0). \tag{6.34}$$

The third jump condition is

$$\left(\frac{1}{2}\rho_0 v_0^2 + \rho_0 e_0 + P_0\right) v_0 = \left(\frac{1}{2}\rho v^2 + \rho e + P\right) v.$$

We want to remove the kinetic energy part and only remain an internal energy relation. From $\rho_0 v_0 = \rho v$, we get

$$\frac{1}{2}v_0^2 + e_0 + P_0 V_0 = \frac{1}{2}v^2 + e + PV.$$

By $v_0^2 = m^2 V_0^2, v^2 = m^2 V^2$, and $m^2 = -\frac{P - P_0}{V - V_0}$, we arrive at

$$H(P, V) := e - e_0 + \frac{P + P_0}{2}(V - V_0) = 0$$

*The case $m = 0$ corresponds to the contact discontinuity.

Using $e = \frac{PV}{\gamma-1}$, we get

$$\frac{PV}{\gamma-1} - \frac{P_0V_0}{\gamma-1} + (\frac{P+P_0}{2})(V-V_0) = 0.$$

We use this equation to express V in terms of P, P_0, V_0:

$$V = \frac{\left(\frac{P+P_0}{2}\right)V_0 + \frac{P_0V_0}{\gamma-1}}{\frac{P+P_0}{2} + \frac{P}{\gamma-1}}$$

then plug it into

$$(u-u_0)^2 = -(P-P_0)(V-V_0).$$

We get an expression of $\mathfrak{S}_1$ and $\mathfrak{S}_3$ on the u-P plane:

$$\begin{aligned} \mathfrak{S}_1: \quad u &= u_0 - \phi_0(P) \\ \mathfrak{S}_3: \quad u &= u_0 + \phi_0(P) \end{aligned}$$

$$\phi_0(P) = (P-P_0)\sqrt{\frac{\frac{2}{\gamma+1}V_0}{P + \frac{\gamma-1}{\gamma+1}P_0}} = \frac{(P-P_0)}{Z_0},$$

$$Z_0 = \sqrt{\frac{P_0}{V_0}}\Phi\left(\frac{P}{P_0}\right), \quad \Phi(w) = \sqrt{\frac{\gamma+1}{2}w + \frac{\gamma-1}{2}}.$$

Admissible rarefaction curves and shock curves On $\mathfrak{R}_1$, only the portion where λ_1 is increasing is admissible, because the rarefaction fan requires the characteristic speed of the left end of the fan should be smaller than that of the right end of the fan. Therefore, we define the admissible rarefaction curves and shock curves for the left state (ℓ) as

$$\begin{aligned} \mathfrak{R}_1^+(\ell) &= u_0 - \psi_0(P) \text{ for } P < P_0 \\ \mathfrak{S}_1^-(\ell) &= u_0 - \phi_0(P) \text{ for } P > P_0 \end{aligned}$$

and the admissible rarefaction curves and shock curves for the right state (r) as

$$\begin{aligned} \mathfrak{R}_3^-(r) &= u_0 + \psi_0(P) \text{ for } P < P_0 \\ \mathfrak{S}_3^+(r) &= u_0 + \phi_0(P) \text{ for } P > P_0. \end{aligned}$$

The admissible wave curves are defined to be

$$\begin{aligned} T_1^{(\ell)} &:= \mathfrak{R}_1^+(\ell) \cup \mathfrak{S}_1^-(\ell) \\ T_3^{(r)} &:= \mathfrak{R}_3^-(r) \cup \mathfrak{S}_3^+(r). \end{aligned}$$

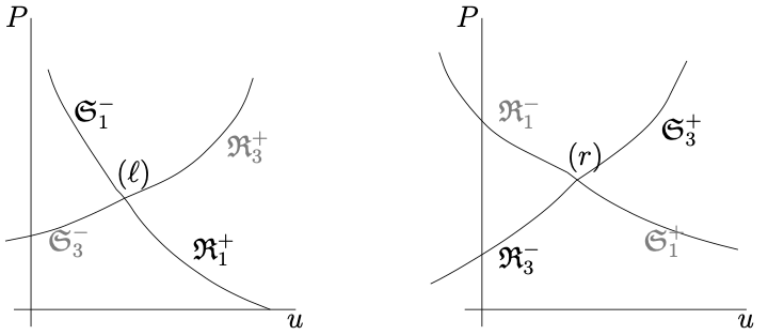


Figure 6.3: The admissible rarefaction curves and shock curves on the u-P plane with left/right states.

Solving Riemann problems Now we are ready to solve the Riemann Problem with initial states (ρ_L, P_L, u_L) and (ρ_R, P_R, u_R). The solution to this Riemann problem consists of three elementary waves:

$$\begin{aligned}
&\text{1-wave}: ((\rho_L, P_L, u_L), (\rho_I, P_I, u_I)),\\
&\text{2-wave}: ((\rho_I, P_I, u_I), (\rho_{II}, P_{II}, u_{II})),\\
&\text{3-wave}: ((\rho_{II}, P_{II}, u_{II}), (\rho_R, P_R, u_R)).
\end{aligned}$$

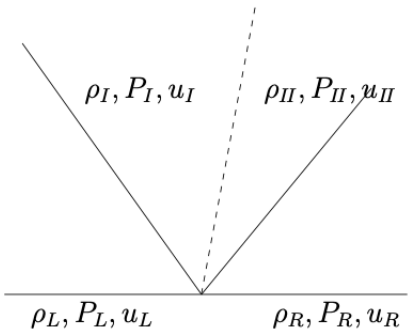


Figure 6.4: The three elementary waves with the left state (ρ_L, P_L, u_L) and the right state (ρ_R, P_R, u_R). The states (ρ_I, P_I, u_I) and $(\rho_{II}, P_{II}, u_{II})$ are called the mid states, which forms a contact discontinuity.

Recall that the second wave is a contact discontinuity, on which $[u] = 0$, $[P] = 0$. Thus, we have

$$\begin{aligned}
u_I &= u_{II} = u_*,\\
P_I &= P_{II} = P_*.
\end{aligned}$$

Finding the mid states (u_*, P_*) Given a left state $U_L := (\rho_L, P_L, u_L)$ and a right state $U_R := (\rho_R, P_R, u_R)$, we want to find two mid states U_I and U_{II} such that (U_L, U_I) forms an 1-wave, and (U_{II}, U_R) forms a 3-wave and (U_I, U_{II}) forms a 2-wave. From the jump condition of the 2-wave, we have $U_I = (\rho_I, P_*, u_*)$ and $U_{II} = (\rho_{II}, P_*, u_*)$. With this, then ρ_I and ρ_{II} can be determined the equation on $T_1^{(\ell)}(U_L)$ and $T_3^{(r)}(U_R)$, respectively. The mid state (u_*, P_*) is the intersection of $T_1^{(\ell)}(U_L)$ and $T_3^{(r)}(U_R)$ on the u-P plane.

Godunov gives a procedure to find the mid state (u_*, P_*). The algorithm to find P_* is to solve

$$\begin{aligned} u_L - f_L(P) &= u_I = u_{II} = u_R + f_R(P) \\ f_0(P) &= \begin{cases} \psi_0(P) & P < P_0 \\ \phi_0(P) & P \geq P_0 \end{cases} \qquad 0 = L, \text{ or } R. \end{aligned}$$

This is equivalent to

$$\begin{cases} -Z_L(u_* - u_L) &= P_* - P_L \\ Z_R(u_* - u_R) &= P_* - P_R, \end{cases} \tag{6.35}$$

where

$$Z_L = \sqrt{\frac{P_L}{V_L}}\Phi\left(\frac{P_*}{P_L}\right), \quad Z_R = \sqrt{\frac{P_R}{V_R}}\Phi\left(\frac{P_*}{P_R}\right)$$

and

$$\Phi(w) = \begin{cases} \sqrt{\frac{\gamma+1}{2}w + \frac{\gamma-1}{2}} & w > 1 \text{ (shock)}, \\ \frac{\gamma-1}{2\sqrt{\gamma}} \frac{1-w}{1-w^{\frac{\gamma-1}{2\gamma}}} & w \leq 1 \text{ (rarefaction)}. \end{cases}$$

System (6.35) is an equation for (u_*, P_*). It can be solved by Newton's method.

The state ρ_{II} can be obtained from (ρ_R, p_R, u_R) and (u_*, P_*) by similar way.

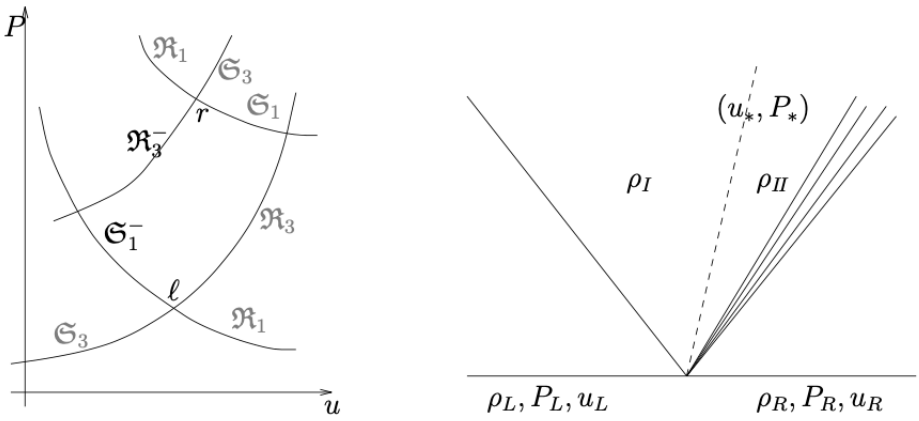


Figure 6.5: This is a solution of the Riemann problem with $p_L < p_R$. In this case, from the left state (ℓ), we follow $\mathfrak{S}_1^-$; and from the right state (r), we follow $\mathfrak{R}_3^-$. Their intersection gives the mid state (u_*, P_*).

Wave structures Given (ρ_L, P_L, u_L) and (ρ_R, P_R, u_R). Let us define

- raref $:= \frac{2}{\gamma-1} c_L \left(1 - \left(\frac{p_R}{p_L}\right)^{\frac{\gamma-1}{2\gamma}}\right)$,
- shk $:= c_L \left(\frac{P_R}{P_L} - 1\right) \sqrt{\frac{2}{\gamma\left((\gamma-1)+(\gamma+1)\frac{P_R}{P_L}\right)}}$,
- $du := u_R - u_L$.

We have the following cases:

(1) $(P_R < P_L)$ & $(du \geq \text{raref})$ or $(p_R \geq P_L)$ & $(du \geq \text{shk}) \Rightarrow R_1 + R_3$.

(2) $(p_R \geq P_L)$ & $(-\text{shk} < du < \text{shk}) \Rightarrow S_1 + R_3$

(3) $(p_R < P_L)$ & $(-\text{shk} < du < \text{shk}) \Rightarrow R_1 + S_3$

(4) $(P_R < P_L)$ & $(du \leq -\text{raref})$ or $(p_R \geq P_L)$ & $(du < -\text{shk}) \Rightarrow S_1 + S_3$.

Note that the transition from (1) to (2) (i.e. $R_1 + R_3$ to $S_1 + R_3$ happens when the left state $(\ell) \in R_3^-(r)$.

Once (u_*, P_*) is found, the full mid state can be obtained by the follows:

- If the 1-wave is a rarefaction wave, then ρ_I can be determined by

$$P_* \rho_I^{-\gamma} = A(S_I) = A(S_L) = P_L \rho_L^{-\gamma}$$

  In the region: $\lambda_1(U_L) < x/t < \lambda_1(U_I)$, the state $U = (\rho, u, S_L)$ is determined by

$$\begin{cases} u - c = \frac{x}{t} \\ u - u_L = \phi_L(P). \end{cases}$$

- If the 1-wave is a shock, then $1/\rho_I = V_I$ can be determined by

$$(u_* - u_L)^2 = -(P_* - P_L)(V_I - V_L).$$

The vacuum State The mid state should satisfy $P_* > 0$. There are situations that the mid state $P_* < 0$. In such cases, we say the mid state contains a vacuum state. The intersections of the admissible wave curves and the axis where $P = 0$ are the vacuum states. Usually, this happens when the two sides of gases running in opposite directions too fast.

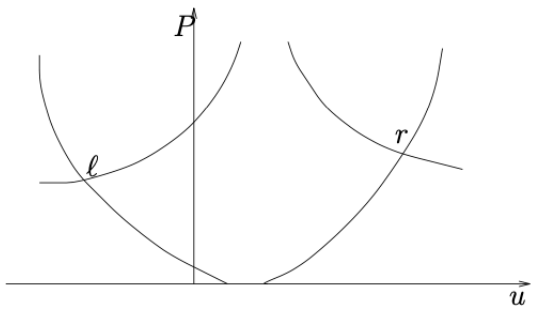


Figure 6.6: The vacuum state appears when $P_* < 0$.

Chapter 7

Finite Difference/ Finite Volume Methods for Hyperbolic Conservation Laws

7.1 General theory of finite difference methods for hyperbolic conservation laws

The important issues of FD schemes for hyperbolic conservation laws are

- Upwinding,
- Interpolating functions with discontinuities,
- Conservative.

7.1.1 Some problems in FD schemes for conservation laws

We list common problems for FD schemes for hyperbolic conservation laws.

- **Stability issue** First of all, we should keep in mind that local stability is necessary in designing finite difference schemes for hyperbolic conservation laws. The scheme has to be stable for hyperbolic conservation laws with frozen coefficients, see Section 5.4. In particular, the CFL condition should be satisfied.
  Stability
- **Spurious oscillation appears around discontinuities in every high order schemes.**. The reason is that the solution of finite difference scheme is closer to a PDE with higher order derivatives. The corresponding dispersion formula demonstrates that oscillation should occur. Also, one may view that it is incorrect to approximate weak

derivative at discontinuity by higher order finite differences. The detail spurious structure can be analyzed by the study of the discrete traveling wave corresponding to a finite difference scheme.

To cure this problem, we have to lower the order of approximation near discontinuities to avoid oscillation. We shall devote to this issue later.
A scheme cannot be high order near discontinuities

- **The approximate solutions may converge to a function which is not a weak solution.** For example, let us apply the Courant-Isaacson-Rees (C-I-R) method to compute a single shock for the inviscid Burgers equation:

$$u_t + uu_x = 0.$$

The C-I-R method is based on characteristic method. Suppose we want to update the state U_j^{n+1}. We draw a characteristic curve back to time t_n. However, the slope of the characteristic curve is not known yet. So, let us approximate it by U_j^n. Then we apply upwind method:

$$U_j^{n+1} - U_j^n = \begin{cases} \frac{\Delta t}{\Delta x} U_j^n (U_{j-1}^n - U_j^n) & \text{if } U_j^n \geq 0 \\ \frac{\Delta t}{\Delta x} U_j^n (U_j^n - U_{j+1}^n) & \text{if } U_j^n < 0 \end{cases}$$

Now, we take the following initial data:

$$U_j^0 = \begin{cases} 1 & \text{for } j < 0 \\ 0 & \text{for } j \geq 0 \end{cases}$$

It is easy to see that $U_j^n = U_j^0$. This is a wrong solution. The reason is that we use a wrong characteristic speed U_j^n when there is a discontinuity passing x_j from t^n to t^{n+1}.

To resolve such a problem, it is advised that one should use a conservative scheme. We shall discuss this issue in the next section.
The scheme is not conservative.

- **The approximate solutions converge to a weak solution, but not an entropy solution.** For example, consider the inviscid Burgers equation $u_t + uu_x = 0$ with the initial data:

$$U_j^0 = \begin{cases} -1 & \text{for } j < 0 \\ 1 & \text{for } j \geq 0 \end{cases}$$

We define the scheme by

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

where

$$F(U, V) = \begin{cases} f(U) & \text{if } U + V \geq 0 \\ f(V) & \text{if } U + V < 0 \end{cases}$$

We find that $F(U_j^n, U_{j+1}^n) = F(U_{j-1}^n, U_j^n)$. Thus, the solution is $U_j^n = U_j^0$ for all $n > 0$. This is a non-entropy solution.
Need entropy satisfying numerical flux.

7.1.2 Conservative schemes

1. **Conservative schemes** A finite difference scheme is called conservative if it can be written as

$$\boxed{U_j^{n+1} = U_j^n + \frac{\Delta t}{\Delta x}\left(F_{j-1/2}^{n+1/2} - F_{j+1/2}^{n+1/2}\right)} \tag{7.1}$$

where $F_{j+1/2}^{n+1/2}$ is called a numerical flux. The advantage of this formulation is that the total mass is conservative:

$$\sum_j U_j^n = \sum_j U_j^{n+1} \tag{7.2}$$

2. **Numerical fluxes** There is a nice interpretation of F if we view U_j^n as an approximation of the cell-average of the solution u over the cell $(x_{j-1/2}, x_{j+1/2})$ at time step n. Let us integrate the conservation law $u_t + f(u)_x = 0$ over the box: $(x_{j-1/2}, x_{j+1/2}) \times (t_n, t_{n+1})$. Using divergence theorem, we obtain

$$\bar{u}_j^{n+1} = \bar{u}_j^n + \frac{\Delta t}{\Delta x}(\bar{f}_{j-1/2}^{n+1/2} - \bar{f}_{j+1/2}^{n+1/2}) \tag{7.3}$$

where

$$\begin{aligned} \bar{u}_j^n &= \frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t_n)\, dx \\ \bar{f}_{j+1/2}^{n+1/2} &= \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f(u(x_{j+1/2}, t))\, dt. \end{aligned}$$

Thus, in a conservative scheme (7.1), we may view

- U_j^n as an approximation of the cell average $\bar{u}_j^n$ and
- $F_{j+1/2}^{n+1/2}$ as an approximation of the flux average $\bar{f}_{j+1/2}^{n+1/2}$.

This formulation is closer to the original integral formulation of conservation laws, and it does not involve derivatives of the unknown quantity u.

3. **Consistency** A conservative scheme is consistent if $F_{j+1/2}(U) = f(u)$, where U is a vector with $U_j = u$. For explicit scheme, $F_{j+1/2}$ is a function of $U_{j-\ell+1}^n, \cdots, U_{j+m}^n$. That is

$$F_{j+1/2}^{n+1/2} = F(U_{j-\ell+1}^n, \cdots, U_{j+m}^n).$$

We usually assume that the function F is a Lipschitz function.

An advantage of conservative schemes is the following Lax-Wendroff theorem. Which says that its approximate solutions, if converge, must to a weak solution.

4. **Conservative schemes guarantee weak solutions**

Theorem 7.1 (Lax-Wendroff)**.** *Suppose $\{U_j^n\}$ be the solution of a conservative scheme (7.1). Define $u_{\Delta x} := U_j^n$ for $[x_{j-1/2}, x_{j+1/2}) \times [t_n, t_{n+1})$. Suppose $u_{\Delta x}$ is uniformly bounded and converges to u almost everywhere. Then u is a weak solution of (6.5).*

Proof. Let ϕ be a smooth test function with compact support on $R \times [0, \infty)$. We multiply (7.1) by ϕ_j^n and sum over j and n to obtain

$$\sum_{n=0}^{\infty} \sum_{j=-\infty}^{\infty} \phi_j^n (U_j^{n+1} - U_j^n) = \frac{\Delta t}{\Delta x} \sum_{n=0}^{\infty} \sum_{j=-\infty}^{\infty} \phi_j^n [F_{j-1/2}(U^n) - F_{j+1/2}(U^n)]$$

Using summation by part, we obtain

$$\sum_{j=-\infty}^{\infty} \phi_j^0 U_j^0 + \sum_{n=1}^{\infty} \sum_{j=-\infty}^{\infty} (\phi_j^n - \phi_j^{n-1}) U_j^n + \sum_{n=0}^{\infty} \sum_{j=-\infty}^{\infty} (\phi_{j+1}^n - \phi_j^n) F_{j+1/2}(U^n) = 0$$

Since ϕ is of compact support and $u_{\Delta x}$, hence $F(U^n)$, are uniformly bounded, we obtain the convergence in the above equation is uniformly in j and n. If $(x_j, t_n) \to (x, t)$, then from the consistency condition, $F_{j+1/2}(U^n) \to f(u(x,t))$. We obtain that u is a weak solution. □

5. **Examples** Below, we show that many scheme can be written in conservation form. We may view $F_{j+1/2}^{n+1/2}$ as a numerical flux at $x_{j+1/2}$ between t_n and t_{n+1}. All schemes below have the form:

$$\boxed{U_j^{n+1} = U_j^n + \frac{\Delta t}{\Delta x} \left(F_{j-1/2}^{n+1/2} - F_{j+1/2}^{n+1/2} \right)} \tag{7.4}$$

(a) **Lax-Friedrichs:**

$$F_{j+1/2}^{n+1/2} = F_{LF}(U_j^n, U_{j+1}^n) \tag{7.5}$$

where

$$F_{LF}(U, V) = \frac{1}{2} \left(f(U) + f(V) + \alpha (U - V) \right). \tag{7.6}$$

The constant $\alpha > 0$ is chosen so that the eigenvalues

$$|f'(u)| \le \alpha.$$

(b) **Godunov scheme:**

$$F_{j+1/2}^{n+1/2} = F_G(U_j^n, U_{j+1}^n). \tag{7.7}$$

where

$$F_G(U, V) = f(\bar{U}), \quad \bar{U} = \text{Riemann solution}_{(U,V)}(0). \tag{7.8}$$

For scalar cases, we claim that

$$F_G(U, V) = \begin{cases} \min_{U \le u \le V} f(u) & \text{if } U \le V \\ \max_{V \le u \le U} f(u) & \text{if } V < U. \end{cases}$$

Let us prove this result only for the case $U \le V$. Suppose $\bar{U} \in (U, V)$. If the solution to the Riemann problem is a rarefaction wave and across $x/t = 0$, then $f'(\bar{U}) = 0$ because it is the solution on the $x/t = 0 = f'(\bar{U})$. The state $\bar{U}$ must be a local minimum of $f(u)$ because the wave structure near $x/t = 0$ is a rarefaction fan across $x/t = 0$, where $f'(u)$ is increasing near $\bar{U}$. For other cases, we have either $\bar{U} = U$ or $\bar{U} = V$. You can check that the above formula for F_G is valid.

(c) **Two-step Lax-Wendroff:**

$$U_{j+1/2}^{n+1/2} = \frac{U_j^n + U_{j+1}^n}{2} + \frac{\Delta t}{2\Delta x}\left[f(U_j^n) - f(U_{j+1}^n)\right],$$
$$F_{j+1/2}^{n+1/2} = f(U_{j+1/2}^{n+1/2}).$$

Note that the state $U_{j+1/2}^{n+1/2}$ is the solution obtained by the Lax-Friedrichs scheme with $\Delta t/2$ and $\Delta x/2$. The Lax-Wendroff scheme is 2nd order accurate.

(d) **MacCormack:**

$$U_j^* = U_j^n + \frac{\Delta t}{\Delta x}\left(f(U_j^n) - f(U_{j+1}^n)\right)$$
$$F_{j+1/2}^{n+1/2} = \frac{1}{2}\left(f(U_{j+1}^n) + f(U_j^*)\right).$$

The MacCormick method is a two-step method. In the first step, the prediction step, it is an upwind method if $f'(u) > 0$, or it is a downwind method if $f'(u) < 0$. In the upwind case, it contains an advection + a diffusion. The second step, a correction step, which removes this numerical diffusion. In the downwind case, it contains an advection + an anti-diffusion, which is removed in the second step. Note that we can switch the indices $(j-1)$ and $(j+1)$ in the above formulation. Namely,

$$U_j^* = U_j^n + \frac{\Delta t}{\Delta x}\left(f(U_j^n) - f(U_{j-1}^n)\right)$$
$$F_{j+1/2}^{n+1/2} = \frac{1}{2}\left(f(U_{j-1}^n) + f(U_j^*)\right).$$

Homeworks 7.1. 1. Construct an example to show that the Lax-Wendroff scheme may produce non-entropy solution.

2. Find the modified equation for the MacCormack scheme for the linear advection equation.

7.1.3 Monotone schemes and entropy-satisfying schemes

1. **Definition of Monotone schemes**

Definition 7.1. A FD scheme expressed as

$$U_j^{n+1} = G(U_{j-\ell}^n, \cdots, U_{j+m}^n) \tag{7.9}$$

is called a *monotone scheme* if

$$\frac{\partial G}{\partial U_{j+k}} \geq 0, k = -\ell, \cdots, m. \tag{7.10}$$

Note that we still need to require the *consistency condition*:

$$U = G(U, ..., U).$$

2. **Conditions of monotone schemes in flux form** Let us consider the conservative schemes:

$$U_j^{n+1} = U_j^n + \frac{\Delta t}{\Delta x}\left(F(U_{j-1}^n, U_j^n) - F(U_j^n, U_{j+1}^n\right) = G(U_{j-1}^n, U_j^n, U_{j+1}^n). \tag{7.11}$$

We want to find the conditions on F so that the corresponding scheme is monotone. Note that

$$\frac{\partial G}{\partial U_{j-1}} \geq 0 \Leftrightarrow \frac{\partial F(U,V)}{\partial U} \geq 0,$$
$$\frac{\partial G}{\partial U_{j+1}} \geq 0 \Leftrightarrow \frac{\partial F(U,V)}{\partial V} \leq 0,$$

Thus, F is increasing in the first argument and decreasing in the second argument. We denotes this as $F(\uparrow, \downarrow)$. Usually, the condition $\frac{\partial G}{\partial U_j} \geq 0$ corresponds to the CFL condition for the scheme (7.11). In the above schemes,

- **Lax-Friedrichs flux**:

$$F_{LF}(U,V) = \frac{1}{2}\left(f(U) + f(V) + \alpha(U-V)\right), \quad |f'(u)| \leq \alpha.$$

We see that the condition $|f'(u)| \leq \alpha$ is equivalent to $F_{LF}(\uparrow, \downarrow)$. The scheme is

$$\begin{aligned} G(U_{j-1}, U_j, U_{j+1}) &= U_j + \frac{\Delta t}{\Delta x}\left[F(U_{j-1}, U_j) - F_{LF}(U_j, U_{j+1})\right] \\ &= U_j + \frac{\Delta t}{\Delta x}\left[\frac{1}{2}\left(f(U_{j-1}) + f(U_j) + \alpha(U_{j-1} - U_j)\right) - \frac{1}{2}\left(f(U_j) + f(U_{j+1}) + \alpha(U_j - U_{j+1})\right)\right] \\ &= U_j + \frac{\Delta t}{2\Delta x}\left[(f(U_{j-1}) - f(U_{j+1}) + \alpha(U_{j-1} - 2U_j + U_{j+1})\right] \end{aligned}$$

The condition $\frac{\partial G}{\partial U_j} \geq 0$ is equivalent to

$$1 - \frac{\Delta t}{\Delta x}\alpha \geq 0.$$

We see that the best choice of α is

$$\alpha = \max |f'(u)|.$$

For such a choice, the above condition is equivalent to the CFL condition. If we choose larger α, then the scheme has stronger numerical diffusion. The forward Euler method for such diffusion equation, a stability condition is needed. The condition $1 - \frac{\Delta t}{\Delta x}\alpha \geq 0$ gives such stability constraint.
Note that the flux f can be decomposed into upwind and downwind part:

$$f(u) = f^+(u) + f^-(u),$$

$$f^\pm(u) := \frac{1}{2}\left(f(u) \pm \alpha u\right), \quad |f'(u)| \leq \alpha.$$

The eigenvalues of f^+ are all non-negative, whereas the eigenvalues of f^- are all non-positive. The Lax-Friedrichs flux can be expressed as

$$F_{LF}(U, V) = \left(f^+(U) + f^-(V)\right).$$

- **Godunov flux**: for scalar,

$$F_G(U, V) = \begin{cases} \min_{U \leq u \leq V} f(u) & \text{if } U \leq V \\ \max_{V \leq u \leq U} f(u) & \text{if } V < U. \end{cases}$$

You can show that $F_G(\uparrow, \downarrow)$.
For system,

$$F_G(U, V) = f(\bar{U}),$$

where $\bar{U}$ is the solution $u(x/t))$ of the Riemann problem (U, V) on the vertical line $x = 0$.

- **Kinetic flux**:

3. In the case of linear equation, a monotone scheme can be expressed as

$$U_j^{n+1} = \sum_{k=-\ell}^{m} a_k U_{j+k}^n$$

with $a_k \geq 0$. The consistency condition gives $\sum_k a_k = 1$. Thus, for linear cases, a *monotone scheme* produces U_j^{n+1} *as an average of* $U_{j-\ell}^n, \cdots, U_{j+m}^n$. Indeed, monotone schemes are generalization of such "averaging schemes" for nonlinear cases. A monotone scheme has the following properties: the sup norm is non-increasing, the solution operator is ℓ^1-contraction, and the total variation is diminishing. To be precise, let us define the following norms for $U = \{U_j\}$:

$$|U|_\infty \quad = \quad \sup_j |U_j|$$

$$\begin{aligned} \|U\|_1 &= \sum_j |U_j|\Delta x \\ T.V.(U) &= \sum_j |U_{j+1} - U_j|. \end{aligned}$$

4. **Properties of monotone schemes** We have the following theorem.

Theorem 7.2. *For a monotone scheme (7.9), we have*

(a) ℓ^∞- bound:

$$|U^{n+1}|_\infty \le |U^n|_\infty$$

(b) ℓ^1-contraction: if U, V are two solutions of (7.1), then

$$\|U^{n+1} - V^{n+1}\|_1 \le \|U^n - V^n\|_1 \tag{7.12}$$

(c) total variation diminishing:

$$T.V._x(U^{n+1}) \le T.V._x(U^n) \tag{7.13}$$

(d) boundedness of total variation: there exists a constant C such that

$$T.V._{x,t}(U) \le C. \tag{7.14}$$

Proof. (a)

$$\begin{aligned} U_j^{n+1} &= G(U_{j-\ell}^n, \cdots, U_{j+m}^n) \\ &\le G(\max U^n, \cdots, \max U^n) \\ &= \max U^n \qquad\qquad \because \text{ consistency} \end{aligned}$$

Hence, we have $\max U^{n+1} \le \max U^n$. Similarly, we also have $\min U^{n+1} \ge \min U^n$.

(b) Let us denote the vector (U_j^n) by U^n and express scheme (7.9) in operator form:

$$U^{n+1} = G(U^n).$$

The notations:

- $U \le V$ means that $U_j \le V_j$ for each j.
- $U \vee V$ for the vector $(\max\{U_j, V_j\})$.
- The monotonicity reads $G(U) \le G(V)$ if $U \le V$.

(c) Clearly we have $G(U \vee V) \ge G(V)$. Hence,

$$(G(U) - G(V))^+ \le ((G(U \vee V) - G(V))^+ = G(U \vee V) - G(V).$$

We take summation in j, and use conservative property of G, namely, $\sum_j (G(U))_j = \sum_j U_j$, we obtain

$$\sum_j (G(U) - G(V))_j^+ \le \sum_j ((U \vee V) - V)_j = \sum_j (U - V)_j^+.$$

Similarly, we have

$$\sum_j (G(V) - G(U))_j^+ \le \sum_j (V - U)_j^+.$$

Adding these two, we obtain the ℓ^1-contraction:

$$\sum_j |G(U)_j - G(V)_j| \le \sum_j |U_j - V_j|.$$

(d) Suppose U_j^n is a solution of (7.9). We take V_j^n to be U_{j+1}^n. Then V_j^n also satisfies (7.9). From the ℓ^1-contraction property, we have

$$\sum_j |U_{j+1}^{n+1} - U_j^{n+1}| \le \sum_j |U_{j+1}^n - U_j^n|$$

This shows the total variation dimishing property of (7.9).

(e) The total variation of U in x, t with $0 \le t \le T$ is defined by

$$\begin{aligned} T.V._{x,t}(U) &= \sum_{n=0}^{N} \sum_{j=-\infty}^{\infty} \left[\frac{|U_{j+1}^n - U_j^n|}{\Delta x} + \frac{|U_j^{n+1} - U_j^n|}{\Delta t} \right] \Delta x \, \Delta t \\ &= \sum_{n=0}^{N} \left[T.V._x U^n \Delta t + \|U^{n+1} - U^n\|_{L^1} \right] \\ &\le T.V._x U^0 T + \sum_{n=0}^{N} \|U^{n+1} - U^n\|_{L^1}. \end{aligned}$$

Here $N\Delta t = T$. We claim that $\|U^{n+1} - U^n\|_{L^1} \le O(\Delta t)$. If so, then we obtain the result with $C \le T + NO(\Delta t) \le T + KT$ for some constant K. Now, we prove this claim:

$$\begin{aligned} \|U^{n+1} - U^n\|_{L^1} &= \sum_j |U_j^{n+1} - U_j^n| \Delta x \\ &= \sum_j |G(U_{j-\ell}^n, \cdots, U_{j+m}^n) - G(U_j^n, \cdots, U_j^n)| \Delta x \\ &\le \sum_j L \left(|U_{j-\ell}^n - U_j^n| + \cdots + |U_{j+m}^n - U_j^n| \right) \Delta x \end{aligned}$$

$$\begin{aligned} &\leq L(\ell+m)T.V._x(U^n)\Delta x \\ &\leq L(\ell+m)T.V._x(U^0)\frac{\Delta t}{\lambda} \end{aligned}$$

Here, $\lambda = \Delta t/\Delta x$. We have used that G is Lipschitz continuous. Hence, we conclude

$$\sum_{n=0}^{N} \|U^{n+1} - U^n\|_{L^1} \leq \frac{L(\ell+m)T}{\lambda} T.V._x(U^0).$$

□

The boundedness of total variation of U in (x,t) implies that we can substract a subsequence $u_{\Delta x}$ which converges in L^1. Below, we show that its limit indeed satisfies entropy condition.

5. **Monotone schemes give entropy solutions**

Theorem 7.3. *The limiting function of an approximate solution constructed from a monotone scheme satisfies Kruzkov's entropy condition.*

Proof. We choose $\eta = (u-c)^+ = u \vee c - c$. The corresponding entropy flux is $q(u) = f(u \vee c) - f(c)$. It is natural to choose the numerical entropy flux to be $Q(U_{j-\ell+1}, \cdots, U_{j+m}) = F(U_{j-\ell+1} \vee c, \cdots, U_{j+m} \vee c) - F(c, \cdots, c)$. We have

$$\begin{aligned} (U^{n+1} \vee c) &= G(U^n_{j-\ell}, \cdots, U^n_{j+m}) \vee G(c, \cdots, c) \\ &\leq G(U^n_{j-\ell} \vee c, \cdots, U^n_{j+m} \vee c) \\ &= U^n_j \vee c + \frac{\Delta t}{\Delta x}\left[F(U^n_{j-\ell} \vee c, \cdots, U^n_{j+m-1} \vee c) - F(U^n_{j-\ell+1} \vee c, \cdots, U^n_{j+m} \vee c)\right] \\ &= U^n_j \vee c + \frac{\Delta t}{\Delta x}\left[Q(U^n_{j-\ell}, \cdots, U^n_{j+m-1}) - Q(U^n_{j-\ell+1}, \cdots, U^n_{j+m})\right] \end{aligned}$$

Multiply this inequality by ϕ^n_j, sum over j and n, and apply "summation-by-part", then take limit $\Delta t, \Delta x \to 0$. We obtain that u is an entropy solution. □

6. **Monotone schemes can only be first order.**

Theorem 7.4 (Harten-Hyman-Lax)**.** *A monotone scheme (7.9) is at most first order.*

Proof. We claim that the modified equation corresponding to a monotone scheme has the following form

$$u_t + f(u)_x = \Delta t[\beta(u,\lambda)u_x]_x \tag{7.15}$$

where $\lambda = \Delta t/\Delta x$,

$$\beta = \frac{1}{2\lambda^2}\sum_{k=-\ell}^{m} k^2 G_k(u, \cdots, u) - \frac{1}{2}f'(u)^2, \quad G_k := \frac{\partial G}{\partial u_k}, \tag{7.16}$$

and $\beta > 0$ except for some exceptional cases. Thus, monotone schemes are at most first order.

To show (7.15), we take Taylor expansion of G about $(u_0, \cdots, u_0)$:

$$\begin{aligned} G(u_{-\ell}, \cdots, u_m) &= G(u_0, \cdots, u_0) + \sum_{k=-\ell}^{m} G_k(u_k - u_0) \\ &\quad + \frac{1}{2} \sum_{j,k=-\ell}^{m} G_{j,k}(u_j - u_0)(u_k - u_0) + O(\Delta x)^3 \\ &= u_0 + \Delta x u_x \sum_{k=-\ell}^{m} kG_k + \frac{1}{2}(\Delta x)^2 u_{xx} \sum_{k=-\ell}^{m} k^2 G_k \\ &\quad + \sum_{j,k} \frac{1}{2}(\Delta x)^2 u_x^2 jk G_{j,k} + O(\Delta x)^3 \\ &= u_0 + \Delta x u_x \sum_{k=-\ell}^{m} kG_k + \frac{1}{2}(\Delta x)^2 \left(\sum_{k=-\ell}^{m} k^2 G_k u_x \right)_x \\ &\quad + \sum_{j,k} \frac{1}{2}(\Delta x)^2 u_x^2 (jk - k^2) G_{j,k} + O(\Delta x)^3 \end{aligned}$$

On the other hand,

$$G(u_{-\ell}, \cdots, u_m) = u_0 + \lambda(F(\bar{u}) - F(T\bar{u}))$$

where $\bar{u} = (u_{-\ell}, \cdot, u_{m-1})$, $T\bar{u} = (u_{-\ell+1}, \cdots, u_m)$. We differentiate this equation to obtain

$$\begin{aligned} G_k &= \delta_{0,k} + \lambda[F_k(\bar{u}) - F_{k-1}(T\bar{u})] \\ G_{j,k} &= \lambda[F_{j,k}(\bar{u}) - F_{j-1,k-1}(T\bar{u})] \end{aligned}$$

We differentiate the consistency condition $F(u_0, \cdots, u_0) = f(u_0)$ to obtain

$$\sum_{-\ell}^{m-1} F_k(u_0, \cdots, u_0) = f'(u_0).$$

Therefore,

$$\begin{aligned} \sum_{k=-\ell}^{m} G_k &= 1 \\ \sum_{k=-\ell}^{m} kG_k &= \lambda \sum (F_k - F_{k-1})k = -\lambda f'(u_0) \end{aligned}$$

$$\sum_{j,k}(j-k)^2 G_{j,k} \;=\; \lambda \sum (j-k)^2 [G_{j-1,k-1} - G_{j,k}] = 0$$

Using this and the symmetry $G_{j,k} = G_{k,j}$, we obtain

$$\sum_{j,k} G_{j,k}(jk - k^2) = -\frac{1}{2}\sum G_{j,k}(j-k)^2 = 0.$$

Hence we obtain

$$G(u_{-\ell}, \cdots, u_m) = u_0 - \Delta x \lambda f'(u) u_x + (\frac{1}{2}\Delta x)^2 u_{xx} \sum_k k^2 G_k + O(\Delta x)^3$$

Now, from the Taylor expansion:

$$\begin{aligned} u_0^1 &= u_0 + \Delta t u_t + \frac{1}{2}(\Delta t)^2 u_{tt} + O(\Delta t)^3 \\ &= u_0 - \Delta t f(u)_x + (\frac{1}{2}\Delta t)^2 [f'(u)^2 u_x]_x + O(\Delta t)^3 \end{aligned}$$

Combine these two, we obtain that smooth solution of the finite difference equation satisfy the modified equation up to a truncation error $(\Delta t)^2$.

To show $\beta \geq 0$, from the monotonicity $G_k \geq 0$. Hence

$$\begin{aligned} \lambda^2 f'(u)^2 &= \left(\sum_k k G_k\right)^2 = \left(\sum k\sqrt{G_k}\sqrt{G_k}\right)^2 \\ &\leq \sum k^2 G_k \cdot \sum G_k = \sum_k k^2 G_k \end{aligned}$$

The equality holds only when $G_k(u, \cdots, u) = 0$ for all k except 1. This means that $G(u_\ell, \cdots, u_m) = u_1$. This is a trivial case.

□

7.2 Flux limiter methods

The main idea for flux limiter methods is to add a switch between a low order scheme and a high order scheme such that the scheme becomes first order near discontinuities and remains high order in the smooth region. For details, See Flux Limiter (Wiki).

Suppose

$$F^L \text{ a lower-order numerical flux}$$

$$F^H \text{ a higher-order numerical flux}$$

Define

$$F_{j+\frac{1}{2}} \;=\; F^L_{j+\frac{1}{2}} + \phi_{j+\frac{1}{2}}(F^H_{j+\frac{1}{2}} - F^L_{j+\frac{1}{2}})$$

$$= F^H_{j+\frac{1}{2}} + (1-\phi_{j+\frac{1}{2}})(F^L_{j+\frac{1}{2}} - F^H_{j+\frac{1}{2}}).$$

Here, $\phi_{j+\frac{1}{2}}$ is a limiter or a switch. It is required such that

- $\phi_{j+\frac{1}{2}} \sim 0$ (i.e. $F_{j+\frac{1}{2}} \sim F^L_{j+\frac{1}{2}}$) near a discontinuity,
- $\phi_{j+\frac{1}{2}} \sim 1$ (i.e. $F_{j+\frac{1}{2}} \sim F^H_{j+\frac{1}{2}}$) in smooth region.

The term $F_H - F_L$ is an anti-diffusion flux. When $\phi \sim 1$, this term cancels the numerical diffusion inherited in the lower order flux F_L.
Below, we devote to design criteria for $\phi_{j+\frac{1}{2}}$

7.2.1 Total Variation Diminishing (TVD)

In this subsection, we will find a condition on the limiter ϕ so that the resulting scheme is total variation diminishing (TVD). We consider the linear advection equation

$$u_t + au_x = 0, \qquad a > 0.$$

We show the ideas by using

$$\begin{aligned} F^L_{j+\frac{1}{2}} &= aU_j && \text{(upwind flux)} \\ F^H_{j+\frac{1}{2}} &= aU_j + \tfrac{1}{2}a(1-\tfrac{a\Delta t}{\Delta x})(U_{j+1}-U_j) && (\text{ 2nd order flux}). \end{aligned}$$

Note that the F_L gives the upwind scheme. It produces a numerical diffusion $\frac{1}{2}\frac{a\Delta t}{\Delta x}(1-\frac{a\Delta t}{\Delta x})u_{xx}$ term in the modified equation. The higher-order flux contains the anti-diffusion flux:

$$(F^H - F^L)_{j-1/2} - (F^H - F^L)_{j+1/2} = \frac{1}{2}a(1-\frac{a\Delta t}{\Delta x})\left[(U_j - U_{j-1}) - (U_{j+1} - U_j)\right],$$

which cancels the numerical diffusion produced by the upwind method.

The numerical flux of a flux limiter method is defined as

$$F_{j+\frac{1}{2}} = aU_j + \phi_{j+\frac{1}{2}}\left[\frac{1}{2}a(1-\frac{a\Delta t}{\Delta x})(U_{j+1}-U_j)\right], \tag{7.17}$$

where

$$\phi_{j+\frac{1}{2}} = \phi(\theta_{j+\frac{1}{2}}), \quad \theta_{j+\frac{1}{2}} := \frac{U_j - U_{j-1}}{U_{j+1}-U_j}.$$

Theorem 7.5. *1. If ϕ is bounded, then the scheme is consistent.*

2. If $\phi(1) = 1$, and ϕ is Lipschitz continuous(or C^1) at $\theta = 1$, then the scheme is second order in smooth monoton regions (i.e. u is smooth and $u_x \neq 0$).

3. *If* $0 \le \frac{\phi(\theta)}{\theta} \le 2$ *and* $0 \le \phi(\theta) \le 2$*, then the scheme is total variation diminishing (TVD).*

Proof. 1. $F_{j+\frac{1}{2}}(u,u) = f(u) = au$.

2. Hint: Apply truncation error analysis.

3. From (7.17), the next time step U_j^{n+1} is

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

where

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

In other words, U_j^{n+1} is the average of U_j^n and U_{j-1}^n with weights $(1 - c_{j-1}^n)$ and c_{j-1}^n.

$$\begin{aligned} U_{j+1}^{n+1} - U_j^{n+1} &= (U_{j+1}^n - c_j^n(U_{j+1}^n - U_j^n)) - (U_j^n - c_{j-1}^n(U_j^n - U_{j-1}^n)) \\ &= (1 - c_j^n)(U_{j+1}^n - U_j^n) + c_{j-1}^n(U_j^n - U_{j-1}^n) \end{aligned}$$

Suppose $0 \le c_j^n \le 1 \ \ \forall j, n$. Then

$$|U_{j+1}^{n+1} - U_j^{n+1}| \le (1 - c_j^n)|U_{j+1}^n - U_j^n| + c_{j-1}^n |U_j^n - U_{j-1}^n|$$

$$\begin{aligned} \sum_j |U_{j+1}^{n+1} - U_j^{n+1}| &\le \sum_j (1 - c_j^n)|U_{j+1}^n - U_j^n| + \sum c_{j-1}^n |U_j^n - U_{j-1}^n| \\ &= \sum_j (1 - c_j^n)|U_{j+1}^n - U_j^n| + \sum c_j^n |U_{j+1}^n - U_j^n| \\ &= \sum_j |U_{j+1}^n - U_j^n|. \end{aligned}$$

Thus, the computed solution U_j^n is total variation diminishing, provided $0 \le c_j^n \le 1$ for all j, n.

Next, we need to find conditions on ϕ such that $0 \le c_j^n \le 1, \ \forall j, n$. Note that the quantity

$$\frac{\phi_{j+\frac{1}{2}}(U_{j+1} - U_j) - \phi_{j-\frac{1}{2}}(U_j - U_{j-1})}{U_j - U_{j-1}} = \frac{\phi_{j+1/2}}{\theta_{j+\frac{1}{2}}} - \phi_{j-1/2} = \frac{\phi(\theta_{j+\frac{1}{2}})}{\theta_{j+\frac{1}{2}}} - \phi(\theta_{j-\frac{1}{2}}),$$

$$\Longrightarrow c_{j-1}^n = \nu + \frac{1}{2}\nu(1-\nu)\left(\frac{\phi(\theta_{j+\frac{1}{2}})}{\theta_{j+\frac{1}{2}}} - \phi(\theta_{j-\frac{1}{2}})\right), \quad 0 \le \nu \le 1$$

Note that $0 < \nu \le 1$. A sufficient condition for $(0 \le c_{j-1}^n \le 1,\ \forall j)$ is

$$\left|\frac{\phi(\theta_{j+\frac{1}{2}})}{\theta_{j+\frac{1}{2}}} - \phi(\theta_{j-\frac{1}{2}})\right| \le 2. \tag{7.18}$$

If $0 \le \frac{\phi(\theta)}{\theta} \le 2$, $0 \le \phi(\theta) \le 2$, then (7.18) is valid.

□

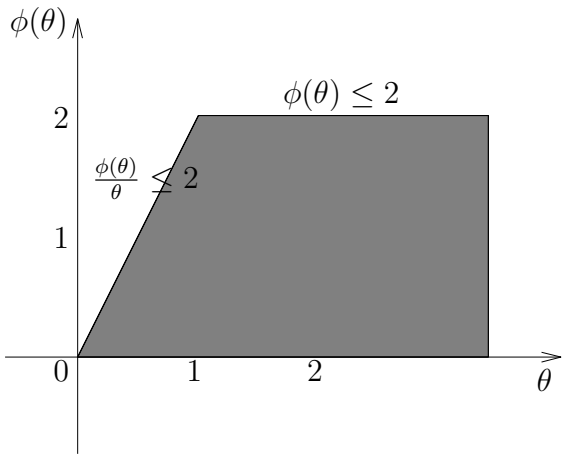


Figure 7.1: The region in which $\phi(\theta)$ should lie so that the scheme is TVD.

We summarize the criteria of a limiter $\phi(\theta)$:

- $\phi(1) = 1$, and ϕ is Lipschitz continuous(or C^1) at $\theta = 1$. This gives second scheme in smooth regions.
- $\phi(0) = 0$, $\phi(\theta) = 0$ for $\theta < 0$. This gives a first order scheme near discontinuities. Note that if $\theta_{j+\frac{1}{2}} := \frac{U_j - U_{j-1}}{U_{j+1} - U_j} < 0$, then U_j is a local extremal. In this case, we should set $\phi(\theta_{j+\frac{1}{2}}) = 0$ to avoid possible spurious oscillations.
- $0 \le \frac{\phi(\theta)}{\theta} \le 2$ and $0 \le \phi(\theta) \le 2$. This is a sufficient condition for TVD.

7.2.2 Examples of limiters $\phi(\theta)$

We list some limiters below. Case 1-3 are second schemes. For more examples of limiters, see Flux Limiter, (Wiki).

1. $\phi_{LW}(\theta) = 1$. This is the Lax-Wendroff scheme.

2. $\phi_{BW}(\theta) = \theta$. This is Beam-Warming.

3. Any ϕ between ϕ_{BW} and ϕ_{LW} with $0 \le \phi \le 2$, $0 \le \frac{\phi(\theta)}{\theta} \le 2$ gives a second order scheme.

4. Van Leer's minmod
$$\phi(\theta) = \frac{\theta + |\theta|}{1 + |\theta|}.$$
It is a smooth limiter with $\phi(1) = 1$ and $\phi(0) = 0$.

5. Roe's superbee
$$\phi(\theta) = \max(0, \min(1, 2\theta), \min(\theta, 2))$$

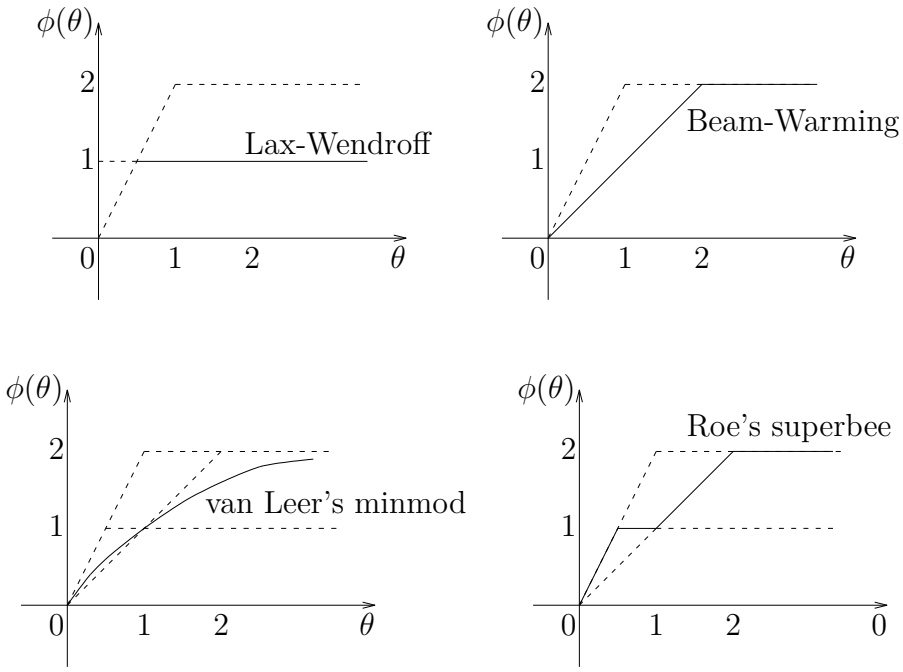


Figure 7.2: The solid lines are the limiters

7.2.3 Extensions

There are two kinds of extensions. One is the case $a < 0$, and the other is the linear system case.

Case: scalar flux with negative characteristic speed For $a < 0$, we choose *

$$F^L_{j+\frac{1}{2}} = \frac{1}{2}\left(a(U_j + U_{j+1}) - |a|(U_{j+1} - U_j)\right)$$

*Let us use $f(u) = au$. We decompose $f(u) = f^+(u) + f^-(u)$, where $f^\pm = \frac{1}{2}\left(f(u) \pm |A|u\right)$, $A = f'(u)$. Then $F_L(u, v) = f^+(u) + f^-(v)$.

$$
\begin{aligned}
&= \begin{cases} aU_j & \text{if } a > 0 \\ aU_{j+1} & \text{if } a < 0 \end{cases} \\
F^H_{j+\frac{1}{2}} &= \frac{1}{2}\left(a(U_j + U_{j+1}) - \nu a(U_{j+1} - U_j)\right), \quad \nu = \frac{a\Delta t}{\Delta x}
\end{aligned}
$$

Then

$$
\begin{aligned}
F_{j+\frac{1}{2}} &= F^L_{j+\frac{1}{2}} + \phi_{j+\frac{1}{2}}(F^H_{j+\frac{1}{2}} - F^L_{j+\frac{1}{2}}) \\
&= F^L_{j+\frac{1}{2}} + \phi_{j+\frac{1}{2}}\frac{1}{2}(\text{sign}(a) - \nu)a(U_{j+1} - U_j).
\end{aligned}
$$

Where $\phi_{j+\frac{1}{2}} = \phi(\theta_{j+\frac{1}{2}})$, $\theta_{j+\frac{1}{2}} = \frac{U_{j'+1} - U_{j'}}{U_{j+1} - U_j}$, and $j' = j - \text{sign}(a) = j \pm 1$.

Linear system case In the linear system case, our equation is

$$
u_t + Au_x = 0. \tag{7.19}
$$

We can decompose A so that $A = R\Lambda R^{-1}$ with $\Lambda = \text{diag}(\lambda_1, \cdots, \lambda_n)$ constituting by A's eigenvalues and $R = [r_1, \cdots, r_n]$ being right eigenvectors.That is, $Ar_i = \lambda_i r_i$. We know that $U_{j+1} - U_j = \sum_{k=1}^{n} \alpha_{j,k} r_k$, let

$$
\begin{aligned}
\nu_k &= \lambda_k \frac{\Delta t}{\Delta x} \\
\theta_{j,k} &= \frac{\alpha_{j',k}}{\alpha_{j,k}} \qquad j' = j - \text{sign}(\lambda_k).
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
F^L &= \frac{1}{2}A(U_j + U_{j+1}) - \frac{1}{2}|A|(U_{j+1} - U_j) \\
F^H &= \frac{1}{2}A(U_j + U_{j+1}) - \frac{1}{2}\frac{\Delta t}{\Delta x}A^2(U_{j+1} - U_j)
\end{aligned}
$$

where $|A| = R|\Lambda|R^{-1}$. The numerical flux is

$$
F_{j+\frac{1}{2}} = F^L_{j+\frac{1}{2}} + \frac{1}{2}\sum_k \phi(\theta_{j,k})(\text{sign}(\lambda_k) - \nu_k)\lambda_k \alpha_{j,k} r_k.
$$

7.3 Higher-order Godunov methods

7.3.1 Piecewise-linear reconstruction (MUSCL reconstruction)

(1) Reconstruction
Given cell average $\{U^n_j\}$, we want to reconstruct a linear function $\tilde{u}(x, t_n)$ in each cell $(x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}})$ under following criterions

- high order approximation in smooth regions.
- total variation non-increasing

(2) Find edge fluxes:

- Solve a linearized equation in $(x_{j-1/2}, x_{j+1/2}) \times (t^n, t^{n+1/2})$ with the linear data $\tilde{u}(x, t^n)$ to find the solutions at $(x_{j-1/2}+, t^{n+1/2})$ and $(x_{j+1/2}-, t^{n+1/2})$.
- Solve the Riemann problems and find the Godunov fluxes: define

$$F_{j+1/2}^{n+1/2} := F_G(\tilde{u}(x_{j+1/2}-, t^{n+1/2}), \tilde{u}(x_{j+1/2}+, t^{n+1/2})).$$

(3) Updating: We use midpoint rule for time integration

$$U_j^{n+1} = U_j^n + \frac{\Delta t}{\Delta x}[F_{j-1/2}^{n+1/2} - F_{j+1/2}^{n+1/2}].$$

Scalar Case

(1) **Reconstruction**
Suppose $\tilde{u}(x, t_n) = a + b(x - x_j) + c(x - x_j)^2$, want to find a, b, c such that the average of $\tilde{u} = U_j$.

$$\begin{aligned}
\frac{1}{\Delta x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \tilde{u}(x, t_n)\ dx &= U_j \\
\frac{1}{\Delta x}\int_{x_{j-\frac{3}{2}}}^{x_{j-\frac{1}{2}}} \tilde{u}(x, t_n)\ dx &= U_{j-1} \\
\frac{1}{\Delta x}\int_{x_{j+\frac{1}{2}}}^{x_{j+\frac{3}{2}}} \tilde{u}(x, t_n)\ dx &= U_{j+1}
\end{aligned}$$

$$\Longrightarrow a = U_j,\ b = \frac{U_{j+1} - U_{j-1}}{2\Delta x},\ c = 0$$

Lemma 7.1. *Given a smooth function $u(x)$, let $U_j = \frac{1}{\Delta x}\int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u(x)\ dx$, and let $\tilde{u}(x) = U_j + \delta U_j \frac{x - x_j}{\Delta x}, \delta U_j = (U_{j+1} - U_{j-1})/2$, then $|\tilde{u}(x) - u(x)| = O(\Delta x)^3$ for $x \in (x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}})$.*

When u has discontinuities or u_x changes sign, we need to put a "limiter" to avoid oscillation of $\tilde{u}$.

Examples of limiters

(a) Minmod limiter:

$$\delta U_j = \text{minmod}(U_{j+1} - U_j, U_j - U_{j-1}), \tag{7.20}$$

$$\text{minmod}(a, b) := \begin{cases} \text{sign}(a)\min\{|a|, |b|\} & \text{if } a,b \text{ have same signs} \\ 0 & \text{otherwise.} \end{cases} \tag{7.21}$$

(b) A shaper minmod:

$$\delta U_j = \text{minmod}(\frac{U_{j+1} - U_{j-1}}{2}, 2(U_j - U_{j-1}), 2(U_{j+1} - U_j)).$$

(2) **A linear solver for half-time step**
Consider the linear advection equation

$$u_t + au_x = 0, \quad a \geq 0,$$

with linear data

$$\tilde{u}(x, t_n) = \begin{cases} U_j + \delta U_j \frac{x - x_j}{\Delta x} & x < x_{j+\frac{1}{2}} \\ U_{j+1} + \delta U_{j+1} \frac{x - x_{j+1}}{\Delta x} & x > x_{j+\frac{1}{2}} \end{cases}$$

Then

$$\begin{aligned} \tilde{u}_{j+\frac{1}{2}}^{n+\frac{1}{2}} &= \tilde{u}(x_{j+\frac{1}{2}} - a(t - t_n), t_n) \quad (a > 0) \\ &= U_j + \delta U_j (x_{j+\frac{1}{2}} - a(t_{n+\frac{1}{2}} - t_n) - x_j)/\Delta x \\ &= U_j + \delta U_j (\frac{1}{2} - \frac{a\Delta t}{2\Delta x}) \quad \text{let } \nu = \frac{a\Delta t}{\Delta x} \\ F_{j+\frac{1}{2}} &= a\tilde{U}_{j+\frac{1}{2}}^{n+\frac{1}{2}} = a(U_j + \delta U_j (\frac{1}{2} - \frac{\nu}{2})) \end{aligned}$$

To compare with the TVD scheme (7.17), let $\delta U_j = \text{minmod}(U_{j+1} - U_j, U_j - U_{j-1})$

$$\begin{aligned} F_{j+\frac{1}{2}} &= aU_j + (\frac{1}{2} - \frac{\nu}{2}) a(U_{j+1} - U_j) \cdot \phi_{j+\frac{1}{2}} \\ \phi_{j+\frac{1}{2}} &= \frac{\text{minmod}(U_{j+1} - U_j, U_j - U_{j-1})}{U_{j+1} - U_j} \end{aligned}$$

$$\phi(\theta) = \begin{cases} 0 & \theta \leq 0 \\ \theta & 0 \leq \theta \leq 1 \\ 1 & \theta \geq 1 \end{cases}, \quad \theta = \frac{U_j - U_{j-1}}{U_{j+1} - U_j}$$

If $a < 0$, then

$$\begin{aligned} \tilde{u}_{j+\frac{1}{2}}^{n+\frac{1}{2}} &= U_{j+1} + \delta U_{j+1} (-\frac{1}{2} - \frac{a\Delta t}{2\Delta x}) \qquad |\frac{a\Delta t}{\Delta x}| \leq 1 \\ F_{j+\frac{1}{2}} &= a(U_{j+1} + \delta U_{j+1} (-\frac{1}{2} - \frac{\nu}{2})) \end{aligned}$$

This second-order Godunov method reproduces the TVD scheme (7.17).

System Case

$$u_t + Au_x = 0 \tag{7.22}$$

(1) **Reconstruction**
Construct $\tilde{u}(x, t_n)$ to be a piecewise linear function.

$$\tilde{u}(x, t_n) = U_j^n + \delta U_j^n(\frac{x - x_j}{\Delta x})$$

The slope is found by $\delta U_j^n = \text{minmod}(U_j - U_{j-1}, U_{j+1} - U_j)$. We can write it characteristic-wisely: let

$$\begin{aligned} \alpha_{j,k}^L &= \ell_k(U_j - U_{j-1}), \\ \alpha_{j,k}^R &= \ell_k(U_{j+1} - U_j), \\ \alpha_{j,k} &= \text{minmod}(\alpha_{j,k}^L, \alpha_{j,k}^R). \end{aligned}$$

Then $\delta U_j = \sum \alpha_{j,k} r_k$.

(2) **A linear solver for half-time step**
We trace back along the characteristic curve to get u in half time step.

$$\begin{aligned} u_{j+\frac{1}{2}}^{n+\frac{1}{2}} &= \sum_k \ell_k \tilde{u}(x_{j+\frac{1}{2}} - \lambda_k(t^{n+\frac{1}{2}} - t^n), t^n) r_k \\ &= \sum_{\lambda_k \geq 0} \ell_k(U_j + \delta U_j(\frac{1}{2} - \frac{\nu_k}{2}))r_k + \sum_{\lambda_k < 0} \ell_k(U_{j+1} + \delta U_{j+1}(-\frac{1}{2} - \frac{\nu_k}{2}))r_k \\ &= \text{initial state of Riemann data } (U_j, U_{j+1}) \\ &\quad + \sum_{\lambda_k \geq 0}(\ell_k(\frac{1}{2} - \frac{\nu_k}{2})r_k)\delta U_j + \sum_{\lambda_k < 0}(\ell_k(-\frac{1}{2}) - \frac{\nu_k}{2})r_k)\delta U_{j+1}. \end{aligned}$$

Another viewpoint. Let $u_{j+\frac{1}{2},L}^{n+\frac{1}{2}}$ be the solution of (7.22) in $(x_{j-1/2}, x_{j+1/2}) \times (t^n, t^{n+1/2})$. The solution at $(x_{j+1/2}+, t^{n+1/2})$ is given by

$$\begin{aligned} u_{j+\frac{1}{2},L}^{n+\frac{1}{2}} &= u_j^n + \sum_{\lambda_k \geq 0} \ell_k \delta U_j^n (\frac{x_{j+\frac{1}{2}} - \lambda_k \frac{\Delta t}{2} - x_j}{\Delta x}) r_k \\ &= u_j^n + \sum_{\lambda_k \geq 0} \ell_k \delta U_j^n (\frac{1}{2} - \frac{\nu_k}{2}) r_k \end{aligned}$$

where ℓ_k, r_k are left / right eigenvector, λ_k is eigenvalue and $\nu_k = \frac{\lambda_k \Delta t}{\Delta x}$.
Similarly, we solve the linearized equation in $(x_{j+1/2}, x_{j+3/2}) \times (t^n, t^{n+1/2})$ and gives the solution at $(x_{j+1/2}+, t^{n+1/2})$ as

$$u_{j+\frac{1}{2},R}^{n+\frac{1}{2}} = u_{j+1}^n - \sum_{\lambda_k < 0} \ell_k \delta U_{j+1}^n (\frac{x_{j+\frac{1}{2}} - \lambda_k \frac{\Delta t}{2} - x_{j+1}}{\Delta x}) r_k$$

$$= u^n_{j+1} - \sum_{\lambda_k<0} \ell_k \delta U^n_{j+1}(-\frac{1}{2} - \frac{\nu_k}{2}) r_k$$

Then we solve (7.22) with$(u^{n+\frac{1}{2}}_{j+\frac{1}{2},L}, u^{n+\frac{1}{2}}_{j+\frac{1}{2},R})$ as the Riemann data. This gives $u^{n+\frac{1}{2}}_{j+\frac{1}{2}}$. Therefore

$$\begin{aligned}
u^{n+\frac{1}{2}}_{j+\frac{1}{2}} &= u^{n+\frac{1}{2}}_{j+\frac{1}{2},L} + \sum_{\lambda_k \geq 0} \ell_k \delta U_{j+\frac{1}{2}}(-\frac{\lambda_k \frac{\Delta t}{2}}{\Delta x}) r_k \\
&= u^{n+\frac{1}{2}}_{j+\frac{1}{2},L} + \sum_{\lambda_k \geq 0} \ell_k \delta U_{j+\frac{1}{2}}(-\frac{\nu_k}{2}) r_k \\
\text{or} \quad u^{n+\frac{1}{2}}_{j+\frac{1}{2}} &= u^{n+\frac{1}{2}}_{j+\frac{1}{2},R} - \sum_{\lambda_k \leq 0} \ell_k \delta U_{j+\frac{1}{2}}(-\frac{\nu_k}{2}) r_k \\
\text{or} \quad u^{n+\frac{1}{2}}_{j+\frac{1}{2}} &= \frac{U^{n+\frac{1}{2}}_{j+\frac{1}{2},L} + U^{n+\frac{1}{2}}_{j+\frac{1}{2},R}}{2} - \frac{1}{2} \sum \text{sign}(\nu_k) \ell_k \delta U_{j+\frac{1}{2}} \frac{\nu_k}{2} r_k, \\
\text{where} \quad \delta U_{j+\frac{1}{2}} &= U^{n+\frac{1}{2}}_{j+\frac{1}{2},R} - U^{n+\frac{1}{2}}_{j+\frac{1}{2},L}.
\end{aligned}$$

(3) We adopt the Godunov flux: $F^{n+1/2}_{j+1/2} = f(U^{n+1/2}_{j+1/2})$. Thus, the scheme is

$$U^{n+1}_j = U^n_j + \frac{\Delta t}{\Delta x}(f(U^{n+\frac{1}{2}}_{j-\frac{1}{2}}) - f(U^{n+\frac{1}{2}}_{j+\frac{1}{2}})).$$

7.3.2 Approximate Riemann Solvers

A detailed exact and approximate Riemann solver can be found in Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, A Practical Introduction.
Some approximate Riemann solvers listed in Wiki.

1. **Characteristic Riemann solver** Consider the Riemann data (u_L, u_R). We look for middle states $u_0 = u_L, u_1, ..., u_n = u_R$. Suppose $u_L \sim u_R$, the original equation can be replaced by

$$u_t + A(\bar{u}) u_x = 0,$$

where $A(u) := f'(u)$ and $\bar{u} = (u_L + u_R)/2$. We will solve this linear hyperbolic equation with Riemann data (u_L, u_R). Let λ_i, ℓ_i, r_i be eigenvalues and eigenvectors of $A(\bar{u})$. Then the solution of the Riemann problem is self-similar and has the form

$$u\left(\frac{x}{t}\right) = u_L + \sum_{\lambda_i < \frac{x}{t}} (\ell_i \cdot (u_R - u_L)) \cdot r_i.$$

One severe error in this approximate Riemann solver is that rarefaction waves are approximated by discontinuities. This will produce non-entropy shocks. This is particularly serious for Godunov method which uses Riemann solution at $x/t = 0$. To

cure this problem, we expand such a linear discontinuity by a linear fan. Precisely, suppose $\lambda_i(u_{i-1}) < 0, \lambda_i(u_i) > 0$, this suggests that there exists rarefaction fan crossing $\frac{x}{t} = 0$. We then expand this discontinuity by a linear fan. At $x/t = 0$, we thus choose the mid state u_m (the solution of Riemann problem at $x/t = 0$) as

$$u_m = (1-\alpha)u_{i-1} + \alpha u_i, \quad \alpha = \frac{-\lambda_i(u_{i-1})}{\lambda_i(u_i) - \lambda_i(u_{i-1})}.$$

2. **Roe's Riemann solver** A final remark, the above $A(\bar{u})$ can be replaced by a Roe's matrix $\tilde{A}(u_L, u_R)$ in the application of gas dynamics. The matrix $\tilde{A}(u_L, u_R)$ satisfies

   - $\tilde{A}(u, u) = f'(u)$,
   - $f(u_R) - f(u_L) = \tilde{A}(u_r - u_L)$,
   - $\tilde{A}$ is diagonalizable by real eigenvalues with independent eigenvectors.

   The advantage of the Roe matrix is that it is exact for shock representation. For gas dynamics, such $\tilde{A}$ exists. Indeed, we can express the conservative vector u (don't be mixed up with the velocity u) and the flux vector $f(u)$

$$u = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix}, \quad f(u) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ (E+p)u \end{bmatrix}$$

   as a *quadratic functions* of

$$w := \rho^{1/2}[1, u, H]^T, \quad \rho H := E + p.$$

   Note that the total energy density $E = \frac{1}{2}\rho|u|^2 + \frac{p}{\gamma-1}$. The expressions are

$$u = \begin{bmatrix} w_1^2 \\ w_1 w_2 \\ \frac{w_1 w_3}{\gamma} + \frac{\gamma-1}{2\gamma} w_2^2 \end{bmatrix}, \quad f(u) = \begin{bmatrix} w_1 w_2 \\ \frac{\gamma-1}{\gamma} w_1 w_3 + \frac{\gamma+1}{2\gamma} w_2^2 \\ w_2 w_3 \end{bmatrix}.$$

   Then

$$u_R - u_L = \tilde{B}(w_R - w_L),$$
$$f(u_R) - f(u_L) = \tilde{C}(w_R - w_L)$$

   and

$$f(u_R) - f(u_L) = \tilde{C}\tilde{B}^{-1}(u_R - u_L) := \tilde{A}(u_R - u_L),$$

   where

$$\tilde{B} = \begin{bmatrix} 2\bar{w}_1 & 0 & 0 \\ \bar{w}_2 & \bar{w}_1 & 0 \\ \frac{\bar{w}_3}{\gamma} & \frac{\gamma-1}{\gamma}\bar{w}_2 & \frac{\bar{w}_1}{\gamma} \end{bmatrix}, \quad \tilde{C} = \begin{bmatrix} \bar{w}_2 & \bar{w}_1 & 0 \\ \frac{\gamma-1}{\gamma}\bar{w}_3 & \frac{\gamma+1}{\gamma}\bar{w}_2 & \frac{\gamma-1}{\gamma}\bar{w}_1 \\ 0 & \bar{w}_3 & \bar{w}_2 \end{bmatrix}.$$

   The $\bar{w} := \frac{1}{2}(w_L + w_R)$. For detail, see Roe's original paper (1981), "Approximate Riemann solvers, parameter vectors and difference schemes". Journal of Computational Physics. 43 (2): 357–372.

3. **Harten-Lax-von Leer (HLL, HLLC) approximate Riemann solver**

7.4 ENO/WENO schemes

This section is a summary of the following paper:

- Chi-Wang Shu, Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws, NASA, ICASE Report, 1997.

Let us consider a system of hyperbolic conservation laws:

$$u_t + f(u)_x = 0.$$

Let us choose an 1D grid $\{x_{j+1/2}\}_{j\in\mathbb{Z}}$. Let $I_j := (x_{j-1/2}, x_{j+1/2})$ be the jth cell, x_j be the mid point of I_j and U_j the average of the unknown u on I_j.

The ENO/WENO schemes consists of 3 steps:

- **Reconstruction**: given $\{U_j^n\}_{j\in\mathbb{Z}}$, reconstruct a piecewise polynomial function $\tilde{u}^n(x) = p_j(x)$ in each cell.
- **Numerical flux**: choosing an entropy satisfying numerical flux $F(u, v)$.
  - Lax-Friedrichs flux: $F_{LF}(U, V) = f^+(U) + f^-(V)$, where we split the flux f into
    $$f(u) = f^+(u) + f^-(u), \quad f^\pm := f(u) \pm \alpha u, \quad |f'(u)| \le \alpha.$$
  - Godunov flux: $F_G(U, V) = f(\bar{U})$, where $\bar{U}$ is the solution to the Riemann problem (U, V) on the array $x/t = 0$. We can use an approximate Riemann solver for finding an approximation of $\bar{U}$.
  - Kinetic flux
- **A TVD-ODE solver for updating**: generate $\{U_j^{n+1}\}_{j\in\mathbb{Z}}$ by using an ODE solver for
  $$\dot{U}_j = \frac{1}{\Delta x}\left(F(\tilde{u}_{j-1/2,-}, \tilde{u}_{j-1/2,+}) - F(\tilde{u}_{j+1/2,-}, \tilde{u}_{j+1/2,+})\right).$$
  The ODE solver is required to be total variation diminishing (TVD). The forward Euler ODE solver is TVD. The RK2 is a convex combination of two forward Euler solvers, and thus is a TVD solver.

7.4.1 Reconstruction for smooth functions

Problem Let $v(\cdot)$ be a smooth function. Suppose we are given $\bar{v}_i := \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} v(x)\,dx$, we would like to find a piecewise smooth function $\tilde{v}$ such that

- $\tilde{v}(x) = p_i(x)$ in I_i, a degree $k-1$ polynomial
- $\bar{\tilde{v}}_i = \bar{v}_i$
- $\tilde{v}(x) = v(x) + O(\Delta x^k)$ in smooth region
- TVB (total variation bounded):
  $$TV(\tilde{v}) \le TV(v) + O(\Delta x^k).$$

Method

1. Let $V(x) = \int_{-\infty}^{x} v(\xi)\, d\xi$. $V_{i+1/2} = V(x_{i+1/2}) = \sum_{j=-\infty}^{i} \bar{v}_j \Delta x_j$.

2. Let P be a polynomial of degree k which interpolates V at $x_{i-r-1/2}, \cdots, x_{i+s+1/2}$ ($r+s+2 = k+1$ points). †

3. Define $p = P'$, then

$$\bar{p}_j = \frac{1}{\Delta x_j}\left(P(x_{j+1/2}) - P(x_{j-1/2})\right) = \frac{1}{\Delta x_j}\left(V(x_{j+1/2}) - V(x_{j-1/2})\right) = \bar{v}_j$$

   for $j = i-r, ..., i+s$.

4. From interpolation theory,

$$P(x) = V(x) + O(\Delta x^{k+1}), \quad p(x) - v(x) = O(\Delta x^k), x \in I_{i-r}, ..., I_{i+s}.$$

Exact formula for edge states We are interested to evaluate $p(x_{i+1/2})$, denoted by $v_{i+1/2}$, in terms of $\bar{v}_{i-r}, ..., \bar{v}_{i-r+k}$:

$$v_{i+1/2} = p(x_{i+1/2}) = \sum_{j=0}^{k} c_{rj} \bar{v}_{i-r+j}.$$

To find the coefficients c_{rj}, $j = 0, ..., k$, we notice that

1. P has the expression:

$$P(x) = \sum_{m=0}^{k} V_{i-r-1/2+m} L_m(x_{i-r-1/2}, ..., x_{i+s+1/2}, x),$$

$$L_m(x_{i-r-1/2}, ..., x_{i+s+1/2}, x) = \prod_{l=0, l\neq m}^{k} \frac{x - x_{i-r-1/2+l}}{x_{i-r-1/2+m} - x_{i-r-1/2+l}}.$$

   The polynomial L_m is the Lagrange interpolation polynomial satisfying $L_m(x_{i-r-1/2+l}) = \delta_{ml}$, $0 \le l, m \le k$.

$$\begin{aligned} P(x) - V_{i-r-1/2} &= \sum_{m=0}^{k} (V_{i-r-1/2+m} - V_{i-r-1/2}) L_m(x) \\ &= \sum_{m=0}^{k} \sum_{j=0}^{m-1} \bar{v}_{i-r+j} \Delta x_{i-r+j} L_m(x). \end{aligned}$$

†Here, the location of the stencil is another degree of freedom. Let us use r to identify the location of the stencil. Thus, P depends on two parameters: r and k.

2. Taking derivative, we obtain p

$$p(x) = \sum_{m=0}^{k} \sum_{j=0}^{m-1} \bar{v}_{i-r+j} \Delta x_{i-r+j} L'_m(x).$$

3. Evaluate $p(x)$ at $x_{i+1/2}$, we obtain

$$v_{i+1/2} = p(x_{i+1/2}) = \sum_{j=0}^{k-1} c_{rj} \bar{v}_{i-r+j},$$

$$c_{rj} = \left(\sum_{m=j+1}^{k} \frac{\sum_{\substack{l=0 \\ l \neq m}}^{k} \prod_{\substack{q=0 \\ q \neq m,l}}^{k} \left(x_{i+1/2} - x_{i-r+q-1/2} \right)}{\prod_{\substack{l=0 \\ l \neq m}}^{k} \left(x_{i-r+m-1/2} - x_{i-r+l-1/2} \right)} \right) \Delta x_{i-r+j}.$$

For uniform grid,

$$c_{rj} = \left(\sum_{m=j+1}^{k} \frac{\sum_{\substack{l=0 \\ l \neq m}}^{k} \prod_{\substack{q=0 \\ q \neq m,l}}^{k} (r - q + 1)}{\prod_{\substack{l=0 \\ l \neq m}}^{k} (m - l)} \right).$$

4. Examples: $k = 3$,

$$\begin{aligned} v_{i+1/2} &= \frac{11}{6}\bar{v}_{i+1} - \frac{7}{6}\bar{v}_{i+2} + \frac{1}{3}\bar{v}_{i+3} + O(\Delta x^3), \\ &= \frac{1}{3}\bar{v}_i + \frac{5}{6}\bar{v}_{i+1} - \frac{1}{6}\bar{v}_{i+2} + O(\Delta x^3) \\ &= -\frac{1}{6}\bar{v}_{i-1} + \frac{5}{6}\bar{v}_i + \frac{1}{3}\bar{v}_{i+1} + O(\Delta x^3) \\ &= \frac{1}{3}\bar{v}_{i-2} - \frac{7}{6}\bar{v}_{i-1} + \frac{11}{6}\bar{v}_i + O(\Delta x^3). \end{aligned}$$

The set of the grid points used to construct $p(x)$ in I_i is called its *stencil*, which is

$$S_r^k(i) := \{x_{i-r}, ..., x_{i-r+k-1}\}.$$

In the first example above, the stencil set is $S_{-1}^3(i) = \{x_{i+1}, x_{i+2}, x_{i+3}\}$.
Table 2.1 in Shu's note has more examples for various r and j.

7.4.2 ENO approximation

The ENO approximation uses Newton's divided differences to represent the interpolation polynomial P. The construction of the stencil set is by successively adding stencil point from left or right, which has the smallest divided difference.

Table 7.1: The ENO coefficient table

$n=2$	$n=3$			$n=4$				$n=5$	
j	$c_j^{(2)}$	$c_j^{(3)}$	$d_j^{(3)}$	$c_j^{(3)}d_j^{(3)}$	$c_j^{(4)}$	$d_j^{(4)}$	$c_j^{(4)}d_j^{(4)}$	$e_j^{(4)}$	$c_j^{(5)}$	$d_j^{(5)}$
0	$\frac{1}{2}$	$-\frac{1}{3}$	$\frac{1}{2}$	$-\frac{1}{6}$	$\frac{1}{4}$	$-\frac{1}{6}$	$-\frac{1}{24}$	$\frac{1}{24}$	$-\frac{1}{5}$	$\frac{1}{20}$
1	$\frac{1}{2}$	$\frac{2}{3}$	$-\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{2}$	0	0	$-\frac{1}{12}$	$\frac{2}{5}$	$-\frac{1}{10}$
2	0	$-\frac{1}{3}$	$\frac{1}{2}$	$-\frac{1}{6}$	$-\frac{1}{4}$	$\frac{1}{6}$	$\frac{1}{24}$	$\frac{1}{24}$	$-\frac{1}{5}$	$\frac{1}{20}$
3	0	0	0	0	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{144}$	$\frac{1}{144}$	$\frac{4}{5}$	$-\frac{1}{5}$
4	0	0	0	0	0	0	0	0	$-\frac{1}{5}$	$\frac{1}{4}$

1. **Divided Difference** In the theory of polynomial interpolation, we have the following polynomial interpolation represented by the Newton's divided difference. Let f be a function and let $\{x_i\}$ be grid points. The Newton's divided difference is defined as

$$\begin{aligned} f[x_0] &:= f(x_0) \\ f[x_0, ..., x_j] &:= \frac{f[x_1, ..., x_j] - f[x_0, ..., x_{j-1}]}{x_j - x_0}. \end{aligned}$$

   We notice that

$$\begin{aligned} f[x_0, x_1] &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} \to f'(x_0) \quad \text{as } x_1 \to x_0 \\ f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \to \frac{f''(x_0)}{2!} \quad \text{as } x_1, x_2 \to x_0 \\ f[x_0, ..., x_j] &:= \frac{f[x_1, ..., x_j] - f[x_0, ..., x_{j-1}]}{x_j - x_0} \to \frac{f^{(j)}(x_0)}{j!}, \quad \text{as } x_1, ..., x_j \to x_0. \end{aligned}$$

2. **Examples of polynomial approximation**

$$\begin{aligned} f(x) &= f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + O(\Delta x^2) \\ &= f(x_0) + f[x_0, x_1](x - x_0) + O(\Delta x^2) \\ &= f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + O(\Delta x^3) \\ &\vdots \end{aligned}$$

3. **Representation of polynomial interpolant in terms of divided differences:**

   Theorem 7.6. *Suppose $P_k(x)$ interpolates f at $x_0, ..., x_k$. Then P_k can be expressed as*

$$P_k(x) = \sum_{j=0}^{k} f[x_0, ..., x_j] \prod_{m=0}^{j-1} (x - x_m). \tag{7.23}$$

Proof by induction. Let P_{k-1} interpolates f at $x_0, ..., x_{k-1}$ and $Q(x)$ interpolates V at $x_1, ..., x_k$. Then

$$Q(x) = \sum_{j=1}^{k} f[x_1, ..., x_j] \prod_{m=1}^{j-1} (x - x_m) \quad \text{(by induction hypotheses)}$$

$$P_k(x) = Q(x) + \frac{x - x_k}{x_k - x_0} \left(Q(x) - P_{k-1}(x) \right).$$

The second line follows from the fact that both sides take same values at $x_0, ..., x_k$ and have the same degree k. From this formula and the induction hypothesis, we get the coefficient of the term $\prod_{m=0}^{k-1}(x - x_m)$ is $(f[x_1, ..., x_k] - f[x_0, ..., x_{k-1}])/(x_k - x_0)$. □

4. **ENO reconstruction:** ENO scheme is to construct a stencil to determine the polynomial interpolant. Let us use $\bar{S}$ for the stencil for V and S for the corresponding stencil $\bar{v}$. For example, $\bar{S}_0^2(i) = \{x_{i-1/2}, x_{i+1/2}, x_{i+3/2}\}$, its corresponding stencil for $\bar{v}$ is $S_0^2(i) = \{x_i, x_{i+1}\}$. In general,

$$\bar{S}_r^k(i) = \{x_{i-r-1/2}, ..., x_{i-r+k-1/2}\}, \quad S_r^k(i) = \{x_{i-r}, ..., x_{i-r+k-1}\}, \quad r = 0, ..., k-1.$$

- Let us first compute the divided differences of V for degree $1, ..., k$ using $\bar{v}$, that is, $V[x_{i-r}, ..., x_{i-r+k}]$. Here, we use indices $(i - r)$ and k with fixed i.
- Let us start from $\bar{S}^1 := \{x_{i-1/2}, x_{i+1/2}\}$. We approximate V by

$$P^1(x) = V[x_{i-1/2}] + V[x_{i-1/2}, x_{i+1/2}](x - x_{i-1/2}).$$

- We compare $|V[x_{i-3/2}, x_{i-1/2}, x_{i+1/2}]|$ and $|V[x_{i-1/2}, x_{i+1/2}, x_{i+3/2}]|$. We add one side grid point into $\bar{S}^1$ according to
  - if $|V[x_{i-3/2}, x_{i-1/2}, x_{i+1/2}]| \le |V[x_{i-1/2}, x_{i+1/2}, x_{i+3/2}]|$, then

$$\bar{S}^2 = \bar{S}^1 \cup \{x_{i-3/2}\}$$
$$P^2 = P^1 + V[x_{i-3/2}, x_{i-1/2}, x_{i+1/2}](x - x_{i-1/2})(x - x_{i+1/2}),$$

  - If $|V[x_{i-3/2}, x_{i-1/2}, x_{i+1/2}]| > |V[x_{i-1/2}, x_{i+1/2}, x_{i+3/2}]|$, then

$$\bar{S}^2 = \bar{S}^1 \cup \{x_{i+3/2}\}$$
$$P^2 = P^1 + V[x_{i-1/2}, x_{i+1/2}, x_{i+3/2}](x - x_{i-1/2})(x - x_{i+1/2}).$$

- We repeat the above procedure until $\bar{S}^k$, we obtain P^k which has degree k. Let us denote this P^k by P_i because the procedure starts from the cell I_i.
- Define $p_i = P_i'$ and compute

$$v_{i-1/2+} = p_i(x_{i-1/2}), \quad v_{i+1/2-} = p_i(x_{i+1/2}).$$

5. **Properties:**

- Accuracy: $P_i = V(x) + O(\Delta x^{k+1})$ if I_i does not contain a discontinuity.
- Monotonicity: P_i is monotone if I_i does not contain a discontinuity.
- The reconstruction is total variation bounded (TVB), i.e.there exists a function z which satisfying

$$z = P_i + O(\Delta x^{k+1}), \quad TV(z) \leq TV(V) \text{ in } I_i.$$

7.4.3 WENO reconstruction

1. Recall that we have k stencils around a cell I_i:

$$S_r^k(i) = \{x_{i-r}, ..., x_{i-r+k-1}\}, \quad r = 0, ..., k-1.$$

These produces k approximate values of $v_{i+1/2}$:

$$v_{i+1/2}^r := \sum_{j=0}^{k-1} c_{rj} \bar{v}_{i-r+j}, \quad r = 0, ..., k-1.$$

The WENO reconstruction takes a convex combinationof $v_{i+1/2}^r$ to a higher order approximation of $v(x_{i+1/2})$, provided v is smooth around $x_{i+1/2}$:

$$v_{i+1/2} = \sum_{r=0}^{k-1} \omega_r v_{i+1/2}^r.$$

$$\omega_r \geq 0, \quad \sum_{r=0}^{k-1} \omega_r = 1.$$

2. If v is smooth, then we should choose $\omega_r = d_r$, where d_r are the coefficients such that

$$v_{i+1/2} = \sum_{r=0}^{k-1} d_r v_{i+1/2}^r = v(x_{i+1/2}) + O(\Delta x^{2k-1}).$$

For $k = 2, 3$,

$$\begin{array}{llll} d_0 = 2/3, & d_1 = 1/3, & & k = 2, \\ d_0 = 3/10, & d_1 = 3/5, & d_2 = 1/10, & k = 3. \end{array}$$

We always have $d_r \geq 0$ and $\sum_{r=0}^{k-1} d_r = 1$.

3. In the smooth region, we would like to choose

$$\omega_r = d_r + O(\Delta x^{k-1}).$$

This would give

$$v_{i+1/2} = \sum_{r=0}^{k-1} \omega_r v_{i+1/2}^r = v(x_{i+1/2}) + O(\Delta x^{2k-1}).$$

In the non-smooth region, where $|v_{i+1/2}^r|$ is big, we would like to have $\omega_r \sim 0$.

4. Finally, we choose

$$\omega_r = \frac{\alpha_r}{\sum_{s=0}^{k-1}\alpha_s}, \quad r = 0, ..., k-1.$$
$$\alpha_r = \frac{d_r}{(\varepsilon + \beta_r)^2}, \quad \varepsilon = 10^{-6}.$$

The coefficients β_r are called a smooth indicators. We require

$$\beta_r = \begin{cases} O(\Delta x^2) & \text{in smooth region} \\ O(1) & \text{in non-smooth region} \end{cases}$$

This leads to $\omega_r = O(1)$ in smooth region, and $\omega = O(\Delta x^4)$ in non-smooth region. For $k = 2$, we choose

$$\beta_0 = (\bar{v}_{i+1} - \bar{v}_i)^2$$
$$\beta_1 = (\bar{v}_i - \bar{v}_{i-1})^2 .$$

For $k = 3$, β_r are chosen as

$$\beta_0 = \frac{13}{12}(\bar{v}_i - 2\bar{v}_{i+1} + \bar{v}_{i+2})^2 + \frac{1}{4}(3\bar{v}_i - 4\bar{v}_{i+1} + \bar{v}_{i+2})^2 ,$$
$$\beta_1 = \frac{13}{12}(\bar{v}_{i-1} - 2\bar{v}_i + \bar{v}_{i+1})^2 + \frac{1}{4}(v_{i-1} - v_{i+1})^2 ,$$
$$\beta_2 = \frac{13}{12}(\bar{v}_{i-2} - 2\bar{v}_{i-1} + \bar{v}_i)^2 + \frac{1}{4}(\bar{v}_{i-2} - 4\bar{v}_{i-1} + 3\bar{v}_i)^2 .$$

5. The case $k = 3$ has stencil set $S_r^3 = \{x_{i-r,...,i-r+2}\}$, $r = 0, 1, 2$. It involves 5 stencil points $\{x_{i-2}, x_{i+2}\}$. Such WENO is called the WENO5.

7.4.4 Finite Volume WENO

The WENO produces an approximation of u at $x_{j+1/2}-$ and $x_{j-1/2}+$ in cell I_j from the cell averages $\{\bar{U}_k\}$. Let us denote them as $U_{j+1/2\pm}$. Then we can use the Lax-Friedrichs flux F_{LF}, or the Godunov flux F_G to compute the flux at $x_{j+1/2}$. We reach

$$\dot{U}_j = \frac{1}{\Delta x}\left(F(U_{j-1/2-}, U_{j-1/2+}) - F(U_{j+1/2-}, U_{j+1/2+})\right).$$

Let us write this system of ODE by

$$\dot{U} = L(U).$$

This ODE can be solved by RK2:

$$U^{(1)} = U^n + \Delta t L(U^n)$$

$$U^{n+1} = U^n + \frac{1}{2}\Delta t \left(L(U^n) + L(U^{(1)})\right)$$
$$= \frac{1}{2}U^n + \frac{1}{2}U^{(1)} + \frac{1}{2}\Delta t L(U^{(1)}).$$

This time-advancing procedure is a convex combination of the forward Euler method. It can be shown that such a method is TVD if a forward Euler method is TVD.

A TVD-RK3 is given by

$$U^{(1)} = U^n + \Delta t L(U^n)$$
$$U^{(2)} = \frac{3}{4}U^n + \frac{1}{4}U^{(1)} + \frac{1}{4}\Delta t L(U^{(1)})$$
$$U^{n+1} = \frac{1}{3}U^n + \frac{2}{3}U^{(2)} + \frac{2}{3}\Delta t L(U^{(2)}).$$

This is a convex combination of a sequence of forward Euler methods. It is proven that there are no further higher-order RK methods that are convex combinations of forward Euler methods. For higher-order RK methods (which not convex combination of forward Euler methods) and multistep methods with TVB property, see sections 4.2.1, 4.2.2 of Shu's Note.

7.4.5 Finite Difference WENO

1. In the finite difference approach, we discretize the flux derivative by finite difference:

$$\dot{U}_j = \frac{\hat{f}_{j-1/2} - \hat{f}_{j+1/2}}{\Delta x} = \hat{L}(U). \tag{7.24}$$

   This method is a conservative method.

2. Computing $\hat{f}_{j+1/2}(U)$:

   (a) Flux splitting: We split the flux f into two parts: positive and negative fluxes

$$f(u) = f^p(u) + f^m(u).$$

   An example is the Lax-Friedrich flux:

$$f^p(u) := \frac{1}{2}\left(f(u) + \alpha u\right), \quad f^m(u) := \frac{1}{2}\left(f(u) - \alpha u\right)$$

   The coefficient α is chosen such that

$$\max|f'(u)| \le \alpha.$$

   Thus, f^p is part of the flux f with positive characteristic speeds.

   (b) Given $U = (U_j)_{j\in\mathbb{Z}}$, we evaluate $f^p_j(U) := f^p(U_j)$ and $f^m_j(U) := f^m(U_j)$.

(c) Using ENO/WENO, we reconstruct $f^p_{j+1/2-}(U)$ from $f^p_j(U)$ and $f^m_{j-1/2+}(U)$ from $f^m_j(U)$.

(d) $\hat{f}_{j+1/2}(U) = f^p_{j+1/2-}(U) + f^m_{j+1/2+}(U)$.

3. Use the TVD-RK2 method for temporal discretization:

$$\begin{aligned} U^{(1)} &= U^n + \Delta t \hat{L}(U^n) \\ U^{n+1} &= U^n + \frac{1}{2}\Delta t \left(\hat{L}(U^n) + \hat{L}(U^{(1)})\right) \\ &= \frac{1}{2}U^n + \frac{1}{2}U^{(1)} + \frac{1}{2}\Delta t \hat{L}(U^{(1)}). \end{aligned}$$

4. Remark. Without the WENO step, the fluxes $f^p_{j+1/2} = f^p(U_j)$ and $f^m_{j+1/2} = f^m(U_{j+1})$.

$$\begin{aligned} F_{j+1/2}(U_j, U_{j+1}) &= f^p(U_j) + f^m(U_{j+1}) \\ &= \frac{1}{2}\left(f(U_j) + \alpha U_j + f(U_{j+1}) - \alpha U_{j+1}\right) \end{aligned}$$

Thus, the forward Euler gives

$$\begin{aligned} U^{n+1}_j &= U^n_j + \frac{\Delta t}{\Delta x}\left(F_{j-1/2}(U^n_{j-1}, U^n_j) - F_{j+1/2}(U^n_j, U^n_{j+1})\right) \\ &= U^n_j + \frac{\Delta t}{\Delta x}\left[\frac{1}{2}\left(f(U_{j-1}) + \alpha U_{j-1} + f(U_j) - \alpha U_j\right) - \frac{1}{2}\left(f(U_j) + \alpha U_j + f(U_{j+1}) - \alpha U_{j+1}\right)\right] \\ &= U^n_j + \frac{\Delta t}{2\Delta x}\left[f(U_{j-1}) - f(U_{j+1}) + \alpha(U_{j-1} + U_{j+1} - 2U_j)\right] \end{aligned}$$

This is exactly the Lax-Friedrichs method.

With the WENO interpolation, the limiter of the interpolation is applied to the fluxes, instead to the state variables. This is the same as the flux limiter methods.

7.4.6 A sample matlab code

A finite difference WENO5-RK3 code for solving the 1D Euler equation is available in GitHub: WENO5 for 1D gas dynamics.

1. The reconstruction can be applied to
   - the conservative quantities: $(\rho, \rho u, E)$,
   - the characteristic coefficients: $\ell_i \cdot (\rho, u, S)$, $i = 1, 2, 3$
   - the primitive variables: (ρ, p, u).
2. The parameters: $r = 3$, $k = 3$. The stencils are

$$S_0 = \{x_{i-2}, x_{i-1}, x_i\}, \quad S_1 = \{x_{i-1}, x_i, x_{i+1}\}, \quad S_2 = \{x_i, x_{i+1}, x_{i+2}\}.$$

3. The coefficients are

$$c_{0,j} = [11/6, -7/6, 2/6], \quad c_{1,j} = [2/6, 5/6, -1/6], \quad c_{2,j} = [-1/6, 5/6, 2/6].$$

The weights are $d = [1/10, 6/10, 3/10], \quad \varepsilon = 10^{-6}$.
The β's are

$$\begin{aligned}
\beta_0 &= \frac{13}{12}\left(\bar{v}_i - 2\bar{v}_{i+1} + \bar{v}_{i+2}\right)^2 + \frac{1}{4}\left(3\bar{v}_i - 4\bar{v}_{i+1} + \bar{v}_{i+2}\right)^2, \\
\beta_1 &= \frac{13}{12}\left(\bar{v}_{i-1} - 2\bar{v}_i + \bar{v}_{i+1}\right)^2 + \frac{1}{4}\left(v_{i-1} - v_{i+1}\right)^2, \\
\beta_2 &= \frac{13}{12}\left(\bar{v}_{i-2} - 2\bar{v}_{i-1} + \bar{v}_i\right)^2 + \frac{1}{4}\left(\bar{v}_{i-2} - 4\bar{v}_{i-1} + 3\bar{v}_i\right)^2.
\end{aligned}$$

4. The boundary conditions:

   - For the Dirichlet boundary condition, we set the correct boundary state after the reconstruction step.
   - For the wall conditions (Neumann boundary condition),we set the ghost states before each reconstruction step.

5. The TVD-RK3 updating:

$$\begin{aligned}
U^{(1)} &= U^n + \Delta t \hat{L}(U^n) \\
U^{(2)} &= \frac{3}{4}U^n + \frac{1}{4}U^{(1)} + \frac{1}{4}\Delta t \hat{L}(U^{(1)}) \\
U^{n+1} &= \frac{1}{3}U^n + \frac{2}{3}U^{(2)} + \frac{2}{3}\Delta t \hat{L}(U^{(2)}).
\end{aligned}$$

The Sod tube problem is a standard test problem for 1D gas code. It has Riemann initial data:

$$\begin{bmatrix} \rho_L \\ P_L \\ u_L \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} \rho_R \\ P_R \\ u_R \end{bmatrix} \begin{bmatrix} 0.125 \\ 0.1 \\ 0 \end{bmatrix}.$$

The solution to the Sod shock tube problem is given in Figure 7.3.

7.5 Multidimensions

There are two kinds of methods.

1. Splitting method.
2. Unsplitting method.

We consider two-dimensional case.

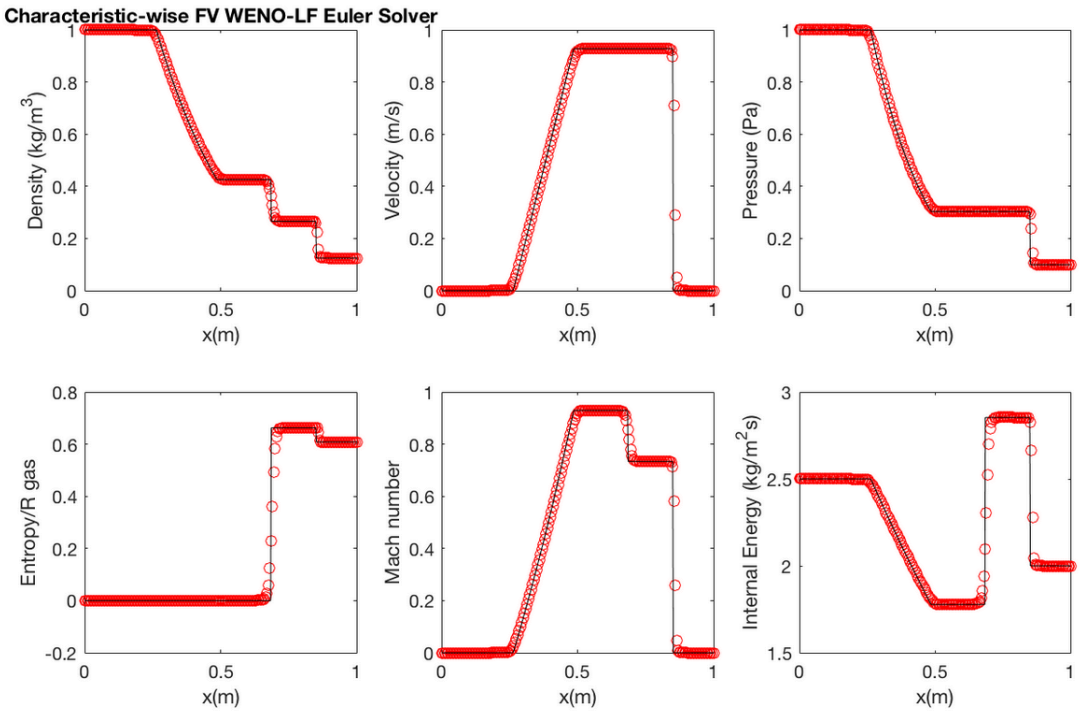


Figure 7.3: The solution to the Sod tube problem. The result was produced by a matlab program coded by Manuel Diaz, 02.10.2012, NTU Taiwan, and last updated on 06.20.2018, NHRI Taiwan.

7.5.1 Splitting Method

We start from

$$u_t + Au_x + Bu_y = 0. \tag{7.25}$$

This equation can be viewed as

$$u_t = (-A\partial_x - B\partial_y)u.$$

Then the solution operator is:

$$e^{-t(A\partial_x + B\partial_y)},$$

which can be approximate by $e^{-tA\partial_x}e^{-tB\partial_y}$ for small t. Let $\mathcal{A} = -A\partial_x, \mathcal{B} = -B\partial_y$, we have

$$u = e^{t(\mathcal{A}+\mathcal{B})}u_0.$$

Consider $e^{t(\mathcal{A}+\mathcal{B})}$,

$$e^{t(\mathcal{A}+\mathcal{B})} = 1 + t(\mathcal{A}+\mathcal{B}) + \frac{t^2}{2}(\mathcal{A}^2 + \mathcal{B}^2 + \mathcal{A}\mathcal{B} + \mathcal{B}\mathcal{A}) + \cdots$$

$$
\begin{aligned}
e^{t\mathcal{B}} \cdot e^{t\mathcal{A}} &= (1 + t\mathcal{B} + \frac{t^2}{2}\mathcal{B}^2 + \cdots)(1 + t\mathcal{A} + \frac{t^2}{2}\mathcal{A}^2 + \cdots) \\
&= 1 + t(\mathcal{A} + \mathcal{B}) + \frac{t^2}{2}(\mathcal{A}^2 + \mathcal{B}^2) + t^2\mathcal{B}\mathcal{A} + \cdots
\end{aligned}
$$

$$
\therefore \quad e^{t(\mathcal{A}+\mathcal{B})} - e^{t\mathcal{B}} \cdot e^{t\mathcal{A}} = \frac{t^2}{2}(\mathcal{A}\mathcal{B} - \mathcal{B}\mathcal{A}) + \mathcal{O}(t^3).
$$

Now we can design splitting method as:
Given $\{U^n_{i,j}\}$,

1. For each j, solve $u_t + Au_x = 0$ with data $\{U^n_j\}$ for Δt step. This gives $\bar{U}^n_{i,j}$.

$$
\bar{U}^n_{i,j} = U^n_{i,j} + \frac{\Delta t}{\Delta x}(F(U^n_{i-1,j}, U^n_{i,j}) - F(U^n_{i,j}, U^n_{i+1,j}))
$$

   where $F(U, V)$ is the numerical flux for $u_t + Au_x = 0$.

2. For each i, solve $u_t + Bu_y = 0$ for Δt step with data $\{\bar{U}^n_{i,j}\}$. This gives $U^{n+1}_{i,j}$.

$$
U^{n+1}_{i,j} = \bar{U}^n_{i,j} + \frac{\Delta t}{\Delta y}(G(\bar{U}^n_{i,j-1}, \bar{U}^n_{i,j}) - G(\bar{U}^n_{i,j}, \bar{U}^n_{i,j+1}))
$$

The error is first order in time $n(\Delta t)^2 = O(\Delta t)$.

To reach higher order time splitting, we may approximate $e^{t(\mathcal{A}+\mathcal{B})}$ by polynomials $P(e^{t\mathcal{A}}, e^{t\mathcal{B}})$ or rationals $R(e^{t\mathcal{A}}, e^{t\mathcal{B}})$. For example, the Strang splitting (or the Trotter product) is given by

$$
e^{t(\mathcal{A}+\mathcal{B})} = e^{\frac{1}{2}t\mathcal{A}} e^{t\mathcal{B}} e^{\frac{1}{2}t\mathcal{A}} + O(t^3).
$$

For $t = n\Delta t$,

$$
\begin{aligned}
e^{t(\mathcal{A}+\mathcal{B})}u_0 &= (e^{\frac{1}{2}\Delta t\mathcal{A}} e^{\Delta t\mathcal{B}} e^{\frac{1}{2}\Delta t\mathcal{A}}) \cdots (e^{\frac{1}{2}\Delta t\mathcal{A}} e^{\Delta t\mathcal{B}} e^{\frac{1}{2}\Delta t\mathcal{A}})(e^{\frac{1}{2}\Delta t\mathcal{A}} e^{\Delta t\mathcal{B}} e^{\frac{1}{2}\Delta t\mathcal{A}})u_0 \\
&= e^{\frac{1}{2}\Delta t\mathcal{A}} e^{\Delta t\mathcal{B}} e^{\Delta t\mathcal{A}} e^{\Delta t\mathcal{B}} e^{\Delta t\mathcal{A}} \cdots e^{\Delta t\mathcal{A}} e^{\Delta t\mathcal{B}} e^{\frac{1}{2}\Delta t\mathcal{A}} u_0
\end{aligned}
$$

The Strang splitting is second order.

7.5.2 Unsplitting Methods

The PDE is

$$
u_t + f(u)_x + g(u)_y = 0 \tag{7.26}
$$

Integrate this equation over $(x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) \times (y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}) \times (t_n, t_{n+1})$. We have

$$
U^{n+1}_{i,j} = U^n_{i,j} + \frac{\Delta t}{\Delta x}(\bar{f}^{n+\frac{1}{2}}_{i-\frac{1}{2},j} - \bar{f}^{n+\frac{1}{2}}_{i+\frac{1}{2},j}) + \frac{\Delta t}{\Delta y}(\bar{g}^{n+\frac{1}{2}}_{i,j-\frac{1}{2}} - \bar{g}^{n+\frac{1}{2}}_{i,j+\frac{1}{2}})
$$

where

$$\begin{aligned}\bar{f}^{n+\frac{1}{2}}_{i+\frac{1}{2},j} &= \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f(u(x_{i+\frac{1}{2}}, y_j, t))\ dt \\ \bar{g}^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} &= \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} g(u(x_i, y_{j+\frac{1}{2}}, t))\ dt.\end{aligned}$$

We consider the second-order Godunov method.

1. Reconstruction

   $$\tilde{u}(x, y, t_n) = u^n_{i,j} + \delta_x U_{i,j}(\frac{x - x_i}{\Delta x}) + \delta_y U_{i,j}(\frac{y - y_j}{\Delta y}) \quad \text{in } I = (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) \times (y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}})$$

   For example, $\delta_x U_{i,j} = \text{minmod}(U_{i,j} - U_{i+1,j}, U_{i+1,j} - U_{i,j})$.

2. We need to solve

   $$u_t + Au_x + Bu_y = 0 \text{ with data } \begin{cases} \tilde{u}(x, y, t_n) & \text{for } (x, y) \in I \\ 0 & \text{otherwise} \end{cases}$$

   For scalar case, where $A = a$ and $B = b$, we have

   $$\begin{aligned}\tilde{u}(x_{j+\frac{1}{2}}, y_j, \frac{\Delta t}{2}) &= U^n_{i,j} + \sum_{a>0} \delta_x U_{i,j}(\frac{x_{i+\frac{1}{2}} - \frac{a\Delta t}{2} - x_i}{\Delta x}) + \delta_y U_{i,j}(\frac{y_j - \frac{b\Delta t}{2} - y_j}{\Delta y}) \\ &= U^n_{i,j} + \sum_{a>0} (\delta_x U^n_{i,j}) \cdot (\frac{1}{2} - \frac{\nu_x}{2}) + (\delta_y U^n_{i,j})(-\frac{\nu_y}{2}),\end{aligned}$$

   where $\nu_x = \frac{a\Delta t}{\Delta x}, \nu_y = \frac{b\Delta t}{\Delta y}$. For system case, the speeds a and b are replaced by λ^x_k, λ^y_k, which are the eigenvalues of A and B, respectively.

   $$U^{n+\frac{1}{2}}_{i+\frac{1}{2},L,j} = U^n_{i,j} + \sum_{\lambda^x_k \geq 0} (\frac{1}{2} - \frac{\nu^x_k}{2})(\ell^x_k \cdot \delta_x U_{i,j}) r^x_k + \sum_k (-\frac{\nu^y_k}{2})(\ell^y_k \cdot \delta_y U_{i,j}) r^y_k.$$

   Similarly,

   $$U^{n+\frac{1}{2}}_{i+\frac{1}{2},R,j} = U^n_{i+1,j} + \sum_{\lambda^x_k < 0} (-\frac{1}{2} - \frac{\nu^x_k}{2})(\ell^x_k \cdot \delta_x U_{i+1,j}) r^x_k + \sum_k (-\frac{\nu^y_k}{2})(\ell^y_k \cdot \delta_y U_{i+1,j}) r^y_k$$

   Finally, solve Riemann problem $u_t + Au_x = 0$ with data $\begin{cases} U^{n+\frac{1}{2}}_{i+\frac{1}{2},L,j} \\ U^{n+\frac{1}{2}}_{i+\frac{1}{2},R,j} \end{cases}$

   $$\therefore \quad f^{n+1/2}_{i+1/2,j} = F(U^{n+1/2}_{i+1/2,L,j}, U^{n+1/2}_{i+1/2,R,j}).$$

   Similarly, we compute

   $$g^{n+1/2}_{i,j+1/2} = G(U^{n+1/2}_{i,j,L}, U^{n+1/2}_{i,j+1/2,R}).$$

3. Updating step:

   $$U^{n+1}_{i,j} = U^n_{i,j} + \frac{\Delta t}{\Delta x}(f^{n+\frac{1}{2}}_{i-\frac{1}{2},j} - f^{n+\frac{1}{2}}_{i+\frac{1}{2},j}) + \frac{\Delta t}{\Delta y}(g^{n+\frac{1}{2}}_{i,j-\frac{1}{2}} - g^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}).$$

7.6 Boundary treatments

7.6.1 1D boundary conditions for hyperbolic conservation laws

1. Let us consider a linear hyperbolic system

$$u_t + A(x)u_x = 0 \quad \text{on } [a, b].$$

We need to impose proper boundary conditions in order to determine the solution in the interior uniquely. Let us discuss the left boundary only. The right boundary can be treated in a similar way.

At $x = a$, suppose the eigenvalues of A satisfy

$$\lambda_1 \leq \cdots \leq \lambda_r < 0 < \lambda_{r+1} \leq \cdots \leq \lambda_n.$$

By the method of characteristics, there are r pieces of information that enter the boundary $x = a$ from the interior. We need to impose $n - r$ boundary conditions to comprise full information at $x = a$. The boundary condition has the form:

$$\sum_{k=1}^{n} b_{ik} u_k = d_i, \quad i = r + 1, ..., n,$$

The matrix $(b_{ik})_{(n-r)\times n}$ has full rank in order to have $n - r$ independent boundary conditions.

For example, for the linear advection equation

$$u_t + au_x = 0, \quad \text{on } [0, 1], \quad a > 0.$$

we should impose u at $x = 0$. But we can not impose any boundary condition at $x = 1$ because the value of u at $x = 1$ is determined from the interior.

2. For nonlinear scalar conservation laws, we need to have consistent boundary conditions. Let us consider the inviscid Burgers equation

$$u_t + uu_x = 0, \quad \text{on } [0, 1].$$

At the boundary $x = 0$, if we impose the value u, then u has to satisfy $u > 0$. For example, we can impose boundary condition

$$u(0, t) = 1, \quad u(1, t) = -1.$$

This will form a shock wave with speed 0 in the interior. However, depending on the initial state, the location of this standing shock may or may not stay in the region $[0, 1]$. Think what will happen when the shock approaching to the boundary. Another similar case is

$$u(0, t) = 3, \quad u(1, t) = -1.$$

The asymptotic shock has speed 2, which will eventually leave the computational domain $[0, 1]$. Suppose we want it moves away from the boundary, what should you do?

Note that we cannot impose a boundary condition $u(0, t)$ with $u(0, t) < 0$, nor $u(1, t) > 0$ with $u(1, t) > 0$, because they are inconsistent to the characteristic direction at the boundary, that is, we can only impose left boundary condition when $\lambda(u) > 0$ and right boundary condition when $\lambda(u) < 0$.

Indeed, if we impose a boundary data u, then the corresponding flux at the boundary is determined. However, this flux has to be consistent to the flux computed from the interior.

3. Boundary conditions for the Euler equation. In a finite gas tube, the boundary conditions are classified into

   - Inflow boundary condition: we impose ρ, p, u. A consistent condition is
   $$u - c > 0.$$
   With this, all characteristic speeds $u - c, u, u + c$ are positive, which is consistent with the three conditions ρ, p, u we impose.
   - Outflow boundary condition: if $u - c, u, u + c$ are all positive, then we cannot impose any more conditions. However, if $u - c < 0$ but $u > 0$, then we should impose one themo condition, say p.
   - Wall condition:
   $$u = 0 \quad \text{on the wall.}$$
   In this case, we impose a reflection boundary condition. This means that we take even function reflection for ρ, p, and odd function reflection for u about the boundary $x = 0$. The Riemann problem to such reflection states gives correct boundary state, i.e. $u = 0$. Thus, the corresponding solution in the extended domain satisfies the wall condition.

4. General hyperbolic conservation laws: $u_t + f(u)_x = 0$. Suppose u has n components. How many boundary conditions we should impose depending on how many characteristic directions go inward to the interior. At the left boundary, the boundary conditions look like
   $$b_L(u) = 0,$$
   where b_L is an $(n - r)$-valued function, it satisfy the following consistent conditions:

   - there are exactly $n - r$ positive characteristic speeds for $f'(u)$.
   - The Jacobian $\frac{\partial b_L}{\partial u}$ has full rank.

   Alternatively we can use the Riemann problem to count on how many waves move into the interior to determine the interior states. The boundary condition can be imposed in terms of the wave strengths of inward waves.

7.6.2 Multidimensional boundary conditions for the Euler equation

Let us consider the 2D case. The unknowns are ρ, p and the velocity $\mathbf{v}$. On the boundary, let us denote the normal velocity by v_n and the tangential velocity by v_t.

- Inflow: we impose ρ, p, v_n, and set $v_t = 0$.
- Outflow: if the flow is subsonic but $v_n > 0$ at the outlet, we need impose p, which is set to be the surrounding pressure. If it is supersonic, then we don't impose any boundary condition.
- Wall: The condition is
$$v_n = 0.$$
We extend the domain for few grid points. The extended grids are called ghost grid. The state at the ghost grid is set to be the reflection of the interior state, which is ρ, p, v_t are even reflection, v_n is odd reflection. The Riemann problem to such reflection states gives correct boundary state, i.e. $v_n = 0$.

  In the ENO/WENO methods, we should set the reflection state at the ghost grids before every ENO/WENO reconstruction steps. For instance, in the splitting methods using splitting WENO-RK2, during the x-sweeping procedure, there are two reconstruction steps, one for LU^n, one for $LU^{(1)}$, we should determine these ghost states before the reconstruction steps. There are two such procedures for the x-sweeping, and another two for the y-sweeping.

  However, for the unsplit WENO-RK2, we only need two such reflection procedures.

Project

1. Implement a WENO5 code to solve the Mach reflection problem for gas dynamics. Imagine a shock impinging a wedge in 2D. We set up initial time is the impinging moment. Set the tip of the wedge to be the origin. At this moment, the data is homogeneous in each array from $(0, 0)$. Since the equations and the initial data are invariant under $(\mathbf{x}, t) \mapsto \lambda(\mathbf{x}, t)$ for all $\lambda > 0$, we expect the solution has the form $U(\mathbf{x}/t)$. We also expect the solution should be symmetric about the axis of the wedge. Thus, we choose the one wedge side to be the x-axis, the computational domain is $[-2, 5] \times [0, 3]$. The shock is 60° to the bottom wall. A computational result from Shu's note is the figure. See
   - Sec. 5.1, example 5 of Shu's note (pp. 57).
   - This benchmark problem was proposed in Paul Woodward and Phillip Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, Journal of Computational Physics, v54, 1984, pp.115-173.

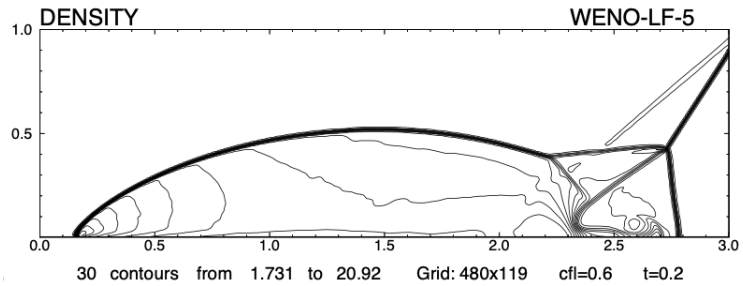


Figure 7.4: Double Mach reflection. The figure is quoted from a Shu's ENO, WENO note

Chapter 8

Finite Element Methods for the Poisson Equation

8.1 Variational formulation of the Poisson problem

8.1.1 Poisson problems in physics

1. **Electric potential** induced by charge density: Let $\rho(x)$ be the charge density. It induces an electric potential in the space. The governing equation is

$$-\triangle \phi = \rho.$$

If there is a closed conductor with boundary $\partial\Omega$ which coberns the support of ρ, then the boundary condition of ϕ is

$$\partial_t \phi = 0.$$

Or equivalently, $\phi = const.$ on $\partial\Omega$.

2. **Streamline of incompressible flows** The incompressible flow satisfies

$$\nabla \cdot \mathbf{v} = 0.$$

The vorticity $\boldsymbol{\omega}$ is defined by

$$\nabla \times \mathbf{v} = \boldsymbol{\omega}.$$

Given vorticity field $\boldsymbol{\omega}$, we can recover a stream function by solving

$$\nabla \times (\nabla \times \boldsymbol{\psi}) = \boldsymbol{\omega}.$$

The streamline is constant on the boundary.

3. **Elasticity** Let u be the displacement of an elastic material. The infinitesimal strain corresponding to u is defined as

$$\mathbf{e} := \nabla u + (\nabla u)^T$$

The stress is a response to the strain. For linear material, it is

$$\boldsymbol{\sigma} = E\mathbf{e}$$

The material is under an exterior force $\mathbf{f}$. The force balance equation is

$$-\nabla \cdot \boldsymbol{\sigma} = \mathbf{f}.$$

There are two kinds of boundary conditions: Dirichlet and Neumann. The boundary $\partial\Omega = \Gamma = \Gamma_D \cup \Gamma_N$.

- Dirichlet: $u = 0$ on Γ_D,
- Neumann: $\boldsymbol{\sigma} \cdot \nu = \mathbf{t}$ on Γ_N. ν is the outer normal of $\partial\Omega$.

4. **Heat balance**. Let u be the temperature of a material in domain Ω. The variation of the temperature will induce a heat conduction $\mathbf{q}$ defined by

$$\mathbf{q} = \kappa \nabla u.$$

The energy balance equation is

$$\nabla \cdot \mathbf{q} = \mathbf{f}.$$

Here, $\mathbf{f}$ is the external heat source.

5. **Hodge decomposition in differential geometry.**

8.1.2 PDE formulation

Consider a domain Ω. The boundary $\partial\Omega$ is decomposed into two disjoint boundaries Γ_D and Γ_N. The Poisson problem we consider is

$$\boxed{-\triangle u = f \text{ in } \Omega,} \tag{8.1}$$

$$\boxed{u = g_D \text{ on } \Gamma_D, \qquad \frac{\partial u}{\partial \nu} = g_N \text{ on } \Gamma_N.} \tag{8.2}$$

8.1.3 Weak form of the Poisson equation

1. Let us introduce some function spaces.

   - $L^2(\Omega) = \{f || \int_\Omega |f(x)|^2\, dx < \infty\}$. In $L^2(\Omega)$, we introduce the inner product structure:

$$(f, g) := \int_\Omega f(x) g(x)\, dx.$$

- Sobolev spaces:

$$H^m(\Omega) := \{f | f, ..., D^\alpha f \in L^2(\Omega), |\alpha| = m\}$$

The index $\alpha = (\alpha_1, ..., \alpha_n)$ is a multi-index. $|\alpha| := \sum_{i=1}^n \alpha_i$. The derivative

$$D^\alpha f(x) := \frac{\partial^{|\alpha|} f(x)}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}}.$$

In $H^m(\Omega)$, we have the inner product structure

$$(f, g)_{H^m} = \sum_{|\alpha|=0}^{m} (D^\alpha f, D^\alpha g).$$

2. We introduce variational formulation for the Poisson problem (8.1), (8.2) with g_D and $g_N \equiv 0$. First, we introduce the space V

$$\boxed{V = \{v \in H^1(\Omega) : \ v|_{\Gamma_D} = 0\}}$$

We take inner product of the Poisson equation with any $v \in V$:

$$(- \triangle u, v) = (f, v).$$

The left-hand side is

$$(- \triangle u, v) = a(u, v) - \int_{\partial\Omega} \frac{\partial u}{\partial \nu} v \, dS,$$

where

$$a(u, v) := \int_\Omega \nabla u(x) \cdot \nabla v(x) \, dx$$

is called the Dirichlet integral. In this step, we have used

$$-(\triangle u)v = -\nabla \cdot (v \nabla u) + (\nabla u)(\nabla v),$$

and the divergence theorem

$$\int_\Omega \nabla \cdot (v \nabla u) \, dx = \int_{\partial\Omega} v \nabla u \cdot \nu \, dS = \int_{\partial\Omega} v \frac{\partial u}{\partial \nu} \, dS.$$

The boundary term

$$\int_{\partial\Omega} \frac{\partial u}{\partial \nu} v \, dS = \int_{\Gamma_D} \frac{\partial u}{\partial \nu} v \, dS + \int_{\Gamma_N} \frac{\partial u}{\partial \nu} v \, dS = 0,$$

because $v = 0$ on Γ_D and $\partial u / \partial \nu = 0$ on Γ_N.

3. Thus, (8.1) (8.2) implies the following *variational formulation*:

$$\boxed{\text{Find } u \in V \text{ such that} \quad a(u,v) = (f,v) \text{ for all } v \in V.} \tag{8.3}$$

This is called the *weak form of the Poisson equation*. Its solution is called a weak solution to (8.1), (8.2).

4. Strong solution and weak solution. Note that a weak solution is only in $H^1(\Omega)$, while a C^2 solution of (8.1), (8.2) is called a classical solution. A weak solution in $H^2(\Omega)$ is also called a strong solution.

5. Suppose u is a strong solution, then u satisfies (8.1), (8.2).
Proof. We choose those $v \in V$ and $v|_{\partial\Omega} = 0$. Then $u \in H^2(\Omega)$, $v \in H_0^1(\Omega)$ and $a(u,v) = (f,v)$ lead to

$$(-\triangle u - f, v) = 0 \text{ for all } v \in H_0^1(\Omega).$$

This implies $-\triangle u = f$. This together with $u|_{\Gamma_D} = 0$, and $a(u,v) = (f,v)$ for all $v \in V$ imply

$$0 = a(u,v) - (f,v) = (-\triangle u - f, v) + \int_{\Gamma_D} \frac{\partial u}{\partial \nu} v \, dS + \int_{\Gamma_N} \frac{\partial u}{\partial \nu} v \, dS = \int_{\Gamma_N} \frac{\partial u}{\partial \nu} v \, dS$$

This implies $\partial u/\partial \nu = 0$ on Γ_N.

6. General boundary conditions. The variational formulation for general Dirichlet boundary data g_D and general Neumann data g_N is: define

$$V_{g_D} := \{u \in H^1(\Omega), \quad u = g_D \text{ on } \Gamma_D\} \tag{8.4}$$

$$(V) \qquad \boxed{\text{Find } u \in V_{g_D} \text{ such that } .a(u,v) = (f,v) + \int_{\Gamma_N} g_N v \, dS \text{ for all } v \in V.} \tag{8.5}$$

One can show that this variational formulation is equivalent to the PDE formulation, provided $u \in H^2(\Omega)$.

7. Existence and uniqueness. The elliptic PDE theory uses coerciveness of a and the Lax-Milgram theorem (or Riesz representation if a is symmetric) to show the existence for weak solution. The uniqueness theory can be obtained by energy estimate, or L^∞ estimate using the maximal principle.

8. The strong solution is obtain by regularity theorem which shows that $u \in H^2$ if $f \in L^2$ and $\partial\Omega \in C^1$.

8.1.4 Variational problem as a minimization problem

1. The above variational problem can be thought as a minimization problem. We consider a functional defined on V by

$$F(v) := \frac{1}{2}a(v,v) - (f,v) - \int_{\Gamma_N} g_N v \, dS := \frac{1}{2}a(v,v) - (f,v) - < g_N, v > \tag{8.6}$$

   We look for

$$(M) \quad \boxed{\min\{F(u) | u \in V_{g_D}\}} \tag{8.7}$$

2. We show (M) $\Rightarrow$ (V). If u is a minimum of F in V_{g_D}, then $u + \varepsilon v \in V_{g_D}$ for $v \in V$ and

$$F(u) \leq F(u + \varepsilon v).$$

   Differentiate in ℓ, we get

$$a(u,v) - (f,v) - < g_N, v >= 0.$$

   This shows that (M) $\Rightarrow$ (V).

3. Next, we show (V) $\Rightarrow$ (M). For any $w \in V_{g_D}$, we have $v := u - w \in V$. Thus,

$$a(u,v) - (f,v) - < g_N, v >= 0.$$

   Now,

$$\begin{aligned} F(w) - F(u) &= \frac{1}{2}\left(a(w,w) - a(u,u)\right) - (f, w-u) - < g_N, w-u > \\ &= \frac{1}{2}\left(a(w,w) - a(u,u)\right) - a(u, w-u) \\ &= \frac{1}{2}\left(a(w,w) + a(u,u) - 2a(u,w)\right) \\ &= \frac{1}{2}a(u-w, u-w) \geq 0. \end{aligned}$$

8.2 1D finite element method

8.2.1 Finite element method

1. Let us consider the Poisson equation in one dimension:

$$-u'' = f \text{ on } (a,b), \; u(a) = u(b) = 0. \tag{8.8}$$

   We shall find an approximate solution by finite element method.

2. First, we discretize the space $[a, b]$ and define the finite element functions. We choose an $n > 0$. Let $h := (b-a)/n$ the mesh size, $x_i = a + ih$, $i = 0, \cdots, n$ the grid point. Each cell $K_i := (x_i, x_{i+1})$ is called an element. We consider linear functions P_1 in K_i. There are two such functions: $\chi_i^0 = (x_{i+1} - x)/h$ and $\chi_i^1 := (x - x_i)/h$, satisfying $\chi_i^0(x_i) = 1$ and $\chi_i^1(x_{i+1}) = 1$.

3. Define the finite element function $\phi_i(x)$ to be $\phi_i(x_j) = \delta_{ij}$ and $\phi(x)$ is continuous and piecewise linear. In cell I_i, $\phi_i = \chi_0^i$, whlle $\phi_i = \chi_1^{i-1}$ in cell I_{i-1}.

4. Let
$$V_h = \text{Span}\{\phi_1, \cdots, \phi_{n-1}\}.$$
It is called the finite element space. An element $v \in V_h$ is a continuous and piecewise linear function and is uniquely expressed by
$$v(x) = \sum_{i=1}^{n-1} v(x_i)\phi_i(x).$$

5. Next, we find an approximate solution $u_h \in V_h$. We express u_h by
$$u_h(x) = \sum_{i=1}^{n-1} U_i\phi_i(x).$$
We project f onto V_h by
$$\pi_h f(x) = \sum_{i=1}^{n-1} f(x_i)\phi_i(x).$$
We project the equation (8.8) onto V_h:
$$(-u'' - f, v) = 0, \text{ for all } v \in V_h$$
This leads to the following equations for $U = (U_1, \cdots, U_{n-1})^T$:
$$\langle u_h, \phi_i\rangle_1 = (f, \phi_i), i = 1, \cdots, n-1.$$
Or
$$\sum_{j=1}^{n-1}(\phi_i', \phi_j')U_j = \sum_{j=1}^{n-1} f(x_j)(\phi_j, \phi_i), i = 1, \cdots, n-1.$$

6. The matrix $A = (\phi_i', \phi_j')_{(n-1)\times(n-1)}$ is called the stiff matrix. The matrix $M := (\phi_i, \phi_j)_{(n-1)\times(n-1)}$ is called the mass matrix. We compute A and M using the representation of ϕ_i in terms of χ_i in each cell. Finally, we obtain
$$A = \frac{1}{h}\text{diag}(-1, 2, -1), \quad M = \frac{h}{6}\,\text{diag}(1, 4, 1).$$
The FE equation reads
$$AU = MF.$$
This can be inverted.

Let us consider another boundary condition:

$$u(a) = 0, \quad u'(b) = u_b.$$

The variational formulation is

$$a(u, v) - (f, v) - u_b v(b) = 0.$$

The trial and test space

$$V_h = \text{Span}\{\phi_1, ..., \phi_n\}.$$

The approximate solution is

$$u_h = \sum_{j=1}^{n} U_j \phi_j.$$

The last row of the stiff matrix is

$$(\phi'_{n-1}, \phi'_n) = -1/h, \quad (\phi'_n, \phi'_n) = 1/h.$$

The last row of the mass matrix is

$$(\phi_{n-1}, \phi_n) = \frac{h}{6}, \quad (\phi_n, \phi_n) = \frac{h}{3}.$$

The last row on the right-hand side is

$$f(x_n)(\phi_n, \phi_n) + u_b.$$

8.2.2 Error analysis

1. **True error is controlled by an approximation error** Let u be the exact solution and $e_h := u - u_h$ be the true error. Since both u and u_h satisfy

   $$(u', v') = (f, v), \ (u'_h, v') = (f, v) \text{ for all } v \in V_h,$$

   we obtain

   $$(e'_h, v') = 0 \text{ for all } v \in V_h.$$

   That is, $(u - u_h) \perp_1 V_h$. * This is equivalent to say that u_h is the $\langle \cdot, \cdot \rangle_1$-orthogonal projection of u on V_h. Thus,

   $$\|u' - u'_h\|_2 \leq \|u' - v'\|_2 \text{ for all } v \in V_h.$$

   In particular, we can choose $v \in V_h$ that equals u at $x_1, \cdots, x_{n-1}$. That is,

   $$v = \pi_h u := \sum_{i=1}^{n-1} u(x_i)\phi_i,$$

*We define $\langle u, v \rangle_1 := (u', v')$. This is an inner product in the space $H_0^1(a, b)$. The zero boundary condition gives $\langle u, u \rangle_1 = 0 \Rightarrow u = 0$.

then

$$\|u' - u_h'\|_2 \leq \|u' - (\pi_h u)'\|_2. \tag{8.9}$$

Thus, the true error is controlled by the approximation error.

2. **Approximation error in terms of $\|u''\|_\infty$**
It is easy to see that π_h is a projection. If $u \in C^2$, then in each cell (x_i, x_{i+1}), the projection error $w(x) = u(x) - \pi_h u(x)$ satisfies $w(x_i) = w(x_{i+1}) = 0$. By applying Rolle's theorem twice, we get that for any $x \in (x_i, x_{i+1})$, there exists an $\xi_i \in (x_i, x_{i+1})$ such that

$$w(x) = \frac{w''(\xi_i)}{2}(x - x_i)(x - x_{i+1}).$$

This leads to

$$|w(x)| \leq \frac{h^2}{8} \max_{\xi \in (x_i, x_{i+1})} |w''(\xi)|.$$

Hence

$$\begin{aligned}
\int_a^b |w(x)|^2 \, dx &= \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} |w(x)|^2 \, dx \\
&\leq \sum_{i=0}^{n-1} h \left(\frac{h^2}{8}\right)^2 \max_{x \in (x_i, x_{i+1})} |w''(x)|^2 \\
&\leq (b-a) \left(\frac{h^2}{8}\right)^2 \max_{x \in [a,b]} |u''(x)|^2.
\end{aligned}$$

Here, we have used that $w''(x) = u''(x)$ on each subinterval (x_i, x_{i+1}). Hence,

$$\|u - \pi_h u\|_2 \leq \sqrt{b-a} \frac{h^2}{8} \|u''\|_\infty$$

We can also estimate $u' - (\pi_h u)'$ by mean value theorem. First, there exists a $\zeta_1 \in (x_i, x_{i+1})$ such that $u'(\zeta_1) = (u(x_{i+1} - u(x_i))/h$. For any $x \in (x_i, x_{i+1})$, there exists $\zeta_2 \in (x_i, x_{i+1})$ such that $u'(x) - u'(\zeta_1) = u''(\zeta_2)(x - \zeta_1)$. Therefore, we get

$$u'(x) - (\pi_h u)'(x) = u'(x) - \frac{u(x_{i+1}) - u(x_i)}{h} = u''(\zeta_2)(x - \zeta_1).$$

Notice that $(\pi_h u)'(x) = \frac{u(x_{i+1}) - u(x_i)}{h}$ for $x \in (x_i, x_{i+1})$. Hence, we obtain

$$\begin{aligned}
\int_a^b |u' - (\pi_h u)'|^2 \, dx &= \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} |u' - (\pi_h u)'|^2 \, dx \\
&\leq \sum_{i=0}^{n-1} h \, h^2 \max_{x \in [a,b]} |u''(x)|^2 \\
&= (b-a) h^2 \|u''\|_\infty^2
\end{aligned}$$

3. **Approximation error in terms of $\|u''\|_2$** The estimate above is in terms of $\|u''\|_\infty$. It is desirable to estimate in terms of $\|u''\|_2$. That is, we want to estimate $\|u - \pi_h u\|_2$ in terms of $\|u''\|_2$. To do so, we should use the integral representation of error. We recall that for $w(x_i) = w(x_{i+1}) = 0$, w has the representation:

$$w(x) = h^2 \int_{x_i}^{x_{i+1}} g\left(\frac{x - x_i}{h}, \frac{y - x_i}{h}\right) w''(y)\, dy$$

$$w'(x) = h \int_{x_i}^{x_{i+1}} g_x\left(\frac{x - x_i}{h}, \frac{y - x_i}{h}\right) w''(y)\, dy$$

where g is the Green's function of d^2/dx^2 on $(0, 1)$ with zero boundary condition. That is,

$$g(x, y) = \begin{cases} -x/2 & \text{if } x < y \\ -y + x/2 & \text{if } x > y \end{cases}$$

Thus, we can estimate $\|w\|_2$ in terms of $\|w''\|_2$ on (x_i, x_{i+1}). Namely,

$$|w(x)|^2 \le h^4 \left(\int_{x_i}^{x_{i+1}} |g\left(\frac{x - x_i}{h}, \frac{y - x_i}{h}\right)|^2\, dy\right) \left(\int_{x_i}^{x_{i+1}} |w''(y)|^2\, dy\right).$$

$$\begin{aligned} \int_{x_i}^{x_{i+1}} |w(x)|^2\, dx &\le \int_{x_i}^{x_{i+1}} \int_{x_i}^{x_{i+1}} |g\left(\frac{x - x_i}{h}, \frac{y - x_i}{h}\right)|^2\, dy\, dx \int_{x_i}^{x_{i+1}} |w''(y)|^2\, dy \\ &\le \frac{1}{90} h^4 \int_{x_i}^{x_{i+1}} |w''(y)|^2\, dy. \end{aligned}$$

As we sum over $i = 1, \cdots, n - 1$, we get

$$\|w\|_2 \le \frac{1}{\sqrt{90}} h^2 \|w''\|_2.$$

Similarly, we get

$$\|w'\|_2 \le \frac{1}{\sqrt{6}} h \|w''\|_2.$$

Theorem 8.1. *For $u \in H^2(a, b) \cap H_0^1[a, b]$, the interpolation error has the following estimates*

$$\|u - \pi_h u\|_2 \le \frac{1}{\sqrt{90}} h^2 \|u''\|_2,$$

$$\|u' - (\pi_h u)'\|_2 \le \frac{1}{\sqrt{6}} h \|u''\|_2.$$

4. **True error of the finite element method**

Theorem 8.2. *For the finite element method for problem (8.8), the true error $u - u_h$ has the following estimate*

$$\|u' - u_h'\|_2 \leq \|u' - (\pi_h u)'\|_2 \leq \frac{1}{\sqrt{6}} h \|u''\|_2,$$

$$\|u - u_h\|_2 \leq \frac{1}{6} h^2 \|u''\|_2.$$

Proof. The first estimate follows from the previous theorem. For the second, the trick is called a duality argument. Let $e_h = u - u_h$. We find the function ϕ_h such that $\phi_h'' = -e_h$ and $\phi(a) = \phi(b) = 0$. Then

$$(e_h, e_h) = -(e_h, \phi_h'') = (e_h', \phi_h') = (e_h', \phi_h' - (\pi_h \phi_h)').$$

Here, I have used

$$(e_h', v') = 0 \text{ for all } v \in Ran(\pi_h).$$

Applying interpolation estimate to ϕ_h, we get

$$\|e_h\|^2 \leq \|e_h'\| \|(\phi_h - \pi_h \phi_h)'\| \leq \frac{1}{\sqrt{6}} \|e_h'\| \, h \, \|\phi_h''\| = \frac{1}{\sqrt{6}} h \|e_h'\| \, \|e_h\|$$

Hence, we get

$$\|e_h\|_2 \leq \frac{1}{\sqrt{6}} h \|e_h'\|_2 \leq \frac{1}{6} h^2 \|u''\|_2.$$

□

Homeworks

1. The error function w on each interval (x_i, x_{i+1}) satisfies $w(x_i) = w(x_{i+1}) = 0$. w can be estimated in terms of w'' in (x_i, x_{i+1}). This is indeed a generalized Poincaré inequality. You can get best estimate via Fourier sin expansion. Find the best constant and the get the best error estimate.

   Given $x_0 < x_1 < x_2$. Let w be a smooth function satisfying $w(x_i) = 0$ for $i = 0, 1, 2$. Find an integral representation of w in terms of w''' on (x_0, x_2).

8.3 Finite element methods for the Poisson problem in 2 dimensions

8.3.1 Framework of finite element method

1. **Triangulation**. This give $\mathcal{T} = \cup_k K_k$ a partition of the domain Ω. The partition consists of nodal points $\mathcal{N} = \{N_1, ..., N_p\}$ and triangles $\mathcal{K} := \{K_1, ..., K_l\}$. The nodes include interior nodes and Neumann boundary nodes, denoted by $\mathcal{N}^i$ and $\mathcal{N}^N$, respectively.

2. **Delaunay triangulation algorithms**: Given a set of points P, the Delaunay triangulation is a triangulation $\mathcal{T} = \cup K_k$ such that no point in P is inside the circumcircle of any triangle in $\mathcal{T}$.

3. **Finite Element** We associate a triangle K with nodes $\mathcal{N} := \{N_1, N_2, N_3\}$ and piecewise linear functions $\mathcal{P}_1 := \{\chi_1, \chi_2, \chi_3\}$ such that $\chi_i(N_j) = \delta_{ij}$. The triple $(K, \mathcal{N}, \mathcal{P}\}$ is called a finite element.

4. **Nodal function** ϕ_i is a continuous function and $\phi_i|_{K_j}$ is a linear function. Moreover, $\phi_i(N_j) = \delta_{ij}$.

   - Interior nodal functions
   - Nodal function on Neumann boundary
   - Nodal function near Dirichlet boundary

5. **The Finite element space**

$$V_h := \text{Span}\{\phi_1, ..., \phi_p\}.$$

6. Representation of the Poisson equation on V_h: The unknown u is approximated by

$$u_h := \sum_{i=1}^{p} U_i \phi_i \in V_h$$

   The test function v is also in V_h. We will uses basis ϕ_i, $i = 1, ..., p$ as test functions. The equation in V_h reads

$$(\sum_{j=1}^{p} U_j \nabla\phi_j, \nabla\phi_i) = (\sum_{j=1}^{p} F_j \phi_j, \phi_i) + (\sum_{k=1}^{N_n} g_{N,k}\phi_k, \phi_i)$$

   In matrix form:

$$SU = MF + TG$$

7. The mass matrix M.

$$\begin{aligned}(\phi_i, \phi_j) &= \int_\Omega \phi_i \phi_j \, dx \\ &= \sum_{k=1}^{l} \int_{K_k} \phi_i \phi_j \, dx \\ &= \sum_{k=1}^{l} \int_{K_k} (\sum_{m=1}^{3} c_{i,m}\chi_{k,m}(x) \cdot \sum_{l=1}^{3} c_{jl}\chi_{k,l}(x)) \, dx\end{aligned}$$

8. The stiff matrix.