COMPUTATIONAL MATHEMATICS

AN INTRODUCTION

I-Liang Chern

Department of Mathematics
National Taiwan University
2015

December 28, 2015

Contents

Chapter 1

Solving Equations of One Variable

1.1 Motivation

This part is taken from QSG's book.

Motivation 1: Investment fund At the beginning of every year a bank customer deposits v euros in an investment fund and withdraws, at the end of the nth year, a capital of M euros. We want to compute the average yearly rate of interest r of this investment. Since M is related to r by the relation

$$M = v\sum_{k=1}^{n}(1+r)^k = v\frac{1+r}{r}[(1+r)^n - 1],$$

we deduce that r is the root of

$$f(r) = 0, \quad f(r) = v\frac{1+r}{r}[(1+r)^n - 1] - M.$$

Motivation 2: State equation of a gas The van der Waals equation of state (i.e. the equation that relates pressure p, volume V and temperature T) is

$$\left[p + a\frac{N}{V}\right](V - Nb) = kNT,$$

where N is the number of molecules and k is the Boltzmann constant. The parameters a measures the attraction between molecules and b measures the volume occupied by the molecules (Van der Waals equation, Wiki).

We want to determine the volume V occupied by a gas at temperature T and pressure p.

Motivation 3: Rods system Let us consider the mechanical system represented by the four rigid rods ai of Figure 2.1. For any admissible value of the angle β, let us determine the value of the corresponding angle α between the rods $\mathbf{a}_1$ and $\mathbf{a}_2$. Starting from the vector identity

$$\mathbf{a}_1 - \mathbf{a}_2 - \mathbf{a}_3 - \mathbf{a}_4 = 0,$$

and noting that the rod $\mathbf{a}_1$ is always aligned with the x-axis, we can deduce the following relationship between β and α:

$$a_1^2 + a_2^2 - a_3^2 + a_4^2 + 2a_1a_4\cos\beta - 2a_1a_2\cos\alpha - 2a_2a_4\cos(\beta - \alpha) = 0.$$

We would also like to mention that a solution does not exist for all values of β, and may not even be unique. To solve the equation for any given β lying between 0 and π we should invoke numerical methods.

Motivation 4: Population dynamics The population of species or bacteria is modeled by

$$x^+ = xR(x)$$

where x and x^+ are respectively the populations of the present and next generations. $R(x)$ is the growth rate.

- Beverton-Holt or discrete Verhulst model $R(x) = \frac{r}{1+Kx}$
- Redator/prey model $R(x) = \frac{rx}{1+(x/K)^2}$.

We look for saturated population where $x^+ = x$.

1.2 Newton's method

Goal: solve $f(x) = 0$

Strategy

- it is an iterative method
- it approximate the equation by a linear equation at each step, use the solution of the linear equation to approximate the root.

$$f(x) \sim f(x_n) + f'(x_n)(x - x_n) = 0.$$

Algorithm

$$\boxed{x_{n+1} = x_n - f'(x_n)^{-1}f(x_n).}$$

Error analysis

Theorem 1.1. *Let x^* be the root. Suppose it is simple. Let $e_n = x_n - x_*$. Then $|e_{n+1}| = O(|e_n|^2)$.*

Proof. Let $g(x) = x - f'(x)^{-1} f(x)$. We have $g(x^*) = x^*$. Subtract $x_{n+1} = g(x_n)$ and $x^* = g(x^*)$, we get

$$e_{n+1} = g(x_n) - g(x^*) = g(x^* + e_n) - g(x^*) = \frac{1}{2} g''(\xi) e_n^2.$$

Here, we have used $g'(x^*) = 0$ and the lemma below. In computing $g'(x^*)$, we use

$$g'(x^*) = 1 - \frac{f'(x^*)^2 - f(x^*) f''(x^*)}{f'(x^*)^2} = 0$$

where $f'(x^*) \neq 0$ is used. □

Lemma 1.1. *If $g \in C^2$ and $g(0) = g'(0) = 0$, then there exists ξ between 0 and x such that*

$$g(x) = \frac{1}{2} g''(\xi) x^2.$$

Proof. We use integral form of mean value theorem.

$$g(x) = \int_0^x (x - t) g''(t)\, dt.$$

Then use mean value theorem:

$$\int_0^x (x - t) g''(t)\, dt = g''(\xi) \int_0^x (x - t)\, dt.$$

□

Remark. The proof above requires too many derivatives of f (it uses g'', which is equivalent to f'''). But we only need f'' for quadratic convergence. Here is a shorter proof.

$$e_{n+1} = e_n - \frac{f(x_n)}{f'(x_n)} = \frac{e_n f'(x_n) - f(x_n)}{f'(x_n)}$$

From

$$0 = f(x^*) = f(x_n - e_n) = f(x_n) - e_n f'(x_n) + \frac{1}{2} f''(\xi) e_n^2,$$

we get

$$e_{n+1} = \frac{1}{2} \frac{f''(\xi)}{f'(x_n)} e_n^2,$$

or

$$\boxed{e_{n+1} \approx \frac{1}{2} \frac{f''(x^*)}{f'(x^*)} e_n^2.}$$

Important example: Solve $x^2 = a$, $a > 0$. The corresponding Newton's method is

$$x_{n+1} = x_n - \frac{x_n^2 - a}{2x_n} := g(x_n)$$

This was known as the Babylonian method. Heron of Alexandria (AD 10 - AD 70) was a Greek mathematician who described an iterative method of computing the square root. Heron's method can also be derived as a special case of the (much) later Newton's method (16th century). In fact, an implementation of this algorithm is found on a Babylonian clay tablet (YBC7289, 1800-1600 BC), hence the Heron's method is also known as the Babylonian method. After thousands of years, today it has been one of the most commonly taught examples in numerical computation and analysis, the basis of many numerical algorithms of nonlinear equations and optimization problems, and in fact the most common algorithm for computing square roots. For this method, you can show that

- $g : [\sqrt{a}, \infty) \to [\sqrt{a}, \infty)$ is a strictly decreasing function.
- $g : (0, \sqrt{a}] \to [\sqrt{a}, \infty)$.

These two properties assure the convergence of $\{x_n\}$ if $x_0 > 0$.

A third order method If we use a parabola to approximate f, we can get a third order method.

$$f(x) \sim f(x_n) + f'(x_n)(x - x_n) + \frac{1}{2}f''(x_n)(x - x_n)^2 = 0.$$

This leads to a root

$$x_{n+1} = x_n + \frac{-f'(x_n) + \sqrt{f'(x_n)^2 - 2f(x_n)f''(x_n)}}{f''(x_n)}$$

You can prove it is third order convergence.

Multiple roots You can check that the method falls to a first order method if the root x^* is a double root. If the multiplicity is p, then the Newton method convergence rate becomes $(1 - 1/p)^n$. This can serve to detect the multiplicity of x^*. There are two ways to gain high order convergence. The first one is to modify the scheme to

$$x_{n+1} = x_n - p\frac{f(x_n)}{f'(x_n)} := g(x_n),$$

provided we have known that the multiplicity of x^* is p. You can check that $g'(x^*) = 0$.

The second method is to consider the new function

$$\mu(x) := \frac{f(x)}{f'(x)},$$

which has a simple root at x^*. You can apply Newton's method to this one.

$$x_n = g(x_{n-1}),$$

where

$$g(x) = x - \frac{\mu(x)}{\mu'(x)} = x - \frac{f(x)f'(x)}{f'(x)^2 - f(x)f''(x)}$$

1.3 Secant method

Goal: solve $f(x) = 0$

Strategy

- it is an iterative method
- it approximate the equation by a linear equation at each step, use the solution of the linear equation to approximate the root.

$$f(x) \sim f(x_n) + \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}(x - x_n) = 0.$$

Algorithm

$$\boxed{x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n)}$$

Start from x_0, x_1.

Avoiding loss of significance The divided difference

$$a_n := \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

loses significant digits. To fix it, we replace it by

$$a_n \leftarrow \frac{f(x_n + h) - f(x_n)}{h} \text{ when } |x_n - x_{n-1}| < h.$$

Here, h is chosen to be a fixed number. For example, we can choose

$$h = \sqrt{\delta} x_n$$

Then

$$f(x_n + x_n\sqrt{\delta}) - f(x_n) \sim f'(x_n) x_n \sqrt{\delta} + O(1)\delta.$$

The function difference loses only one digit provided $O(1) = 1$.

Efficiency In Newton's method, we need to evaluate both $f(x_n)$ and $f'(x_n)$. In secant method, we only evaluate $f(x_n)$. So, if it is time consuming to evaluate $f'(x_n)$, the secant method is more efficient.

Error Analysis and Convergence Rate We use Newton's divided difference to approximate a function by polynomials. Define the Newton divided difference by

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0},$$
$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}.$$

From this definition, we have the interpolation formulae

$$f(x) = f(x_0) + f[x_0, x](x - x_0)$$
$$f[x_0, x] = f[x_0, x_1] + f[x_0, x_1, x](x - x_1)$$

The latter leads to

$$f(x) - f(x_0) = f[x_0, x_1](x - x_0) + f[x_0, x_1, x](x - x_0)(x - x_1).$$

In general,

$$f(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \cdots$$
$$+ f[x_0, x_1, ..., x_n, x](x - x_0) \cdots (x - x_n).$$

Now, consider secant method. At x_n, it approximates $f(x)$ by the linear function

$$\ell(x) = f(x_n) + f[x_n, x_{n-1}](x - x_n)$$

Therefore,

$$0 = \ell(x_{n+1}) = f(x_n) + f[x_n, x_{n-1}](x_{n+1} - x_n) = 0. \tag{1.1}$$

By definition,

$$f(x) = f(x_n) + f[x_n, x_{n-1}](x - x_n) + f[x_n, x_{n-1}, x](x - x_n)(x - x_{n-1}).$$

At $x = x^*$,

$$0 = f(x^*) = f(x_n) + f[x_n, x_{n-1}](x^* - x_n) + f[x_n, x_{n-1}, x^*](x^* - x_n)(x^* - x_{n-1}). \tag{1.2}$$

Use $e_n = x_n - x^*$, subtract the above two equations (1.1) (1.2), we get

$$f[x_n, x_{n-1}]e_{n+1} - f[x_n, x_{n-1}, x^*]e_n e_{n-1}$$

Hence

$$\boxed{e_{n+1} = \frac{f[x_n, x_{n-1}, x^*]}{f[x_n, x_{n-1}]} e_n e_{n-1}}$$

By mean value theorem, we have

$$f[x_n, x_{n-1}] = f'(\xi), \text{ for some } \xi \in (x_n, x_{n-1}),$$

$$f[x_n, x_{n-1}, x^*] = \frac{1}{2} f''(\zeta) \text{ for some } \zeta \text{ in the interval containing } x_n, x_{n-1}, x^*.$$

Let

$$M = \frac{1}{2} \frac{\max_I f''(x)}{\min_I f'(x)}$$

Then

$$|e_{n+1}| \leq M |e_n e_{n-1}|.$$

Now, define $\rho_n = M|e_n|$, then we have

$$\rho_{n+1} \leq \rho_n \rho_{n-1}.$$

We choose x_0 and x_1 so that $|\rho_0| < 1$ and $|\rho_1| < 1$. Now, we choose

$$\rho = \min\{\rho_0, \rho_1\} < 1.$$

It is easy to see by induction that $\rho_n < 1$ for all n. Furthermore

$$\begin{aligned}
\rho_2 &\leq \rho_1 \rho_0 \leq \rho^2, \\
\rho_3 &\leq \rho_2 \rho_1 \leq \rho^2 \rho = \rho^3 \\
\rho_4 &\leq \rho_3 \rho_2 \leq \rho^3 \rho^2 = \rho^5 \\
&\vdots \\
\rho_{n+1} &\leq \rho^{q_{n+1}} = \rho^{q_n + q_{n-1}}
\end{aligned}$$

where $q_{n+1} = q_n + q_{n-1}$, $q_0 = q_1 = 1$. The solution

$$q_n = \frac{1}{\sqrt{5}} \left(\lambda_1^{n+1} - \lambda_2^{n+1} \right),$$

$$\lambda_1 = \frac{1 + \sqrt{5}}{2}, \quad \lambda_2 = \frac{1 - \sqrt{5}}{2}$$

are the two roots of $\lambda^2 - \lambda - 1 = 0$. The solution $q_n \sim \frac{1}{\sqrt{5}} \lambda_1^{n+1}$. Thus,

$$\rho_n \leq \left(\rho^{1/\sqrt{5}} \right)^{\lambda_1^{n+1}},$$

or

$$|e_n| \leq M^{-1} \left(\rho^{1/\sqrt{5}} \right)^{\lambda_1^{n+1}}.$$

1.4 A dynamical system point of view of iterative map

The iterative $x_{n+1} = g(x_n)$ can be viewed as a discrete dynamical system, which serves as an important model for a class of physical world. For instance, the discrete logistic map

$$x_{n+1} = a x_n (1 - x_n).$$

models some population dynamics of animals.

Discrete logistic map For iterative map $g(x)$ starting from x_0, those x_0 which leads x_n to converge to a fixed point x^* is called the basin of convergence of x^*. So, each fixed point has its own basin of attraction. They are disjoint. However, not all points belong to the basins of the fixed points. For instance, for certain range of a, the intersection of the discrete logic map $g(x) = ax(1-x)$ has period 2 stable solution. Then period 4 stable solutions, etc. More precisely, when $0 < a < 1$, then 0 is the only fixed point, and it is stable. For $1 < a < 3$, the fixed points are 0 and $(a-1)/a$. The latter is a stable one and the whole region $(0, 1)$ is its basin of attraction. When $3 < a < 1 + \sqrt{6}$, there are fixed points of $g \circ g$. These two solutions are period 2 solutions. They are stable. This means that all points in $(0, 1)$ are the basin of these period 2 solution. The bifurcation phenomenon from fixed point to a period 2 solution as a across 3 is called period doubling. As a keeps increasing, it exhibits period doubling from period 2 to period 4, to period 8 and so forth. As $a \sim 3.56995$, It exhibits so called chaos, where solutions of all periods appear. (see logistic map, wiki)

Newton's method on complex plane Given a polynomial $p(z)$ on the complex plane, the Newton method gives the iteration

$$z_{n+1} = z_n - \frac{p(z_n)}{p'(z_n)} = g(z_n).$$

For every root ξ of $p(z)$, those z_0 which generates a convergent sequence to ξ by Newton's method is called the basin of convergence of ξ. The basins of attraction of roots of $p(z) = 0$ are disjoint. However, not all points on $\mathbb{C}$ belongs to one of the basin of the roots. There are some points which do not belong to any of these basin sets. The set of these exceptional points are called *Julia set of* p. For example, you can study the Julia set of the polynomial $p(z) = z^3 - 1$. You can partition the domain $[-2, 2] \times [-2, 2]$ into 1000×1000 small cells uniformly. For each cell, run Newton's iteration for 20 step starting from the cell center to classify the class of the cell center. Do the experiment and analyze what you obtain.

1.5 Fixed Point Method

Goal: Solve $f(x) = 0$.

Strategy : We change this to a fixed point problem:

$$x = x - \lambda f(x) := g(x)$$

$\lambda \neq 0$. λ is chosen so that

$$|\lambda f'(x)| < 1.$$

Algorithm

$$x_{n+1} = x_n - \lambda f(x_n).$$

Remarks

1. λ can vary in each step, i.e.
$$\boxed{x_{n+1} = x_n - \lambda_n f(x_n)}$$
In numerical ODE, this λ_n can be viewed as Δt. In this sense, such fixed point method can be thought as a forward Euler method for the ODE: $\dot{x} = -f(x)$. Practically, λ_n is chosen so that $|\lambda_n f'(x_n)| < 1$.

2. We can also choose λ to be a function of x. That is,
$$\boxed{x_{n+1} = x_n - \lambda(x_n) f(x_n).}$$
In particular, $\lambda(x) = 1/f'(x)$ gives the Newton method.

Error Analysis

Definition 1.1. *A function g is called a contraction map if there exists a constant $0 \leq \rho < 1$ such that*
$$|g(x) - g(y)| \leq \rho |x - y|$$
for any x, y under consideration.

A contraction map is certainly Lipschitz continuous.

Theorem 1.2. *If g is a contraction map, then g has a unique fixed point x^*. Moreover, the iterative map*
$$x_{n+1} = g(x_n)$$
converges to x^* linearly in the sense
$$|x_n - x^*| \leq \rho^n |x_0 - x^*|$$

Proof. 1. $\{x_n\}$ is Cauchy. We can write
$$x_n = x_0 + (x_1 - x_0) + (x_2 - x_1) + \cdots + (x_n - x_{n-1}) = x_0 + \sum_{i=1}^{n} (x_i - x_{i-1})$$
The series $\sum_{i=1}^{n} (x_i - x_{i-1})$ converges absolutely:
$$\sum_{i=1}^{n} |x_i - x_{i-1}| \leq \sum_{i=1}^{n} \rho^{i-1} |x_1 - x_0| = \frac{\rho^n - 1}{\rho - 1} |x_1 - x_0| < \infty.$$
Hence x_n is Cauchy and thus converges.

2. Subtracting $x_{n+1} = g(x_n)$ and $x^* = g(x^*)$. This leads to
$$\begin{aligned} |x_{n+1} - x^*| = |g(x_n) - g(x^*)| &\leq \rho |x_n - x^*| \leq \rho^2 |x_{n-1} - x^*| \\ &\leq \cdots \leq \rho^{n+1} |x_0 - x^*| \end{aligned}$$

3. If both x^* and y^* are fixed points, then

$$|x^* - y^*| = |g(x^*) - g(y^*)| \le \rho|x^* - y^*|,$$

we get $(1-\rho)|x^* - y^*| \le 0$. Since $\rho < 1$, we obtain $x^* = y^*$.

□

Example We consider linear equation $ax - b = 0$. The corresponding fixed point method is

$$x_{n+1} = x_n - \lambda(ax_n - b) = (1 - \lambda a)x_n + \lambda b$$

We see that

$$x_{n+1} - x_n = (1 - \lambda a)(x_n - x_{n-1}).$$

Thus, the map $x_n \to x_{n+1}$ is a contraction if and only if

$$|1 - \lambda a| < 1.$$

Applications

Theorem 1.3 (Implicit Function Theorem)**.** *If $F : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$. $F(x_0, y_0) = 0$ and F is continuously differentiable in a neighborhood of (x_0, y_0). Further, suppose the Jacobian $F_y(x_0, y_0)$ is invertible. Then there exist neighborhoods of x_0 and y_0, called U and V respectively, and a continuously differentiable function $f : U \to V$ such that*

$$F(x, f(x)) = 0 \quad \textit{for all } x \in U.$$

We demonstrate the proof for $n = m = 1$. It is easy extended to general case. To solve an equation

$$F(x, y) = 0$$

for y with given x, we linearize it about (x_0, y_0):

$$F(x, y) = F(x_0, y_0) + a\xi + b\eta + h(\xi, \eta).$$

Here, $\xi := x - x_0, \eta := y - y_0$, $a = F_x(x_0, y_0)$, $b = F_y(x_0, y_0)$,

$$h(\xi, \eta) := F(x_0 + \xi, y_0 + \eta) - F(x_0, y_0) - a\xi - b\eta = o(\xi, \eta).$$

Since $F(x_0, y_0) = 0$, we solve the perturbation equation:

$$a\xi + b\eta + h(\xi, \eta) = 0.$$

for η with small ξ. This can be rewritten as

$$\eta = -\frac{a}{b}\xi - \frac{1}{b}h(\xi, \eta_n).$$

We use fixed point method to solve this equation:

$$\eta_{n+1} = -\frac{a}{b}\xi - \frac{1}{b}h(\xi, \eta_n) := g(\xi, \eta_n).$$

We find that as long as ξ is small, then this iterative map $g(\xi, \cdot)$ is a contraction map. Thus, it has a fixed point η. I shall not go into detail of the proof.

Acceleration technique In the fixed point method, near the fixed point, we want to find find a better approximation $\hat{x}_n$ to the limit x^*. Since the sequence converges linearly, we expect the three points $(x_{n-2}, x_{n-1}), (x_{n-1}, x_n)$ and $(\hat{x}_n, \hat{x}_n)$ are co-linear. This leads to

$$\frac{x_{n-1} - \hat{x}_n}{x_{n-2} - \hat{x}_n} = \frac{x_n - \hat{x}_n}{x_{n-1} - \hat{x}_n}$$

We then get a good candidate

$$\hat{x}_n = x_{n-2} - \frac{(x_{n-1} - x_{n-2})^2}{x_n - 2x_{n-1} + x_{n-2}}$$

This is called *Aitken's extrapolation formula*, or the Steffensen algorithm.

$$x_{n+1} = G(x_n)$$

where

$$G(x) = x - \frac{(g(x) - x)^2}{g(g(x)) - 2g(x) + x}.$$

Theorem 1.4. *The Steffensen algorithm $\{x^k\}$ converges quadratically.*

Proof. Without loss of generality, we may assume $x^* = 0$. WE can write $g(x) = \ell x + O(x^2)$. We claim $G(x) = O(x^2)$. By direct computation, we have

$$\begin{aligned} G(x) &= x - \frac{(g(x) - x)^2}{g(g(x)) - 2g(x) + x} \\ &= x - \frac{((\ell - 1)x + O(x^2))^2}{\ell(\ell x + O(x^2)) + O((\ell x + O(x^2))^2 - 2(\ell x + O(x^2)) + x} \\ &= x - \frac{(\ell - 1)^2 x^2 + O(x^3)}{(\ell - 1)^2 x + O(x^2)} \\ &= \frac{O(x^3)}{(\ell - 1)^2 x + O(x^2)} = O(x^2). \end{aligned}$$

This shows $G(x) = O(x^2)$ provided $g'(x^*) \neq 1/2$. □

For program in matlab, see QSG, pp. 65, program 2.4.

Homeworks 1.1. *1. QSG: pp. 74, Ex 2.9*

2. Write a program to solve the Wilkinson problem:

$$f(x) := \prod_{i=1}^{20} (x - i) + \epsilon x^{20}$$

Input: ϵ, output, the largest 6 roots in magnitude.

Chapter 2

Basic Numerical Linear Algebra

2.1 Motivations

Hydraulic Network The hydraulic network can be modeled as a graph (V, E), where $V = \{1, 2, ..., n\}$ is the nodes, $E = \{(i, j)\} \subset V \times V$ are the edges. Each edge $e = (i, j)$ is a pipe connecting nodes i and j, on which, we associate a flow velocity u_e, area of cross section A_e and length L_e. We assume they are uniform along pipe e. The flow direction gives a natural direction (orientation) of the pipe. So, the graph is a directed graph: each $e = (i, j) \in E$ has a direction from i to j. For $(i, j) \in E$, we define $\text{sign}(i, j) = 1$ and $\text{sign}(j, i) = -1$. On each edge e, we compute the flow rate

$$Q_e := \rho u_e A_e,$$

which is the amount of water passing through the pipe per unit time. At each node i, we associate with a pressure p_i. On each pipe (edge), there is a momentum equation which balances the flux in the pipe and the pressure difference at the two ends of the pipe. Physically, this is Darcy's law. It can be expressed as

$$p_x = -\alpha_e \rho u_e$$

where α_e is the friction coefficient, and positive x is the same direction of e. We integrate the Darcy law over the pipe e and get

$$p_j - p_i = \frac{\alpha_e L_e}{A_e} Q_e \text{sign}(j, i).$$

Here, we have assume $\rho \equiv 1$. This is the momentum balance equation on each pipe e. At each node i, we conservation of water. To describe this equation, let $E_i = \{e \in E, i \text{ is one end of } e\}$, $\mathcal{N}_i = \{j | (i, j) \in E\}$ be the neighboring nodes of i. At each interior node i,

$$\sum_{e \in E_i} \text{sign}(j, i) Q_e = \sum_{j \in \mathcal{N}_i} \left(\frac{A_e}{\alpha_e L_e} \right) (p_j - p_i) = 0.$$

At each boundary node i, a pressure p_i^b is prescribed. There are two kinds of boundary nodes, one is node of the end user. The pressure $p_i^b = 0$ there. The other is the water pump node, where

$p_i^b > 0$ is also prescribed. This equation is a *discrete Laplacian* for $(p_i)_{i\in N}$ with Dirichlet boundary condition.

A mass-spring system Consider a spring-mass system which consists of n masses placed vertically between two walls. The n masses and the two end walls are connected by $n+1$ springs. If all masses are zeros, the springs are "at rest" states. When the masses are greater than zeros, the springs are elongated due to the gravitation force. The mass m_i moves down u_i distance, called the displacement. The goal is to find the displacements u_i of the masses m_i, $i = 1, ..., n$.

In this model, the nodes are the masses m_i. We may treat the end walls are the fixed masses, and call them m_0 and m_{n+1}, respectively. The edges (or the bonds) are the springs. Let us call the spring connecting m_i and m_{i+1} by edge (or spring) i, $i = 1, ..., n+1$. Suppose the spring i has spring constant c_i. Let us call the downward direction the positive direction.

Let me start from the simplest case: $n = 1$ and no bottom wall. The mass m_1 elongates the spring 1 by a displacement u_1. The elongated spring has a *restoration force* $-c_1u_1$ acting on m_1.[1] This force must be balanced with the gravitational force on m_1.[2] Thus, we have

$$-c_1u_1 + f_1 = 0,$$

where $f_1 = m_1g$, the gravitation force on m_1, and g is the gravitation constant. From this, we get

$$u_1 = \frac{f_1}{c_1}.$$

Next, let us consider the case where there is a bottom wall. In this case, both springs 1 and 2 exert forces upward to m_1. The balance law becomes

$$-c_1u_1 - c_2u_1 + f_1 = 0.$$

This results $u_1 = f_1/(c_1 + c_2)$.

Let us jump to a slightly more complicated case, say $n = 3$. The displacements

$$u_0 = 0,\ u_4 = 0,$$

due to the walls are fixed. The displacements u_1, u_2, u_3 cause elongations of the springs:

$$e_i = u_i - u_{i-1}, i = 1, 2, 3, 4.$$

The restoration force of spring i is

$$w_i = c_ie_i.$$

The force exerted to m_i by spring i is $-w_i = -c_ie_i$. In fact, when $e_i < 0$, the spring is shortened and it pushes downward to mass m_i (the sign is positive), hence the force is $-c_ie_i > 0$. On the other hand, when $e_i > 0$, the spring is elongated and it pull m_i upward. We still get the force

[1] The minus sign is due to the direction of force is upward.

[2] The mass m_1 is in equilibrium.

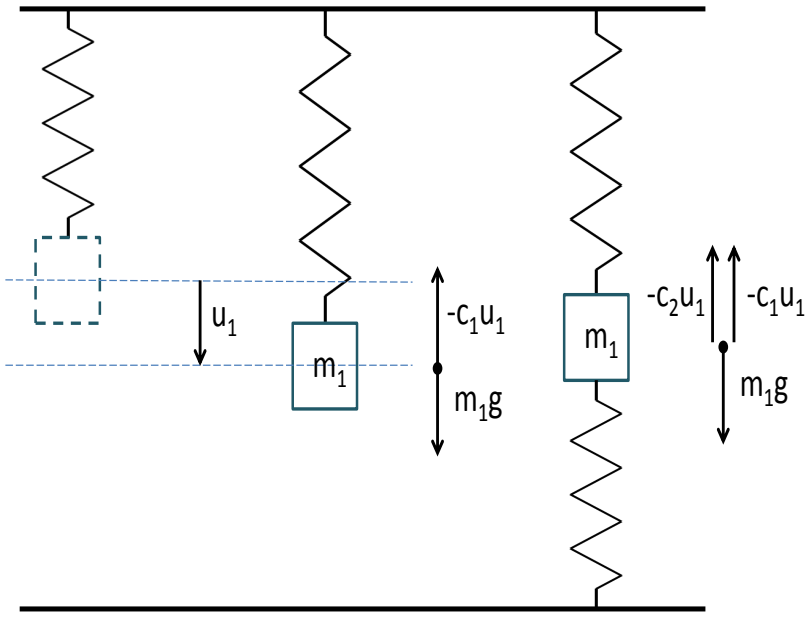


Figure 2.1: The left one is a spring without any mass. The middle one is a spring hanging a mass m_1 freely. The right one is a mass m_1 with two springs fixed on the ceiling and floor.

$-w_i = -c_i e_i < 0$. Similarly, the force exerted to m_i by spring $i+1$ is $w_{i+1} = c_{i+1}e_{i+1}$. When $e_{i+1} > 0$, the spring $i+1$ is elongated and it pulls m_i downward, the force is $w_{i+1} = c_{i+1}e_{i+1} > 0$. When $e_{i+1} < 0$, it pushes m_i upward, and the force $w_{i+1} = c_{i+1}e_{i+1} < 0$. In both cases, the force exterted to m_i by spring $i+1$ is w_{i+1}.

Thus, the force balance law on m_i is

$$w_{i+1} - w_i + f_i = 0, i = 1, 2, 3.$$

There are three algebraic equations for three unknowns u_1, u_2, u_3. In principle, we can solve it.

Let us express the above equations in matrix form. First, the elongation:

$$e = Au, \text{ or } \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix} = \begin{pmatrix} 1 & & \\ -1 & 1 & \\ & -1 & 1 \\ & & -1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$$

the restoration force:

$$w = Ce, \text{ or } \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} = \begin{pmatrix} c_1 & & & \\ & c_2 & & \\ & & c_3 & \\ & & & c_4 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix}$$

the force balance laws:

$$A^t w = f, \text{ or } \begin{pmatrix} 1 & -1 & & \\ & 1 & -1 & \\ & & 1 & -1 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}$$

where A^t is the transpose of A.

We can write the above equations in block matrix form as

$$\begin{pmatrix} C^{-1} & A \\ A^t & 0 \end{pmatrix} \begin{pmatrix} -w \\ u \end{pmatrix} = \begin{pmatrix} 0 \\ -f \end{pmatrix}.$$

This kind of block matrix appears commonly in many other physical systems, for instance, network flows, fluid flows. In fact, any optimization system with constraint can be written in this form. Here, the constraint part is the second equation. We shall come back to this point in the next section.

One way to solve the above block matrix system is to eliminate the variable w and get

$$Ku := A^t C A u = f.$$

The matrix $K := A^t C A$ is a symmetric positive definite matrix. It is called the *stiffness matrix*. For $n = 4$, we get

$$K := A^t C A = \begin{pmatrix} c_1 + c_2 & -c_2 & 0 \\ -c_2 & c_2 + c_3 & -c_3 \\ 0 & -c_3 & c_3 + c_4 \end{pmatrix}$$

Mimimum principle Consider the functional

$$P(u) := \frac{1}{2}(Ku, u) - (f, u),$$

where K is a symmetric positive definite matrix in $\mathbb{R}^n$. The directional derivative of P at u in the direction v is defined as

$$P'(u)v = \left.\frac{d}{dt}\right|_{t=0} P(u + tv)$$

$P'(u)$ is called the gradient (or the first variation) of P at u. We can compute this gradient: [3]

$$\begin{aligned} P'(u)v &= \frac{d}{dt}\Big|_{t=0} \frac{1}{2}(K(u+tv), u+tv) - (f, u+tv) \\ &= \frac{1}{2}\Big((Kv, u) + (Ku, v)\Big) - (f, v) \\ &= (Ku - f, v). \end{aligned}$$

Here, we have used K being symmetric. Thus,

$$P'(u) = Ku - f.$$

The second derivative is the Hessian. It is

$$P''(u) = K.$$

If u^* is a minimum of $P(v)$, then $P'(u^*) = 0$. This is called the Euler-Lagrange equation of P.

Conversely, If u^* satisfies the Euler-Lagrange equation $Ku^* = f$, then u^* is the minimum of $P(v)$. In fact, for any v, we compute $P(v) - P(u^*)$. We claim

$$P(v) - P(u^*) = \frac{1}{2}(K(v - u^*), v - u^*).$$

To see this, since $P(v)$ is a quadratic function of v, we can complete the squares:

$$\begin{aligned} P(v) - P(u^*) &= \frac{1}{2}(Kv, v) - (f, v) - \frac{1}{2}(Ku^*, u^*) + (f, u^*) \\ &= \frac{1}{2}(Kv, v) - \frac{1}{2}(Ku^*, u^*) - (f, v - u^*) \\ &= \frac{1}{2}(Kv, v) - \frac{1}{2}(Ku^*, u^*) - (Ku^*, v - u^*) \\ &= \frac{1}{2}(Kv, v) + \frac{1}{2}(Ku^*, u^*) - (Ku^*, v) \\ &= \frac{1}{2}(K(v - u^*), v - u^*) \geq 0. \end{aligned}$$

Hence we get that u^* is a minimum. In fact, u^* is the only minimum because $P(v) = P(u^*)$ if and only if $(K(v - u^*), v - u^*) = 0$. Since K is positive definite, we get $v - u^* = 0$.

We conclude the above discussion as the follows.

Theorem 2.1. *Let $P(u) := \frac{1}{2}(Ku, u) - (f, u)$ and K is symmetric positive definite. The vector u^* which minimizes $P(v)$ must satisfy the Euler-Lagrange equation $P'(u^*) = Ku^* - f = 0$. The converse is also true.*

[3] Here, I use the following properties: $(f, g)' = (f', g) + (f, g')$. This is because $(f, g) = \sum_i f_i g_i$ and $(f, g)' = \sum_i \left(f_i' g_i + f_i, g_i'\right) = (f', g) + (f, g')$.

The physical meaning of P is the *total potential energy* of the spring-mass system. Indeed,

$$\frac{1}{2}(CAu, Au) = \sum_{i=1}^{n} \frac{1}{2} c_i (u_i - u_{i-1})^2$$

is the sum of the *potential energy stored in the spring,* whereas the term

$$(f, u) = \sum_{i=1}^{n} f_i u_i$$

is the sum of the *works* done by the mass m_i with displacement u_i for $i = 1, ..., n$. The term $-(f, u)$ is the *gravitational potential* due to the masses m_i with displacements u_i.

Principal Component Analysis In statistics, the data set is usually represented as a matrix A. The data are collected by n experiments. Each experiment has p items, represented by a p row vector. For instance, a row vector is a biological records of a person, containing blood pressure, glucose, etc. Suppose there are p items. The data set has n peoples' data. Another example is the pattern recognition of a word. The image of a word is represented by a 20 pixel image. We transform it into a row vector. Suppose there are n experiments (say 100 images with the same words written repeatedly.)

To analysis the data set, we first normalize it:

$$\bar{a}_j := (\sum_{i=1}^{n} a_{ij}, \quad A_0 := \bar{\mathbf{a}} \mathbf{1}^T.$$

$$A_1 := (a_{ij} - \bar{a}_j)_{n \times p}$$

The matrix A_1 has zero mean in each column (item). The matrix

$$C := A_1^* A_1 = (\langle a_{ki} - \bar{a}_i, a_{kj} - \bar{a}_j \rangle)_{p \times p}.$$

is called the covariance matrix. It measures the covariance between item i and item j. The principal component analysis is to decompose A_1 into

$$A_1 = \sum_{i=1}^{p} \sigma_i \mathbf{u}_i \mathbf{v}_i^T.$$

Here $\mathbf{u}_i$ is an $n \times 1$ vector, $\mathbf{v}_i$ a $p \times 1$ vector. The vectors $\mathbf{v}_i$ are indeed the eigenvectors of $A_1^* A_1$. They are orthogonal. This means that $\mathbf{v}_i$ and $\mathbf{v}_j$ are uncorrelated. Thus, we can decompose the data set A_1 in terms of p uncorrelated items $(\mathbf{v}_i)_{i=1}^p$. These $\mathbf{v}_i$ are recombination of the items. We may call them the features. In principal analysis, we want to approximate the data set A_1 by only few such features.

In matrix completion problem, a typical application is to complete an incomplete table of data set. The video company Netflix proposed her incomplete table: each row is the rating record of a

person on a list of videos. This is an incomplete matrix A. The row m are member list, which is about 500,000. The column n is the video list, which is about $20,000$. The rating is from 1 star to 5 stars. The matrix is certainly incomplete. We want to fill in the vacancy based on an assumption that the completed matrix is low rank. The singular value decomposition of A is

$$A = \sum_{i=1}^{p} \sigma_i \mathbf{u}_i \mathbf{v}_i^T.$$

Each class $\mathbf{u}_i \mathbf{v}_i^T$ represents certain types of videos. For instance, the drama, the action, etc. In a class $\mathbf{u}_i \mathbf{v}_i^T$, the row $\mathbf{v}_i^T$ lists those videos with this type (say drama), and $\mathbf{u}_i$ lists those members who rate these videos higher. The completed matrix can be used for recommendation to the members.

2.2 Introduction and overview

There are three kinds of linear problems we encounter in applications:

- solving large linear system: $Ax = b$
- solving least squares problem: $\min_x \|Ax - b\|^2$.
- solving eigenvalue problems: $Ax = \lambda x$
- solving singular value decomposition problem.

In solving linear systems, there are two classes of methods:

- Direct methods: which solves the equation directly. This is usually for small system. Basically, the solving process is a factorization of the matrix A such as LU-factorization.
- Iterative methods: the basic idea is to decompose $A = M - N$, where M is a major part and easy to invert, while N is a minor part. Then perform an iteration $Mx_{n+1} - Nx_n = b$ to get an approximate solution. Usually, a preconditioning is needed, which means that we replace $Ax = b$ by $PAx = Pb$ so that it is easy to have above major-minor decomposition.

In solving eigenvalue problems, I shall discuss the power method and QR algorithm. For least square problem, I shall discussed weighted iterative method.

2.3 *Matrix Algebra

Spectral Decomposition We assume $\mathbf{A}$ is an $n \times n$ matrix in $\mathbb{C}^n$.

Theorem 2.2 (Caley-Hamilton)**.** *Let $p_A(\lambda) := det(\lambda \mathbf{I} - \mathbf{A})$ be the characteristic polynomial of $\mathbf{A}$. Then $p_A(\mathbf{A}) = \mathbf{0}$.*

Theorem 2.3. *There exists a minimal polynomial p_m which is a factor of p_A and $p_m(\mathbf{A}) = 0$.*

Theorem 2.4 (Fundamental Theorem of Algebra)**.** *Any polynomial $p(\lambda)$ over $\mathbb{C}$ of degree m can be factorized as*

$$p(\lambda) = a\prod_{i=1}^{m}(\lambda - \lambda_i)$$

for some constant $a \neq 0$ and $\lambda_1, ..., \lambda_m \in \mathbb{C}$. This factorization is unique.

Definition 2.2. *Let $\mathbf{A} : \mathbb{C}^n \to \mathbb{C}^n$. A subspace $\mathcal{V} \subset \mathbb{C}^n$ is called an invariant subspace of the linear map $\mathbf{A}$ if $\mathbf{A}\mathcal{V} \subset \mathcal{V}$.*

Definition 2.3. *Let $\mathbf{A} : \mathbb{C}^n \to \mathbb{C}^n$. A vector $\mathbf{v}$ is called an eigenvector of $\mathbf{A}$ if there exists a λ such that*

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}.$$

Definition 2.4. *For a matrix $\mathbf{A}$, the set of all its eigenvalues $\sigma(A) := \{\lambda_1, ..., \lambda_n\}$ is called the spectra of $\mathbf{A}$.*

Definition 2.5. *A vector space $\mathcal{V}$ is said to be the direct sum of its two subspaces $\mathcal{V}_1$ and $\mathcal{V}_2$ if for any $\mathbf{v} \in \mathcal{V}$ there exist two unique vectors $\mathbf{v}_i \in \mathcal{V}_i$, $i = 1, 2$ such that $\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2$. We denote it by $\mathcal{V} = \mathcal{V}_1 \oplus \mathcal{V}_2$.*

Remark 2.1. *We also use the notation $\mathcal{V} = \mathcal{V}_1 + \mathcal{V}_2$ for the property: any $\mathbf{v} \in \mathcal{V}$ can be written as $\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2$ for some $\mathbf{v}_i \in \mathcal{V}_i$, $i = 1, 2$. Notice that $\mathcal{V} = \mathcal{V}_1 \oplus \mathcal{V}_2$ if and only if $\mathcal{V} = \mathcal{V}_1 + \mathcal{V}_2$ and $\mathcal{V}_1 \cap \mathcal{V}_2 = \{0\}$.*

Lemma 2.1. *Suppose p and q are two polynomials over $\mathbb{C}$ and are relatively prime (i.e. no common roots). Then there exist two other polynomials a and b such that*

$$ap + bq = 1.$$

Lemma 2.2. *Suppose p and q are two polynomials over $\mathbb{C}$ and are relatively prime (i.e. no common roots). Let $\mathcal{N}_p := Ker(p(\mathbf{A}))$, $\mathcal{N}_q := Ker(q(\mathbf{A}))$ and $\mathcal{N}_{pq} := Ker(p(\mathbf{A})q(\mathbf{A}))$. Then*

$$\mathcal{N}_{pq} = \mathcal{N}_p \oplus \mathcal{N}_q.$$

Proof. From $ap + bq = 1$ we get

$$a(\mathbf{A})p(\mathbf{A}) + b(\mathbf{A})q(\mathbf{A}) = \mathbf{I}.$$

For any $\mathbf{v} \in \mathcal{N}_{pq}$, acting the above operator formula to $\mathbf{v}$, we get

$$\mathbf{v} = a(\mathbf{A})p(\mathbf{A})\mathbf{v} + b(\mathbf{A})q(\mathbf{A})\mathbf{v} := \mathbf{v}_2 + \mathbf{v}_1.$$

We claim that $\mathbf{v}_1 \in \mathcal{N}_p$, whereas $\mathbf{v}_2 \in \mathcal{N}_q$. This is because

$$p(A)\mathbf{v}_1 = p(A)b(\mathbf{A})q(\mathbf{A})\mathbf{v} = b(\mathbf{A})p(\mathbf{A})q(\mathbf{A})\mathbf{v} = 0.$$

Similar argument for proving $\mathbf{v}_2 \in \mathcal{N}_q$. To see this is a direct sum, suppose $\mathbf{v} \in \mathcal{N}_p \cap \mathcal{N}_q$. Then

$$\mathbf{v} = a(\mathbf{A})p(\mathbf{A})\mathbf{v} + b(\mathbf{A})q(\mathbf{A})\mathbf{v} = 0.$$

Hence $\mathcal{N}_p \cap \mathcal{N}_q = \{0\}$. □

Corollary 2.1. *Suppose a polynomial p is factorized as $p = p_1 \cdots p_s$ with $p_1, ..., p_s$ are relatively prime (no common roots). Let $\mathcal{N}_{p_i} := Kerp_i(\mathbf{A})$. Then*

$$\mathcal{N}_p = \mathcal{N}_{p_1} \oplus \cdots \oplus \mathcal{N}_{p_s}.$$

Theorem 2.5 (Spectral Decomposition)**.** *Let p_m be the minimal polynomial of $\mathbf{A}$. Suppose p_m can be factorized as*

$$p_m(\lambda) = \prod_{i=1}^{s} p_i(\lambda) = \prod_{i=1}^{s} (\lambda - \lambda_{k_i})^{m_i}$$

with $\lambda_{k_i} \neq \lambda_{k_j}$ for $i \neq j$. Let $\mathcal{N}_{k_i} = Ker(\mathbf{A} - \lambda_{k_i}\mathbf{I})^{m_i}$. Then

- *$\mathcal{N}_{k_i}$ is invariant under $\mathbf{A}$,*
- *$\mathbb{C}^n = \mathcal{N}_{k_1} \oplus \cdots \oplus \mathcal{N}_{k_s}$.*

Jordan matrix A matrix $\mathbf{J}$ is called a Jordan normal form of a matrix $\mathbf{A}$ if we can find matrix $\mathbf{V}$ such that

$$\mathbf{AV} = \mathbf{VJ},$$

where

$$\mathbf{J} = \mathbf{J}_{k_1} \otimes \cdots \otimes \mathbf{J}_{k_p} := \begin{pmatrix} \mathbf{J}_{k_1} & & & \\ & \mathbf{J}_{k_2} & & \\ & & \ddots & \\ & & & \mathbf{J}_{k_p} \end{pmatrix}, \quad \mathbf{V} = [\mathbf{V}_{k_1}, \mathbf{V}_{k_2}, \cdots, \mathbf{V}_{k_p}].$$

$$\mathbf{J}_k(\lambda_k) = \begin{pmatrix} \lambda_k & 1 & & & \\ & \lambda_k & 1 & & \\ & & \ddots & \ddots & \\ & & & \lambda_k & 1 \\ & & & & \lambda_k \end{pmatrix}_{k \times k}, \quad \mathbf{V}_k = [\mathbf{v}_k^1, \cdots, \mathbf{v}_k^k], \quad k = k_1, ..., k_s,$$

$$\sum_{i=1}^{s} k_i = n.$$

Here, λ_{k_i} are the eigenvalues of $\mathbf{A}$, $\mathbf{v}_k^j \in \mathbb{C}^n$ are called the generalized eigenvectors of $\mathbf{A}$, the matrices $\mathbf{J}_k$ are called Jordan blocks of size k of $\mathbf{A}$. The matrix $\mathbf{V}_k = [\mathbf{v}_k^1, \cdots, \mathbf{v}_k^k]$ is an $n \times k$ matrix. We can restrict $\mathbf{A}$ to $\mathbf{V}_k$, $k = k_1, ..., k_s$ as

$$\mathbf{AV}_k = \mathbf{A}[\mathbf{v}_k^1, \cdots, \mathbf{v}_k^k] = [\mathbf{v}_k^1, \cdots, \mathbf{v}_k^k]\mathbf{J}_k, \ k = k_1, ..., k_s.$$

For each generalized vector,

$$\begin{aligned}(\mathbf{A}-\lambda_k\mathbf{I})\mathbf{v}_k^1 &= 0\\ (\mathbf{A}-\lambda_k\mathbf{I})\mathbf{v}_k^2 &= \mathbf{v}_k^1\\ &\vdots\\ (\mathbf{A}-\lambda_k\mathbf{I})\mathbf{v}_k^k &= \mathbf{v}_k^{k-1}, \quad k=k_1,...,k_s.\end{aligned}$$

This implies

$$\begin{aligned}(\mathbf{A}-\lambda_k\mathbf{I})\,\mathbf{v}_k^1 &= 0\\ (\mathbf{A}-\lambda_k\mathbf{I})^2\mathbf{v}_k^2 &= 0\\ &\vdots\\ (\mathbf{A}-\lambda_k\mathbf{I})^k\mathbf{v}_k^k &= 0, \quad k=k_1,...,k_s.\end{aligned}$$

The set $\{\mathbf{v}_{k_i}^j\}$ form a basis in $\mathbb{C}^n$. Therefore, $\mathbf{V}$ is invertible, and

$$\mathbf{A}=\mathbf{V}\mathbf{J}\mathbf{V}^{-1}.$$

We call $\mathbf{A}$ is similar to $\mathbf{J}$, and is denoted by $\mathbf{A}\sim\mathbf{J}$.

Notice that the matrix $\mathbf{N}_k := \mathbf{J}_k - \lambda_k\mathbf{I}$ is called a Nilpotent matrix, which has the form

$$\mathbf{N}_k = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ & & & 0 \end{pmatrix}_{k\times k}.$$

It is easy to check that

$$\mathbf{N}_k^2 = \begin{pmatrix} 0 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 0 & 0 & 1 \\ & & & 0 & 0 \\ & & & & 0 \end{pmatrix}_{k\times k}, \quad \cdots, \mathbf{N}_k^k = \mathbf{0}.$$

Theorem 2.6. *Any matrix* $\mathbf{A}$ *over* $\mathbb{C}$ *is similar to a Jordan normal form. The structure of this Jordan normal form is unique.*

Example Suppose $\mathbf{A}$ is a 2×2 matrix with double eigenvalue λ. Let $\mathcal{N}_1 = Ker(\mathbf{A}-\lambda\mathbf{I})$ and $\mathcal{N}_2 = Ker(\mathbf{A}-\lambda\mathbf{I})^2$. We assume $dim\mathcal{N}_1 = 1$. Then $\mathcal{N}_1 \subset \mathcal{N}_2 = \mathbb{C}^2$. Let us choose any $\mathbf{v}_2 \in \mathcal{N}_2 \setminus \mathcal{N}_1$. We define $\mathbf{v}_1 = (\mathbf{A}-\lambda\mathbf{I})\mathbf{v}_2$. Then $(\mathbf{A}-\lambda\mathbf{I})\mathbf{v}_1 = (\mathbf{A}-\lambda\mathbf{I})^2\mathbf{v}_2 = 0$. Thus, under $[\mathbf{v}_1, \mathbf{v}_2]$, the matrix $\mathbf{A}$ is transformed to $\mathbf{J}_2(\lambda)$.

Orthogonality, Self-adjoint operators There are some other decomposition, mainly when the under space $\mathbb{R}^n$ or $\mathbb{C}^n$ endowed with inner product structure.

Below V and W are vector spaces.

1. Orthogonal Projection: Given $W \subset V$, there is an orthogonal projection $P : V \to W$ such that (i) $Pw = w$ for all $w \in W$, (ii) $(I - P)v \perp W$ for all $v \in V$.

2. For any $W \subset V$, there is a subspace $W^\perp$ such that (i) $V = W \oplus W^\perp$, (ii) $W \cap W^\perp = \{0\}$, (iii) $W \perp W^\perp$.

3. Self adjoint operator: we define $A^* = (\bar{a}_{ji})$. A matrix A is called self-adjoint if $A^* = A$.

4. Alternatively, A^* is defined by
$$\langle v, A^* w\rangle = \langle Av, w\rangle,$$
and A is self-adjoint if $\langle Av, w\rangle = \langle v, Aw\rangle$.

5. A matrix U is unitary if $U^*U = UU^* = I$. This is equivalent to that $U = [u_1, ..., u_n]$ and $\{u_i\}_{i=1}^n$ are orthonormal.

Theorem 2.7. *If A is self adjoint, then A is diagonalizable by a unitary matrix U and all eigenvalues are real.*

Proof. 1. Suppose μ is an eigenvalue of A. By the spectral decomposition theorem, we can find the maximal invariant subspace W corresponding to $\mu I - A$. Let $J = \mu I - A$. We claim that $J = 0$ on W.

2. Since A is self-adjoint, so is J.

3. If the minimal polynomial of J in W is $p_m(\lambda) = \lambda^m$. If $m > 1$, this means that there exists v_1 and v_2 which are independent such that
$$Jv_1 = 0, Jv_2 = v_1.$$
Then we have
$$\langle v_1, v_1\rangle = \langle Jv_2, v_1\rangle = \langle v_2, Jv_1\rangle = 0.$$
This is a contradiction. Hence, $m = 1$. This also means $J = 0$.

4. The eigenvalues are real. Suppose λ, v are a pair of eigenvalue/eigenvector.
$$\begin{aligned}\lambda\langle v, v\rangle &= \langle \lambda v, v\rangle = \langle Av, v\rangle \\ &= \langle \lambda v, Av\rangle = \langle \lambda v, \lambda v\rangle = \bar{\lambda}\langle v, v\rangle\end{aligned}$$

5. The eigenspace corresponding to two distinct eigenvalues $\lambda \neq \mu$ are orthogonal to each other. Suppose
$$Av = \lambda v, \quad Aw = \mu w, \quad \lambda \neq \mu.$$

Then

$$\lambda\langle v, w\rangle = \langle Av, w\rangle = \langle v, Aw\rangle = \mu\langle v, w\rangle$$

Hence, we get $\langle v, w\rangle = 0$.

□

The Rayleigh quotient method is a constructive method to find eigenvalues of self-adjoint operator.

$$\lambda_1 = \max_v \frac{\langle Av, v\rangle}{\langle v, v\rangle}.$$

Suppose V_1 be the corresponding eigenspace.

$$\lambda_2 = \max_{v \perp V_1} \frac{\langle Av, v\rangle}{\langle v, v\rangle}.$$

This process can be proceeded inductively and find all eigenvalues and eigenvectors.

Singular Value Decomposition

Theorem 2.8. *Let $A : \mathbb{R}^n \to \mathbb{R}^m$(or $\mathbb{C}^n \to \mathbb{C}^m$). Then there exist orthonormal bases $V = [v_1, ..., v_n]$ in $\mathbb{R}^n$ and $U = [u_1, ..., u_m]$ and non-negative numbers*

$$\sigma_1 \geq ... \geq \sigma_p > 0, \quad p \geq \min(m, n)$$

such that

$$Av_i = \sigma_i u_i, \quad i = 1, ..., p,$$
$$Av_i = 0 \quad \text{for } p < i \leq n$$

Or in matrix form

$$AV = U\Sigma,$$

where V is $n \times n$ unitary matrix, U is $m \times m$ unitary matrix, Σ is $m \times n$ diagonal matrix:

$$\Sigma = \begin{cases} (diag(\sigma_1, ..., \sigma_p), \mathbf{0}) & \text{if } m \leq n \\ (diag(\sigma_1, ..., \sigma_p), \mathbf{0})^T & \text{if } m > n. \end{cases}$$

Proof. 1. The matrix A^*A is self-adjoint. All its eigenvalues are real. They are also non-negative because if λ and v is a pair of eigenvalue/eigenvector, then from Rayleigh quotient

$$\lambda\langle v, v\rangle = \langle A^*Av, v\rangle = \langle Av, Av\rangle \geq 0.$$

2. From the spectral decomposition for the self-adjoint matrix A^*A, we can find unitary matrix $[v_1, ..., v_n]$ and $\Lambda = \text{diag}(\lambda_1, ..., \lambda_p, 0, ..., 0)$ such that $AV = V\Lambda$. Here, $\lambda_1 \geq \cdots \geq \lambda_p > 0$, the rest eigenvalues are 0. The corresponding eigenspace spanned by $< v_{p+1}, ..., v_n >$ is the kernel $N(A^*A)$.

3. We note that $A^*Av = 0$ if and only if $Av = 0$. Therefor, $N(A)$ is also spanned by $[v_{p+1}, ..., v_n]$.

4. Let $\sigma_i = \sqrt{\lambda_i}$. For $\sigma_i > 0$, define u_i by

$$Av_i = \sigma_i u_i.$$

Then,

$$\langle u_i, u_j \rangle = \frac{1}{\sigma_i \sigma_j} \langle Av_i, Av_j \rangle = \frac{1}{\sigma_i \sigma_j} \langle A^*Av_i, v_j \rangle = \frac{\sigma_i}{\sigma_j} \langle v_i, v_j \rangle = 0.$$

$$\langle u_i, u_i \rangle = \frac{1}{\sigma_i^2} \langle Av_i, Av_i \rangle = \frac{1}{\sigma_i^2} \langle A^*Av_i, v_i \rangle = \langle v_i, v_j \rangle = 1.$$

The space spanned by $[u_1, ..., u_p]$ is $R(A)$. Let us choose $[u_{p+1}, ..., u_m]$ to be an orthonormal basis in $R(A)^\perp \subset \mathbb{R}^m$. Then $[u_1, ..., u_m]$ is unitary matrix.

□

Remarks.

1. A^* has the following representation:

$$A^*u_i = \frac{1}{\sigma_i} A^*(Av_i) = \sigma_i v_i.$$

2. The domain and range of A can be decomposed into

$$\mathbb{R}^n = [v_1, ..., v_p] \oplus [v_{p+1}, ..., v_n] = [v_1, ..., v_p] \oplus N(A),$$

$$R^m = [u_1, ..., u_p] \oplus [u_{p+1}, ..., u_m] = R(A) \oplus [u_{p+1}, ..., u_m].$$

3. An $m \times n$ matrix is of rank 1 if and only if it has the form

$$u \otimes v, \quad uv^T$$

where u is $m \times 1$ and v^T is $1 \times n$ matrices. The SVD is to decompose A into a sum of rank 1 matrices

$$A = [u_1, ..., u_m] \begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_p & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix} = \sum_{i=1}^{p} \sigma_i u_i v_i^T,$$

with both $U = [u_1, ..., u_p]$ and $V = [v_1, ..., v_p]$ are orthonormal.

4. The least-squares solution for $Ax = b$ is the minimizer of

$$\frac{1}{2}\|Ax - b\|_2^2$$

With the singular value decomposition, we can represent

$$b = \sum_{i=1}^{p} \langle b, u_i \rangle u_i + \sum_{i=p+1}^{m} \langle b, u_i \rangle u_i = \sum_{i=1}^{p} \langle b, u_i \rangle u_i + b^{\perp}$$

The least squares solution is

$$x^* = \sum_{i=1}^{p} \frac{1}{\sigma_i} \langle b, u_i \rangle v_i + N(A),$$

which minimize $\|Ax - b\|^2$ with minimal value

$$\|Ax^* - b\|^2 = \|b^{\perp}\|^2.$$

The solution $x^{\dagger} := \sum_{i=1}^{p} \frac{1}{\sigma_i} \langle b, u_i \rangle v_i$, denoted by $A^{\dagger} b$, is called the Moore-Penrose solution, where

$$A^{\dagger} := V \Sigma^{\dagger} U^*$$

$\Sigma^{\dagger}$ has the same structure as Σ and replacing σ_i by σ_i^{-1}. The matrix $A^{\dagger}$ is called the pseudo inverse of A.

2.4 Matrix Analysis

2.4.1 Matrix Norm

Norm in vector space In analysis, we need to measure how close of two vectors, the concept of convergence. A natural way is to define the concept of norm for vectors.

Definition 2.6. *Let V be a vector space. A mapping $\|\cdot\| : V \to \mathbb{R}$ is called a norm if*

(i) $\|x\| \geq 0$ *and* $\|x\| = 0$ *if and only if* $x = 0$*;*

(ii) $\|\lambda x\| = |\lambda| \|x\|$ *for any* $\lambda \in \mathbb{R}$ *and any* $x \in \mathbb{R}^n$*;*

(iii) $\|x + y\| \leq \|x\| + \|y\|$.

A vector space endowed with a norm $\|\cdot\|$ is called a normed vector space.

In $\mathbb{R}^n$, we define the norms

$$|x|_p := \left(\sum_{j=1}^{n} |x_j|^p \right)^{1/p}, \ 1 \leq p < \infty, \quad |x|_\infty = \max_i |x_i|$$

One can see that $|x|_p \to |x|_\infty$ as $p \to \infty$.

Matrix Norm A $m \times n$ matrix A is viewed as a linear map from $\mathbb{R}^n \to \mathbb{R}^m$ (or $\mathbb{C}^n \to \mathbb{C}^m$). The set of all $m \times n$ matrices is denoted by $\mathcal{M}_{m\times n}$, which is a linear space.

The norms in domain and range may be different. Let us call the norm in the domain by $\|\cdot\|_a$ and range by $\|\cdot\|_b$. The linear map $A : (\mathbb{R}^n, \|\cdot\|_a) \to (\mathbb{R}^m, \|\cdot\|_b)$ induces an operator norm on the matrix A defined by

$$\|A\|_{a\to b} := \max_{x\neq 0} \frac{\|Ax\|_b}{\|x\|_a} \equiv \max_{\|x\|_a=1} \|Ax\|_b.$$

Most of the time, we drop the subindex ($a \to b$) when it is clear from the context. It is easy to see that $\|\cdot\|$ is a norm in the vector space $\mathcal{M}_{m\times n}$. Let us check the triangle inequality:

$$\begin{aligned}\|A+B\| &= \max_{\|x\|=1} \|(A+B)x\| \\ &\le \max_{\|x\|=1} \|Ax\| + \max_{\|x\|=1} \|Bx\| \\ &= \|A\| + \|B\|.\end{aligned}$$

The operator has the following two important properties

- $\|Ax\| \le \|A\|\,\|x\|$
- $\|AB\| \le \|A\|\,\|B\|$ when both $A, B \in \mathcal{M}_{n\times n}$.

Examples

1. $A : (\mathbb{R}^n, |\cdot|_\infty) \to (\mathbb{R}^m, |\cdot|_\infty)$ Then $\|A\|_\infty = \max_i \sum_j |a_{ij}|$

$$\begin{aligned}\|A\|_\infty &= \sup_{|x|_\infty=1} \max_i |\sum_j a_{ij}x_j| = \max_i \sup_{|x|_\infty=1} |\sum_j a_{ij}x_j| \\ &\le \max_i \sup_{|x|_\infty=1} (\sum_j |a_{ij}|)(\max_j |x_j|) = \max_i \sum_j |a_{ij}|\end{aligned}$$

   Conversely, if $\max_i \sum_j |a_{ij}| = \sum_j |a_{i_0 j}|$, we choose

$$x_j = \operatorname{sign} a_{i_0 j} = \pm 1.$$

   Then $|x|_\infty = 1$ and

$$\|A\|_\infty \ge |Ax|_\infty = \max_i |\sum_j a_{ij}x_j| = \sum_j |a_{i_0 j}|.$$

2. $A : (\mathbb{R}^n, |\cdot|_1) \to (\mathbb{R}^m, |\cdot|_1)$, then $\|A\|_1 = \max_j \sum_i |a_{ij}|$.

$$\begin{aligned} |Ax|_1 &= \sum_i |\sum_j a_{ij}x_j| \le \sum_i \sum_j |a_{ij}|\,|x_j| \\ &= \sum_j \left(\sum_i |a_{ij}|\right)|x_j| \le \sum_j \left(\max_k \sum_i |a_{ik}|\right)|x_j| \\ &= \left(\max_k \sum_i |a_{ik}|\right)|x|_1 \end{aligned}$$

Thus, we obtain $\|A\|_1 \le \max_j \sum_i |a_{ij}|$. Conversely, if $\max_j \sum_i |a_{ij}| = \sum_i |a_{ij_0}|$, then we choose

$$x = (\delta_{jj_0})_{j=1}^n = (0, \cdots, 1, \cdots, 0)^T.$$

We have $|x|_1 = 1$ and

$$Ax = (a_{1j_0}, a_{2j_0}, \cdots, a_{nj_0})^T.$$

Thus,

$$\|A\|_1 \ge |Ax|_1 = \sum_{i=1}^n |a_{ij_0}| = \max_j \sum_i |a_{ij}|.$$

3. $\|A\|_2 = \sqrt{\rho(A^*A)}$, where $\rho(B)$ is the spectral radius of a matrix B, which is defined for a general square matrix B, by

$$\rho(B) = \max_i \{|\lambda_1(B)|, ..., |\lambda_n(B)|\},$$

the largest eigenvalues of B in magnitude.

Proof. Since A^*A is hermitian, its eigenvalues are real. Let us order them by

$$\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n.$$

Then the spectral radius $\rho(A^*A) = \lambda_1$. Suppose x is the unit eigenvector corresponding to λ_1, then

$$\|A\|_2^2 \ge |Ax|_2^2 = \langle Ax, Ax\rangle = \langle A^*Ax, x\rangle = \lambda_1\langle x, x\rangle = \lambda_1$$

We get $\|A\|_2 \ge \sqrt{\lambda_1}$. On the other hand, for any $|x|_2 = 1$, we have

$$|Ax|^2 = \langle Ax, Ax\rangle = \langle A^*Ax, x\rangle \le \lambda_1\langle x, x\rangle = \lambda_1.$$

We get $\|A\|_2^2 \le \lambda_1$. □

4. Frobenius norm: it is define to be

$$\|A\|_F^2 := \sum_{i,j} |a_{ij}|^2 = tr(A^*A).$$

This norm is easy to compute. It has the following properties

- $|Ax|_2 \leq \|A\|_F |x|_2$, $\|A\|_2 \leq \|A\|_F$;
- $\|AB\|_F \leq \|A\|_F \|B\|_F$;
- $\|AR\|_F = \|RA\|_F = \|A\|_F$ for any rotation matrix R.

The proofs of the first and second follow from Cauchy-Schwarz inequality. For the proof of the third statement,

$$\|AR\|_F^2 = tr(R^T A^T AR) = tr(RR^T A^T A) = tr(A^T A) = \|A\|_F^2.$$

Here, we have used the cyclic formula for trace:

$$tr(ABC) = tr(BCA) = tr(CAB).$$

5. Nuclear norm:

$$\|A\|_* = \sum_{i=1}^{\min(m,n)} |\sigma_i(A)|,$$

where $\sigma_1 \geq \sigma_2 \geq \cdots$ are the singular values of A, equivalently, $\sigma(A) = \sqrt{\lambda(A^*A)}$. The nuclear norm is also called Ky Fan 'n'-norm. It is used, for instance, in compressive sensing, in principal component analysis in statistics, to find a low rank matrix approximation to a given matrix.

6. *Schatten norm: the above nuclear norm, Frobenius norm, L^2 operator norm can all be unified as special cases of Schatten norm, which is defined as

$$\|A\|_p := \left(\sum_{i=1}^{\min(m,n)} \sigma_i(A)^p \right)^{1/p}.$$

From the functional calculus theory,

$$\|A\|_p^p = tr(|A|^p), \quad |A| := \sqrt{A^*A}.$$

Thus, the L^2 operator norm is the Schatten maximum norm. The nuclear norm is the Schatten 1-norm. The Frobenius norm is the Schatten 2-norm.

Remarks.

- The Nilponent matrix is defined to be

$$N = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

has $\rho(N) = 0$, but $\|N\|_1 = \|N\|_2 = \|N\|_\infty = 1$. The matrix

$$N^*N = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

Thus, the singular values are 1 and 0. Their Schatten norms $\|N\|_p = 1$.

2.4.2 Condition number

Consider

$$Ax = b,$$

where A is an $n \times n$ matrix and assume it is invertible. We want to measure the sensitivity of solving x from b. Suppose $\tilde{b}$ is a perturbation of b and $\tilde{x}$ the corresponding solution of $A\tilde{x} = \tilde{b}$. Then

$$\|x - \tilde{x}\| = \|A^{-1}(b - \tilde{b})\| \leq \|A^{-1}\|\|b - \tilde{b}\|$$

$$\frac{\|x - \tilde{x}\|}{\|x\|} \leq \|A^{-1}\|\|b - \tilde{b}\|\frac{1}{\|x\|}\frac{\|Ax\|}{\|b\|} \leq \|A^{-1}\|\|A\|\frac{\|b - \tilde{b}\|}{\|b\|}$$

Condition number $\kappa(A) := \|A\|\|A^{-1}\|$ measures the sensitivity of x w.r.t. b.

1. Find the condition number of
$$A = \begin{pmatrix} 1 & 1+\epsilon \\ 1-\epsilon & 1 \end{pmatrix}$$
Ans: $\kappa(A) \geq 4/\epsilon^2$.

2. The Hilbert matrix is given by
$$H = \left(\frac{1}{i+j+1}\right)_{0 \leq i,j \leq n}$$
Its condition number has estimate: $\kappa(H) \approx O((1+\sqrt{2})^{4n}/\sqrt{n})$.

3. The discrete Laplacian in one dimension with Dirichlet boundary condition is
$$A = \text{diag}(-1, 2, -1).$$
Since A is symmetric positive definite, we have
$$\|A\| = \max_i \lambda_i(A) = \lambda_1(A),$$
$$\|A^{-1}\| = \max_i |\lambda_i(A^{-1})| = \max_i |\lambda_i(A)^{-1}| = \lambda_n(A)^{-1}.$$
Thus,
$$\kappa(A) = \frac{\lambda_1(A)}{\lambda_n(A)}.$$

Homework

1. What is the explicit expression of the operator norm of $A : (\mathbb{R}^n, |\cdot|_1) \to (\mathbb{R}^m, |\cdot|_\infty)$?
2. Show that $\kappa(A) = \sup_{\|x\|=\|y\|} \frac{\|Ax\|}{\|Ay\|}$.

2.4.3 *Functional Calculus

Given an $n \times n$ matrix $\mathbf{A}$, we can define $p(\mathbf{A})$ for any polynomial p. Let $\sigma(\mathbf{A})$ denote the spectra of $\mathbf{A}$, we have the following spectral mapping theorem.

Theorem 2.9 (Spectral Mapping Theorem for polynomial functions)**.** *Let $\mathbf{A}$ be an $n \times n$ matrix over $\mathbb{C}$ and $\sigma(\mathbf{A})$ be its spectra. Then for any polynomial p, we have*

$$\sigma(p(\mathbf{A})) = p(\sigma(\mathbf{A})).$$

Proof. 1. By spectral decomposition theorem, there exists V and J such that

$$A = VJV^{-1}.$$

From this expression, we get that

$$A^k = (VJV^{-1}) \cdot (VJV^{-1}) \cdots (VJV^{-1}) = VJ^kV^{-1}.$$

The Jordan matrix has the form

$$J = J_1 \otimes \cdots \otimes J_m.$$

This implies

$$J^k = J_1^k \otimes \cdots \otimes J_m^k.$$

2. For each Jordan block above, say $J_p = \mu_p I + N$, N is Nilpotent, we can get that J_p^k is always upper triangular with diagonal $\mu_p^k I$. Thus, the eigenvalue of J_p^k is μ_p^k. This shows

$$\sigma(J_p^k) = (\sigma(J_p))^k.$$

3. Since

$$\sigma(J) = \sigma(J_1^k \otimes \cdots \otimes J_m^k) = \bigcup_{p=1}^{m} \sigma(J_p^k)$$

we then get

$$\sigma(A^k) = \sigma(J^k) = \bigcup_{p=1}^{m} \sigma(J_p^k) = \sigma(A)^k,$$

and thus

$$\sigma(p(A)) = p(\sigma(A))$$

for any polynomial function $p(\cdot)$. □

Remark. In applications, we will need $f(\mathbf{A})$ for more general functions. For example, A^{-1}, $\exp(\mathbf{A})$, $\sin(\mathbf{A})$, etc. These operator-valued functions can be defined through the helps of resolvent $(\lambda I - \mathbf{A})^{-1}$ and the Cauchy integral formula.

The resolvent $(\lambda I - A)^{-1}$ can be defined in $|\lambda| > \rho(A)$, where $\rho(A)$ is called the spectral radius of A. The spectral radius $\rho(A)$ is useful in the power series expansion of a matrix. We have the following theorem.

Theorem 2.10. *$A^k \to 0$ if and only if $\rho(A) < 1$. $\|A^k\|$ is unbounded as $k \to \infty$ if and only if $\rho(A) > 1$.*

Proof. If $A^k \to 0$, and suppose λ/x be a pair of eigenvalue/eigenvector of A, then

$$A^k x = \lambda^k x \to 0.$$

This implies $\lambda^k \to$. Hence $|\lambda| < 1$. This implies $\rho(A) < 1$.

Conversely, let us suppose $\rho(A) < 1$, which means that all eigenvalues $|\lambda(A)| < 1$. Let us decompose A into direct product of Jordan blocks: $AV = VJ$ with V invertible and $J = J_1 \otimes \cdots \otimes J_\ell$. The power

$$A^k = V J^k V^{-1}, \quad J^k = J_1^k \otimes \cdots \otimes J_\ell^k.$$

We can see that $\lambda(A) < 1 \Leftrightarrow J(\lambda(A))^k \to 0$, which is equivalent to $|\lambda(A)| < 1$.

Suppose $|\lambda(A)| > 1$ for some eigenvalue $\lambda(A)$, then the corresponding Jordan block

$$J^k = (\lambda I + N)^k = \sum_{m=0}^{k} \begin{pmatrix} k \\ m \end{pmatrix} \lambda^{k-m} I N^m \to \infty$$

if and only if $|\lambda| > 1$. □

Theorem 2.11 (Gelfand formula)**.** *For any matrix norm $\|\cdot\|$, we have*

$$\rho(A) = \lim_{k \to \infty} \|A^k\|^{1/k}.$$

Proof. For any $\epsilon > 0$, we have $\rho(A/(\rho(A) + \epsilon)) < 1$. Hence

$$\left(\frac{A}{\rho(A) + \epsilon} \right)^k \to 0 \text{ as } k \to \infty.$$

Thus, there exists N_1 such that for all $k > N_1$, we have

$$\left\| \left(\frac{A}{\rho(A) + \epsilon} \right)^k \right\| < 1$$

This means

$$\|A^k\| \le (\rho(A) + \epsilon)^k$$

or

$$\|A^k\|^{1/k} \leq \rho(A) + \epsilon.$$

Similarly, $\rho(A/(\rho(A) - \epsilon)) > 1$. From

$$\left\| \left(\frac{A}{\rho(A) - \epsilon} \right)^k \right\| \to \infty \text{ as } k \to \infty,$$

there exists N_2 such that for all $k > N_2$,

$$\|A^k\|^{1/k} \geq \rho(A) - \epsilon.$$

This completes the proof. □

Theorem 2.12. *The series $\sum_{n=0}^{\infty} A^n$ converges if and only if $\rho(A) < 1$. In the convergence case, the series equals $(I - A)^{-1}$.*

Proof. Suppose $\rho(A) < 1$, we want to show $(I - A)$ is invertible. The key is to expand

$$(I - A)^{-1} = \sum_{n=0}^{\infty} A^n.$$

This is called Neumann series. From $\rho(A) < 1$, we chooser $\epsilon > 0$ such that $\rho(A) + \epsilon = \eta < 1$. From $\|A^n\|^{1/n} \to \rho(A)$, there exists N such that for all $n > N$, we have

$$\|A^n\|^{1/n} \leq \rho(A) + \epsilon = \eta,$$

or $\|A^n\| \leq \eta^n$. Thus, $\sum_{n=0}^{\infty} A^n$ converges absolutely and uniformly in any operator norm.

It is also easy to see that this series commutes with A because the finite part of the Neumann series commutes with A. Thus, we get

$$(I - A)\,(\sum_{n=0}^{\infty} A^n) = (\sum_{n=0}^{\infty} A^n)\,(I - A) = I.$$

□

Corollary 2.2. *The operator $R_\lambda(A) := (\lambda I - A)^{-1}$ is well-defined and analytic in $|\lambda| > \rho(A)$.*

Proof. This is because the series

$$\sum_{n=0}^{\infty} \left(\frac{A}{\lambda} \right)^n$$

converges absolutely and uniformly in any operator norm $\|\cdot\|$ and uniformly in λ for λ in the compact region in $|\lambda| > \rho(A)$. Since the finite sum is analytic in λ, so is their uniform limit. □

The operator $R_\lambda(A)$ is called the resolvent of A.

Example Suppose $J = \mu I_n + N_n$ be a Jordan matrix, find the exact formula of $(\lambda I_n - J)^{-1}$.

Definition 2.7. *Let f be a holomorphic function on $\mathbb{C}$ and $\mathbf{A}$ be an $n \times n$ matrix over $\mathbb{C}$. We define*

$$f(\mathbf{A}) := \int_C f(\lambda)(\lambda \mathbf{I} - \mathbf{A})^{-1}\, d\lambda$$

where C is any closed contour that winds once around $\sigma(\mathbf{A})$.

Theorem 2.13 (Spectral Mapping Theorem)**.** *Let f be a holomorphic function on $\mathbb{C}$ and $\mathbf{A}$ be an $n \times n$ matrix over $\mathbb{C}$. We have*

$$\sigma(f(\mathbf{A})) = f(\sigma(\mathbf{A})).$$

2.5 Direct Methods for Solving Linear Equations

2.5.1 LU Decomposition

Goal : Solv small size linear system

$$Ax = b.$$

Small size means that n is at most few hundreds.

Strategy

- Decompose $A = LU$ by Gaussian elimination method, where L is lower triangular matrix and U is a upper triangular matrix.
- Solve $LUx = b$ by solving

  $$Ly = b \quad Ux = y.$$

  These two equations can be solved by forward and backward substitution, respectively.

Procedure If the matrix is upper triangular, i.e. $a_{ij} = 0$ if $j < i$, then we can solve this equation by backward substitution:

Algorithm 1 Backward substitution

```
1: procedure BkSbstitut(n, A = (a_ij), b)
2:     for i = n : 1 do
3:         x_i ← (b_i − Σ_{j=i+1}^{n}) / a_ii
4:     end for
5: end procedure
```

If the matrix is lower triangular, i.e. $a_{ij} = 0$ if $j > i$, then we can solve this equation by forward substitution:

Algorithm 2 Forward substitution

1: **procedure** FWDSBSTITUT($n, A = (a_{ij}), b$)
2: **for** $i = 1 : n$ **do**
3: $x_i \leftarrow \left(b_i - \sum_{j=1}^{i-1}\right) / a_{ii}$
4: **end for**
5: **end procedure**

For general matrix A, we factorize it into the product of a lower triangular matrix L and an upper triangular matrix U:

$$A = LU,$$

called LU factorization. By direct calculation, we get

$$a_{ij} = \sum_{s=1}^{\min(i,j)} \ell_{is} u_{sj}.$$

The procedure to obtain L and U is by the Gaussian elimination method. It is an inductive procedure. At step k,

- we assume that we have computed rows $1, ..., k-1$ of U and columns $1, ..., k-1$ of L
- we want to update $u_{kj}, j \geq k$ and $\ell_{ik}, i \geq k$.
- From
$$a_{kk} = \sum_{s=1}^{k-1} \ell_{ks} u_{sk} + \ell_{kk} u_{kk},$$
we can determine ℓ_{kk} or u_{kk} if one of them is chosen. So there are three approaches:
  - choose $\ell_{kk} = 1$ for all k. Such L is a unit lower triangular matrix, the factorization is called Doolittle's factorization;
  - choose $u_{kk} = 1$ for all k. Such U is a unit upper triangular matrix, the factorization is called Crout's factorization;
  - For symmetric matrix, we can choose $\ell_{kk} = u_{kk}$ for all k. Such factorization for symmetric matrices is called the Cholesky factorization.

  Let us choose $\ell_k = 1$ here. With this, we determine u_{kk}.
- We proceed to compute u_{kj} for $j > k$ and ℓ_{ik} for $i > k$ as the follows.

$$a_{kj} = \sum_{s=1}^{k-1} \ell_{ks} u_{sj} + \ell_{kk} u_{kj} \quad (k+1 \leq j \leq n)$$
$$a_{ik} = \sum_{s=1}^{k-1} \ell_{is} u_{sk} + \ell_{ik} u_{kk} \quad (k+1 \leq i \leq n)$$

The corresponding pseudocode is

Algorithm 3 LU Decomposition

procedure GAUSSIANELIMINATION($n, A = (a_{ij}), b$)
  for $k = 1 : n$ **do**
    $\ell_{kk} = 1$
    $u_{kk} = a_{kk} - \sum_{s=1}^{k-1} \ell_{ks} u_{sk}$
    for $j = k + 1 : n$ **do**
      $u_{kj} \leftarrow \left(a_{kj} - \sum_{s=1}^{k-1} u_{sj}\right) / \ell_{kk}$
    end for
    for $i = k + 1 : n$ **do**
      $\ell_{ik} \leftarrow \left(a_{ik} - \sum_{s=1}^{k-1} \ell_{is} u_{sk}\right) / u_{kk}$
    end for
  end for
end procedure

With a LU factorization, the system $Ax = b$ can be solved by

$$Ly = b$$
$$Ux = y.$$

In practice, we can store L and U in matrix A. At step k, the matrix $A^{(k)}$ has the form

$$A^{(k)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & \cdots & \cdots & a_{1n}^{(1)} \\ \ell_{21} & a_{22}^{(2)} & & & & a_{2n}^{(2)} \\ \vdots & \ddots & \ddots & & & \vdots \\ \ell_{k1} & \cdots & \ell_{k,k-1} & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ \vdots & & \vdots & \vdots & & \vdots \\ \ell_{n1} & \cdots & \ell_{n,k-1} & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{pmatrix}$$

Algorithm 4 Gaussian Elimination

procedure GAUSSIANELIMINATION($n, A = (a_{ij}), b$)
  for $k = 1 : n - 1$ **do**
    for $i = k + 1 : n$ **do**
      $\ell_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$
      for $j = k + 1 : n$ **do**
        $a_{ij}^{(k+1)} = a_{ij}^{(k)} - \ell_{ik} a_{kj}^{(k)}$
      end for
      $b_i^{(k+1)} = b_i^{(k)} - \ell_{ik} b_k^{(k)}$
    end for
  end for
end procedure

Variants of LU Decomposition

- $A = LDU$, where L and D are unit lower/upper triangular matrices, D is a diagonal matrix.
- If A is symmetric, then we can factor A into $A = LL^T$. This is called Cholesky factorization.

Stability and Pivoting LU Decomposition It is possible that the LU factorization fails at some iteration k. In performing Gaussian elimination for the matrix $A^{(k)}$:

$$A^{(k)} = \begin{pmatrix} a_{kk}^{(k)} & \cdots & a_{km}^{(k)} \\ \vdots & & \vdots \\ a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{pmatrix}.$$

it is possible that $a_{kk}^{(k)}$ is zero, or very small. In this case, the Gaussian elimination is either fails or unstable. To avoid this, we can perform row permutation to move the largest $|a_{ik}^{(k)}|$, $i = k, ..., n$ to the kth row of the matrix $A^{(k)}$. Let us denote such row permutation by P_k. The factorization now becomes

$$PA = LU,$$

where $P = P_{n-1} \cdots P_1$ is the product of these row permutations. Such process involving only row permutation is called *partial pivoting*.

In the above pivoting process, it is also possible to find the largest $|a_{ij}^{(k)}|$ for $k \leq i, j \leq n$ then perform a row permutation P_k and a column permutation Q_k. Then the factorization becomes

$$PAQ = LU,$$

where $P = P_{n-1} \cdots P_1$ and $Q = Q_1 \cdots Q_{n-1}$. This is called *total pivoting*.

Matlab Commands

- $[LU] = lu(A)$ gives the LU decomposition with partial pivoting;
- $[L, U, P] = lu(A)$ gives the LU decomposition with partial pivoting;
- $[L, U, P, Q] = lu(A)$ gives the LU decomposition with total pivoting.

If A is symmetric, then Cholesky method is adopted:

- $R = chol(A)$ gives $A = R'R$;
- $L = chol(A,' lower')$ gives $A = LL'$.

Computational Complexity :

- It is in general $O(n^3)$ for full matrices.
- For banded matrices with band size b, the computational complexity is $O(b^2 n)$, provided there is no pivoting.
- Theoretically, if the matrix-matrix multiplication is $M(n)$, then the LU factorization is also $M(n)$. There are some fast algorithms for matrix-matrix multiplication:
  - Strassen algorithm: $O(n^{2.807355})$,
  - Coppersmith?Winograd algorithm: $O(n^{2.375477})$.

  The latter may be impractical because large constant.

2.5.2 *Other direct methods

- Cyclic Reduction Method: This is for tridiagonal matrix

$$A = \text{diag}(a_j, 1, c_j)$$

The jth equation is

$$a_j x_{j-1} + x_j + c_j x_{j+1} = b_j.$$

We will reduce to half size by eliminating the odd index terms. Let us write three consecutive equations

$$\begin{cases} a_{2j-1}x_{2j-2} & +x_{2j-1} & +c_{2j-1}x_{2j} & & & = b_{2j-1} \\ & a_{2j}x_{2j-1} & +x_{2j} & +c_{2j}x_{2j+1} & & = b_{2j} \\ & & a_{2j+1}x_{2j} & +x_{2j+1} & +c_{2j+1}x_{2j+2} & = b_{2j+1} \end{cases}$$

We can eliminate x_{2j-1} and x_{2j+1} and then obtain an equation only involves x_{2j-2}, x_{2j} and x_{2j+2}:

$$-a_{2j-1}a_{2j}x_{2j-2} + (1 - a_{2j}c_{2j-1} - a_{2j+1}c_{2j})x_{2j} - c_{2j}c_{2j+1}x_{2j+2} = b_{2j} - a_{2j}b_{2j-1} - c_{2j}b_{2j+1}.$$

The problem now can be reduced to half size:

$$a_j^{(1)} x_{j-1}^{(1)} + x_j^{(1)} + c_j^{(1)} = b_j^{(1)}.$$

Here, all original x_j, b_j, a_j, c_j will be denoted by $x_j^{(0)}$,etc. as an initialization. The new variables and coefficients are

$$a_j^{(1)} = \frac{-a_{2j-1}^{(0)} a_{2j}^{(0)}}{1 - a_{2j}^{(0)} c_{2j-1}^{(0)} - a_{2j+1}^{(0)} c_{2j}^{(0)}}, \quad c_j^{(1)} = \frac{-c_{2j}^{(0)} c_{2j+1}^{(0)}}{1 - a_{2j}^{(0)} c_{2j-1}^{(0)} - a_{2j+1}^{(0)} c_{2j}^{(0)}}$$

$$x_j^{(1)} = x_{2j}^{(0)}, \quad b_j^{(1)} = \frac{b_{2j}^{(0)} - a_{2j}^{(0)} b_{2j-1}^{(0)} - c_{2j}^{(0)} b_{2j+1}^{(0)}}{1 - a_{2j}^{(0)} c_{2j-1}^{(0)} - a_{2j+1}^{(0)} c_{2j}^{(0)}}.$$

We perform this process recursively. Suppose the number of unknowns is $2^L M$ at level 0. The solution $(x_j^{(k)})$ is called at level k. We perform the above reduction procedure for $k = 0$ to L. This is a $M \times M$ system, a small system, which can the solve it exactly. Once we have the solution at the coarsest level L, we can go backward to obtain solutions at finer grid. Indeed, suppose we have solutions at level $k+1$, that is $x_j^{(k+1)}$. These are also the solutions $x_{2j}^{(k)}$. Then the solution at odd grids at level k can be obtained from the odd equation:

$$a_{2j+1}^{(k)} x_{2j}^{(k)} + x_{2j+1}^{(k)} + c_{2j+1}^{(k)} x_{2j+2}^{(k)} = b_{2j+1}^{(k)}$$

which can be solved for $x_{2j+1}^{(k)}$ once $x_{2j}^{(k)}$ and $x_{2j+2}^{(k)}$ are obtained. But these two are obtained from the previous iteration steps.

The cyclic reduction method is very similar to multi grid method. If the matrix is diagonally dominant, then off-diagonal coefficients $a_j^{(k)}$, $c_j^{(k)}$, which changes during the level reduction, converges to zeros quadratically.

- Block Cyclic Reduction Method: this is particular useful for two dimension problems.

2.6 Classical Iterative Methods

The target problem we can have in mind is the discrete Poisson equation

$$-u_{j-1} + 2u_j - u_{j+1} = f_j, \quad j = 1, ..., N-1.$$

The boundary conditions are

$$u_0 = u_N = 0.$$

This can be written in matrix form

$$Ax = b,$$

where $A = \text{diag}(-1, 2, -1)$.

2.6.1 Splitting iterative methods

Problem Solve $Ax = b$. A is large size and usually sparse.

Ideas This class of iterative methods split A into

$$A = M - N,$$

where M and N satisfy

- M is the major part, M is easy to invert
- N is the minor part.

Then perform iteration:

$$Mx_{n+1} - Nx_n = b.$$

This is supposed to be solved easily because we assume M is easy to invert.

Splitting examples For instance, we can express $A = D + L + U$, where D is diagonal, L lower triangular, and U upper triangular. Then we perform the following splitting

- Jacobi method: choose $M = D$, $N = -L - U$;
- Gauss-Seidel: choose $M = D + L$, $N = -U$.

Theory The iteration can be rewritten as

$$x_{n+1} = M^{-1}Nx_n + M^{-1}b := Gx_n + M^{-1}b.$$

The matrix G is called amplification matrix.

Theorem 2.14. *The sequence $x_{n+1} = Gx_n + c$ converges if and only if $\rho(G) < 1$.*

Proof. 1. Subtracting $x_{n+1} = Gx_n + c$ and $x_n = Gx_{n-1} + c$, we get

$$x_{n+1} - x_n = G(x_n - x_{n-1}) = G^n(x_1 - x_0).$$

The convergence of the sequence $\{x_n\}$ is equivalent to the convergence of the series $\sum_n (x_{n+1} - x_n)$, which is also equivalent to that of $\sum_n G^n(x_1 - x_0)$.

2. If $\rho(G) < 1$, then, from Gelfand formula, we can choose an $\rho(G) < \eta < 1$ and there exists an N, such that for all $n \geq N$, we have

$$\|G^n\| \leq \eta^n$$

Thus, the series

$$\sum_n (x_{n+1} - x_n) = \sum_n G^n(x_1 - x_0)$$

converges absolutely.

3. Conversely, suppose $\rho(G) > 1$. Let λ_1 be the largest eigenvalue in magnitude and v_1 be the corresponding eigenvector. From $\rho(G) > 1$, we have $|\lambda_1| > 1$.We choose x_0 such that $x_1 - x_0 = v_1$. This means that $(I - G)x_0 + c = v_1$. As long as 1 is not an eigenvalue of G, this is possible. With this x_0, we see that

$$\sum_n (x_{n+1} - x_n) = \sum_n G^n(x_1 - x_0) = \sum_n G^n v_1 = \sum_n \lambda_1^n$$

is unbounded.

4. If 1 happens to be an eigenvalue with v being the corresponding eigenvector. We choose $x_0 = v$, we see that $x_n = v + nc$ is still unbounded.

□

Remark A sufficient condition for $\sum_n G^n(x_1 - x_0)$ converges is $\|G\| < 1$ for *some norm*. But this is not a necessary condition.

Definition 2.8. *A matrix A is called*

- *strictly diagonally dominant if*

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \quad \textit{for all } i = 1, ..., n.$$

- *irreducible diagonally dominant if A is diagonally dominant:*

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}| \quad \textit{for all } i = 1, ..., n,$$

  and A is irreducible, i.e. A cannot be similar via permutation to a block upper triangular matrix.

Theorem 2.15. *The Jacobi method or the Gauss-Seidel method converge if one of the following cases holds*

- *A is symmetric positive definite;*
- *A is strictly diagonally dominant;*
- *A is irreducible and diagonally dominant.*

The convergence rate is linear.

Proof. I shall only give the convergence proof for Jacobi method for strictly diagonally dominant matrices. For Jacobi method, $A = D + (L + U) = M - N$. The iteration algorithm is $Mx_{n+1} =$

$Nx_n + b$. We get $x_{n+1} = Gx_n + M^{-1}b$ with $G = M^{-1}N$. When A is strictly row diagonally dominant, we use operator sup norm $\|G\|_\infty =$, which is

$$\|G\|_\infty = \max_i \sum_j |g_{ij}| = \max_i \sum_{j \neq i} \left| \frac{a_{ij}}{a_{ii}} \right| = \eta < 1$$

Thus, the series

$$\sum_n (x_{n+1} - x_n)$$

converges absolutely in $|\cdot|_\infty$. This leads to x_n converges. Suppose x^* is its limit. Then $x^* = Gx^* + M^{-1}b$. Subtracting this from $x_{n+1} = Gx_n + M^{-1}b$, we obtain

$$e_{n+1} = Ge_n,$$

where $e_n := x_n - x^*$. Since $\|G\|_\infty = \eta < 1$, we get

$$|e_n|_\infty \leq \|G^n\|_\infty |e_0| \leq \eta^n |e_0|_\infty \to 0.$$

This shows that the convergence is linear. □

Acceleration techniques: Richardson Extrapolation In the late 50's, people think that one can accelerate the convergent rate by performing Richardson extrapolation techniques. Let us take Jacobi method as an example.

1. Damped Jacobi method. Suppose we use Jacobi method to produce x_{n+1} from x_n. We can extrapolate it to $\hat{x}_{n+1}$ by
$$\hat{x}_{n+1} = (1 - \omega)x_n + \omega x_{n+1},$$
where ω will be properly chosen. Usually it will be larger than 1 for extrapolation. Now, suppose x_{n+1} is produced by Jacobi method. Then
$$\hat{x}_{n+1} = (1 - \omega)x_n + \omega D^{-1}(-(L + U)x_n + b).$$
We drop hat in $\hat{x}_{n+1}$. The resulting scheme is:
$$x_{n+1} = (1 - \omega)x_n + \omega D^{-1}(-(L + U)x_n + b) = x_n + \omega D^{-1}(b - Ax_n).$$
This is called damped Jacobi method.

2. Successive over relaxation method (SOR). Suppose x_{n+1} is produced by Gauss-Seidel method. We extrapolate it from x_n, x_{n+1} to a new $\hat{x}_{n+1}$ by
$$\hat{x}_{n+1} = (1 - \omega)x_n + \omega x_{n+1}.$$
In this case,
$$\hat{x}_{n+1} = (1 - \omega)x_n + \omega(D + L)^{-1}(-Ux_n + b) = x_n + \omega(D + L)^{-1}(b - Ax_n).$$

3. Symmetric Successive Over Relaxation (SSOR). For symmetric matrix, the above amplification matrix G is not symmetric. But, we can perform Gauss-Seidel method twice in one iteration, one use lower triangular matrix L, the other uses the upper triangular matrix. In this procedure, we can maintain symmetric of the amplification matrix. So, the scheme reads

$$\begin{aligned} x_{n+1/2} &= x_n + \omega(D+L)^{-1}(b - Ax_n) \\ x_{n+1} &= x_{n+1/2} + \omega(D+U)^{-1}(b - Ax_{n+1/2}). \end{aligned}$$

$$x_{n+1} = x_n + \omega\left[(D+L)^{-1} + (D+U)^{-1} - \omega(D+U)^{-1}A(D+L)^{-1}\right](b - Ax_n).$$

Note that $U = L^T$ and the amplification matrix

$$G = \omega\left[(D+L)^{-1} + (D+U)^{-1} - \omega(D+U)^{-1}A(D+L)^{-1}\right]A$$

is symmetric.

The goal is to find proper ω which minimize $\rho(G(\omega))$. It depends on specific A. For discrete Laplacian on box, one can compute the spectrum explicitly, then obtain an optimal ω. For symmetric positive definite matrix, we can also do similar things.

Remarks.

1. SOR in the standard textbook is not expressed in the form above. It is derived and expressed as below. Originally, the Gauss-Seidel can be written as

$$x_{n+1} = D^{-1}(b - Lx_{n+1} - Ux_n).$$

Hence, one can design SOR as

$$x_{n+1} = (1-\omega)x_n + \omega D^{-1}(b - Lx_{n+1} - Ux_n).$$

From this, we obtain

$$Dx_{n+1} = (1-\omega)Dx_n + \omega(b - Lx_{n+1} - Ux_n)$$

$$(D + \omega L)x_{n+1} = ((1-\omega)D - \omega U)x_n + \omega b.$$

Thus, we split $A = M - N$,

$$M = D + \omega L, \quad N = (1-\omega)D - \omega U.$$

Or we can express it as

$$x_{n+1} = x_n + \omega(D + \omega L)^{-1}(b - Ax_n).$$

2. SSOR: We perform two SORs:

$$\begin{aligned} x_{n+1/2} &= x_n + \omega(D+\omega L)^{-1}(b - Ax_n) \\ x_{n+1} &= x_{n+1/2} + \omega(D+\omega U)^{-1}(b - Ax_{n+1/2}). \end{aligned}$$

This gives

$$\begin{aligned} x_{n+1} &= x_n + \omega\left[(D+\omega L)^{-1} + (D+\omega U)^{-1} - \omega(D+\omega U)^{-1}A(D+\omega L)^{-1}\right](b - Ax_n) \\ &= x_n + \omega P^{-1}(b - Ax_n) \end{aligned}$$

where

$$\begin{aligned} P^{-1} &= (D+\omega L)^{-1} + (D+\omega U)^{-1} - \omega(D+\omega U)^{-1}A(D+\omega L)^{-1} \\ &= (D+\omega U)^{-1}\left[D+\omega L - \omega A + D + \omega U\right](D+\omega L)^{-1} \\ &= (2-\omega)(D+\omega U)^{-1}D(D+\omega L)^{-1}. \end{aligned}$$

2.6.2 Preconditioned iterative methods

To solve

$$Ax - b = 0,$$

we shall solve the equation

$$P^{-1}(Ax - b) = 0$$

instead, where P is called a preconditioner, which is designed to satisfy

- P^{-1} is easy to compute,
- P^{-1} is an approximation of A^{-1} in the sense that $P^{-1}A$ has smaller condition number of that A has.

With a preconditioner P, we can design a fixed point method as

$$\boxed{x_{n+1} = x_n + \omega_n P^{-1}(b - Ax_n).}$$

One can see all classical iterative methods can be expressed as this preconditioned iterative method.

- Jacobi method: $P = D, \omega = 1$
- Damped Jacobi method: $P = D, \omega \in (0, 2)$,
- Gauss-Seidel: $P = D + L, \omega = 1$,
- SOR: $P = D + \omega L$
- SSOR: $P = \frac{1}{2-\omega}(D+\omega U)D^{-1}(D+\omega L)$.

Homework

1. Discretize the one-dimension Poisson equation $-u'' = f$ by central finite difference method. Solve the resulting linear system

$$\text{diag}(-1, 2, -1)u = f$$

by above classical iterative methods. Choose proper ω, Compare them.

2.6.3 Conjugate Gradient Method

Goal: Solve $Ax = b$, A is symmetric positive definite and $b \neq 0$.

Ideas and derivation: Solve the problem successively in the space spanned by $\{b, Ab, ..., A^{k-}b\}$. The reason is the follows. Suppose p is the minimal polynomial of A, that is $p(A) = 0$ and $deg(A) \leq n$. If A is invertible, then $p(0) \neq 0$. Otherwise 0 would be an eigenvalue. We may normalize p such that $p(0) = -1$. Thus

$$0 = p(A) = -I + A\,q(A).$$

This shows that $A^{-1} = q(A)$ with $deg(q) \leq n - 1$. Thus,

$$x = A^{-1}b = q(A)b.$$

To derive an iterative method, the above observation suggests us to solve this equation iteratively in the spaces

$$V_0 \subset V_1 \subset \cdots \subset V_n = \mathbb{R}^n.$$

where

$$V_k =< b, Ab, ..., A^{k-1}b >,$$

the space spanned by $\{b, Ab, ..., A^{k-1}b\}$, called the Krylov spaces. The details of the derivation are the follows.

1. The problem can be written in variation form:

$$\min \phi(x) := \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle.$$

The minimum occurs at

$$\nabla\phi(x) = Ax - b = 0.$$

2. Let us look for optimal solution in V_k. Let us call it

$$x_k = \text{arg min } \{\phi(x)|x \in V_k\}.$$

3. We start from $x_0 = 0$. Let us define the residual

$$r_0 := b - Ax_0,$$

which is not zero. We define $V_1 =< r_0 >=< b >$. We look for optimal solution in V_1. The search direction is called p_1, which is $p_1 = b$. An element in V_1 can be expressed as $x = \alpha_1 p_1$. Plug it into $\phi(x)$, we get

$$\phi(x) = \frac{\alpha^2}{2}\langle Ap_1, p_1\rangle - \langle b, \alpha p_1\rangle.$$

The optimal solution is

$$x_1 = \alpha_1 p_1, \quad \alpha_1 = \frac{\langle b, p_1\rangle}{\langle Ap_1, p_1\rangle}.$$

4. The residual $r_1 := b - Ax_1$. Suppose $r_1 \neq 0$. We find

$$r_1 \in V_2 :=< b, Ab > .$$

Hence,

$$V_2 =< r_0, r_1 > .$$

Furthermore, the residual $r_1 = -\nabla\phi(x_1)$, it is orthogonal to V_1 because x_1 is the optimal solution of ϕ in V_1. Thus,

$$r_1 \perp r_0.$$

5. Now, suppose we have found an optimal solution $x_k \in V_k$ and suppose the residual

$$r_k := b - Ax_k \neq 0.$$

We extend V_k by adding r_k. Then

$$V_{k+1} = V_k + < r_k >=< b, Ab, ..., A^k b > .$$

For any $x \in V_{k+1}$, we express it as

$$x = y + \alpha_{k+1} p_{k+1}$$

where $y \in V_k$, $p_{k+1} \in V_{k+1} \setminus V_k$ is the search direction to be determined later. We plug it into ϕ

$$\phi(x) = \phi(y) + \langle y, Ap_{k+1}\rangle + \frac{\alpha^2}{2}\langle p_{k+1}, Ap_{k+1}\rangle - \alpha\langle p_{k+1}, b\rangle.$$

If $\langle y, Ap_{k+1}\rangle = 0$, then the above minimization in V_{k+1} is separable:

$$\min_{x\in V_{k+1}} \phi(x) = \min_{y\in V_k} \phi(y) + \min_{\alpha\in\mathbb{R}} \frac{\alpha^2}{2}\langle p_{k+1}, Ap_{k+1}\rangle - \alpha\langle p_{k+1}, b\rangle.$$

This gives

$$x_{k+1} = x_k + \alpha_{k+1} p_{k+1}$$

with

$$p_{k+1} \perp_A V_k, \quad \alpha_{k+1} = \frac{\langle b, p_{k+1}\rangle}{\langle Ap_{k+1}, p_{k+1}\rangle}.$$

6. The search direction p_k is chosen to be

$$p_{k+1} = r_k + \beta_{k+1} p_k.$$

Since it is required

$$\langle Ap_{k+1}, p_k \rangle = 0.$$

This gives

$$\beta_{k+1} = -\frac{\langle Ap_k, r_k \rangle}{\langle Ap_k, p_k \rangle}.$$

7. Residual: The residual of x_k is defined as

$$r_{k+1} = b - Ax_{k+1}.$$

8. If $r_{k+1} = 0$, then we are done. If not, we repeat the above procedure. This will continue at most to $k = n$, where the search space V_n is the whole space $\mathbb{R}^n$.

Theory Below, we assume A is symmetric positive definite $n \times n$ matrix. The matrix A defines an inner product

$$\langle x, y \rangle_A := \langle Ax, y \rangle.$$

Theorem 2.16. *Given $r_0 \neq 0$. Let $V_k = [r_0, Ar_0, ..., A^{k-1}r_0]$ be the Krylov spaces. Set $p_0 = 0$. For $k = 0, ..., n-1$, define*

$$p_{k+1} = r_k + \beta_{k+1} p_k, \quad \beta_{k+1} = -\frac{\langle Ap_k, r_k \rangle}{\langle Ap_k, p_k \rangle}$$

$$r_{k+1} = r_k - \alpha_{k+1} Ap_{k+1}, \quad \alpha_{k+1} = \frac{\langle r_k, p_{k+1} \rangle}{\langle Ap_{k+1}, p_{k+1} \rangle},$$

then,

1. *either $r_{k+1} = 0$, or*
2. *(i) $V_{k+1} = [r_0, r_1, ..., r_k] = [p_1, p_2, ..., p_{k+1}]$, $dim V_{k+1} = k+1$.*
   (ii) $\{p_1, ..., p_{k+1}\}$ are A-orthogonal,
   (iii) $r_{k+1} \perp V_{k+1}$.

Proof. 1. We prove (i), (ii), (iii) by induction.
For $k = 1$, since $p_1 = r_0$, it is clear that $V_1 = [r_0] = [p_1]$ and $dim V_1 = 1$. Moreover,

$$\langle r_1, p_1 \rangle = \langle r_0 - \alpha_1 Ap_1 \rangle = 0 \text{ because } \alpha_1 = \frac{\langle r_0, p_1 \rangle}{\langle Ap_1, p_1 \rangle}.$$

2. Suppose (i) (ii) (iii) are true for k, that is,

(i) $V_k = [r_0, r_1, ..., r_{k-1}] = [p_1, p_2, ..., p_k]$, $dim V_k = k$.

(ii) $\{p_1, ..., p_k\}$ are A-orthogonal,

(iii) $r_k \perp V_k$,

we want to show they are also true for $k+1$.

3. To show (i), we show that $r_k \in V_{k+1}$ and $r_k \notin V_k$. These two together give $V_{k+1} = V_k + < r_k >$ and $dim V_{k+1} = k+1$. The reason for $r_k \in V_{k+1}$ is due to

$$r_k = r_{k-1} - \alpha_k A p_k \in V_k + AV_k = V_{k+1}.$$

The reason why $r_k \notin V_k$ is due to the induction hypothesis $r_k \perp V_k$, unless $r_k = 0$, which is also contradicts our assumption.

4. We show that $V_{k+1} = V_k + < p_{k+1} >$. We have $p_{k+1} = r_k + \beta_{k+1} p_k \in V_{k+1}$. Since $r_k \in V_{k+1} \setminus V_k$ and $p_k \in V_k$, we get that $p_{k+1} \in V_{k+1} \setminus V_k$. These two show $V_{k+1} = V_k + < p_{k+1} >$.

5. We show $\langle Ap_{k+1}, p_k \rangle = 0$. From $p_{k+1} = r_k + \beta_{k+1} p_k$, we see that β_{k+1} is chosen so that $\langle Ap_{k+1}, p_k \rangle = 0$.

6. We show that $\langle Ap_{k+1}, p_\ell \rangle = 0$ for $\ell = 1, ..., k-1$. We have

$$\begin{aligned} \langle Ap_{k+1}, p_\ell \rangle &= \langle r_k + \beta_{k+1} p_k, Ap_\ell \rangle \\ &= \langle r_k, Ap_\ell \rangle + \beta_{k+1} \langle p_k, Ap_\ell \rangle = 0. \end{aligned}$$

For $\langle r_k, Ap_\ell \rangle$, we have used (a) $Ap_\ell \in V_k$ if $\ell < k$, (b) the induction hypothesis $r_k \perp V_k$. For $\langle p_k, Ap_\ell \rangle = 0$ for $\ell = 1, ..., k-1$, this is the induction hypothesis.

7. We show $\langle r_{k+1}, p_{k+1} \rangle = 0$. We have

$$r_{k+1} = r_k - \alpha_{k+1} A p_{k+1}.$$

The coefficient α_{k+1} is chosen so that

$$\langle r_{k+1}, p_{k+1} \rangle = \langle r_k - \alpha_{k+1} A p_{k+1}, p_{k+1} \rangle = 0.$$

That is, r_{k+1} is obtained by removing the errors of r_k in the direction of p_{k+1}.

8. We show $\langle r_{k+1}, p_\ell \rangle = 0$ for $\ell \leq k$. We have

$$\langle r_{k+1}, p_\ell \rangle = \langle r_k - \alpha_{k+1} A p_{k+1}, p_\ell \rangle = 0.$$

Here, we have used the induction hypothesis $r_k \perp V_k$ and $\{p_1, ..., p_k\}$ are A-orthogonal.

□

Remarks

1. In conjugate gradient method, if we start from any x_0 and define

$$x_k = x_{k-1} + \alpha_k p_k, \quad r_k = b - Ax_k.$$

then $r_n = 0$. This means that the exact solution can always achieved in n step iterations.

2. We can avoid some matrix-vector multiplication in CG method as shown below. We claim that

$$\alpha_k := \frac{\langle r_{k-1}, p_k\rangle}{\langle Ap_k, p_k\rangle} = \frac{\langle r_{k-1}, r_{k-1}\rangle}{\langle Ap_k, p_k\rangle}, \tag{2.1}$$

$$\beta_{k+1} := -\frac{\langle Ap_k, r_k\rangle}{\langle Ap_k, p_k\rangle} = \frac{\langle r_k, r_k\rangle}{\langle r_{k-1}, r_{k-1}\rangle}. \tag{2.2}$$

This means that we only need to evaluate matrix-vector multiplication Ap_k once in each iteration. To show (2.1), we use $p_k = r_{k-1} + \beta_k p_{k-1}$,

$$\langle r_{k-1}, p_k\rangle = \langle r_{k-1}, r_{k-1} + \beta_k p_{k-1}\rangle = \langle r_{k-1}, r_{k-1}\rangle.$$

To show (2.2), from $r_k = r_{k-1} - \alpha_k Ap_k$, we have

$$\langle r_k, r_k\rangle = \langle r_{k-1} - \alpha_k Ap_k, r_k\rangle = -\alpha_k\langle Ap_k, r_k\rangle.$$

Hence,

$$\langle Ap_k, r_k\rangle = -\frac{1}{\alpha_k\langle r_k, r_k\rangle} = -\frac{\langle Ap_k, p_k\rangle}{\langle r_{k-1}, p_k\rangle}\langle r_k, r_k\rangle.$$

Thus,

$$\beta_{k+1} = -\frac{\langle Ap_k, r_k\rangle}{\langle Ap_k, p_k\rangle} = \frac{\langle r_k, r_k\rangle}{\langle r_{k-1}, p_k\rangle} = \frac{\langle r_k, r_k\rangle}{\langle r_{k-1}, r_{k-1}\rangle}.$$

Algorithm 5 Conjugate Gradient Algorithm

procedure CG($n, A = (a_{ij}), b$)
  $r_0 = b, p = b$
  $\alpha = \frac{\langle b,p\rangle}{\langle Ap,p\rangle}$
  $x = \alpha p$
  $r_1 = b - Ax$
  $C0 = \langle r_0, r_0\rangle$
  $C1 = \langle r_1, r_1\rangle$
  while $|r_1| \geq Tol$ **do**
    $\beta = -\frac{C1}{C0}$
    $p = r_1 + \beta p$
    $\alpha = \frac{C1}{\langle Ap,p\rangle}$
    $x = x + \alpha p$
    $r_0 = r_1$
    $r_1 = b - Ax$
    $C0 = C1$
    $C1 = \langle r_1, r_1\rangle$
  end while
end procedure

Algorithm

Theorem 2.17. *Suppose A is symmetric positive definite. Then the conjugate gradient method converges and has the following estimate*

$$\|x^* - x_k\|_A \leq 2\left[\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}\right]^k \|x^* - x_0\|_A.$$

Proof. 1. x_k is the best solution of $Ax - b = 0$ in the space V_k. This means that

$$0 = \langle r_k, v\rangle = \langle b - Ax_k, v\rangle = \langle Ax^* - Ax_k, v\rangle \text{ for all } v \in V_k.$$

That is,

$$(x^* - x_k) \perp_A V_k.$$

This implies

$$\langle A(x^* - x_k), x^* - x_k\rangle \leq \langle A(x^* - v), (x^* - v)\rangle \text{ for all } v \in V_k.$$

2. Now, we choose $v = p_{k-1}(A)b = p_{k-1}(A)Ax^*$.

$$\begin{aligned}\langle A(x^* - x_k), x^* - x_k\rangle &\leq \min_{p_{k-1}} \langle A(I - p_{k-1}(A)A)x^*, (I - p_{k-1}(A)A)x^*\rangle \\ &= \min_{q_k(0)=1} \langle Aq_k(A)x^*, q_k(A)x^*\rangle \\ &= \min_{q_k(0)=1} \max_{\lambda\in\sigma(A)} |q_k(\lambda)|^2 \langle Ax^*, x^*\rangle \\ &\leq \min_{q_k(0)=1} \max_{\lambda\in[a,b]} |q_k(\lambda)|^2 \langle Ax^*, x^*\rangle\end{aligned}$$

where

$$a = \lambda_{min}(A), \quad b = \lambda_{max}(A).$$

3. We choose

$$q_k(\lambda) = \frac{T_k\left(\frac{b+a-2\lambda}{b-a}\right)}{T_k\left(\frac{b+a}{b-a}\right)},$$

where

$$T_k(t) = \begin{cases} \cos\left(k\cos^{-1}t\right) & \text{if } |t| < 1 \\ \cosh\left(k\cosh^{-1}t\right) & \text{if } |t| \geq 1. \end{cases}$$

Then

$$q(0) = 1$$

and

$$\left|T_k\left(\frac{b+a-2\lambda}{b-a}\right)\right| \leq 1 \text{ for all } \lambda \in [a,b].$$

Thus,

$$\max_{\lambda\in[a,b]} |q_k(\lambda)| \leq \left[T_k\left(\frac{b+a}{b-a}\right)\right]^{-1}.$$

4. We set

$$\frac{b+a}{b-a} = \cosh\sigma = \frac{e^\sigma + e^{-\sigma}}{2}.$$

This implies

$$e^\sigma = \frac{\sqrt{\kappa(A)}+1}{\sqrt{\kappa(A)}-1}, \quad \kappa(A) = \frac{b}{a}.$$

We get

$$\begin{aligned}\cosh(k\sigma) = \frac{e^{k\sigma} + e^{-k\sigma}}{2} &\geq \frac{1}{2}e^{k\sigma} \\ &= \frac{1}{2}\left[\frac{\sqrt{\kappa(A)}+1}{\sqrt{\kappa(A)}-1}\right]^k\end{aligned}$$

Hence, we get

$$\min_{q_k(0)=1} \max_{\lambda\in[a,b]} |q_k(\lambda)| \leq 2\left[\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right]^k.$$

□

Preconditioned Conjugate Gradient Method In the CG method, the convergent rate depends on the condition number. The convergence performs faster if we have a smaller condition. This is why the preconditioned CG method (PCG) is favored.

Proposition 1. *Suppose A is symmetric positive definite. Suppose P is a symmetric pre-conditioner of A. Then PA is symmetric positive definite with respective to the norm*

$$[x, y] := \langle P^{-1}x, y\rangle.$$

That is,

$$[PAx, y] = [x, PAy] \geq 0.$$

Proof.

$$\begin{aligned}[PAx, y] &= \langle P^{-1}PAx, y\rangle = \langle Ax, y\rangle = \langle x, Ay\rangle \\ &= \langle PP^{-1}x, Ay\rangle = \langle P^{-1}x, PAy\rangle = [x, PAy].\end{aligned}$$

□

Now, we solve

$$PAx = Pb.$$

The PCG is as the follows:

1. $x_0 = 0$, $r_0 = b$, $p_1 = Pr_0$.
2. $\alpha_1 = \langle Pr_0, r_0\rangle/\langle Ap_+1, p_1\rangle$, $x_1 = \alpha_1 p_1$, $r_1 = b - Ax_1$.
3. For $k = 1, ...,$ the residue is $Pr_k := Pb - PAx_k$,

$$\begin{aligned}\beta_{k+1} &= \frac{[Pr_k, Pr_k]}{[Pr_{k-1}, Pr_{k-1}]} = \frac{\langle Pr_k, r_k\rangle}{\langle Pr_{k-1}, r_{k-1}\rangle} \\ p_{k+1} &= Br_k + \beta_{k+1}p_k. \\ \alpha_{k+1} &= \frac{[Pr_k, Pr_k]}{[PAp_{k+1}, p_{k+1}]} = \frac{\langle P^{-1}Pr_k, Pr_k\rangle}{\langle P^{-1}PAp_{k+1}, p_{k+1}\rangle} = \frac{\langle Pr_k, r_k\rangle}{\langle Ap_{k+1}, p_{k+1}\rangle} \\ x_{k+1} &= x_k + \alpha_{k+1}p_{k+1} \\ r_{k+1} &= b - Ax_{k+1}\end{aligned}$$

2.7 Power Method for Finding Eigenvalues

Goal Find the largest eigenvalue in magnitude of A.

Algorithm Let $\lambda_i(A)$ and v_i be its eigenvalues / eigenvectors. Suppose

$$|\lambda_1(A)| > |\lambda_2(A)| \geq |\lambda_3(A)| \geq \cdots$$

and suppose x_0 has nonzero component in the direction v_1. Then

$$x_{k+1} = \frac{Ax_k}{\|Ax_k\|}$$

has a subsequence converges to v_1, and

$$\mu_k = \frac{\langle x_k, Ax_k \rangle}{\langle x_k, x_k \rangle}$$

converges to $\lambda_1(A)$ with geometric rate $|\lambda_2/\lambda_1|$.

Proof. 1. A can be expressed in Jordan form:

$$A = VJV{-}1$$

where $V = [v_1, ..., v_n]$ are the generalized eigenvectors. We can express

$$x_0 = \sum_j c_j v_j = Vc, \quad c = [c_1, ..., c_n]^T.$$

By our assumption, $c_1 \neq 0$.

2. One can prove by induction that

$$x_k = \frac{A^k x_0}{\|A^k x_0\|}.$$

From this, and $A^k = VJ^kV{-}1$, we obtain

$$\begin{aligned} x_k &= \frac{VJ^kV^{-1}x_0}{\|VJ^kV^{-1}x_0\|} = \frac{VJ^kV^{-1}Vc}{\|VJ^kV^{-1}Vc\|} = \frac{VJ^kc}{\|VJ^kc\|} \\ &= \left(\frac{\lambda_1}{|\lambda_1|}\right)^k \frac{c_1v_1 + V(J/\lambda_1)^k(\sum_{j>1} c_j e_j)}{\|c_1v_1 + V(J/\lambda_1)^k(\sum_{j>1} c_j e_j)\|} \\ &= e^{ik\phi}\frac{c_1}{|c_1|}v_1 + r_k \end{aligned}$$

where

$$e^{i\phi} = \frac{\lambda_1}{|\lambda_1|}, \quad r_k = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right).$$

3. We see that the sequence is bounded and thus has a convergent subsequence. Indeed, as k large enough, x_k is closed to v_1 up to a scalar.

4. The Rayleigh quotient is

$$\begin{aligned}\frac{\langle Ax_k, x_k\rangle}{\langle x_k, x_k\rangle} &= \frac{\langle \lambda_1 e^{ik\phi}\frac{c_1}{|c_1|}v_1 + Ar_k, e^{ik\phi}\frac{c_1}{|c_1|}v_1 + r_k\rangle}{\langle e^{ik\phi}\frac{c_1}{|c_1|}v_1 + r_k, e^{ik\phi}\frac{c_1}{|c_1|}v_1 + r_k\rangle} \\ &= \lambda_1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right).\end{aligned}$$

□

Algorithm 6 Power Method

```
1: procedure POWERMTHD(n, A = (a_ij), μ, x, Iter)
2:     x = rand(n)
3:     x ← x/‖x‖
4:     for i=1:Iter do
5:         y = Ax
6:         μ = ⟨y, x⟩
7:         x ← y/‖y‖
8:     end for
9: end procedure
```

Algorithm

Remark

1. One can also use $\|x\|_\infty$ for normalization.

2. The convergence of μ_k is linear. One can use Aitken's acceleration technique to speed up convergence of μ_k. But we cannot speed up x_k. We have

$$\mu_k = \frac{\langle x_k, Ax_k\rangle}{\langle x_k, x_k\rangle} \approx \lambda_1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$$

We use μ_k, μ_{k-1}, μ_{k-2} to find a better approximation of μ_k:

$$\frac{\mu_{k-1} - \hat{\mu}}{\mu_{k-2} - \hat{\mu}} \approx \frac{\mu_k - \hat{\mu}}{\mu_{k-1} - \hat{\mu}}$$

This leads to

$$\hat{\mu} \approx \mu_{k-2} - \frac{(\mu_{k-1} - \mu_{k-2})^2}{\mu_k - 2\mu_{k-1} + \mu_{k-2}}$$

2.7.1 Inverse Power Method

Goal Find the eigenvalue λ_k of A which is close to a prescribed number q.

Idea The eigenvalues of

$$(A - qI)^{-1}$$

are

$$\frac{1}{\lambda_1 - q}, \cdots, \frac{1}{\lambda_n - q}.$$

We can apply power method to $(A - qI)^{-1}$ to find λ_k.

How to locate eigenvalues

Theorem 2.18 (Gershgorin's Theorem)**.** *Let $A = (a_{ij})_{n\times n}$. Then*

$$\sigma(A) \subset \bigcup_{i=1}^{n} \left\{ z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \right\}$$

Proof. Suppose λ and x be an eigen pair. We can normalize x such that $\|x\|_\infty = 1$. Suppose the maximum component is $|x_i| = 1$. We have

$$\sum_{j=1}^{n} a_{ij}x_j = \lambda x_i.$$

$$(\lambda - a_{ii})x_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_j.$$

Thus,

$$|\lambda - a_{ii}| \leq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|$$

□

Remark. We can also apply this proof to the left eigenvector and obtain

$$\sigma(A) \subset \bigcup_{i=1}^{n} \left\{ z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ji}| \right\}$$

Thus, $\sigma(A)$ is contained in the intersection of the row Gershgorin disks and column Gershgorin disks.

Chapter 3

Approximation Theory

3.1 Motivations

Data and signal representation In experiments, we collect data, which are usually discrete. We want to use a function to connect them. This can be in one dimension such as planetary orbits, asset values in market, in two dimensions, such as images, or in three dimensions, such as video, or molecular energy plots in chemistry, or in general, just a data cloud in some high dimensions.

Numerical approximation to functions, partial differential equations In numerical partial differential equations (PDEs), we approximate our solutions by splines, nodal functions, Fourier modes, etc. in order to project the equations to finite dimensions to solve. All of these are *to represent our objects in terms of some known atoms.*

- The objects can be signals, images, solutions of PDEs, or in general, a functions, or a unordered data.
- The atoms can be polynomials, splines, Fourier modes, wavelets, some special functions, or even object-dependent atoms.

The classical approximation deals with approximation of smooth functions by polynomials, splines, Fourier functions, wavelets. It can be used for numerical differentiation, integration, solving PDE problems, etc.

3.1.1 Basic Notion of function spaces

Normed linear spaces Let $\mathcal{X}$ be a vector space over $\mathbb{R}$ (or $\mathbb{C}$). A norm $\|\cdot\|$ is a function maps $\mathcal{X}$ to $\mathbb{R}$ which satisfies

1. $\|x\| \geq 0$ for all $x \in \mathcal{X}$, and $\|x\| = 0$ if and only if $x = 0$;
2. $\|\alpha x\| = |\alpha|\|x\|$;
3. $\|x + y\| \leq \|x\| + \|y\|$.

A vector space $\mathcal{X}$ endowed with a norm $\|\cdot\|$ is called a normed linear space. We are interested in those function spaces.

Examples of function spaces

- Space of continuous functions:

$$C[a,b] = \{u : [a,b] \to \mathbb{R} \text{ is continuous.}\}, \quad \|u\|_\infty := \sup_x |u(x)|.$$

- Space of continuous differentiable functions

$$C^m[a,b] = \{u : [a,b] \to \mathbb{R} | u, u', ..., u^{(m)} \text{ are continuous on } [a,b]\}.$$

  The norm is

$$\|u\|_{m,\infty} = \sum_{i=0}^{m} \|u^{(i)}\|_\infty.$$

- It is also common to use L^p norm in these spaces. The L^p norm is defined as

$$\|u\|_p = \left(\int_a^b |u(x)|^p \, dx \right)^{1/p}, \quad 1 \le p < \infty.$$

  Similarly, in $C^m[a,b]$, we can define

$$\|u\|_{m,p} = \sum_{i=0}^{m} \|u^{(i)}\|_p$$

Limiting processing in normed linear spaces

- A sequence $\{x_n\}$ in $\mathcal{X}$ is called a Cauchy sequence if for any $\epsilon > 0$ there exists an N such that for any $n, m > N$, $\|x_n - x_m\| < \epsilon$.
- A sequence $\{x_n\}$ is called convergence if there exists an $x \in \mathcal{X}$ such that for any $\epsilon > 0$ there exists an N such that for any $n \ge N$, $|x_n - x\| < \epsilon$.
- A normed linear space is called complete if all its Cauchy sequence $\{x_n\}$ has a limit x in $\mathcal{X}$.
- A complete normed linear space is called a Banach space.
- Theorem: Given a normed linear space $\mathcal{X}$, there exists an extension space $\overline{\mathcal{X}}$ such that
  - (i) $\mathcal{X} \subset \overline{\mathcal{X}}$,
  - (ii) $\|\cdot\|$ can also be extended to $\overline{\mathcal{X}}$,
  - (iii) $\overline{\mathcal{X}}$ is complete,
  - (iv) $\overline{\mathcal{X}}$ is the smallest such kind space.
- The space $(C[a,b], \|\cdot\|_p)$ is not complete. Its completion is called L^p space, denoted by $L^p(a,b)$.
- The completion of $(C^m(a,b), \|\cdot\|_{m,p})$ is called Sobolev spaces, denoted by $W^{m,p}(a,b)$.

L^p functions Below, we want to give examples and characterization of L^p functions without having background on measure theory. However, we do need the concept of measure 0 set.

- The function $1/|x|^\alpha \in L^p(-1,1)$ if and only if $-\alpha p + 1 > 0$. This is because the improper integral
$$\int_{-1}^{1} \left(\frac{1}{|x|^\alpha}\right)^p dx = |x|^{-\alpha p+1}\Big|_{-1}^{1} < \infty \quad \Leftrightarrow \quad -\alpha p + 1 > 0.$$
- The function $1/|x|^\alpha \in L^p(1,\infty)$ if and only if $-\alpha p + 1 < 0$. The improper integral now is
$$\int_{1}^{\infty} \left(\frac{1}{|x|^\alpha}\right)^p dx = |x|^{-\alpha p+1}\Big|_{1}^{\infty} < \infty \quad \Leftrightarrow \quad -\alpha p + 1 < 0.$$

Two L^p functions f and g are identical in L^p sense if they differ only on a measure zero set.

Measure 0 sets A set $S \subset \mathbb{R}$ is measure 0 if for any $\epsilon > 0$, there exists a sequence of intervals I_n such that $S \subset \bigcup_n I_n$ and $\sum_n |I_n| < \epsilon$.

- Countable union of measure zero sets is measure 0.
- $\mathbb{Q}$ is a measure zero set.
- The Cantor set is a measure zero set.

3.2 Approximation by polynomials: Interpolation Theory

In this section, we approximate a function by polynomial through interpolation at some prescribed nodes. We are concerned with the approximation in $C[a,b]$. An important example is the Runge phenomenon, which shows that Chebeshev nodes are better over the uniformly distributed nodes on an interval.

Goal Given $x_0, x_1, ..., x_n$ distinct and $f_0, f_1, ..., f_n$, find polynomial $P_n(x)$ such that $P_n(x_i) = f_i$ for $i = 0, ..., n$.

Uniqueness If we express $P_n(x) = \sum_{j=0}^n a_j x_j$, then $P_n(x_i) = f_i$ gives the linear equation

$$\begin{pmatrix} 1 & \cdots & 1 \\ x_0 & \cdots & x_n \\ \vdots & \ddots & \vdots \\ x_0^n & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix}$$

The matrix is called a Vandermonde matrix. Its determinant is $\prod_{0 \le i < j \le n}(x_i - x_j)$, which is nonzero, provided $x_i \neq x_j$ for $0 \le i < j \le n$. Thus, we get uniqueness of P_n.

The Vandermonde matrix has very poor condition number. Below, the Newton's approach is more stable.

3.2.1 Newton's interpolation

Newton's Interpolation Formula We express

$$\boxed{P_n(x) = \sum_{j=0}^{n} c_j q_j(x), \quad q_j(x) = \prod_{\ell=0}^{j-1}(x - x_\ell).}$$

Then the condition $P_n(x_i) = f_i$ gives the linear equation with lower triangular matrix:

$$\begin{pmatrix} 1 & 0 & \cdots & & \\ 1 & x_1 - x_0 & 0 & \cdots & \\ 1 & x_2 - x_0 & (x_2 - x_0)(x_2 - x_1) & 0 & \cdots \\ \vdots & \vdots & \vdots & & \\ 1 & x_n - x_0 & (x_n - x_0)(x_n - x_1) & & \cdots \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix}$$

This gives

$$c_0 = f_0, \quad P_0(x) = c_0$$
$$c_1 = \frac{f_1 - c_0}{x_1 - x_0} := f[x_0, x_1], \quad P_1(x) = P_0(x) + f[x_0, x_1](x - x_0)$$
$$c_2 = \frac{f_2 - P_1(x_2)}{(x_2 - x_0)(x_2 - x_1)} := f[x_0, x_1, x_2], \quad P_2(x) = P_1(x) + f[x_0, x_1, x_2](x - x_0)(x - x_1).$$

In general,

$$c_m = \frac{f_m - P_{m-1}(x_m)}{q_m(x_m)} := f[x_0, \cdots, x_m]$$
$$P_m(x) := P_{m-1}(x) + f[x_0, \cdots, x_m] q_m(x).$$

Thus, the polynomial P_n which interpolates f at $x_0, ..., x_n$ can be expressed as

$$\boxed{P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + \cdots + f[x_0, ..., x_n](x - x_0) \cdots (x - x_{n-1}).}$$

Lemma 3.1.

$$f[x_0, \cdots, x_m] = \frac{f[x_1, \cdots, x_m] - f[x_0, \cdots, x_{m-1}]}{x_m - x_0}.$$

Proof. We prove this lemma by induction. Suppose q interpolates f at $x_1, ..., x_n$. Then,

$$P_n(x) = q(x) + \frac{(x - x_n)}{x_n - x_0}\left(q(x) - P_{n-1}(x)\right).$$

By comparing the coefficient of x^n, we get

$$f[x_0, ..., x_n] = \frac{f[x_1, ..., x_n] - f[x_0, ..., x_{n-1}]}{x_n - x_0}.$$

□

Lemma 3.2. *Let σ be any permutation on $\{0, 1, ..., n\}$. Then*

$$f[x_{\sigma(0)}, ..., x_{\sigma(n)}] = f[x_0, ..., x_n].$$

Proof. The interpolating polynomial is unique, independent of the order of the interpolateing points. □

Theorem 3.1. *We have the expression of the interpolation error:*

$$\boxed{f(x) - P_n(x) = f[x_0, ..., x_n, x] \prod_{j=0}^{n} (x - x_j).} \tag{3.1}$$

Furthermore, there exists an $\xi \in [\min_{0\le i\le n}\{x_i, x\}, \max_{0\le i\le n}\{x_i, x\}]$ such that

$$\boxed{f[x_0, ..., x_n, x] = \frac{f^{n+1}(\xi)}{(n+1)!}.} \tag{3.2}$$

Proof. 1. Consider $\tilde{P}$ which interpolate f at $x_0, ..., x_n$ and x. Then

$$\tilde{P}(y) = P_n(y) + f[x_0, ..., x_n, x] \prod_{j=0}^{n} (y - x_j).$$

Evaluate y at x, we get (3.2).

2. Consider

$$\phi(y) := f(y) - P_n(y) - f[x_0, ..., x_n, x] \prod_{j=0}^{n} (y - x_j).$$

We have

$$\phi(x_0) = \cdots \phi(x_n) = \phi(x) = 0.$$

By Rolle's theorem, there exists $\xi \in [\min_{0\le i\le n}\{x_i, x\}, \max_{0\le i\le n}\{x_i, x\}]$ such that

$$\phi^{(n+1)}(\xi) = 0.$$

By direct calculation,

$$\phi^{(n+1)}(\xi) = f^{(n+1)}(\xi) - (n+1)! f[x_0, ..., x_n, x].$$

Thus, we obtain (3.2).

□

Hermite Interpolation Suppose x_i is no longer distinct. This is so-called Hermite interpolation.

Goal : Given $f^{(j)}(x_i)$, $j = 0, ..., k_i$, $i = 0, ..., n$, find a polynomial P_m such that

$$P_m^{(j)}(x_i) = f^{(j)}(x_i), \quad j = 0, ..., k_i, \ i = 0, ..., n.$$

Uniqueness The polynomial P_m has $m+1$ coefficients which are determined by the interpolation conditions. There are $\sum_{i=0}^{n} k_i$. Thus,

$$\sum_{i=0}^{n} k_i = m + 1.$$

The polynomial P_m is unique based on the non-zero determinant of the corresponding system for finding the coefficients.

Divided Difference with Repetitions When the interpolation points $x_0, ..., x_n$ cluster to a point, the divided differences are reduced to ordinary differentiation, and the New expansion formula is reduced to ordinary Taylor expansion formula. You can check:

- $f[x_0, x_0] = \lim_{x_1 \to x_0} f[x_0, x_1] = f'(x_0)$.
- $f[x_0, ..., x_0] = \frac{1}{k!} f^{(k)}(x_0)$
- $f[x_0, ..., x_n] = \frac{1}{n!} f^{(n)}(\xi)$, even with repetition.

Thus, Newton's interpolation formula holds with repetition.

Lagrange Interpolation Formula Lagrange takes the following expression

$$\boxed{P_n(x) = \sum_{i=0}^{n} f_i \ell_i(x), \quad \ell_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}.}$$

The polynomial ℓ_i is called the Lagrange characteristic polynomial. It satisfies

$$\ell_i(x_k) = \delta_{ik}, \quad 0 \leq i, k \leq n.$$

Barycentric interpolation formula The interpolation polynomial P_n can be computed by the following barycentric formula, which can lead to smaller interpolation errors as n increases.

$$\boxed{P_n(x) = \frac{\sum_{i=0}^{n} \frac{w_i}{x - x_i} f_i}{\sum_{i=0}^{n} \frac{w_i}{x - x_i}}, \quad w_i = \left(\prod_{\substack{j=0 \\ j \neq i}}^{n} (x_i - x_j) \right)^{-1}.}$$

To derive this formula, we rewrite $\ell_i(x)$ as

$$\ell_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} = \left(\prod_{j=0}^{n} (x - x_j) \right) \frac{w_i}{x - x_i}.$$

Noting $\sum_{i=0}^{n} \ell_i(x) = 1$, we can get the barycentric interpolation formula.

Usually, for $n < 50$, there is no difference between Newton interpolation formula, Lagrange interpolation formula and barycentric interpolation formula. But for $n > 50$, there is a significant difference.

3.2.2 Runge Phenomenon

If we perform polynomial interpolation with uniformly distributed nodes $x_0, ..., x_n$ on some interval, it is found that the error increase after n increases. The Runge example is

$$f(x) = \frac{1}{1 + x^2} \text{ on } I = [-5, 5], \text{ with } x_0, ..., x_n \text{ evenly distributed.}$$

We have seen that the error is

$$E_n f(x) := f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i).$$

For uniformly distributed nodes, the behavior of the function

$$q_{n+1}(x) = \prod_{i=0}^{n} (x - x_i)$$

oscillates near boundary. In fact,

$$|q_{n+1}(x)| \leq n! \frac{h^{n+1}}{4} \approx \frac{\sqrt{2\pi}}{4} n^{n+1/2} e^{-n} 10^n n^{-n-1} \to \infty, \quad h = \frac{10}{n}.$$

The maximal value can be achieved near the boundary. The error

$$\max_{x \in I} |E_n f(x)| \leq \frac{\max_{x \in I} |f^{(n+1)}(x)|}{4(n+1)} h^{n+1}.$$

Unfortunately, the quantity

$$\max_{x \in I} |f^{(n+1)}(x)|$$

grows very fast as n increases, while $q_{n+1}(x)/(n+1)!$ decays to 0 slower. We observe

$$\lim_{n \to \infty} \max_{x \in I} |E_n f(x)| = \infty.$$

The lack of convergence is indicated by the severe oscillation of the interpolating polynomial versus to the original function near the boundary. Such phenomenon is called the Runge phenomenon.

Homework

1. Reproduce this result, plot the functions $f^{(n+1)}(x)$ and $q_{n+1}(x)$ on interval $[-5, 5]$ to observe their behaviors.

Chebyshev-Gauss-Lobatto nodes As we have seen that the interpolating polynomial on a uniform nodes on an interval I oscillating severely on the boundary. The error containing the term

$$q_{n+1}(x) = \prod_{j=0}^{n}(x - x_j)$$

oscillates on the boundary. However, the Chebeshev function

$$T_n(x) := \cos(n \cos^{-1} x)$$

is a polynomial of degree n. We have $|T_n(x)| \leq 1$ for all $x \in [-1, 1]$. Its roots are

$$x_j = -\cos\left(\frac{2j+1}{n+1}\frac{\pi}{2}\right), \quad j = 0, ..., n.$$

They are called the Chebyshev-Gauss-Lobatto nodes. With these nodes, omen can show that the leading coefficient of $T_{n+1}(x)$ is 2^n. Thus, both $2^{-n}T_{n+1}$ and $q_{n+1}(x)$ have leading coefficient 1 and they have same roots and same degree. We have

$$q_{n+1}(x) = \prod_{j=0}^{n}(x - x_j) = 2^{-n}T_{n+1}(x).$$

The interpolation error $E_n f$ for Chebyshev-Gauss-Lobatto nodes on $[-1, 1]$ is

$$E_n f(x) := f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}2^{-n}T_{n+1}(x).$$

Homework

1. Plot the functions $f^{(n+1)}(x)$ and $q_{n+1}(x)$ on interval $[-5, 5]$ over Chebyshev-Gauss-Lobatto nodes. Find the error $E_n f(x)$. Observe their behaviors.
2. Prove that T_n is a polynomial of degree n. Prove that its leading coefficient is 2^{-n+1}.

Stability of Polynomial Interpolation The issue is: if f_i is perturbed to $\hat{f}_i$ for $i = 0, ..., n$, what is the error between the interpolating polynomials P_n and $\hat{P}_n$ on the interval I. The answer is the error depends on where you interpolate. Let $x_i \in I$, $i = 0, ..., n$ be the interpolation nodes. P_n and $\hat{P}_n$ be the polynomials which interpolate f and $\hat{f}$ at these nodes. The error is

$$\begin{aligned}\max_{x\in I}|P_n(x) - \hat{P}_n(x)| &\leq \max_{x\in I}\left|\sum_{i=0}^{n}\left(f(x_i) - \hat{f}_i(x_i)\right)\ell_i(x)\right| \\ &\leq \Lambda_n(x)\max_{0\leq i\leq n}\left|f(x_i) - \hat{f}(x_i)\right|,\end{aligned}$$

where

$$\Lambda_n(x) = \max_{x \in I} \sum_{i=0}^{n} |\ell_i(x)|$$

is called Lebesgue's constant. It serves as a condition number which measures the stability of the interpolation. It depends on the nodal points $x_0, ..., x_n$. The Lebesgue constant has the following estimates [1]:

- For equispaced node,
$$\Lambda_n(x) \approx \frac{2^{n+1}}{en(\log n + \gamma)},$$
where $e \approx 2.718$, $\gamma \approx 0.548$;
- For Chebyshev-Gauss- Lobatto nodes,
$$\Lambda_n(x) < \frac{2}{\pi}\left(\log n + \gamma + \log \frac{8}{\pi}\right) + \frac{\pi}{72n^2}.$$

Homework

1. Plot $\Lambda_n(x)$ for equi-spaced nodes and for Chebyshev-Gauss- Lobatto nodes.

Best approximation

Definition 3.1. *A modulus of continuity is a function $\omega : [0, \infty) \to [0, \infty)$ with $\lim_{t \to 0} \omega(t) = 0$. A function admits ω as a modulus of continuity if*

$$|f(x) - f(y)| \leq \omega(|x - y|)$$

for all x, y.

- If $\omega(t) = O(|t|)$, then this is Lipschitz continuity.
- If $\omega(t) = C|t|^\alpha$, $0 < \alpha < 1$, then this is Hölder continuity.

A function is uniformly continuous on $[0, 1]$ if and only if it admits a modulus of continuity ω.

Theorem 3.2 (Jackson)**.** *Let $\mathbb{P}_n$ be the set of polynomials of degree less or equal to n. For any $f \in C^r[0, 1]$, for any $n > 0$ integer,*

$$dist_\infty(f, \mathbb{P}_n) \leq C_r h^r \omega(f^{(r)}, h).$$

where $h = 1/n$ and ω is the modulus of continuity.

Applications

[1] For reference, see Quarteroni's book.

3.3 Approximation by Trigonometric polynomials

Motivations

- Trigonometric polynomials can approximate smooth periodic functions very efficiently.
- Fourier transform can diagonalize differential operators, convolution integral operators.
- Fourier expansion can be used to analyze data and signals. For instance, image debarring, image denoising.
- Fourier transform is a fundamental tool in magnetic resonance imaging (MRI).

3.3.1 Definition and examples

Definition We study Fourier expansion for 2π-periodic functions. Suppose f is a 2π-periodic function. Let us expand f as

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

To find the coefficients a_k, we take the following inner product, defined by

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

with e^{imx}. Using

$$(e^{imx}, e^{inx}) = \delta_{mn},$$

we can get

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

The coefficient a_m is called the Fourier coefficient, or Fourier multiple, of f at wave number m. We usually denote it by $\hat{f}_m$.

Examples

1. $$f(x) = \begin{cases} 1 & \text{for } 0 < x < \pi \\ -1 & \text{for } -\pi < x < 0 \end{cases}$$
2. $f(x) = \frac{1}{\pi}|x|$

3.3.2 Basic properties

A 2π-periodic function can be identified as a function on circle, that is $\mathbb{T} = \mathbb{R}/(2\pi\mathbb{Z})$. An important properties of Fourier transform are

- 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.

Differentiation

Lemma 3.3. *If $f \in C^1[\mathbb{T}]$, then*

$$\boxed{\widehat{f'}_k = ik\hat{f}_k.}$$

Proof.

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

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

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

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

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 smooth 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

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

and

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

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

Lemma 3.4. *If $f, g \in C(\mathbb{T})$, then*

$$\boxed{\widehat{\left(f * g\right)}_k = 2\pi\hat{f}_k\hat{g}_k.}$$

Proof.

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

Here, we have used Fubini theorem. □

Remarks The above two lemmae are valid for f, g are in L^2. The proof is based on the L^2 convergence for nice functions and the density theorem in the next section.

Regularity and decay: Riemann-Lebesgue lemma If f is smooth, then its Fourier coefficients decays very fast. Indeed, by taking integration by part n times, we have

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

Thus, if $f \in C^n$, we see $\hat{f}_k = O(|k|^{-n})$.[2] This can also be observed by the following arguments. We notice that

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

Hence,

$$\begin{aligned}
\hat{f}_k &= \frac{1}{2\pi}\int_{-\pi}^{\pi} f(x)e^{-ikx}\,dx \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{f(x)-f(x-\pi/k)}{2}e^{-ikx}\,dx \\
&:= \frac{1}{2\pi}\int_{-\pi}^{\pi} D_{\pi/k}f(x)e^{-ikx}\,dx \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} D^n_{\pi/k}f(x)e^{-ikx}\,dx
\end{aligned}$$

[2] If fact, we shall see later from the Riemann-Lebesgue lemma that $\hat{f}_k = o(|k|^{-n})$.

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 oscallation property of f at scale π/k. We conclude with the following lemma.

Lemma 3.5. *If $f \in C^n(\mathbb{T})$, then $\hat{f}_k = o(|k|^{-n})$.*

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.

Lemma 3.6 (Riemann-Lebesgue)**.** *If f is in $L^1(a,b)$, then*

$$\boxed{\hat{f}_A := \int_a^b f(x)\sin(Ax)\,dx \to 0, \text{ as } A \to \infty.}$$

Proof. (i) First, we show that the lemma holds for uniformly continuous functions. We have

$$\begin{aligned} 2\hat{f}_A &= \int_a^b f(x)\sin(Ax)\,dx - \int_a^b f(x)\sin(A(x-\pi/A))\,dx \\ &= -\int_{a-A/\pi}^{a} f(x+\pi/A)\sin(Ax)\,dx + \int_{b-\pi/A}^{b} f(x)\sin(Ax)\,dx \\ &\quad + \int_a^{b-\pi/A} (f(x) - f(x+A/\pi))\sin(Ax)\,dx \end{aligned}$$

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 $\|f-g\|_{L^1} < \epsilon$.

(iii) It holds for any A

$$|\widehat{(f-g)}_A| \le \int_a^b |f(x)-g(x)|\,dx := \|f-g\|_{L^1} < \epsilon.$$

From (i), there exists M such that for $A > M$, $|\hat{g}_A| < \epsilon$. (iv) Given $f \in L^1(a,b)$, and given any $\epsilon > 0$, from (ii), we can find a smooth function g such that $\|f-g\|_{L^1} < \epsilon$. From (i), there exists an $M > 0$ such that for any $A > M$ we have $|\hat{g}_A| < \epsilon$. From (iii), we have $|\widehat{(f-g)}_A| \le \|f-g\|_{L^1} < \epsilon$. Combining all these together, we get

$$|\hat{f}_A| \le |\hat{g}_A| + |\widehat{(f-g)}_A| \le 2\epsilon.$$

□

Remarks.

1. 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 $|k| \to \infty$.
2. If f is a piecewise smooth function with finite many jumps, then it holds that $\hat{f}_k = O(1/k)$. One may consider f has only one jump first. Then f is a superposition of a step function g and a smooth function h. We have seen that $\hat{h}_k$ decays fast. For the step function g, we have $\hat{g}_k = O(1/k)$.

3.4 Convergence Theory

Let 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.

3.4.1 Convergence theory for Smooth function

Theorem 3.3. *If f is a 2π-periodic, C^∞-function, then for any $n > 0$, there exists a constant C_n such that*
$$\boxed{|f_N(x) - f(x)| \le C_n N^{-n}.} \tag{3.3}$$

Proof.
$$\begin{aligned}
f_N(x) &:= \sum_{|k|\le N} \hat{f}_k e^{ikx} \\
&= \sum_{|k|\le N} \frac{1}{2\pi}\int_{-\pi}^{\pi} f(y)e^{ik(x-y)}\,dy \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\sin(N+\frac{1}{2})(x-y)}{\sin(\frac{1}{2}(x-y))} f(y)\,dy \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\sin(N+\frac{1}{2})t}{\sin\frac{t}{2}} f(x+t)\,dt \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} D_N(t) f(x+t)\,dt
\end{aligned}$$

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 $g(t) := (f(x+t)-f(x))/\sin(t/2) = \int_0^1 f'(x+st)\,ds \cdot t/\sin(t/2)$ is 2π periodic and in C^∞. We can apply integration-by-part n times to arrive

$$f_N(x) - f(x) = (N+\frac{1}{2})^{-n}\frac{(-1)^{n/2}}{2\pi}\int_{-\pi}^{\pi} g^{(n)}(t)\sin((N+\frac{1}{2})t)\,dt$$

for even n. Similar formula for odd n. This completes the proof. □

Remark. The constant C_n, which depends on $\int |g^{(n)}|\,dt$, is in general not big, as compared with the term N^{-n}. Hence, the approximation (3.3) is highly efficient for smooth functions. For example, $N = 20$ is sufficient in many applications. The accuracy property (3.3) is called spectral accuracy.

3.4.2 L^2 Convergence Theory

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. The function spaces L^2 and ℓ^2 are defined below.

$$L^2(\mathbb{T}) := \{f \mid f \text{ is } 2\pi \text{ periodic and } \int_{-\pi}^{\pi} |f(x)|^2\,dx < \infty\}$$

with the inner product

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

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 $f \in L^p(\mathbb{T})$, define

$$f_\epsilon(x) := \frac{1}{\epsilon}\int \rho\left(\frac{x-y}{\epsilon}\right) f(y)\,dy$$

Then f_ϵ is a C^∞ function and $f_\epsilon \to f$ in L^p. This is called the density theorem. We shall not prove here.

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 $(a, b) := \sum_k a_k \overline{b_k}$.

It is easy to check that e^{ikx} are orthogonal in L^2. From this, we have for any N,

$$0 \le (f - f_N, f - f_N) = \|f\|^2 - \sum_{|k|\le N} |\hat{f}_k|^2.$$

Or equivalently,

$$\sum_{|k|\le N} |\hat{f}_k|^2 \le \|f\|^2. \tag{3.4}$$

This is called the Bessel inequality. It says that the Fourier transform maps continuously from $L^2(\mathbb{T})$ to $\ell^2(\mathbb{Z})$.

Theorem 3.4 (Isometry property)**.** *The Fourier transform is an isometry from $L^2(\mathbb{T})$ to $\ell^2(\mathbb{Z})$:*

$$(f, g) = \sum_k \hat{f}_k \overline{\hat{g}_k}.$$

Proof. To show this, we first assume that f is a smooth function. We can apply the convergence theorem for f. This yields

$$\begin{aligned}(f, g) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x)\overline{g(x)}\, dx \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_k \hat{f}_k e^{ikx} \overline{g(x)}\, dx \\ &= \sum_k \hat{f}_k \overline{\hat{g}_k}.\end{aligned}$$

To show this formula is valid for all $f, g \in L^2$, we notice that any function in L^2 can be approximated by smooth functions.

The isometry property is valid for f_ϵ and g: $(f_\epsilon, g) = (\widehat{f_\epsilon}, \hat{g})$. As $\epsilon \to 0$,

$$|(f_\epsilon - f, g)| \le \|f_\epsilon - f\|\|g\| \to 0,$$

and

$$|(\widehat{f_\epsilon} - \hat{f}, \hat{g})| \le \|\widehat{f_\epsilon} - \hat{f}\|\|\hat{g}\| \le \|f_\epsilon - f\|\|g\| \to 0.$$

The last inequality is from the Bessel inequality. □

The isometry property says that the Fourier transformation preserves the inner product. When $g = f$ in the above isometry property, we obtain the following Parseval identity.

Corollary 3.3 (Parseval identity). *For $f \in L^2$, we have*

$$\|f\|^2 = \sum_k |\hat{f}_k|^2.$$

Theorem 3.5 (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 $\|f_N - f_M\|^2 = \sum_{N \le |k| < M} |\hat{f}_k|^2$ and the Bessel inequality. Suppose f_N converges to g. We see that

$$\widehat{(f - f_N)}_k = \frac{1}{2\pi} \int_{\mathbb{T}} (f - f_N)(x) e^{-ikx}\, dx = 0 \text{ if } |k| < N.$$

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 Parvesal identity, we have $f = g$. □

3.4.3 BV Convergence Theory

A function is called a BV function on an interval (a, b), that is, function of finite total variation, if for any partition $\pi = \{a = x_0 < x_1 < \cdots < x_n = b\}$,

$$\|f\|_{BV} := \sup_{\pi} \sum_i |f(x_i) - f(x_{i-1})| < \infty.$$

An important property of BV function is that its singularity can only be 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 and f_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.

Theorem 3.6 (Fourier inversion theorem for BV functions). *If f is in BV (function of bounded variation), then*

$$f_N(x) := \sum_{k=-N}^{N} \hat{f}_k e^{ikx} \to \frac{1}{2}(f(x+) + f(x-)).$$

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 $N\to\infty$. □

3.4.4 Pointwise estimate of rate of convergence

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 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. The 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

$$f_N(x)=\frac{1}{2\pi}\int_{x-\pi}^{x}\frac{\sin((N+\frac{1}{2})t)}{\sin(t/2)}\,dt-\frac{1}{2\pi}\int_{x}^{x+\pi}\frac{\sin((N+\frac{1}{2})t)}{\sin(t/2)}\,dt$$

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 $z > 0$,

$$\int_z^\infty \text{sinc}(t)\, dt = O\left(\frac{1}{z}\right).$$

To see the latter inequality, we rewrite

$$\int_z^\infty \text{sinc}(t)\, dt = \left(\int_z^{n\pi} + \sum_{k\geq n}\int_{n\pi}^{(n+1)\pi}\right) \text{sinc}(t)\, dt,$$

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 $\int_0^z \text{sinc}(t)\, dt$ by $\text{Si}(z)$.

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

$$\begin{aligned}
f_N(\frac{z}{(N+1/2)}) &= \frac{1}{\pi}\int_{z-(N+1/2)\pi}^{z} \text{sinc}(t)\, dt - \frac{1}{\pi}\int_z^{z+(N+1/2)\pi} \text{sinc}(t)\, dt + o(1/N) \\
&= \frac{1}{\pi}\int_{-\infty}^{z} \text{sinc}(t)\, dt - \frac{1}{\pi}\int_z^{\infty} \text{sinc}(t)\, dt + O(1/(z+N)) + O(1/(z-N)) \\
&= \frac{1}{\pi}\int_{-\infty}^{z} \text{sinc}(t)\, dt + \frac{1}{\pi}\int_{-z}^{-\infty} \text{sinc}(t)\, dt + O(1/(z+N)) + O(1/(z-N)) \\
&= \frac{1}{\pi}\int_{-z}^{z} \text{sinc}(t)\, dt + (1/(z+N)) + O(1/(z-N)) \\
&= \frac{2}{\pi}\int_{0}^{z} \text{sinc}(t)\, dt + (1/(z+N)) + O(1/(z-N)) \\
&= 1 - \frac{2}{\pi}\int_z^{\infty} \text{sinc}(t)\, dt + (1/(z+N)) + O(1/(z-N))
\end{aligned}$$

In general, for function f with arbitrary jump at 0, we have

$$\begin{aligned}
f_N(\frac{z}{(N+1/2)}) &= f(0+) - \frac{[f]}{\pi}\int_z^{\infty} \text{sinc}(t)\, dt + (1/(z+N)) + O(1/(z-N)) \\
&= f(0+) + [f]O(1/z) + O(1/(z-N)).
\end{aligned}$$

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 $\text{Si}(z)$ indeed occurs at $z = \pi$ where

$$\frac{1}{\pi}\text{Si}(\pi) \approx 0.58949$$

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 Gibbs phenomenon.

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 $|x| \le 1$ 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 $x = 0$?

3.4.5 Fourier Expansion of Real Valued Functions

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 real-valued,

$$\hat{f}_n = \overline{\hat{f}_{-n}}.$$

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

$$\begin{aligned} f(x) &= \sum_{n\in\mathbb{Z}} \hat{f}_n e^{inx} \\ &= \frac{1}{2}a_0 + \frac{1}{2}\sum_{n=1}^{\infty}(a_n - ib_n)e^{inx} + \frac{1}{2}\sum_{n=1}^{\infty}(a_n + ib_n)e^{-inx} \\ &= \frac{1}{2}a_0 + \sum_{n=1}^{\infty}(a_n \cos nx + b_n \sin nx) \end{aligned}$$

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,

$$a_n = \frac{1}{\pi}\int_0^{2\pi} f(x)\cos nx\,dx,\ b_n = \frac{1}{\pi}\int_0^{2\pi} f(x)\sin nx\,dx.$$

The functions $\{\cos nx, \sin mx\}$ are orthogonal to each other. Further,

$$\frac{1}{\pi}\int_0^{2\pi}\cos^2 nx\,dx = \frac{1}{\pi}\int_0^{2\pi}\sin^2 nx\,dx = 1 \text{ for all } n.$$

The Parseval equality reads

$$\frac{1}{2\pi}\int_{\mathbb{T}} f(x)^2\,dx = 2\sum_n \left(a_n^2 + b_n^2\right).$$

3.5 Discrete Fourier Transform

3.5.1 Definition and inversion formula

Given a 2π-periodic function f. Let us sample f by $f_j := f(x_j)$, where $x_j := 2\pi j/N$. Define the discrete Fourier transform for the sampled data by

$$\boxed{\tilde{f}_k := \frac{1}{N}\sum_{j=0}^{N-1} f_j e^{-ikx_j}.}$$

This is exactly the trapezoidal approximation for numerical integration of the Fourier multiples:

$$\frac{1}{2\pi}\int_0^{2\pi} f(x)e^{-ikx}\,dx.$$

When $f \in C^\infty$, according to the Euler-MacLaurin summation formula for periodic functions,

$$\left|\frac{1}{2\pi}\int_0^{2\pi} f(x)e^{-ikx}\,dx - \frac{1}{N}\sum_{j=0}^{N-1} f_j e^{-ikx_j}\right| = O(N^{-n})$$

for any n. Thus, the discrete Fourier multiples can approximate Fourier multiples with spectral accuracy, provided the underlying function is C^∞.

From $\tilde{f}_k$, we define

$$\boxed{I_N f(x) := \sum_{k=-N/2}^{N/2-1} \tilde{f}_k e^{ikx}.}$$

We claim that

$$\boxed{I_N f(x_j) = f(x_j).}$$

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:

$$\begin{aligned} I_N f(x) &= \sum_{k=-N/2}^{N/2-1} \frac{1}{N}\sum_{k=0}^{N-1} f_j e^{ik(x-x_j)} \\ &= \frac{1}{N}\sum_{j=0}^{N-1} D_N(x-x_j) f_j \end{aligned}$$

where

$$D_N(x) = \sum_{k=-N/2}^{N/2-1} e^{ikx} = e^{-ix/2}\frac{\sin(Nx/2)}{\sin(x/2)}$$

We find that

$$D_N(x_j) = \begin{cases} 1 & \text{for } j \equiv 0(\bmod N) \\ 0 & \text{for } j \not\equiv 0(\bmod N). \end{cases}$$

Hence, $I_N f(x_j) = f_j$.

Let $\mathcal{S}_N$ be the space of the trigonometric polynomial of degree $N/2$:

$$\mathcal{S}_N := \text{span}\{E_k(x) = e^{ikx} \mid -N/2 \le k < N/2\}.$$

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 orthonomal in both inner products. Hence, these two inner products are identical any $f, g \in S_N$.

Again, from orthonomality of $\{E_k(x)\}$, we have the isometry property:

$$(f, g)_N = \sum_{-N/2 \le k < N/2} \tilde{f}_k \overline{\tilde{g}_k},$$

and the Parseval identity:

$$\frac{1}{N} \sum_{j=0}^{N} |f_j|^2 = \sum_{-N/2 \le k < N/2} |\tilde{f}_k|^2.$$

3.5.2 Approximation issues

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 $\mathcal{S}_N$. Similarly, the interpolation operator $I_N f$:

$$I_N f(x) := \sum_{-N/2 \le k < N/2} \tilde{f}_k e^{ikx}$$

is a projection from $C(-\pi, \pi)$ onto $\mathcal{S}_N$, and is characterized by $I_N f(x_j) = f(x_j)$, $j = 0, \cdots, N-1$. 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

$$\begin{aligned}
\|R_N f\|^2 &= \sum_{-N/2\le k<N/2} |\tilde{f}_k - \hat{f}_k|^2 \\
&\le \sum_{-N/2\le k<N/2} \sum_{\substack{-\infty<m<\infty \\ m\neq 0}} |\hat{f}_{k+mN}|^2 \\
&= \sum_{k\ge N/2, k<-N/2} |\hat{f}_k|^2.
\end{aligned}$$

Since P_N is an orthogonal projection, we have

$$\|f - I_N f\|^2 = \|f - P_N f\|^2 + \|R_N f\|^2.$$

It is not difficult to find the approximation error for P_N. Indeed, let H^s denote the Sobolev space of order s:

$$H^s := \{f \text{ is } 2\pi\text{-periodic}, f, \cdots, f^{(s)} \in L^2\}$$

with the norm $\|f\|_{H^s}^2 := \sum_{m=0}^s \|f^{(m)}\|^2$. From the Parseval identity, this norm is equivalent to $\sum_k (1+|k|^2)^s |\hat{f}_k|^2$. We have the following approximation theorem.

Theorem 3.7. *If $f \in H^s$, then*

$$\|f - P_N f\| \leq CN^{-s}\|f^{(s)}\|$$

Proof. We use the facts that $\{E_k\}_{k\in Z}$ is a basis in L^2 and the Parseval identity:

$$\begin{aligned}\|f - P_N f\|^2 &= \sum_{|k|\geq N/2} |\hat{f}_k|^2 \\ &= \sum_{|k|\geq N/2} |k|^{-2s}|k|^{2s}|\hat{f}_k|^2 \\ &\leq O(N^{-2s})\|f^{(s)}\|^2.\end{aligned}$$

□

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 detail proof to (p.280**??**).

Theorem 3.8. *If $f \in H^s$, $s \geq 1$, then*

$$\|f - I_N f\| \leq CN^{-s}\|f^{(s)}\|.$$

3.6 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.

3.6.1 The FFT algorithm

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 $\mathcal{F}_N$ of the discrete Fourier transformation is

$$\mathcal{F}_N = \left(\omega_N^{ij}\right)_{\substack{0\leq i<N \\ 0\leq j<N}}, \ \omega_N = e^{-2\pi\sqrt{-1}/N}.$$

Note that

$$\boxed{\bar{\mathcal{F}}_N \mathcal{F}_N = N I_{N\times N}.}$$

For simplicity, below let us call $\tilde{f} = \mathcal{F}_N f$ instead of $\tilde{f} = \frac{1}{N}\mathcal{F}_N f$. Let us suppose N is even, say $2M$. 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 $f' = (f_0, f_2, \cdots, f_{2N-2})$, $f'' = (f_1, f_3, \cdots, f_{2N-1})$. For $0 \le k < M$, 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

$$\omega_N^2 = \omega_M.$$

For $\tilde{f}_{M+k}$, $k = 0, ..., M-1$, we have

$$\begin{aligned}\tilde{f}_{k+M} &= \sum_{j=0}^{M-1} \omega_N^{(M+k)2j} f_{2j} + \sum_{j=0}^{M-1} \omega_N^{(M+k)(2j+1)} f_{2j+1} \\ &= \sum_{j=0}^{M-1} \omega_M^{kj} f_{2j} + \omega_N^{M+k} \sum_{j=0}^{M-1} \omega_M^{(M+k)j} f_{2j+1} \\ &= \sum_{j=0}^{M-1} \omega_M^{kj} f_{2j} - \omega_N^k \sum_{j=0}^{M-1} \omega_M^{kj} f_{2j+1} \\ &= (\mathcal{F}_M f')_k - \omega_N^k (\mathcal{F}_M f'')_k\end{aligned}$$

Here, we have used

$$\omega_N^{2M} = 1,\ \omega_N^M = -1.$$

Thus, the discrete Fourier transform can be calculated as the follows.

1. Split $f = (f_0, \cdots, f_{N-1})$ into

$$f' = (f_0, f_2, \cdot, f_{N-2}),\ f'' = (f_1, f_3, \cdots, f_{N-1})$$

2. Perform

$$\tilde{f}' = \mathcal{F}_M f',\ \tilde{f}'' = \mathcal{F}_M f''$$

3. For $0 \le k < M$, compute

$$\begin{aligned} \tilde{f}_k &= \tilde{f}'_k + \omega_N^k \tilde{f}''_k \\ \tilde{f}_{k+M} &= \tilde{f}'_k - \omega_N^k \tilde{f}''_k \end{aligned}$$

In matrix form, $\mathcal{F}_N$ can be splitted into

$$\boxed{\mathcal{F}_N = Q_N \begin{bmatrix} \mathcal{F}_{N/2} & 0 \\ 0 & \mathcal{F}_{N/2} \end{bmatrix} P_N.} \tag{3.5}$$

Here, $\mathcal{P}_N$ is a permutation matrix which maps

$$\mathcal{P}_N : (f_0, f_1, \cdots, f_{N-1})^t \mapsto (f_0, f_2, \cdots, f_{N-2}, f_1, f_3, \cdots, f_{N-1})^t;$$

the matrix Q_N is defined as

$$Q_N = \begin{bmatrix} I_{N/2} & D_{N/2} \\ I_{N/2} & -D_{N/2} \end{bmatrix},\ I : \text{identity matrix}, D_{N/2} = \text{diag}(1, \omega, \cdots, \omega^{N/2-1})$$

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 count to perform $\mathcal{F}_{2^L}$. Then we have

$$C_{2^L} = C2^L + 2\, C_{2^{L-1}} = 2C2^L + 2^2 C_{2^{L-2}} = \cdots = LC2^L + 2^L C_{2^0}.$$

Thus,

$$C_N = CL2^L = CN \log_2 N.$$

3.6.2 Variants of FFT

Trigonometric representation

When all $f_j \in \mathbb{R}$, then, similar to the continuous case where $\overline{\widehat{f}_k} = \widehat{f}_{-k}$, we also have

$$\bar{\tilde{f}}_k = \tilde{f}_{-k} = \tilde{f}_{N-k}, \quad \text{for } k = 0, ..., N-1.$$

Let

$$M = \begin{cases} N/2 & \text{for even } N \\ (N+1)/2 & \text{for odd } N \end{cases}$$

and let

$$\tilde{f}_k = c_{2k-1} - ic_{2k}, \quad k = 1, \cdots, M-1,$$

$$c_0 = \tilde{f}_0, \quad c_{N-1} = f_{N/2}.$$

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

$$\begin{aligned} c_0 &= \frac{1}{N}\sum_{j=0}^{N-1} f_j \\ c_{2k-1} &= \frac{1}{N}\sum_{j=0}^{N-1} f_j \cos(kx_j), k = 1, \cdots, N/2-1 \\ c_{2k} &= \frac{1}{N}\sum_{j=0}^{N-1} f_j \sin(kx_j), k = 1, \cdots, N/2 \\ c_{N-1} &= \frac{1}{N}\sum_{j=0}^{N-1} (-1)^j f_j \end{aligned}$$

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

$$\begin{aligned} f_j &= \sum_{k=-N/2}^{N/2-1} \tilde{f}_k e^{ikx_j} \\ &= f_0 + (-1)^j \tilde{f}_{N/2} + \sum_{k=1}^{N/2-1} 2\tilde{f}_k \cos(kx_j) \end{aligned}$$

Fourier Sine Transform

When f_j is an odd sequence, i.e. $f_{N-j} = -f_j$, $j = 0, \cdots, N/2$, then for $k = 1, \cdots N/2 - 1$,

$$\begin{aligned} \tilde{f}_k &= \frac{1}{N} \sum_{j=-N/2}^{N/2-1} f_j e^{-ikx_j} \\ &= \frac{1}{N} \sum_{j=1}^{N/2-1} 2 f_j \sin(kx_j) \end{aligned}$$

Its inverse transform is, for $j = 1, \cdots, N/2 - 1$,

$$\begin{aligned} f_j &= \sum_{k=-N/2}^{N/2-1} \tilde{f}_k e^{ikx_j} \\ &= \sum_{k=1}^{N/2-1} 2\tilde{f}_k \sin(kx_j). \end{aligned}$$

3.7 Fast Chebyshev Transformation

For boundary value problems, it is more favorable to use another representation for functions on bounded intervals, the Chebyshev representation. Without loss of generality, we consider the domain $[-1, 1]$. The Chebyshev polynomials are defined as

$$T_n(x) = \cos(n \cos^{-1} x).$$

The Chebyshev expansion for functions f defined on $[-1, 1]$ is

$$f \sim \sum_{n=0}^{\infty} a_n T_n(x).$$

The Chebyshev expansion can be view as the Fourier expansion through a composition of the transformation:

$$\theta := \cos^{-1} x.$$

For f define on $[-1, 1]$, we can define a function F on the unit circle by

$$F(\theta) = f(\cos\theta).$$

Then F is 2π-periodic and even. The Fourier expansion of F is

$$F(\theta) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(n\theta),$$

where

$$a_n = \frac{1}{\pi}\int_0^\pi F(\theta)\cos n\theta\, d\theta.$$

The corresponding Chebyshev expansion of f is

$$\begin{aligned} f(x) &= \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(n\cos^{-1} x) \\ &= \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n T_n(x). \end{aligned}$$

The coefficient

$$\begin{aligned} a_n &= \frac{1}{\pi}\int_0^\pi F(\theta)\cos(n\theta)\, d\theta \\ &= \frac{1}{\pi}\int_{-1}^{1} f(x)T_n(x)\frac{dx}{\sqrt{1-x^2}} \end{aligned}$$

We may perform discrete Chebyshev transformation through the help of discrete Fourier transformation. Let

$$\theta_j = j\pi/N,\ x_j = \cos(\theta_j),\ 0 \le j < N.$$

3.8 Approximation by Splines

A general function on an interval can be approximated by piecewise polynomials.pii These piecewise polynomials are called splines. To be precise, let us consider an interval $[a, b]$. It is partitioned into $\Pi = \{x_0 = a < x_1 < \cdots < x_n = b\}$. Let

$$h = \max_i (x_i - x_{i-1}),$$

which measures the maximal size of the partition. A spline function S of degree k is a function on $[a, b]$ which satisfies

- $S(\cdot)$ is a polynomial of degree $\le k$ on each (x_{i-1}, x_i) for $i = 1, ...n$;
- $S \in C^{k-1}[a, b]$.

The hat function at node x_i is a piecewise linear continuous function which satisfies

$$\phi_i(x_j) = \delta_{ij}, i = 0, ...n.$$

Piecewise linear functions Let

$$\mathbb{S}_1 := \{ \text{ piecewise linear continuous functions on the partition } \Pi\}$$

The space $\mathbb{S}_1$ has a basis $\{\phi_0, ..., \phi_n\}$. Indeed, any function $s \in \mathbb{S}_1$ can be represented as

$$s(x) = \sum_{i=0}^{n} s(x_i)\phi_i(x).$$

Question: Given $f \in C^2[a,b]$, we want to find a function $s \in \mathbb{S}_1$ which is closed to f. There are many choices of such s. Let us introduce two:

- $I_1(f) := \sum_{i=0}^n f(x_i)\phi_i$;
- $\pi_1(f) := \arg\min_{g\in\mathbb{S}_1} \|f-g\|_2$.

It is clear that both mappings are projections. The first one is an interpolation projection, while the second one is an L^2 projection, or the best least squares approximation. We shall discuss their approximation errors.

Approximation Error for $I_1(f)$

Theorem 3.9.

$$\|f - I_1 f\|_\infty \le \frac{1}{8}h^2\|f''\|_\infty.$$

$$\|f - I_1 f\|_2 \le \frac{1}{\sqrt{90}}h^2\|f''\|_2.$$

Proof. 1. On (x_{i-1}, x_i), there exists an ξ_i such that

$$f(x) - I_1 f(x) = \frac{f''(\xi)}{2}(x - x_{i-1})(x - x_i).$$

Thus, for $x \in (x_{i-1}, x_i)$, we have

$$|f(x) - I_1 f(x)| \le \left(\frac{x_i - x_{i-1}}{2}\right)^2 \frac{|f''(x_i)|}{2}.$$

This shows

$$\|f - I_1 f\|_\infty \le \frac{h^2}{8}\|f''\|_\infty.$$

2. Let $w = f - I_1 f$. Then (i) $w(x_i) = 0$ for $i = 0, ..., n$, (ii) $w'' = f''$. Our goal is to estimate $\|w\|_2$ in terms of $\|w''\|_2$ We can express the interpolation error w on (x_{i-1}, x_i) in integral form. We recall that for $w(x_i) = w(x_{i+1}) = 0$, w has the representation:

$$w(x) = -h^2 \int_{x_i}^{x_{i+1}} g\left(\frac{x - x_i}{h}, \frac{y - x_i}{h}\right) w''(y)\, dy$$

where g is the Green's function of $-d^2/dx^2$ on (x_i, x_{i+1}). Thus, we can estimate $\|w\|_2$ in terms of $\|w''\|_2$ on (x_i, x_{i+1}). Namely,

$$|w(x)|^2 \le h^4 \left(\int_{x_i}^{x_{i+1}} \left|g\left(\frac{x - x_i}{h}, \frac{y - x_i}{h}\right)\right|^2 dy\right) \left(\int_{x_i}^{x_{i+1}} |w''(y)|^2\, dy\right).$$

$$\begin{aligned}\int_{x_i}^{x_{i+1}} |w(x)|^2\,dx &\le \int_{x_i}^{x_{i+1}}\int_{x_i}^{x_{i+1}} \left|g\left(\frac{x-x_i}{h},\frac{y-x_i}{h}\right)\right|^2 dy\,dx \int_{x_i}^{x_{i+1}} |w''(y)|^2\,dy \\ &\le \frac{1}{90}h^4 \int_{x_i}^{x_{i+1}} |w''(y)|^2\,dy.\end{aligned}$$

As we sum over $i = 1, \cdots, n-1$, we get

$$\|w\|_2 \le \frac{1}{\sqrt{90}} h^2 \|w''\|_2.$$

Remark The expression for $w(x)$ in terms of w'' on an interval (x_i, x_{i+1}) with $w(x_i) = w(x_{i+1}) = 0$ is equivalent to solve

$$-w''(x) = f(x), x \in (0,1)$$

$$w(0) = w(1) = 0.$$

We integrate it once to get

$$w'(y) = -\int_1^y f(s)\,ds + C_1.$$

Next, we integrate it from 0 to x and use $w(0) = 0$ to get

$$w(x) = -\int_0^x \int_1^y f(s)\,ds\,dy + C_1 x.$$

By integration-by-part,

$$\begin{aligned}-\int_0^x \int_1^y f(s)\,ds\,dy &= -\int_0^x F(y)\,dy = -\left[yF(y)\right]_0^x + \int_0^x yF'(y)\,dy \\ &= x\int_x^1 f(y)\,dy + \int_0^x yf(y)\,dy\end{aligned}$$

From $w(1) = 0$, we obtain $C_1 = -\int_0^1 yf(y)\,dy$. Hence

$$w(x) = \int_0^x y(1-x)f(y)\,dy + \int_x^1 x(1-y)f(y)\,dy$$

Let us define

$$g(x,y) := \begin{cases} x(1-y) & \text{if } 0 \le x \le y \le 1 \\ y(1-x) & \text{if } 0 \le y \le x \le 1. \end{cases}$$

Then the solution above can be represented as

$$w(x) = \int_0^1 g(x,y)f(y)\,dy = -\int_0^1 g(x,y)w''(y)\,dy.$$

□

Cubic Spline The cubic spline g approximate a function f on $[a,b]$ by piecewise cubic polynomial. Suppose $[a,b]$ is partitioned into

$$x_0 = a < x_1 < \cdots < x_n = b.$$

On $[x_i, x_{i+1}]$, $g(x) = p_i(x)$, a cubic polynomial, which has 4 parameters. Natural conditions for them are

$$p_i(x_i) = f_i, \quad p_i(x_{i+1}) = f(x_{i+1}).$$

We need another two conditions. Let us choose them to be

$$p_i'(x_i) = s_i, \quad p_i'(x_{i+1}) = s_{i+1}.$$

With these 4 parameters, p_i can be expressed as

$$p_i(x) = f_i + p_i[x_i, x_i](x-x_i) + p_i[x_i, x_i, x_{i+1}](x-x_i)^2 + p_i[x_i, x_i, x_{i+1}, x_{i+1}](x-x_i)^2(x-x_{i+1})$$

where

$$p_i[x_i, x_i] = s_i, \quad p_i[x_{i+1}, x_{i+1}] = s_{i+1}$$

$$\begin{aligned} p_i[x_i, x_i, x_{i+1}] &= \frac{p_i[x_i, x_{i+1}] - p_i[x_i, x_i]}{\Delta x_i} \\ p_i[x_i, x_i, x_{i+1}, x_{i+1}] &= \frac{p_i[x_i, x_{i+1}, x_{i+1}] - p_i[x_i, x_i, x_{i+1}]}{\Delta x_i}. \end{aligned}$$

Or

$$p_i(x) = c_0 + c_1(x - x_i) + c_2(x - x_i)^2 + c_3(x - x_i)^3,$$

where

$$\begin{aligned} c_0 &= f_i \\ c_1 &= s_i \\ c_2 &= \frac{f[x_i, x_{i+1}] - s_i}{\Delta x_i} - c_3 \Delta x_i \\ c_3 &= \frac{s_i + s_{i+1} - 2f[x_i, x_{i+1}]}{(\Delta x_i)^2}. \end{aligned}$$

Choices of s_i We introduce two choices to determine s_i:

- Hermite: $s_i = f_i'$: with this, $I_3^H f(x) := g(x)$ has the following estimate

$$\begin{aligned} \|f - I_3^H f\|_\infty &\le \max_i \|(x - x_i)^2 (x - x_{i+1})^2 \frac{f^{(4)}(\xi_i)}{4!}\|_\infty \\ &= \left(\frac{h}{2}\right)^4 \frac{\|f^{(4)}\|_\infty}{4!}. \end{aligned}$$

- Cubic Bessel interpolation: we choose

$$s_i = \tilde{p}_i'(x_i),$$

where $\tilde{p}_i$ interpolates f at x_{i-1}, x_i, x_{i+1}. Hence

$$s_i = \frac{\Delta x_i f[x_{i-1}, x_i] + \Delta x_{i-1} f[x_i, x_{i+1}]}{\Delta x_i + \Delta x_{i-1}}$$

With this interpolation, call such g by $I_3^B f$, we have

$$\|f - I_3^B f\|_\infty \leq Ch^3(1 + h\|f^{(4)}\|_\infty).$$

- Cubic spline: $s_0, s_1, ..., s_n$ are determined so that $g \in C^2$. This requires

$$p_{i-1}''(x_i) = p_i''(x_i).$$

Or

$$h_{i-1}s_{i-1} + 2(h_i + h_{i-1})s_i + h_i s_{i+1} = b_i, \quad i = 1, ..., n-1,$$
$$b_i = 3\left(f[x_{i-1}, x_i]h_i + f[x_i, x_{i+1}]h_i\right).$$

This is a second order finite difference equation. There are $n+1$ unknowns $s_0, ..., s_n$, $n-1$ equations. We need two boundary conditions. There are different choices of boundary conditions:

  - Complete cubic spline:

$$s_0 = f'(x_0), \quad s_n = f'(x_n).$$

  - Natural spline:

$$s''(x_0) = s''(x_n) = 0.$$

    Not-a-knot condition

$$p_0 = p_1, p_{n-2} = p_{n-1},$$

    or g''' is continuous across x_1 and x_{n-1}.

Let us denote this cubic spline approximation of f by $I_3 f$. We have the following approximation estimates: suppose $f \in C^4[a, b]$, then

$$\|f - I_3 f\|_\infty = O(h^4)\|f^{(4)}\|_\infty.$$
$$\|f'' - (I_3 f)''\|_\infty = O(h^2)\|f^{(4)}\|_\infty.$$

3.8.1 Splines on uniform grid systems

In this approach, we first set up a grid system

$$x_{jk} = 2^{-j}k$$

on $\mathbb{R}$. The index j represents the scale, whereas k is the location index. In the spline approach, we fix the scale index j, say $j = J$. We choose a spline function ϕ. For instance, the box function $1_{[0,1)}$.

Box spline We define

$$\phi_{jk} = 2^{j/2}\phi(2^j x - k).$$

It is clear that

- $\|\phi_{jk}\|_2 = 1$
- $\langle \phi_{jk}, \phi_{jl} \rangle = \delta_{kl}$

If we define

$$\mathcal{S}_0^j = \{f : \mathbb{R} \to \mathbb{R} | f \text{ is constant on each } (x_{j,k}, x_{j,k+1})\}$$

Then a general function can be approximated by

$$f(x) \approx f_j(x) := \sum_{k \in \mathbb{Z}} \langle f, \phi_{j,k} \rangle \phi_{j,k}(x).$$

If $f \in H^1(\mathbb{R})$, then we have

$$\|f - f_j\|_2 = O(2^{-j}).$$

Hat splines We define

$$\phi_2(x+1) := 1_{[0,1)} * 1_{[0,1)}(x).$$

This function is the hat spline

$$\phi_2(x) = \begin{cases} x & \text{if } -1 < x < 0 \\ 2 - x & \text{if } 0 \le x < 1 \\ 0 & \text{otherwise.} \end{cases}$$

We define $\phi_{jk} = \phi(2^j x - k)$. We also define

$$\mathcal{S}_1^j := \{f : \mathbb{R} \to \mathbb{R} \text{ continuous} | f \text{ is linear on each } (x_{j,k}, x_{j,k+1})\}$$

As we have seen before that any $C^1 \cap H^2$ function can be approximated by

$$f(x) \approx f_j := \sum_{i \in \mathbb{Z}} f(x_{ji}) \phi_{ji}(x)$$

with approximation error

$$\|f - f_j\|_2 \le O(2^{-2j}).$$

General B-splines In general, we consider splines $\phi_r = 1_{[0,1)} * ... * 1_{[0,1)}$. Support of ϕ_r is $[0, r]$. ϕ_r has the following properties

- ϕ_r is a polynomial of degree $r-1$ on each interval $(i, i+1)$ for all $i \in \mathbb{Z}$;
- $\phi_r \in C^{r-2}$;

- ϕ_r satisfies the following scaling relation:

$$\phi_r(x) = 2\sum_{k=0}^{r} h_k \phi_r(2x-k)$$

where

$$h(z) = \sum_{k=0}^{r} h_k z^k = \left(\frac{1+z}{2}\right)^r.$$

In particular, for the box function, $h(z) = \frac{1+z}{2}$, $h_0 = 1/2$ and $h_1 = 1/2$. The box function satisfies

$$\phi(x) = 2\left(\frac{1}{2}\phi(2x) + \frac{1}{2}\phi(2x-1)\right).$$

For the hat function,

$$h(z) = \left(\frac{1+z}{2}\right)^2 = \frac{1}{4} + \frac{1}{2}z + \frac{1}{4}z^2.$$

The hat function satisfies

$$\phi(x) = 2\left(\frac{1}{4}\phi(2x) + \frac{1}{2}\phi(2x-1) + \frac{1}{4}\phi(2x-2)\right).$$

When $r = 0$, $h(z) = 1$. We find that the function that satisfies the scaling relation

$$\phi(x) = 2\phi(2x)$$

is the Dirac delta function $\delta(x)$.

Scaling function and mask The above box spline functions are called scaling functions. A general scaling function on $\mathbb{R}$ is defined as the follows.

Definition 3.2. *Given a series $h(z) = \sum_{k=-\infty}^{\infty} h_k z^k$ with $h(1) = 1$, the function ϕ satisfying*

$$\phi(x) = 2\sum_{k=-\infty}^{\infty} h_k \phi(2x-k) \tag{3.6}$$

is called a scaling function with mask $h(z)$.

Remark. The factor 2 here is for the consistence condition $h(1) = 1$. Indeed, we have

$$\int \phi(x)\,dx = 2\sum_{k=-\infty}^{\infty} h_k \int \phi(2x-k)\,dx = (\sum_k h_k)\int \phi(x)\,dx.$$

Thus, the consistence requires $h(1) = \sum_k h_k = 1$.

Construction of scaling function There are two standard ways to construct scaling functions.

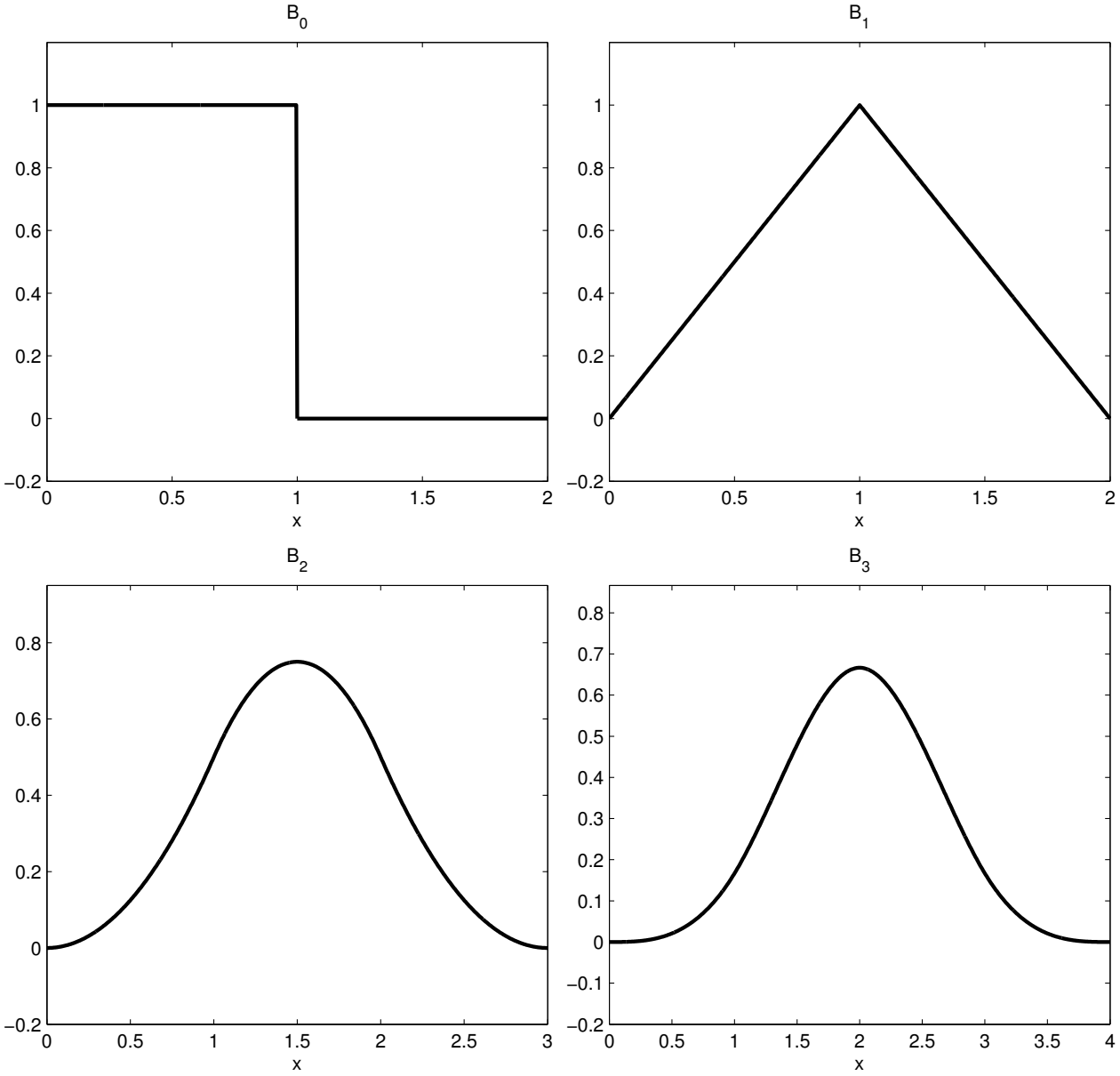
- Cascade algorithm:
$$\phi_{n+1}(x) = 2\sum_{k=-\infty}^{\infty} h_k \phi_n(2x-k)$$
with $\phi_0 = 1_{[0,1)}$.
- Fourier method:
$$\hat{\phi}(\xi) = h(e^{i\xi/2})\hat{\phi}(\xi/2).$$
We can find
$$\hat{\phi}(\xi) = \prod_{j=1}^{\infty} h\left(e^{\frac{\xi}{2^j}}\right).$$
- Subdivision scheme. This is a simple to construct a general function. We notice that if $\phi(x)$ is known at all integer points, then the scaling relation immediate gives us its values at half integer points. We can proceed this inductively and get the values of ϕ at $2^{-j}i$ points for all i and j. Eventually, the points $x_{ji} := 2^{-j}i$ is dense in $\mathbb{R}$, we obtain ϕ by a limiting process. This algorithm is called the subdivision algorithm: if we have computed $\phi(2^{-j+1}i)$ for all i, then we compute
$$\phi(2^{-j}i) = 2\sum_k h_k \phi(2^{-j+1}i - k) \text{ for all } i.$$
But how do we obtain $\phi(k)$ for $k \in \mathbb{Z}$? From the scaling relation
$$\phi(i) = 2\sum_{k=-\infty}^{\infty} h_k \phi(2i-k) = 2\sum_j h_{2i-j}\phi(j)$$
This is an eigenvalue problem. We can write it in matrix form:
$$\begin{pmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \\ \vdots \end{pmatrix} = 2\begin{pmatrix} h_0 & & & & \\ h_2 & h_1 & h_0 & & \\ h_4 & h_3 & h_2 & h_1 & h_0 \\ & & & & \end{pmatrix}\begin{pmatrix} \phi_0 \\ \phi_1 \\ \phi_2 \\ \vdots \end{pmatrix}$$

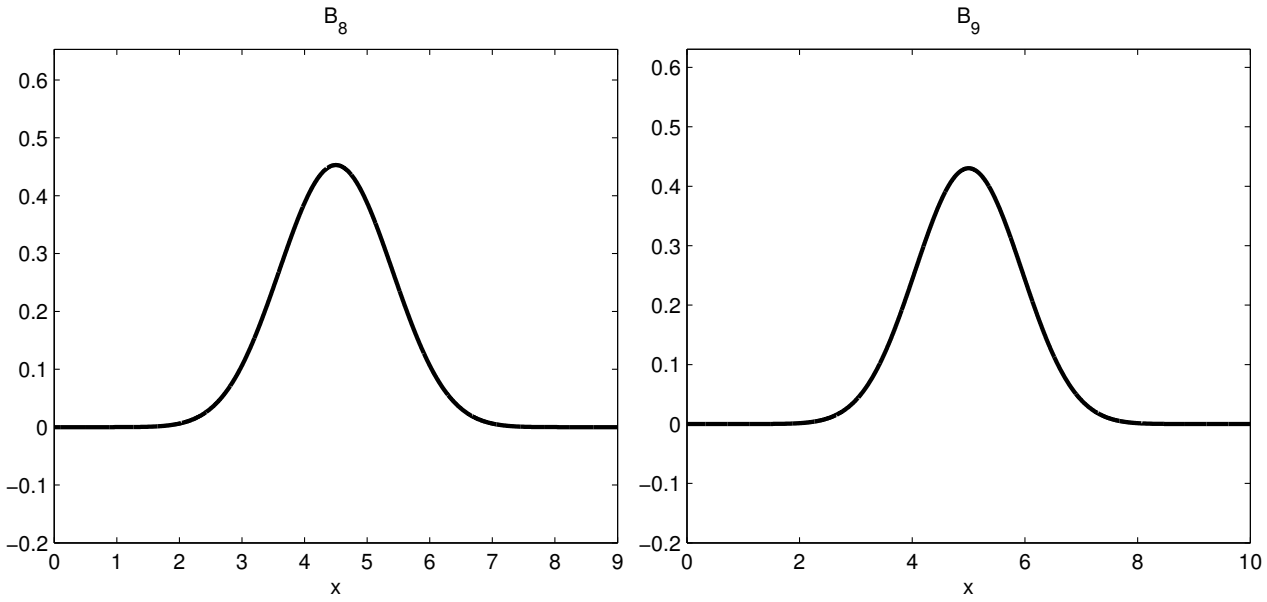
Lemma 3.7. *Suppose $h_k = 0$ for $k < 0$ and $k > r$, then the corresponding scaling function ϕ has support on $[0, r]$.*

Regularity We call a scaling function ϕ with mask $h(z)$ is of order r if -1 is a multiple root of $h(z) = 0$ with multiplicity r. In other words,
$$h(z) = \left(\frac{1+z}{2}\right)^r Q(z).$$
for some Q with $Q(-1) \neq 0$. We shall study the regularity of scaling functions later.

Example B-splines. The figures are B-spines, $B_{r-1} = 1_{[0,1)} * \cdots * 1_{[0,1)}$, r times convolution. The mask of B_{r-1} is

$$h(z) = \left(\frac{1+z}{2}\right)^r.$$





We shall come back to the construction of scaling function later.

Dual scaling functions ϕ (with mask $h(z)$) and $\tilde{\phi}$ (with mask $\tilde{h}(z)$).

Definition 3.3. *Two scaling functions ϕ and $\tilde{\phi}$ are dual to each other if*

$$\int \phi(x)\tilde{\phi}(x-i)\,dx = \delta_{0,i}$$

Proposition 2. *Two scaling functions ϕ and $\tilde{\phi}$ with masks $h(z)$ and $\tilde{h}(z)$ are dual to each other if and only if*

$$h(z)\tilde{h}(z^{-1}) + h(-z)\tilde{h}(-z^{-1}) = 1.$$

Let us postpone the construction of $\tilde{h}$ and $\tilde{\phi}$ later. Suppose we have such dual scaling function. We can define

- translation: $\phi(x-i)$
- dilation: $\phi(2^j x)$
- dilation and translation: $\phi_{j,i} := \phi(2^j x - i)$. It is a local function at $2^j i$ at scale 2^{-j}.

We can represent a function f successively by

$$f(x) \sim f_j(x) := \sum_{i\in\mathbb{Z}} (f, \tilde{\phi}_{j,i})\phi_{j,i}(x)$$

The meaning of $(f, \phi_{j,i})$: local average of f at scale 2^{-j} at $x_{j,i} := 2^{-j}i$.

Theorem 3.10. *Strang-Fix theorem:*

$$\|f - f_j\|_{L^2} = O(2^{-jr}).$$

3.9 Approximation by Wavelets and Framelets

3.9.1 Motivations

Wavelets are a local oscillators. They are designed to provide *multi-resolution analysis* for functions, signals, images and data. The flamelets are similar stuffs except they form redundant basis instead of basis in function space.

Images are multiscale Images can be composed of

- macroscopic parts
- microscopic parts (texture, fractal,...)
- noises

Image can be presented more effectively by multi-resolution representation. For example, the image of Tiffany (Figure 3.9.1) has 256×256 pixels. The wavelet representation Figure 3.9.1 of this image has the same number of coefficients, but most of them are closed to zeros. Therefore, this image can be compressed by setting those small coefficients to zeros. The compressed image here uses only $1/7.4873$ of the original wavelet coefficients and the corresponding recovered image shown in Figure 3.9.1, which has almost no difference from the original one from the view of human eyes.

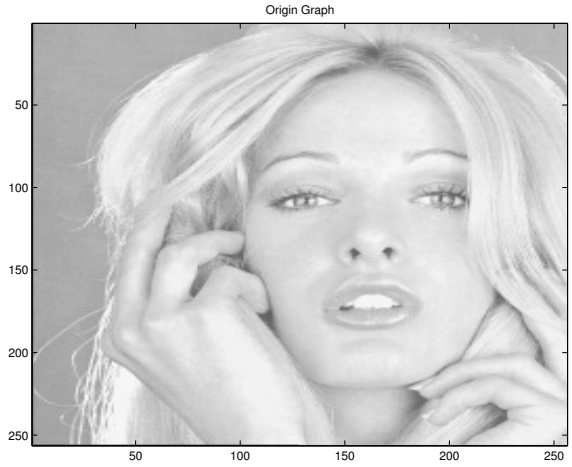


Figure 3.1: Original image

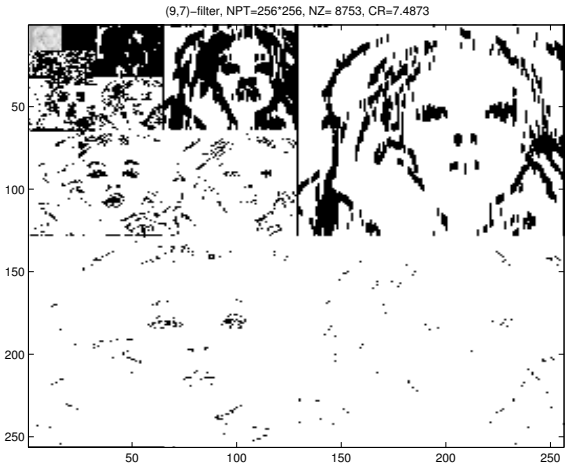


Figure 3.2: Wavelet transform of the image.

Tools for multiscale representation of functions Signals can be viewed as one-dimensional functions, whereas Images and video can be thought as two-dimensional and three-dimensional functions. A function in time can be represented by atoms in time-domain, or in frequency domain, or in both time-frequency domain. Examples of these atoms are

- Time (space) domain representation: by local functions such as splines, scaling functions
- Frequency (scale) domain representation:
  - Fourier analysis
  - Spectral representation (by eigen-modes of special systems such as Chebyshev, Legendre, etc)
- Time-frequency (Spatial-Scale) representation:
  - Variants of Fourier methods:
    - windowed Fourier (Garbor transform)
    - Wigner distribution (Wigner transform)
    - Empirical modes decomposition (Hilbert-Huang transform)
  - Wavelets
    - continuous wavelet
    - discrete wavelets
    - curvelets, shearlets,
  - Framelets, redundent bases

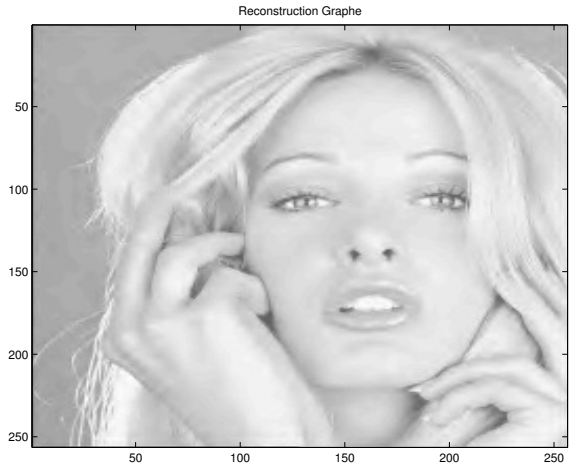


Figure 3.3: Image reconstructed from a truncated wavelet coefficients with compression ratio 7.4873.

Below, we give a simple example of single-resolution and multi-resolution representations of an one-dimensional function (or one-dimensional data). Imagine we have a function $c(x)$ defined on $\mathbb{R}$. Let us define the grid points

$$x_{j,k} := 2^{-j}k.$$

- Single resolution representation: The data $c_{J,k} := c(x_{J,k})$, $k = 0, ..., 2^J - 1$ can be thought as sampling the continuous signal $c(x)$ on $[0, 1]$ at rate 2^{-J}. This is a single-resolution representation:

  $$c_J = (c_{J,k}), k = 0, ..., 2^J - 1.$$

  We may also choose

  $$c_{J,k} := 2^{-J} \int_{x_{j,k}}^{x_{j,k+1}} c(x)\, dx$$

  which is the average of c in a 2^{-J} neighborhood of $x_{j,k}$.

- Multi-resolution representation: We perform the following transformation recursively for $j = J$ to $j = 1$. At scale j, we assume that we are given $c_{j,k}$. We define

  $$\begin{aligned} c_{j-1,k} &= (c_{j,2k} + c_{j,2k+1}) \\ d_{j-1,k} &= (c_{j,2k} - c_{j,2k+1}) \end{aligned}$$

  $$k = 0, ..., 2^{j-1} - 1.$$

  The meaning of these quantities is

  $$\begin{aligned} c_{j,k} &:= \text{averages of data at resolution level } j \\ d_{j,k} &:= \text{fluctuations of data at resolution level } j \end{aligned}$$

Here, j is the scale index and k is the location index. The representation

$$(c_0, d_0, d_1, \cdots, d_{J-1}), \quad c_j = (c_{j,k})_k,\ d_j = (d_{j,k})_k.$$

is called an multi-resolution of c. It means that c_J is represented as

local averages at coarsest level + fluctuations at various levels.

The transformation

$$T_J : (c_J) \mapsto (c_0, d_0, d_1, \cdots, d_{J-1}).$$

is called the discrete Haar wavelet transform. It is a simple example of discrete wavelet transform. Its inverse transform T_j^{-1} can be obtained recursively by performing

$$\begin{aligned} c_{j,2k} &= \frac{1}{2}\left(c_{j-1,k} + d_{j-1,k}\right) \\ c_{j,2k+1} &= \frac{1}{2}\left(c_{j-1,k} - d_{j-1,k}\right). \end{aligned}$$

An advantages of multi-resolution representation is that the representation is more efficient (or sparse) if the underlying data are piecewise smooth. Therefore, it is useful for image compression. We can also have other kinds of multi-resolution representations.

Example 1 Let $c_{j,k}$ be the value of $c(x)$ at $x_{j,k} := 2^{-j}k$. We define

$$\begin{aligned} &c_{j-1,k} = c_{j,2k} \\ &d_{j-1,k} = c_{j,2k+1} - \frac{1}{2}\left(c_{j,2k} + c_{j,2k+2}\right), k = 0, ..., 2^j - 1,\ j = L, ..., J. \end{aligned}$$

Thus,

- $c_{j,k}$: data of c at resolution level j
- $d_{j,k}$: interpolation error (use piecewise linear interpolant) at resolution level j

Example 2

$$\begin{aligned} c_{j-1,k} &= c_{j,2k} \\ d_{j-1,k} &= c_{j,2k+1} - L(x_{j,2k+1}; x_{j,2k-2}, x_{j,2k}, x_{j,2k+2}, x_{j,2k+4}) \end{aligned}$$

where $L(x; x_{j,2k-2}, x_{j,2k}, x_{j,2k+2}, x_{j,2k+4})$ is the Lagrange interpolant of degree 3 which interpolates $c(\cdot)$ at $x_{j,2k-2}, x_{j,2k}, x_{j,2k+2}, x_{j,2k+4}$.

3.9.2 General Discrete Wavelet Transform

Discrete Wavelet transform In wavelet representation of an one-dimensional data, the data are represented multiscale form $(c_{j,k}, d_{j,k})$, where j is a scale index and k is a location index. The data $c_j = (c_{j,k})_{k\in\mathbb{Z}}$ are the "averages" of $c(\cdot)$ at scale j. The data $d_j = (d_{j,k})_{k\in\mathbb{Z}}$ are the "fluctuations" of $c(\cdot)$ at scale j. A wavelet transform decomposes averages at finest scale into averages and fluctuations at various scales. Such transformation depends on two sets of coefficients $\{h_k\}, \{g_k\}$. Usually, only finite of them are non-zero. The data at finest scale, say c_J can be decomposed into averages and fluctuations at coarser scales by the following recursive process:

$$\begin{cases} \text{the low-pass data: } c_{j-1,i} = \sqrt{2}\sum_k h_k c_{j,2i-k}, \\ \text{the high-pass data: } d_{j-1,i} = \sqrt{2}\sum_k g_k c_{j,2i-k}. \end{cases} \tag{3.7}$$

We perform this recursively for $j = J, ..., 1$ and obtain the transformation

$$T_J : \ c_J \mapsto (c_0, d_0, d_1, \cdots, d_{J-1}).$$

Thus, c_J is decomposed into

local averages at coarsest scale + local fluctuations at various scales.

We call the transformation T_J the *discrete wavelet transform*.

The inverse transformation T_J^{-1}, which depends on two sets of coefficients $\{\tilde{h}_k\}$ and $\{\tilde{g}_k\}$, can be performed recursively by the following reconstruction process. For $j = 1, ..., J$,

$$c_{j,i} = \sqrt{2}\sum_k \left[\tilde{h}_{2k-i} c_{j-1,k} + \tilde{g}_{2k-i} d_{j-1,k}\right]. \tag{3.8}$$

We can obtain c_J from $(c_0, d_0, d_1, ..., d_{J-1})$.

Condition for perfect reconstruction The discussion below is to find conditions on $h, g, \tilde{h}$ and $\tilde{g}$ such that we can have perfect reconstruction. This means that we can transform c_j into (c_{j-1}, d_{j-1}) by h and g, and transform them back perfectly by $\tilde{h}$ and $\tilde{g}$. In order to find perfect reconstruction condition, we introduce the notion of generating functions.

Definition 3.4. *Given a sequence of coefficients $\{h_k\}_{k\in\mathbb{Z}}$, we define the generating function:*

$$h(z) = \sum_{k\in\mathbb{Z}} h_k z^k.$$

The generating function $h(z)$ is sometimes called mask or filter bank.

Lemma 3.8. *It holds*

$$\sum_i (\sum_j a_{i-j} b_j) z^i = a(z)b(z)$$
$$\sum_i (\sum_j a_{i+j} b_j) z^i = a(z)b(z^{-1})$$
$$\sum_i (\sum_j a_{2i-j} b_j) z^{2i} = \frac{1}{2}(a(z)b(z) + a(-z)b(-z))$$
$$\sum_i (\sum_j a_{i-2j} b_{2j}) z^i = a(z)\frac{1}{2}(b(z) + b(-z))$$

Proposition 3. *The perfect reconstruction condition for (3.7) and (3.8) is*

$$g(z) = z\tilde{h}(-z^{-1}) \tag{3.9}$$
$$\tilde{g}(z) = zh(-z^{-1}), \tag{3.10}$$

and

$$h(z)\tilde{h}(z^{-1}) + h(-z)\tilde{h}(-z^{-1}) = 1. \tag{3.11}$$

Proof. 1. The decomposition (3.7) can be expressed in terms of generating functions by

$$c_{j-1}(z^2) = \frac{\sqrt{2}}{2}\left(h(z)c_j(z) + h(-z)c_j(-z)\right)$$
$$d_{j-1}(z^2) = \frac{\sqrt{2}}{2}\left(g(z)c_j(z) + g(-z)c_j(-z)\right)$$

The reconstruction (3.8) can be expressed as

$$\begin{aligned} c_j(z) &= \sqrt{2}\left(\tilde{h}(z^{-1})c_{j-1}(z^2) + \tilde{g}(z)d_{j-1}(z^2)\right) \\ &= \tilde{h}(z^{-1})\left(h(z)c_j(z) + h(-z)c_j(-z)\right) \\ &\quad + \tilde{g}(z^{-1})\left(g(z)c_j(z) + g(-z)c_j(-z)\right) \end{aligned}$$

This gives

$$h(z)\tilde{h}(z^{-1}) + g(z)\tilde{g}(z^{-1}) = 1$$
$$h(-z)\tilde{h}(z^{-1}) + g(-z)\tilde{g}(z^{-1}) = 0.$$

2. If we choose

$$g(z) = z\tilde{h}(-z^{-1})$$
$$\tilde{g}(z) = zh(-z^{-1}),$$

then the second equation is satisfied automatically

$$h(z)\tilde{h}(-z^{-1}) + z\tilde{h}(-z^{-1})(-z)^{-1}h(z) = 0.$$

The first equation becomes

$$h(z)\tilde{h}(z^{-1}) + h(-z)\tilde{h}(-z^{-1}) = 1.$$

□

Design filter banks To design a wavelet transform, we need to design the four sets of filter banks $h(z)$, $g(z)$, $\tilde{h}(z)$ and $\tilde{g}(z)$ such that they satisfy the perfect reconstruction condition. The filter banks $h(z)$ and $g(z)$ are called analysis filter banks, while $\tilde{h}(z)$ and $\tilde{g}(z)$ are called synthesis filter banks. The mask $h(z)$ plays the role of averaging. A simple average filter is $h(z) = (1+z)/2$. This means that -1 is a root of $h(z)$. The mask $g(z)$ plays a role of differencing. A simple differencing filter is $g(z) = (1-z)/2$. In general, we look for $h(z)$ and $g(z)$ having the following properties

$$h(z) \text{ contains a factor } \left(\frac{1+z}{2}\right)^r$$
$$g(z) \text{ contains a factor } \left(\frac{1-z}{2}\right)^{\tilde{r}}$$

for some positive integers r and $\tilde{r}$. Example of these analytic masks are

- $h(z) = \frac{1+z}{2}$: averaging
- $h(z) = z^{-1}\left(\frac{1+z}{2}\right)^2$: 2nd order averaging
- $g(z) = \frac{1-z}{2}$: differencing
- $g(z) = z^{-1}\left(\frac{1-z}{2}\right)^2$: 2nd order differencing
- The filter bank $g(z) = \left(\frac{1-z}{2}\right)^p$ annihinate monomials $x^0, x^1, ..., x^{p-1}$.

We can obtain $g(z)$ and $\tilde{g}(z)$ from $h(z)$ and $\tilde{h}(z)$ via the equations (3.9) and (3.10). Equation (3.11) gives condition on $h(z)$ and $\tilde{h}(z)$. In this equation, only the product $h(z)\tilde{h}(z^{-1})$ is involved. Thus, we call

$$h(z)\tilde{h}(z^{-1}) = H_0(z)$$

and (3.11) reads

$$H_0(z) + H_0(-z) = 1.$$

Notice that from (3.9) that $g(z)$ containing a factor $(1-z)/2$ is equivalent to $\tilde{h}(z)$ containing a factor $(1+z)/2$. Thus, the desired property that $h(z)$ and $g(z)$ contain factors $(1+z)/2$ and $(1-z)/2$ respectively is equivalent to that H_0 contains a factor $(1+z)/2$.

Let us summarize the procedure of designing filter banks as the follows.

- Find a mask $H_0(z)$ which satisfies

$$H_0(z) + H_0(-z) = 1.$$

  and also containing a factor $((1+z)/2)^{r+\tilde{r}}$.

- Split $H_0(z)$:

$$H_0(z) = h(z)\tilde{h}(z^{-1})$$

  such that $h(z)$ containing $((1+z)/2)^r$ and $\tilde{h}(z)$ containing $((1+z)/2))^{\tilde{r}}$.

- Define

$$g(z) = z\tilde{h}(-z^{-1}) \quad \tilde{g}(z) = zh(-z^{-1}).$$

Design of $H_0(z)$ We have seen that $H_0(z)$ should satisfies $H_0(z) + H_0(-z) = 1$. This is equivalent to

- $H_0(z)$ has no even order terms except $k = 0$
- The constant coefficient is 1/2.

In addition, $H_0(z)$ should contains factor $(1+z)/2$. Here are two tricks to design $H_0(z)$.

Method I Let $H(z) = z^{-1}\left(\frac{1+z}{2}\right)^2$, $K(z) = H(-z)$, we have

$$H(z) + K(z) = 1$$

We raise it to nth power

$$(H+K)^n = 1^n.$$

From the binomial expansion, we can easily find H_0. Let us see the following examples.

1. $n = 2$:

$$\begin{aligned}(H+K)^2 &= H^2 + HK + KH + K^2 \\ &= H(H+K) + K(K+H)\end{aligned}$$

   We choose $H_0(z) = H(H+K)$

2. $n = 3$:

$$\begin{aligned}(H+K)^3 &= H^3 + 3H^2K + 3K^2H + K^3 \\ &= H^2(H+3K) + K^2(K+3H)\end{aligned}$$

   We choose $H_0(z) = H^2(H+3K)$

3. $n = 5$:

$$(H + K)^5 = H^3(H^2 + 5HK + 10K^2) + K^3(K^2 + 5HK + 10H^2)$$

We choose $H_0(z) = H^3(H^2 + 5HK + 10K^2)$.

4. (Homework) Derive general formulae for general n.

Example:

- $r = 1$: $H_0 = 2^{-2}(1, 2, 1)$
- $r = 2$: $H_0 = 2^{-4}(-1, 0, 9, 16, 9, 0, -1)$

Method II We shall construct $H_0(z)$ having the following form

$$H_0(z) = H(z)^r Q_r(z),$$

where

$$H(z) = z^{-1}\left(\frac{1+z}{2}\right)^2 = \frac{z + 2 + z^{-1}}{4}$$

is the symmetric average polynomial, $Q_r(z)$ is the polynomial of lowest degree such that $H_0(z) + H_0(-z) = 1$.

Proposition 4. *The polynomial $Q_r(z)$ has the form:*

$$Q_r(z) = \sum_{n=0}^{r-1} \begin{pmatrix} r+n-1 \\ n \end{pmatrix} \left(\frac{2 - z - z^{-1}}{4}\right)^n$$

Lemma 3.9 (Bazout)**.** *Two polynomial p_1 and p_2 with degree n_1 and n_2 are relativily prime, then there exist unique q_1 and q_2 of degree $n_2 - 1$ and $n_1 - 1$, respectively, such that $p_1(x)q_1(x) + p_2(x) + q_2(x) = 1$.*

Proof. of the Proposition.

1. Proof I: Denote $H(-z)$ by $K(z)$ From

$$(H(z) + K(z))^{2r-1} = 1$$

We get

$$\left(\sum_{n=0}^{r-1} + \sum_{n=r}^{2r-1}\right) \begin{pmatrix} 2r-1 \\ n \end{pmatrix} H^{2r-1-n}K^n = H_0(z) + H_0(-z)$$

where

$$H_0(z) = H(z)^r \sum_{n=0}^{r-1} \begin{pmatrix} 2r-1 \\ n \end{pmatrix} H^{r-1-n}K^n = H(z)^r Q(z)$$

We notice that $Q(z)$ has no factor $H(z)$. Since the degree of Q_r is from $(-r+1)$ to $r-1$, we obtain this $Q(z)$ must be the unique $Q_r(z)$ of the lowest degree which satisfies $H^r(z)Q_r(z) + H^r(-z)Q_r(-z) = 1$.

2. Proof II. Notice that

$$H(z) = 1 - H(-z)$$

The polynomial $H(z)$ and $H(-z)$ are relatively prime. By Bezout's theorem and the symmetric property of $H(z)$, we see that there exists a unique $Q_r(z)$ of degree from $-r+1$ to $r-1$ such that

$$H^r(z)Q_r(z) + H^r(-z)Q_r(-z) = 1$$

we get

$$\begin{aligned} Q_r(z) &= \frac{1}{H^r(z)}\left(1 - H^r(-z)Q_r(-z)\right) \\ &= \frac{1}{(1-H(-z))^r}\left(1 - H^r(-z)Q_r(-z)\right) \\ &= \sum_{n=0}^{\infty} \binom{-r}{n} (-H(-z))^n \left(1 - H^r(-z)Q_r(-z)\right) \\ &= \sum_{n=0}^{r-1} \binom{r+n-1}{n} H^n(-z) + O(H^r(-z)) \end{aligned}$$

Since degree of $Q_r(z)$ is from $-r+1$ to $r-1$, we obtain that the $O(H^r(-z))$ term is identical to zero.

□

3.9.3 Examples of filter banks

In this subsection, we introduction concrete examples that are popular in use. These include

- Lagrange interpolation filter banks,
- Daubechies orthogonal filter banks
- Cohen-Daubechies-Feauveau filter banks.

Their designs are all based on the basic filter bank that we construct in the last section, namely $H_0(z) = H^r(z)Q_r(z)$.

Lagrange interpolation filter banks In the Lagrange interpolation filter bank, we split

$$H_0(z) = 1 \cdot H_0(z) = h(z) \cdot \tilde{h}(z).$$

That is,

$$\begin{aligned} h(z) &= 1, \quad \tilde{g}(z) = z \\ \tilde{h}(z) &= \frac{1}{2}\left(1 + \sum_{m=-r+1}^{r} \prod_{\substack{-r < n \le r \\ n \neq m}} \frac{1-2n}{2m-2n} z^{2n-1}\right) \\ g(z) &= \frac{1}{2}\left(z - \sum_{m=-r+1}^{r} \prod_{\substack{-r < n \le r \\ n \neq m}} \frac{1-2n}{2m-2n} z^{2n}\right) = z\tilde{h}(-z^{-1}) \end{aligned}$$

Proposition 5. *The Lagrange filter bank has the properties*

$$\begin{aligned} c_{j-1,k} &= c_{j,2k} \\ d_{j-1,k} &= c_{j,2k+1} - L(c_{j,2(k-r+1)}, \cdots, c_{j,2(k+r)}) \end{aligned} \tag{3.12}$$

the Lagrange interpolation error.

Proof. 1. Let us compare the two filter bank: $g(z)$ above and the interpolation error filter bank (3.12).

2. Since $g(z)$ contains a factor $z^{-r}\left(\frac{1-z}{2}\right)^{2r}$, it annihilates polynomials of degree less than $2r-1$. This means that
$$\sum_k g_k x_k^m = 0, \text{ for all } m = 0, 1, ..., 2r-1,$$
where $x_k := k$.

3. We notice that degree of $g(z)$ is from $-2r+1$ to $2r-1$. Thus, $g(z)$ is the shortest filter bank that can annihilate x^m for $m = 0, ..., 2r-1$.

4. The Lagrange interpolation error (3.12) has the same property. Because this interpolation is unique. Thus, $g(z)$ must be the Lagrange interpolation filter bank (3.12).

□

Daubechies orthogonal filter banks

- Since $Q_r(z) = Q_r(z^{-1})$, we can split
$$Q_r(z) = Q(z)Q(z^{-1})$$
where $Q(z) = a_0 + a_1 z + \cdots + a_{r-1} z^{r-1}$.

- We can split $H(z)^r$ into
$$z^{-r}\left(\frac{1+z}{2}\right)^{2r} = \left(\frac{1+z}{2}\right)^r \left(\frac{1+z}{2}\right)^r z^{-r} = \left(\frac{1+z}{2}\right)^r \left(\frac{1+z^{-1}}{2}\right)^r$$

- Then we can split $H_0(z)$ into $h(z)h(z^{-1})$:

$$H_0(z) = \underbrace{Q(z^{-1})\left(\frac{1+z^{-1}}{2}\right)^r}_{h(z^{-1})} \underbrace{Q(z)\left(\frac{1+z}{2}\right)^r}_{h(z)}$$

We shall see in the next section that the scaling function $\phi(x)$ and $\tilde{\phi}(x)$ associated with the masks $h(z)$ and $\tilde{h}(z)$ are identical, and $\phi(\cdot - k)$ and $\phi(\cdot - \ell)$ are orthogonal for $k \neq \ell$.

Cohen-Daubechies-Feauveau biorthogonal filter banks

- Split $H^r Q_r$ into

$$\underbrace{Q_r(z)\left(\frac{1+z^{-1}}{2}\right)^{\tilde{d}}}_{\tilde{h}(z^{-1})} \underbrace{\left(\frac{1+z}{2}\right)^d}_{h(z)}$$

  where $d + \tilde{d} = 2r$.

- $c_{j,k}$ local averages of order d at level j
- $d_{j,k}$ local fluctuations of order $\tilde{d}$ at level j

3.9.4 Multi-resolution Analysis framework

In this section, we shall construct scaling functions and wavelets corresponding to the masks $h(z)$ and $g(z)$. They are basis (or atoms) of the $L^2(\mathbb{R})$ space. These bases give a multi-resolution structure of the $L^2(\mathbb{R})$ space.

Scaling functions and Wavelets Given four set of masks $h(z)$, $g(z)$, $\tilde{h}(z)$ and $\tilde{g}(z)$ satisfying the perfect reconstruction condition. Associate with them, we define

$$\begin{aligned}
\phi(x) &= 2\sum_k h_k \phi(2x-k), \\
\psi(x) &= 2\sum_k g_k \phi(2x-k). \\
\tilde{\phi}(x) &= 2\sum_k \tilde{h}_k \tilde{\phi}(2x-k), \\
\tilde{\psi}(x) &= 2\sum_k \tilde{g}_k \tilde{\phi}(2x-k),
\end{aligned}$$

The functions ϕ and $\tilde{\phi}$ are called scaling and dual scaling functions, respectively, whereas $\psi(x)$ and $\tilde{\psi}(x)$ are called wavelet and dual wavelet, respectively. Through dilation and translation, we define

$$\psi_{j,i}(\cdot) = 2^{\frac{j}{2}} \psi(2^j \cdot - i).$$

We define $\phi_{j,i}$, $\tilde{\phi}_{j,i}$ and $\tilde{g}_{j,i}$ similarly. They are called atoms in a vague terminology.

The regularity of ϕ depends on the factor $(1+z)/2$.

Definition 3.5. *A scaling function with mask $h(z)$ is called of order r if -1 is a multiple root of $h(z)$ with multiplicity r.*

Our goal is to represent a function $u \in L^2(\mathbb{R})$ in terms of $\phi_{j,i}$ or $\psi_{j,i}$, etc. Below, we outline the main results. Their proofs will be given later.

Dual properties

Proposition 6. *For each fixed j, it holds*

$$(\phi_{j,i}, \tilde{\phi}_{j,k}) = \delta_{i,k}. \tag{3.13}$$

For any j, ℓ and i, k, it holds

$$(\psi_{j,i}, \tilde{\psi}_{\ell,k}) = \delta_{i,k}\delta_{j,\ell}. \tag{3.14}$$

Proof. 1. We only need to prove

$$\int \phi(x-i)\tilde{\phi}(x-l)\,dx = \delta_{il}$$

We assume that ϕ can be obtain by the cascade algorithm (see next subsection)

$$\phi^n = 2\sum_k h_k \phi^{n-1}(2x-k), \quad \phi^0 = \chi_{[0,1)}$$

and $\phi^n \to \phi$.

2. We prove for all $n \geq 0$, it holds

$$\int \phi^n(x-i)\tilde{\phi}^n(x-l)\,dx = \delta_{il}. \tag{3.15}$$

This can be proved by induction. For $n = 0$,

$$\int \phi^0(x-i)\tilde{\phi}^0(x-l)\,dx = \int \chi_{[0,1)}(x-i)\chi_{[0,1)}(x-l)\,dx = \delta_{il}.$$

Suppose (**??**) holds up to n,

$$\begin{aligned}(\phi^{n+1}(x-i), \tilde{\phi}^{n+1}(x-j)) &= \int 4\sum_k h_k \phi^n(2(x-i)-k)\sum_\ell \tilde{h}_\ell \tilde{\phi}^n(2x-j)-\ell)\,dx \\ &= 2\sum_k \sum_\ell h_k \tilde{h}_\ell \delta_{2i+k,2j+\ell} \\ &= 2\sum_k h_k \tilde{h}_{2(i-j)+k}\end{aligned}$$

The generating function of $2\sum_k h_k \tilde{h}_{2i+k}$ is

$$\sum_i \sum_k 2\tilde{h}_{2i+k} h_k z^{-2i} = h(z)\tilde{h}(z^{-1}) + h(-z)\tilde{h}(-z^{-1}) = 1.$$

□

Multi-resolution structure We define

$$\begin{aligned} V_j &= \text{span}\,\{\phi_{j,k}\}_{k\in Z}, \quad W_j = \text{span}\,\{\psi_{j,k}\}_{k\in Z} \\ \tilde{V}_j &= \text{span}\,\{\tilde{\phi}_{j,k}\}_{k\in Z}, \quad \tilde{W}_j = \text{span}\,\{\tilde{\psi}_{j,k}\}_{k\in Z} \end{aligned}$$

Then we have

Proposition 7.

$$\begin{aligned} V_{j+1} &= V_j + W_j, \quad \tilde{V}_{j+1} = \tilde{V}_j + \tilde{W}_j \\ W_j &\perp \tilde{V}_j, \quad \tilde{W}_j \perp V_j \\ L^2(\mathbb{R}) &= \overline{\cup V_j}, \quad L^2(\mathbb{R}) = \overline{\cup \tilde{V}_j} \\ L^2(\mathbb{R}) &= \oplus W_j, \quad L^2(\mathbb{R}) = \oplus \tilde{W}_j \end{aligned}$$

In other words, we can expand $u \in L^2(\mathbb{R})$ as

$$u = \sum_j \sum_k (u, \psi_{j,k})\tilde{\psi}_{j,k} = \sum_j \sum_k (u, \tilde{\psi}_{j,k})\psi_{j,k}$$

In connection with the wavelet coefficients in the previous section, we have

$$\begin{aligned} c_{j.k} &:= (u, \phi_{j,k}) \text{ (local averaging)} \\ d_{j.k} &:= (u, \psi_{j,k}) \text{ (local differencing)} \end{aligned}$$

Riesz basis property

Theorem 3.11. *If ϕ is a scaling function of order r with $r \geq 1$, then $\phi_{0,k}$ constitute a Riesz basis in $V_0 :=$ span $\{\phi_{0,k}\}_{k\in Z}$, i.e. there exists two constants $A > 0$ and $B < \infty$ such that*

$$A \sum_k |c_k|^2 \leq \| \sum_k c_k \phi_{0,k}^{[r]} \|^2 \leq B \sum_k |c_k|^2$$

Theorem 3.12. *There exist positive constant γ, $\tilde{\gamma}$, Γ and $\tilde{\Gamma}$ such that for any $u \in L^2(\mathbb{R})$, it holds*

$$\begin{aligned} \gamma \sum_{i,j\in Z} |d_{j,i}|^2 &\leq \| \sum_{i,j\in Z} d_{j,i}\psi_{j,i} \|^2 \leq \Gamma \sum_{i,j\in Z} |d_{j,i}|^2 \\ \tilde{\gamma} \sum_{i,j\in Z} |\tilde{d}_{j,i}|^2 &\leq \| \sum_{i,j\in Z} \tilde{d}_{j,i}\tilde{\psi}_{j,i} \|^2 \leq \tilde{\Gamma} \sum_{i,j\in Z} |\tilde{d}_{j,i}|^2 \end{aligned}$$

Approximation power

Theorem 3.13 (Strang-Fix, Unser)**.** *Suppose ϕ is of pth order and V_j is the span of $\{\phi_{j,k}\}_{k\in Z}$. Let Q_j be any projection from L^2 onto V_j. Then*

$$\|Q_j u - u\|_{L^2} = C_Q 2^{-jp} + O(2^{-j(p+1)})$$

3.9.5 Construction of scaling functions and wavelets

Goal: To construct a function ϕ such that $V_j := \text{span}\,\{\phi_{j,k}\}_{k\in Z}$ constitutes a multi-resolution analysis. It is also served as a basic element to build wavelet functions.

Definition 3.6. *A function ϕ is called refinable (or a scaling function) if there exist coefficients $\{h_k\}_{k\in\mathbb{Z}}$ such that*

$$\phi(x) = 2\sum_k h_k \phi(2x-k)$$

Let $h(z) = \sum_{k\in\mathbb{Z}} h_k z^k$, called the generating function of $\{h_k\}$. It is necessarily that $h(1) = 1$. $h(z)$ is called the mask of ϕ.

We outline the theory:

- Existence and construction of the scaling functions
- Three examples of scaling functions
  - B-splines
  - Lagrange interpolation functions
  - Daubechies orthogonal scaling functions.
- Properties of scaling functions:
  - Support
  - Regularity
- Riesz basis property
- Approximation power

Construction and Existence of scaling function We introduce three methods to construct scaling functions.

- **Cascade algorithm**

$$\begin{aligned}\phi^n(x) &= 2\sum_k h_k \phi^{n-1}(2x-k)\\ \phi^0 &= 1_{[0,1)}\end{aligned}$$

  The function ϕ^n will converge to ϕ.

- **Fourier method:** Taking Fourier transform on the refinable equation, we obtain

$$\widehat{\phi}(\xi) = m(\frac{\xi}{2})\widehat{\phi}(\frac{\xi}{2})$$

where $m(\xi) = h(e^{i\xi})$. Performing this successively and taking the normalization $\widehat{\phi}(0) = 1$ (i.e. $\int \phi(x)\,dx = 1$), we obtain

$$\widehat{\phi}(\xi) = \prod_{j=1}^{\infty} m\left(\frac{\xi}{2^j}\right)$$

- **Subdivision scheme**

  1. Step 1: Find $\{\phi(k)\}_{k\in Z}$ by solving the eigen system:

  $$\phi(i) = 2\sum_k h_{2i-k}\phi(k)$$

  2. Step 2: Find the value of ϕ at $x_{j+1,k}$ points recursively by the subdivision scheme

  $$\phi(2^{-(j+1)}i) = 2\sum_k h_k \phi(2^{-j}i - k)$$

  Or equivalently,

  $$\phi(x_{j+1,i}) = S_h(\phi(x_{j,\cdot}))_i$$

  where $S_h : \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z})$ defined by

  $$(S_h b)_i = \sum_k 2h_{i-2k}b_k.$$

Remarks on subdivision scheme

1. The subdivision scheme can be understood through the superposition of S_h. First, a general $b : \mathbb{Z} \to \mathbb{R}$ can be written as $b = b_i\delta_i$, where $\delta_i \in \ell^2(\mathbb{Z})$ is defined by

   $$\delta_i(j) = \delta(i-j) = \begin{cases} 1 & mboxifi = j \\ 0 & \text{otherwise.} \end{cases}$$

   Thus, $S_h b = \sum_i b_i S_h \delta_i$.

2. $S_h\delta_0 = h$.

Lemma 3.10. *If $h(1) = 1$ and $m(\xi) = h(e^{i\xi})$ is Lipschitz continuous at 1, then the corresponding scaling function ϕ exists as a distribution.*

Proof. The assumption for $m(\xi)$ is

$$|m(\xi)| = |m(0) + (m(\xi) - m(0))| \le 1 + C|\xi| \le e^{C|\xi|}.$$

Hence

$$\prod_{j=1}^{\infty}\left|m\left(\frac{\xi}{2^j}\right)\right| \le \exp\left(\sum_{j=1}^{\infty} C|2^{-j}\xi|\right) \le e^{C|\xi|}.$$

The convergence is absolute and uniformly for ξ in any compact set in $\mathbb{C}$. Thus, the Fourier inversion of this infinite product function exists as a distribution. □

Proposition 8. *Cascade algorithm is equivalent to subdivision: algorithm*

$$\phi^n(x) = \sum_i (S_h^n \delta_0)_i \phi^0(2^n x - i)$$

Proof. 1. The subdivision scheme S_h is defined by

$$(S_h b)_i = \sum_k 2h_{i-2k} b_k$$

In particular, $2h = S_h \delta_0$.

2. A cascade algorithm is

$$\begin{aligned}
\phi^n(x) &= 2\sum_i h_i \phi^{n-1}(2x - i) \\
&= \sum_i (S_h \delta_0)_i \phi^{n-1}(2x - i) \\
&= 2^2 \sum_i h_i \sum_k h_k \phi^{n-2}(2^2 x - 2i - k) \\
&= \sum_\ell (\sum_i 2h_{\ell-2i} 2h_i) \phi^{n-2}(2^2 x - \ell) \\
&= \sum_\ell (S_h(2h))_e l \phi^{n-2}(2^2 x - \ell) \\
&= \sum_\ell S_h^2 \delta_0 \phi^{n-2}(2^2 x - \ell)
\end{aligned}$$

3. In general, we have

$$\phi^n(x) = \sum_i (S_h^n \delta_0)_i \phi^0(2^n x - i)$$

□

Three examples:

- $h(z) = \left(\frac{1+z}{2}\right)^r$, the corresponding ϕ is the B-spline of order r.
- $h(z) = z^{-r}\left(\frac{1+z}{2}\right)^{2r} Q_r(z)$ the corresponding refinable function is the Lagrange interpolating function. Here,

$$Q_r(z) = \sum_{n=0}^{r-1} \begin{pmatrix} r+n-1 \\ n \end{pmatrix} \left(\frac{2 - z - z^{-1}}{4}\right)^n$$

- $h(z) = \left(\frac{1+z}{2}\right)^r Q(z)$, where $Q_r(z) = Q(z)Q(z^{-1})$. This corresponds to Daubechies orthogonal scaling function.

B-spline

Proposition 9. *The scaling function corresponding to* $h(z) = \left(\frac{1+z}{2}\right)^r$ *is* $1_{[0,1)} * \cdots * 1_{[0,1)}$ *(r times)*

Proof. 1. Using Fourier method,

$$\widehat{\phi}(\xi) = \prod_{j=1}^{\infty} m\left(\frac{\xi}{2^j}\right)$$

where $m(\xi) = h(e^{i\xi})$. One can show that when $h(z) = \frac{1+z}{2}$, then $m(\xi) = e^{i\xi/2}\cos\left(\frac{\xi}{2}\right)$.

2. Using $\sin(2\xi) = 2\sin\xi\cos\xi$, we obtain

$$\begin{aligned}
\widehat{\phi}(\xi) &:= \prod_{j=1}^{\infty} m\left(\frac{\xi}{2^j}\right) \\
&= \exp\left[i\left(\sum_{j=2}^{\infty}\frac{\xi}{2^j}\right)\right]\prod_{j=2}^{\infty}\cos\left(\frac{\xi}{2^j}\right) \\
&= e^{i\xi/2}\prod_{j=2}^{\infty}\frac{\sin\left(\frac{\xi}{2^{j-1}}\right)}{2\sin\left(\frac{\xi}{2^j}\right)} \\
&= e^{i\xi/2}\frac{\sin\left(\frac{\xi}{2}\right)}{\frac{\xi}{2}}.
\end{aligned}$$

□

B-spline is refinable:

- *The* r^{th} *order B-spline* $\phi^{[r]}$ *is refinable and the corresponding mask is* $\left(\frac{1+z}{2}\right)^r$.

  1. Suppose the mask of $\phi^{[r]}$ is $h^{[r]}$ with coefficients $h_k^{[r]}$.
  2. From definition, $\phi^{[r]} = \phi^{[1]} * \cdots * \phi^{[1]}$, we have

$$\begin{aligned}
\phi^{[r]}(x) &= \phi^{[r-1]} * \phi^{[1]}(x) \\
&= \int \phi^{[r-1]}(y)\phi^{[1]}(x-y)dy \\
&= 2\int \sum_k h_k^{[r-1]}\phi^{[r-1]}(2y-k)\left(\phi^{[1]}(2x-2y) + \phi^{[1]}(2x-2y-1)\right)dy \\
&= 2\sum h_k^{[r-1]}\int \phi^{[r-1]}(2y-k)\phi^{[1]}(2x-2y)dy \\
&\quad +2\sum h_k^{[r-1]}\int \phi^{[r-1]}(2y-k)\phi^{[1]}(2x-2y-1)dy
\end{aligned}$$

$$\begin{aligned} &= \sum h_k^{[r-1]} \int \phi^{[r-1]}(y)\phi^{[1]}(2x-k-y)dy \\ &\quad + \sum h_k^{[r-1]} \int \phi^{[r-1]}(y)\phi^{[1]}(2x-k-1-y)dy \\ &= \sum h_k^{[r-1]}\phi^{[r]}(2x-k) + \sum h_k^{[r-1]}\phi^{[r]}(2x-k-1) \\ &= \sum (h_k^{[r-1]} + h_{k-1}^{[r-1]})\phi^{[r]}(2x-k). \end{aligned}$$

3. This implies $h_k^{[r]} = (h_k^{[r-1]} + h_{k-1}^{[r-1]})/2$.
4. It is easy to see that $h^{[1]}(z) = \left(\frac{1+z}{2}\right)$.

- Support $\phi^{[r]} = [0, r]$
- $\phi^{[r]} \in C^{r-1-\epsilon}$ for any $\epsilon > 0$.
- We may define $\phi^{[0]} = \delta$ which satisfies

$$\delta(x) = 2\delta(2x)$$

  i.e. $h^{[0]} = 1$.

- Riesz basis property: for $r \geq 1$, $\phi_{0,k}$ constitute a Riesz basis in $V_0 :=$ span $\{\phi_{0,k}\}_{k\in Z}$, i.e. there exists two constants $A > 0$ and $B < \infty$ such that

$$A \sum_k |c_k|^2 \leq \| \sum_k c_k \phi_{0,k}^{[r]} \|^2 \leq B \sum_k |c_k|^2$$

Example 2. The Lagrange interpolation function

- **Definition.**
  1. Initially, define $\tilde{\phi}(k) = \delta_{0,k}$
  2. Using subdivision scheme, define $\tilde{\phi}$ at $x_{j+1,2k+1}$ by Lagrange interpolation using data at $x_{j,k-r+1}, \cdots, x_{j,k+r}$, i.e.

$$\tilde{\phi}(x_{j+1,2k+1}) = \sum_u \prod_{\substack{-r<v\leq r \\ v\neq u}} \frac{x_{j+1,2k+1} - x_{j,k+v}}{x_{j,k+u} - x_{j,k+v}} \tilde{\phi}(x_{j,k+u})$$

- The mask
  1. $h(z) = z^{-r}\left(\frac{1+z}{2}\right)^{2r} Q_r(z)$, where

$$Q_r(z) = \sum_{n=0}^{r-1} \begin{pmatrix} r+n-1 \\ n \end{pmatrix} \left(\frac{2-z-z^{-1}}{4}\right)^n$$

2. Example:
   - $r = 1$: $2h = 2^{-2}(1, 2, 1)$
   - $r = 2$: $2h = 2^{-4}(-1, 0, 9, 16, 9, 0, -1)$

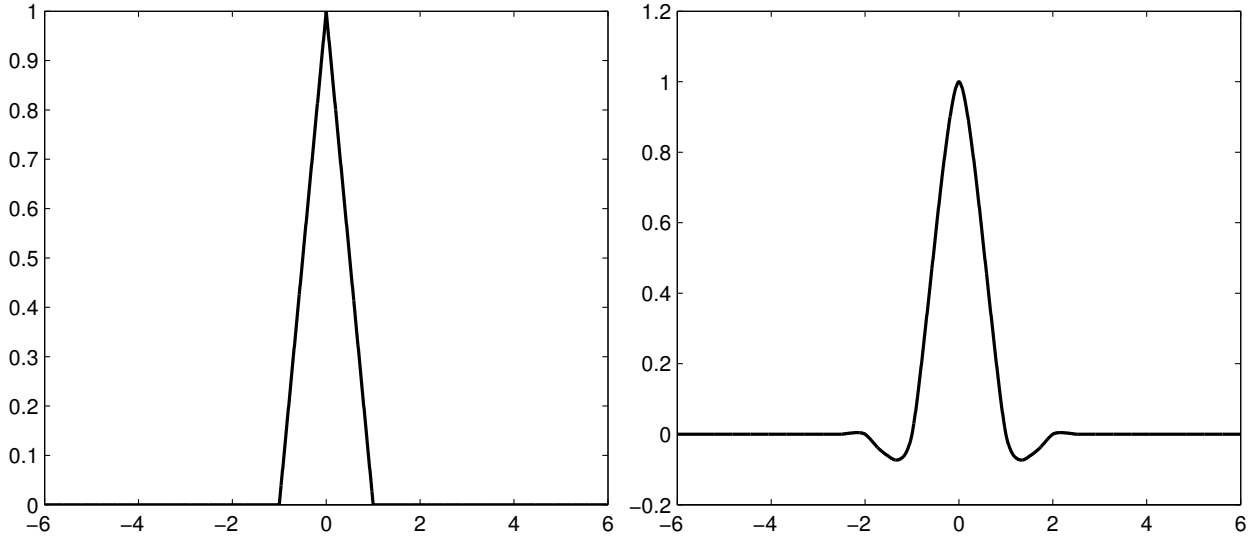
- Property:

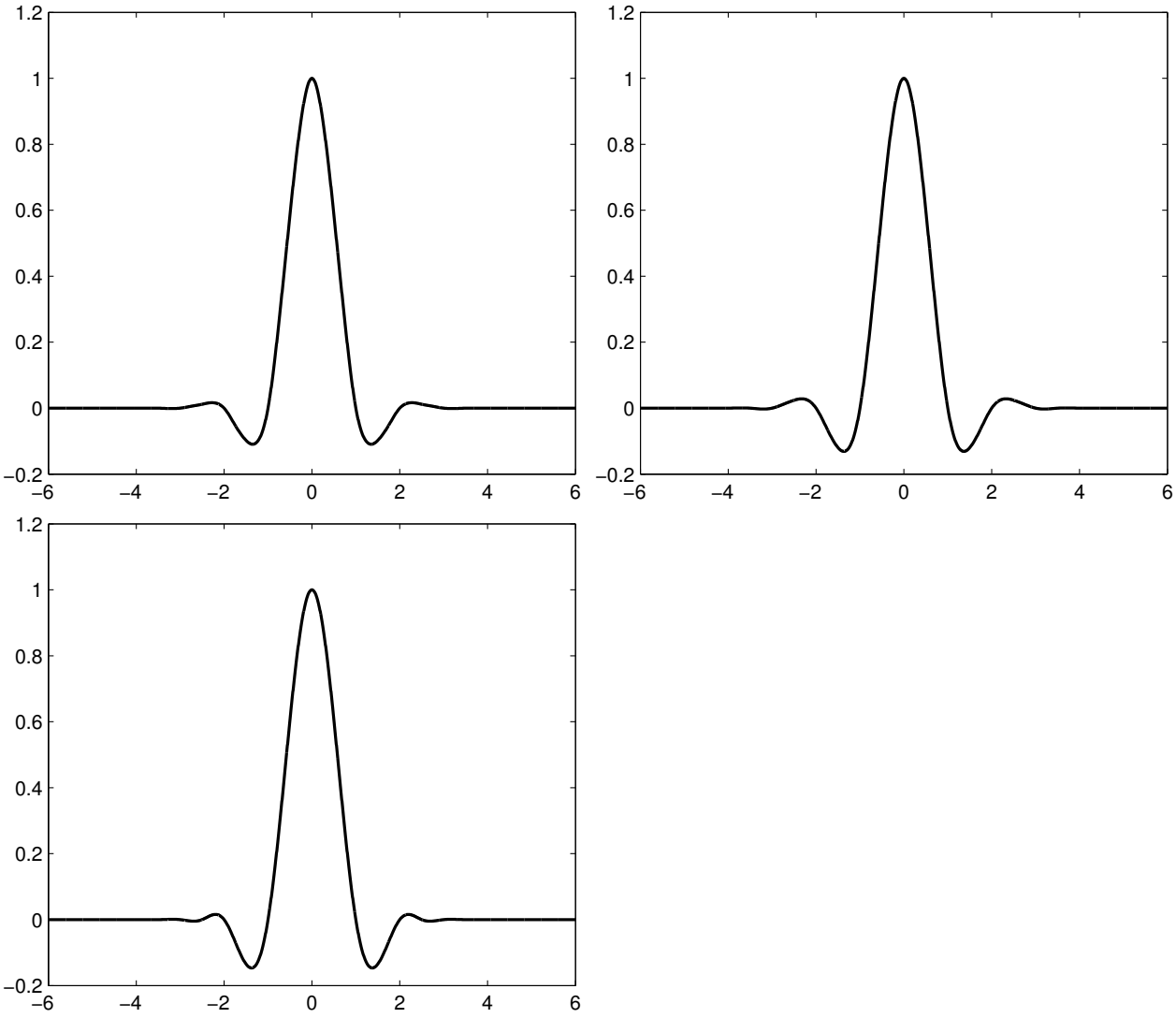
  1. Lagrange interpolation mask of order $2r$ can annihinate polynomials of degree less than $2r$.
  2. *The Lagrange interpolation mask has the smallest length among all interpolatory mask of order* $2r$.
  3. Support of $\tilde{\phi}$ of order $2r$ is $[-2r + 1, 2r - 1]$.
  4. Regularity: the order of differentiability is linearly propotional to r

- Riesz basis property

- Approximation power: *If Q_j is any projection onto V_j which is spanned by the Lagrange interpolant $\tilde{\phi}$ of order $2r$, then*

$$\|u - Q_j u\| = C_{\tilde{\phi}} 2^{-2rj} + O(2^{-(2r+1)j})$$





Example 3. Daubechies orthogonal wavelets

- Since $Q_r(z) = Q_r(z^{-1})$, we can split

$$Q_r(z) = Q(z)Q(z^{-1})$$

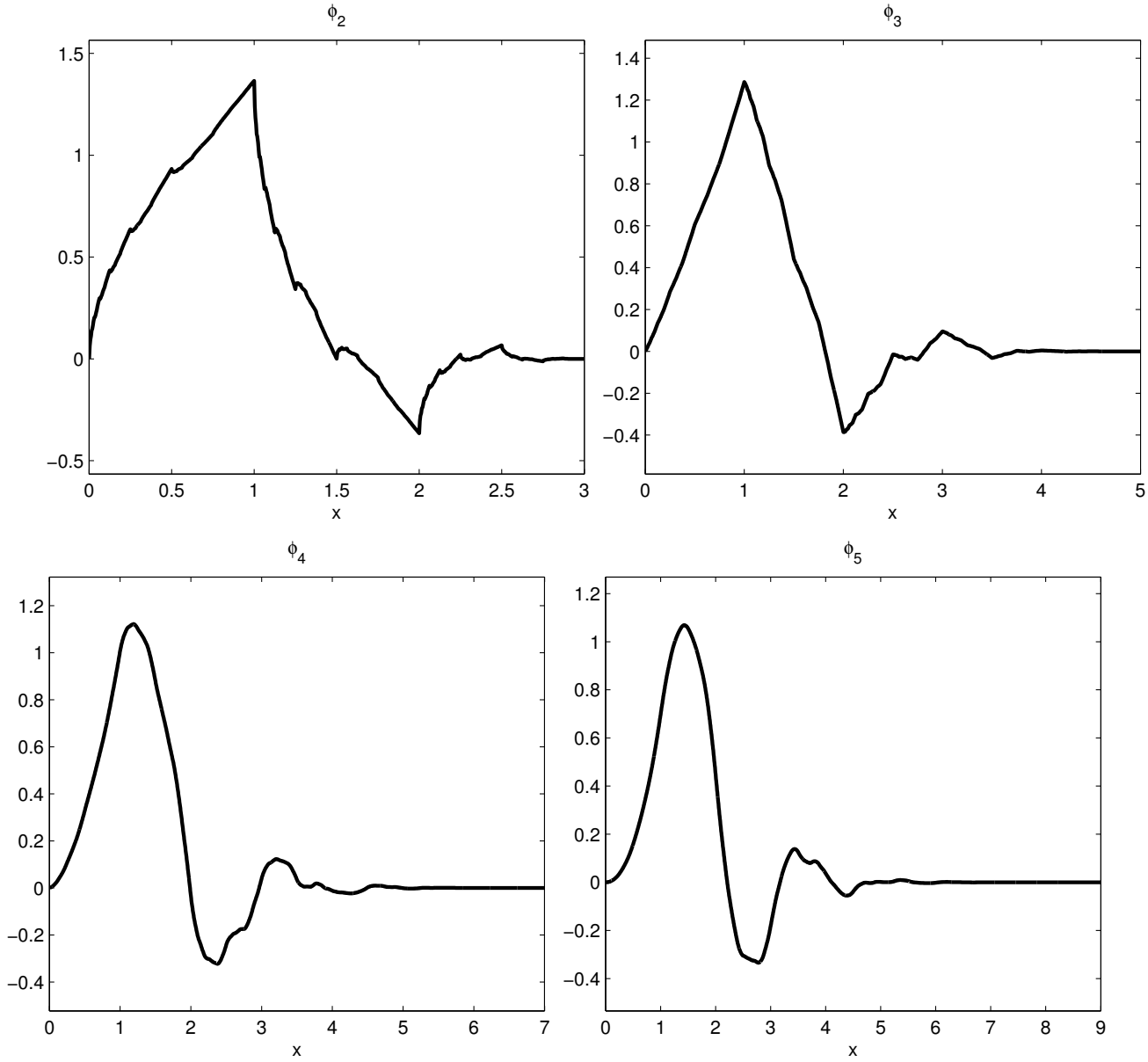
  where $Q(z) = a_0 + a_1 z + \cdots + a_{r-1} z^{r-1}$.

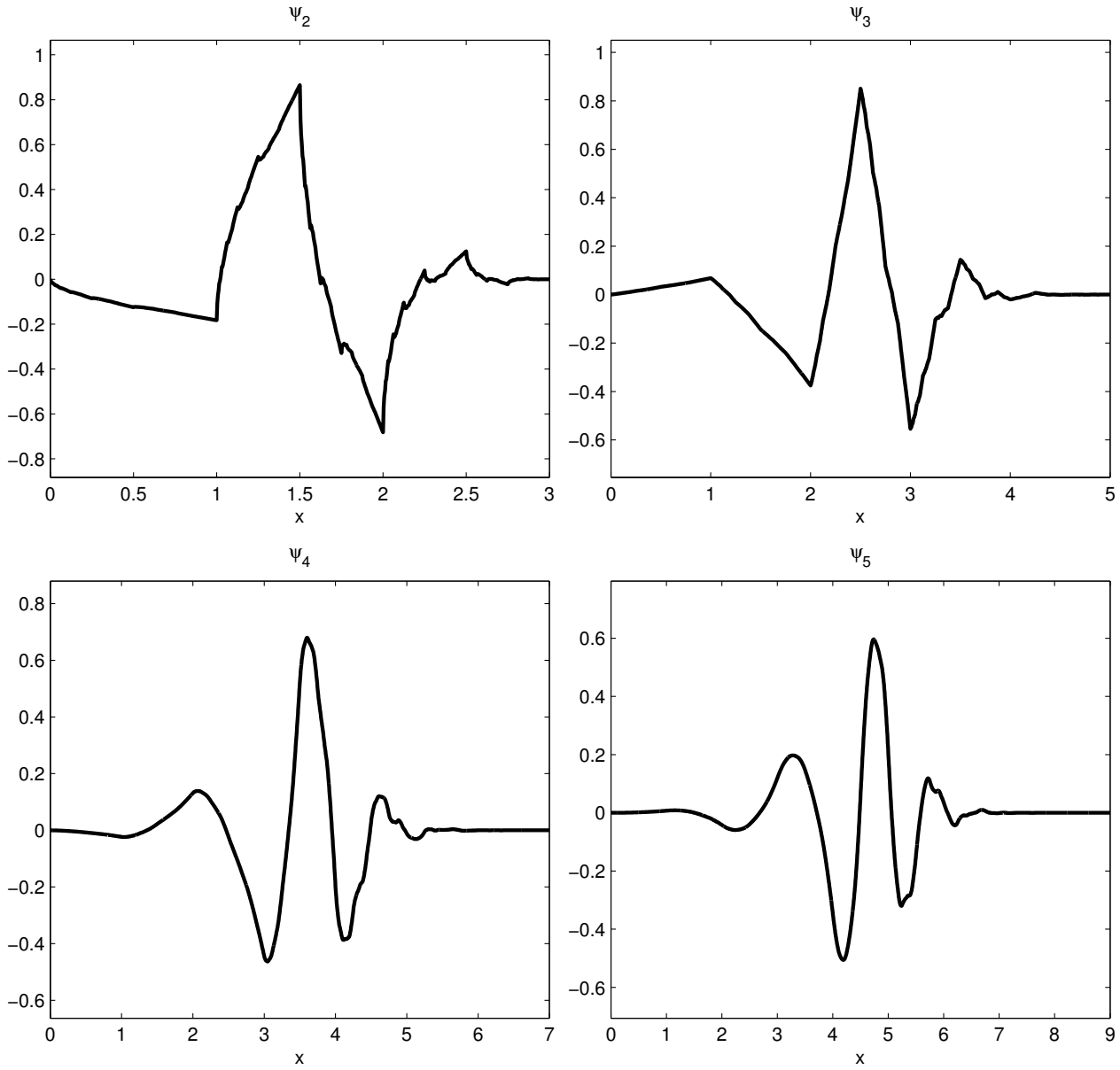
- We can split $H(z)^r$ into

$$z^{-r}\left(\frac{1+z}{2}\right)^{2r} = \left(\frac{1+z}{2}\right)^{r}\left(\frac{1+z}{2}\right)^{r} z^{-r} = \left(\frac{1+z}{2}\right)^{r}\left(\frac{1+z^{-1}}{2}\right)^{r}$$

- Then we can split $H_0(z)$ into $h(z)h(z^{-1})$:

$$H_0(z) = \underbrace{Q(z^{-1})\left(\frac{1+z^{-1}}{2}\right)^r}_{h(z^{-1})} \underbrace{Q(z)\left(\frac{1+z}{2}\right)^r}_{h(z)}$$



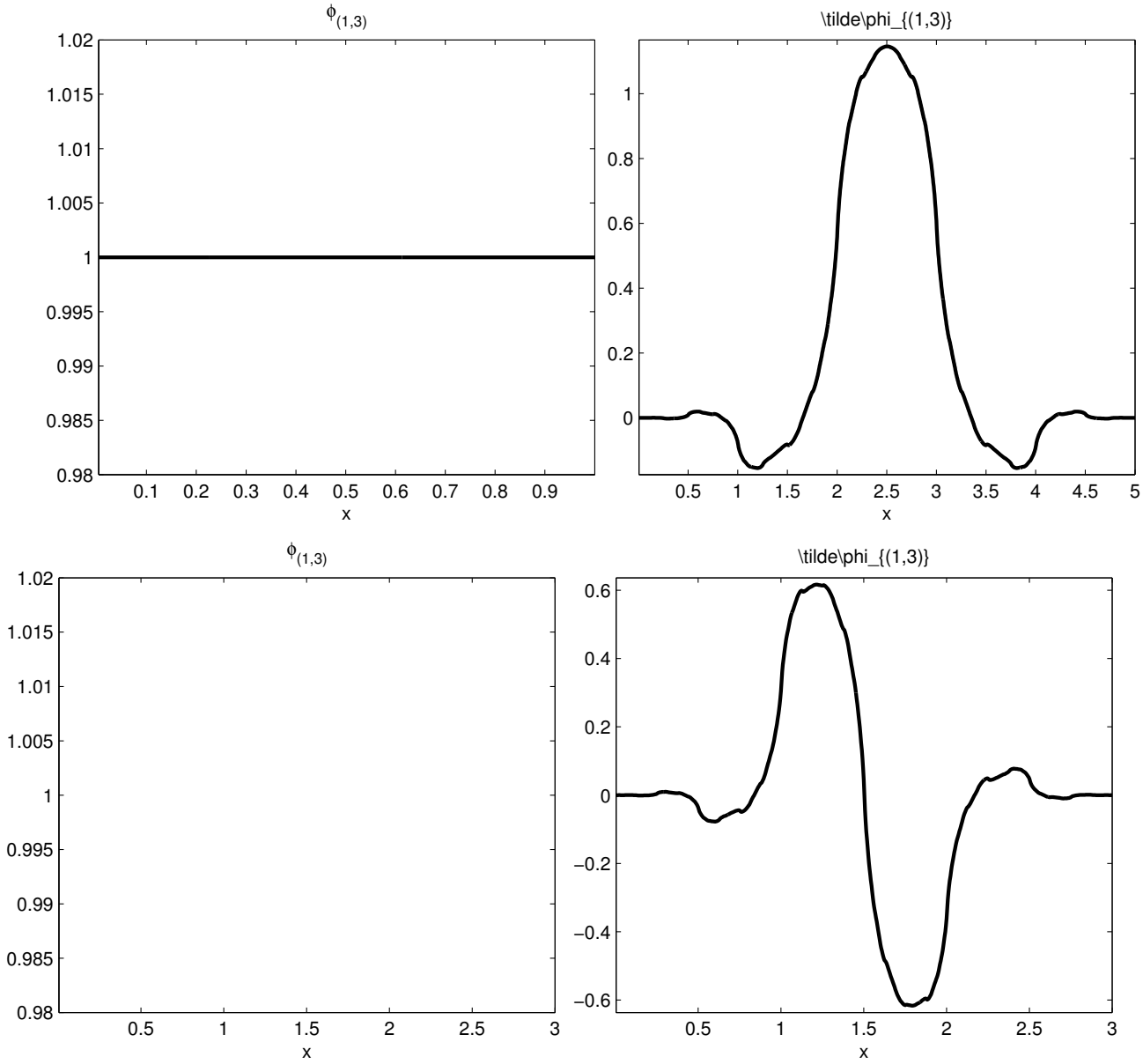


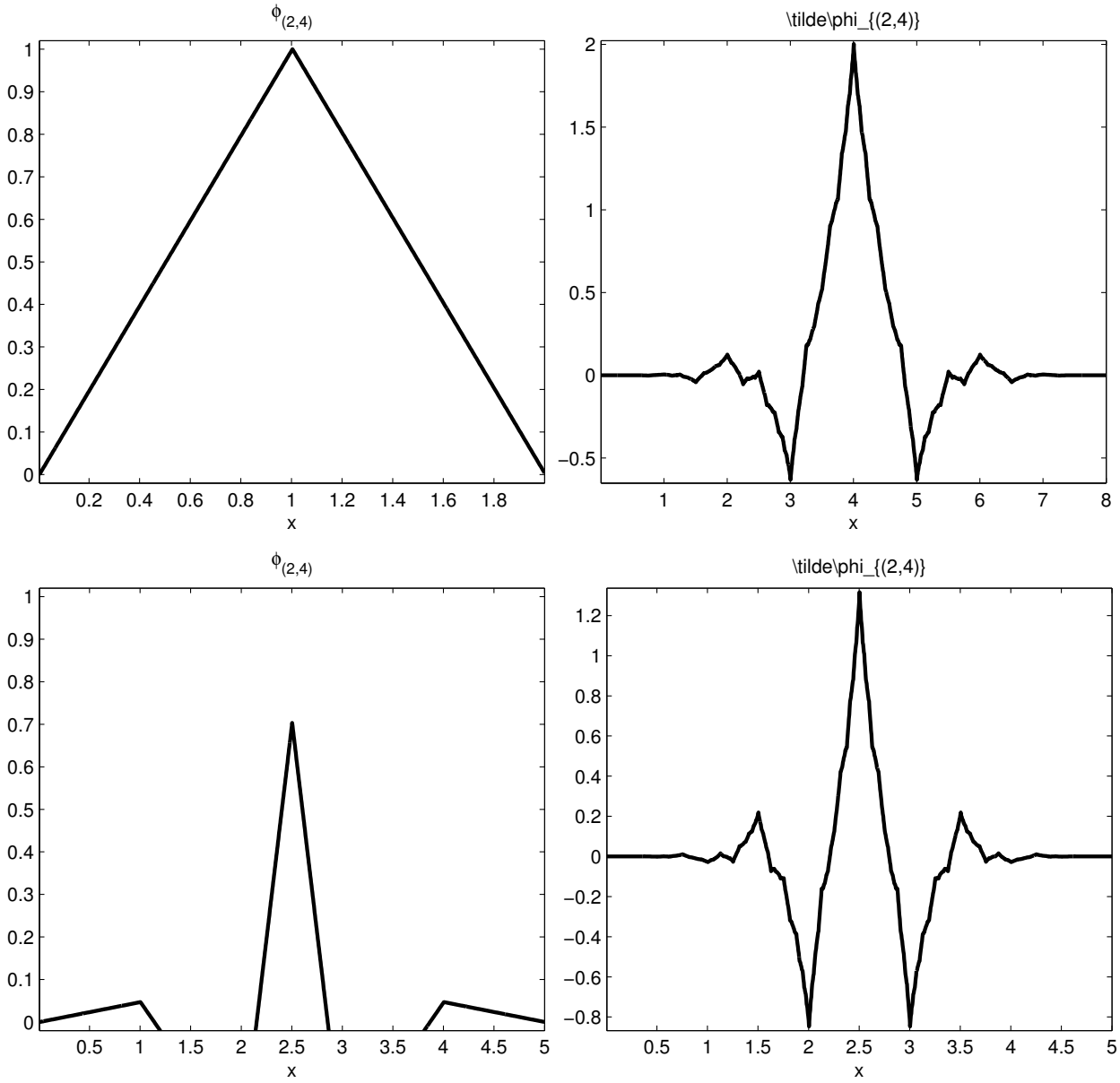
Example 4. Cohen-Daubechies-Feauveau biorthogonal wavelets

- Split $H^r Q_r$ into

$$\underbrace{Q_r(z)\left(\frac{1+z^{-1}}{2}\right)^{\tilde{d}}}_{\tilde{h}(z^{-1})}\underbrace{\left(\frac{1+z}{2}\right)^{d}}_{h(z)}$$

  where $d + \tilde{d} = 2r$.


$\phi_{(1,3)}$
\tilde\phi_{(1,3)}
x
x
$\phi_{(1,3)}$
\tilde\phi_{(1,3)}
x
x



Properties

- Convergence and regularity
  - If $h(1) = 1$, then ϕ exists as a distribution
  - Convergence in L^2 is related to the regularity of ϕ.
  - Regularity is related to the the "approximation order" of ϕ
  - If $h(z) = \left(\frac{1+z}{2}\right)^p H(z)$ with $H(1) \neq 0$, then we say that $\phi(z)$ has approximation order p.

  - If ϕ has approximation order p, then $\pi_p \subset V_0$, where π_p be the set of all polynomials of order less than p.
  - The higher the p is, the more regular the ϕ is

- Support and Decay
  - A mask is called of finite length if there exists an integer M such that $h(k) = 0$ for $|k| > M$
  - ϕ is of finite support if and only if its mask is of finite length
  - ϕ decay expontially at $x = \pm\infty$ iff h_k decays exponetially at $k = \pm\infty$.

- Riesz basis property:
  Under certain assumption on h, the corresponding refinable function ϕ satisfies the Riesz basis property:
$$A \sum_k |c_k|^2 \leq \| \sum_k c_k \phi_{j,k} \|^2 \leq B \sum_k |c_k|^2$$

- Approximation power:
  Theorem. (Strang-Fix, Unser) Suppose ϕ is of pth order and V_j is the span of $\{\phi_{j,k}\}_{k \in Z}$. Let Q_j be any projection from L^2 onto V_j. Then
$$\|Q_j u - u\|_{L^2} = C_Q 2^{-jp} + O(2^{-j(p+1)})$$

- The main technique is by Fourier method.

Duality

- $h(z)$ and $\tilde{h}(z)$ correspond to scaling functions ϕ and $\tilde{\phi}$
- Perfect reconstruction of $h(z)$ and $\tilde{h}(z)$:
$$h(z)\tilde{h}(z^{-1}) + h(-z)\tilde{h}(-z^{-1}) = 1$$
  is equivalent to duality of ϕ and $\tilde{\phi}$:
$$\int \phi(x)\tilde{\phi}(x-k)dx = \delta_{0,k}$$

- Proof by induction: Use
  1. $\phi^{n+1}(x) = \sum_k h_k \phi^n(2x-k)$, $\phi^0 = 1_{[0,1)}$,
  2. $\tilde{\phi}^{n+1}(x) = \sum_k \tilde{h}_k \tilde{\phi}^n(2x-k)$, $\tilde{\phi}^0 = 1_{[0,1)}$,
  3. $\int 1_{[0,1)}(x) 1_{[0,1)}(x-k)\, dx = \delta_{0,k}$

Biorthogonality

- Biorthogonality:
$$\int \psi(x)\tilde{\phi}(x-k)\,dx = 0.$$
- Proof by induction
  1. $\psi^{n+1}(x) = \sum_k g_k \phi^n(2x-k)$, $\phi^0 = 1_{[0,1)}$,
  2. $\tilde{\phi}^{n+1}(x) = \sum_k \tilde{h}_k \tilde{\phi}^n(2x-k)$, $\tilde{\phi}^0 = 1_{[0,1)}$,
  3. $\int 1_{[0,1)}(x) 1_{[0,1)}(x-k)\,dx = \delta_{0,k}$
  4. $g(z^{-1})\tilde{h}(z) + g(-z^{-1})\tilde{h}(-z) = 0$.

Biorthogonality

-
$$\begin{aligned}(\phi_{j,i}, \tilde{\phi}_{j,k}) &= \delta_{i,k}\\ (\psi_{j,i}, \tilde{\phi}_{j,k}) &= 0\\ (\psi_{j,i}, \tilde{\psi}_{j',k}) &= \delta_{j,j'}\delta_{i,k}\end{aligned}$$

References:

- I. Daubechies, Ten Lectures on Wavelets, SIAM Lecture Notes
- S. Mallat, A Wavelet Tool for Signal Processing, the Sparse Way, Academic Press, 2009.

Chapter 4

Numerical Integration

4.1 Motivations

Solving integral equations In applications, we encounter integral equations such as

$$\int_a^b K(x,y)u(y)\,dx = f(x)$$

or

$$u(x) + \int_a^b K(x,y)u(y)\,dx = f(x).$$

In solving Laplace equation, Helmholtz equation, we sometimes change it into integral equations, which is of the previous form. This is called boundary integral method.

Performing integral transform We need to perform the following transformations in applications

- Fourier transform
$$\int_{-\infty}^{\infty} f(x)e^{-ix\xi}\,dx$$
- Legendre transform
$$\int_{-1}^{1} f(x)P_m^n(x)\,dx$$
- Wavelet transform
$$\int_{-\infty}^{\infty} f(x)\psi_{jk}(x)\,dx.$$

In general, numerical integrations on intervals, curves, surfaces, and, in general, manifolds are needed in many places.

There are three classes of methods for numerical integration:

- Newton-Cotes method (which is based on uniformly spaced grid points)
- Gaussian-Quadrature methods
- Monte-Carlo methods

4.2 Newton-Cotes Method for numerical integration

Goal Numerical integration of $I(f) := \int_a^b f(x)\,dx$. Suppose the grid points are evenly spaced:

$$x_i = a + ih, \quad h = \frac{b-a}{n}.$$

and $f_i = f(x_i)$ are given.

Method

1. Approximate f by f^k, the Lagrange interpolation of f at x_i, $i = 0, ..., k$:
$$f^k(x) = \sum_{i=0}^{k} f_i \ell_i(x), \quad \ell_i(x) = \frac{\prod_{j\neq i}(x - x_j)}{\prod_{j\neq i}(x_i - x_j)}.$$
2. Approximate $I(f)$ by $I^k(f) := \int_{x_0}^{x_k} f^k(x)\,dx$.
$$I_k(f) = \int_{x_0}^{x_k} \sum_{i=0}^{k} f_i \ell_i(x)\,dx = \sum_{i=0}^{k} f_i w_i,$$
where
$$w_i = \int_{x_0}^{x_k} \ell_i(x)\,dx$$

We will not compute the weights w_i directly. Instead, we will use Romberg rule to compute these weights successively in k.

Examples

1. $k = 1$, Trapezoidal rule:
$$I_n^1(f) = \sum_{i=1}^{n} \frac{1}{2}\left(f_{i-1} + f_i\right)\frac{b-a}{n}$$
2. $k = 2$, Simpson rule:
$$I_{2n}^2(f) = \sum_{i=0}^{n-1} \frac{1}{6}\left(f_{2i} + 4f_{2i+1} + f_{2i+2}\right)\frac{(b-a)}{2n}.$$
3. $k = 3$, Boole's rule:
$$I_{2^2n}^3(f) = \sum_{i=0}^{n-1} \frac{1}{90}\left(7f_{4i} + 32f_{4i+1} + 12f_{4i+2} + 32f_{4i+3} + 7f_{4i+4}\right)\frac{(b-a)}{2^2n}$$

Error analysis Let us denote

$$E_n^1(f) = I(f) - I_n^1(f).$$

- Trapezoidal rule. By the interpolation analysis,

$$f(x) - f^1(x) = f[x_j, x_{j+1}, x](x - x_j)(x - x_{j+1})$$

Thus,

$$\begin{aligned} E_n^1(f) &= \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} (f - f^1)\, dx \\ &= \sum_{j=0}^{n-1} \frac{1}{2} f''(\eta_j) \int_{x_j}^{x_{j+1}} (x - x_j)(x - x_{j+1})\, dx \\ &= \sum_{j=0}^{n-1} \frac{1}{2} f''(\eta_j) \left(-\frac{1}{6}(x_{j+1} - x_j)^3 \right) \\ &= h^3 n \left[\frac{1}{n} \sum_{j=0}^{n-1} -\frac{1}{12} f''(\eta_j) \right] \\ &= -\frac{(b-a)}{12} h^2 f''(\eta) \text{ for some } \eta \in (a, b). \end{aligned}$$

- For Simpson rule, we partition the interval $[a, b]$ into $2n$ subintervals evenly. On the interval (x_{i-1}, x_{i+1}),

$$f - f^2 = f[x_{i-1}, x_i, x_{i+1}, x](x - x_{i-1})(x - x_i)(x - x_{i+1}).$$

Let

$$w(x) = \int_{x_{i-1}}^{x} (t - x_{i-1})(t - x_i)(t - x_{i+1})\, dt.$$

Because the grids x_i are evenly spaced and $(t - x_{i-1})(t - x_i)(t - x_{i+1})$ is an odd function, we obtain [1]

$$w(x_{i-1}) = w(x_{i+1}) = 0.$$

Thus,

$$\begin{aligned} \int_{x_{i-1}}^{x_{i+1}} (f - f^2)\, dx &= \int_{x_{i-1}}^{x_{i+1}} w'(x) f[x_{i-1}, x_i, x_{i+1}, x]\, dx \\ &= -\int_{x_{i-1}}^{x_{i+1}} w(x) f[x_{i-1}, x_i, x_{i+1}, x, x] \\ &= -\frac{4}{15} h^5 \left(-\frac{f^{(4)}(\eta_i)}{24} \right) \\ &= \frac{h^5}{90} f^{(4)}(\eta_i). \end{aligned}$$

[1] Indeed, if we choose $x_{i-1} = -h$, $x_i = 0$, $x_{i+1} = h$, then $w(x) = \frac{1}{4}(x - h)^2(x + h)^2$.

We obtain

$$I(f) - I_n^2(f) = \frac{h^4}{180}(b-a)f^{(4)}(\eta) \text{ for some } \eta \in (a,b).$$

Romberg Method One can derive the coefficients (weights) of the Newton-Cotes method successively starting from the trapezoidal rule by the Romberg method. We begin with the trapezoidal rule.

For the trapezoidal rule, the Euler-MacLaurin formula gives

$$I(f) - I_n^1(f) = \frac{d_1}{n^2} + \frac{d_2^1}{n^4} + \cdots$$

We shall use Richardson extrapolation formula to remove the error term: $\frac{d_1}{n^2}$ then get a high order method. We have

$$I(f) - I_{n/2}^1(f) = 4\frac{d_1}{n^2} + 2^4\frac{d_2^1}{n^4} + \cdots$$

From $4(I - I_n^1) - (I - I_{n/2}^1)$, we obtain

$$4(I - I_n^1) - (I - I_{n/2}^1) = \frac{-12d_2^1}{n^4} + \cdots.$$

Thus, let us define

$$I_n^2 = \frac{4I_n^1 - I_{n/2}^1}{3},$$

then we have

$$I(f) - I_n^2(f) = \frac{d_2}{n^4} + \cdots.$$

This numerical integration rule is exactly the Simpson rule. We can check the simplest case: $n = 2$. Suppose $b - a = 1$

$$I_2^1(f) = \frac{1}{2}\left(f_0 + f_1 + f_1 + f_2\right)\frac{1}{2}$$

$$I_1^1(f) = \frac{1}{2}\left(f_0 + f_2\right)$$

$$I_2^2(f) = \frac{4I_2^1(f) - I_1^1(f)}{3} = \frac{1}{6}\left(f_0 + 4f_1 + f_2\right)$$

We can continue this extrapolation:

$$I_n^3 = \frac{2^4 I_n^2 - I_{n/2}^2}{2^4 - 1}$$

$$I - I_n^3 = \frac{d_3}{n^6} + \cdots.$$

$$I_n^4 = \frac{2^6 I_n^3 - I_{n/2}^3}{2^6 - 1}$$

$$I - I_n^4 = \frac{d_4}{n^8} + \cdots .$$

In general,

$$I - I_n^k = \frac{d_k}{n^{2k}} + \cdots$$

$$I_n^{k+1} = \frac{2^{2k} I_n^k - I_{n/2}^k}{2^{2k} - 1}$$

and

$$I - I_n^{k+1} = \frac{d_{k+1}}{n^{2(k+1)}} + \cdots .$$

Error analysis One can show that

$$I - I_n^{k+1} = \frac{d_{k+1}}{n^{2(k+1)}} + \cdots$$

with

$$d_{k+1} = A_{k+1}(b-a)h^{2k+2} f^{(2k+2)}, \quad h = \frac{b-a}{2^k n},$$

where A_{k+1} is independent of f and n. In particular,

$$|I - I_{2^\ell}^\ell| = O\left(\frac{1}{2^\ell}\right)^{2(\ell+1)},$$

which gives the spectral accuracy, or the Newton-Cotes method. However, high order method is not so stable, as we have seen in Runge's example that high order interpolation polynomial on evenly spaced grid points produces large error on the boundary in general.

Euler-MacLausin expansion

Theorem 4.1. *Suppose $f \in C^{2n}[0,1]$. Then*

$$\int_a^b f(x)\,dx = \frac{1}{2}\left(f(0) + f(1)\right) - \sum_{k=0}^{n-1} \frac{b_{2k}}{(2k)!}\left[f^{(2k-1)}(1) - f^{(2k-1)}(0)\right] + R$$

where

$$R = -\frac{b_{2n}}{(2n)!} f^{(2n)}(\xi) \text{ for some } \xi \in (0,1),$$

b_{2k} are Bernoulli constants.

4.3 Gaussian Quadrature Methods

Goal: Let $w(x) > 0$ be a positive weighted function on (a, b). The goal is to numerically compute

$$I(f) = \int_a^b w(x) f(x)\, dx$$

by

$$I_n(f) = \sum_{i=1}^{n} w_{i,n} f(x_{i,n}),$$

where the $2n$ parameters $\{w_{i,n}, x_{i,n}\}$ are chosen so that

$$E_n(p) := I(p) - I_n(p) = 0$$

for all polynomials $p \in \Pi_{2n-1}$, the space of all polynomials with degree $\leq 2n-1$. Unlike the Newton-Cotes method, where the nodal points $\{x_1, ..., x_n\}$ are evenly spaced, we give freedom to choose these nodal points in order to increase the accuracy of numerical integration.

Difficulty Let us take $[-1, 1]$ with weight $w = 1$ as an example to illustrate difficulty. We choose $p(x) = x^i$, then $E_n(x^i) = 0$ for $i = 0, ..., 2n-1$ gives

$$\sum_{j=1}^{n} w_{j,n} x_{j,n}^i = \int_{-1}^{1} x^i\, dx = \begin{cases} 0 & i = 1, 3, ..., 2n-1 \\ \frac{2}{i+1} & i = 0, 2, ..., 2n-2. \end{cases}$$

There are $2n$ equations for the $2n$ unknowns $\{w_{j,n}, x_{j,n} | i = 1, ..., n\}$. However, this nonlinear equation is difficult to solve.

Orthogonal polynomials The main idea is to use an important property of orthogonal polynomial. Let us brief describe the theory of orthogonal polynomial, then explain how it is used to design these Gaussian-Quadrature points.

Let us recall $w(x) > 0$ on (a, b) be a positive weighted function. We introduce the space $L^2_w(a, b)$ with the inner product

$$\langle f, g \rangle := \int_a^b f(x) g(x)\, w(x)\, dx.$$

Proposition 10. *There exist sequence of orthogonal polynomials ϕ_n such that (i) deg $\phi_n = n$, (ii) $\langle \phi_n, \phi_m \rangle = \delta_{nm}$.*

Proof. These ϕ_n can be obtained from Gram-Schmidt process on $\{1, x, x^2, \cdots, x^n, \cdots\}$ inductively in n with $\phi_0 \equiv 1$. □

Examples:

- $(-1,1)$ with $w(x) \equiv 1$: Legendre polynomial

$$P_n(x) = \frac{(-1)^n}{2^n n!} \frac{d^n}{dx^n} \left[(1-x^2)^n \right], \quad n \geq 1.$$

$$P_0 \equiv 1, \quad \phi_n = \sqrt{\frac{2n+1}{2}} P_n.$$

- $(-1,1)$ with $w(x) = 1/\sqrt{1-x^2}$: Chebeshev polynomials

$$T_n(x) = \cos(n \cos^{-1} x).$$

- $(0,\infty)$ with $w(x) = e^{-x}$: Laguerre polynomials

$$L_n(x) = \frac{1}{n!} e^x \frac{d^n}{dx^n} \left[x^n e^{-x} \right].$$

- $(-\infty,\infty)$ with $w(x) = e^{-x^2/2}$: Hermite polynomials

$$H_n(x) = (-1)^n e^{x^2/2} \frac{d^n}{dx^n} e^{-x^2/2}.$$

Theorem 4.2. *In $L^2_w(a,b)$, the orthogonal polynomial ϕ_n has exactly n distinct real roots on (a,b).*

Proof. 1. Suppose $x_1, ..., x_m \in (a,b)$ are distinct roots such that ϕ_n changes sign at x_i. That is,

$$\phi_n(x) = \prod_{j=1}^{m} (x - x_j)^{r_j} h(x),$$

where $r_j \geq 1$ are odd and $h(x)$ does not change sign in (a,b).

2. Let $B(x) = \prod_{j=1}^{m} (x - x_j)$. If $m < n$, then $deg\, B < n$. Thus, by the orthogonal property: $\phi_n \perp \Pi_{n-1}$, we get

$$\int_a^b \phi_n(x) B(x) w(x)\, dx = 0.$$

3. On the other hand,

$$\phi_n B = \prod_{j=1}^{m} (x - x_j)^{r_j+1} h(x)$$

does not change sign on (a,b) and $w(x) > 0$ on (a,b). Hence

$$\int_a^b w(x) \phi_n(x) B(x)\, dx \neq 0.$$

This is a contradiction. Thus, we must have $m = n$.

4. Because $deg\, \phi_n = n$, we obtain that all $r_j = 1$, $j = 1, ..., n$. This means that ϕ_n has n distinct simple roots.

□

Gaussian quadrature method

- Choose $x_1, ..., x_n$ to be the roots of ϕ_n, where ϕ_n is the orthogonal polynomial in $L^2_w(a,b)$ with degree n;
- The Gaussian-quadrature formula is

$$I_n(f) = \sum_{j=1}^{n} w_j f(x_j),$$

where

$$w_j = \int_a^b w(x)\ell_j(x)\,dx, \quad \ell_j(x) = \frac{\prod_{i\neq j}(x - x_i)}{\prod_{i\neq j}(x_j - x_i)}.$$

The explicit formulae of x_j, w_j can be found in Wiki (see Gaussian quadrature, Wiki).

We will introduce two derivations of the Gaussian quadrature method.

Derivation I Let us write

$$\psi_n(x) = \prod_{j=1}^{n}(x - x_j) = c_n\phi_n.$$

Theorem 4.3. *Let $w(x) > 0$ be a weight in (a,b) and let ϕ_n be the orthogonal polynomial of degree n. Then $E_n(f) := I(p) - I_n(p) = 0$ for all $p \in \Pi_{2n-1}$.*

Proof. 1. For any $p \in \Pi_{2n-1}$, we can divide it by ψ_n and get

$$p(x) = \psi_n(x)q(x) + r(x),$$

where $deg\, q \leq n-1$, $deg\, r < n$. From $\psi_n(x_i) = 0$, we obtain $p(x_i) = r(x_i)$ for $i = 1, ..., n$.

2. Since $\psi_n \perp \Pi_{n-1}$ and $deg\, q < n$, we get

$$\begin{aligned} I(p) &= \int_a^b p(x)w(x)\,dx = \int_a^b (\psi_n(x)q(x) + r(x))\,w(x)\,dx = \int_a^b r(x)w(x)\,dx \\ &= \int_a^b \sum_{i=1}^{n} r(x_i)\ell_i(x)w(x)\,dx = \sum_{i=1}^{n} r(x_i)w_i = \sum_{i=1}^{n} p(x_i)w_i = I_n(p). \end{aligned}$$

□

Lemma 4.1. *The weights $w_i > 0$ for all $i = 1, ..., n$. Moreover, $\sum_i w_i = \int_a^b w(x)\,dx$.*

Proof. 1. Since $\ell_i^2 \in \Pi_{2n-1}$, we get $E_n(\ell_i^2) = 0$. That is

$$0 < \int_a^b \ell_i^2(x)w(x)\,dx = \sum_{j=1}^{n} w_j\ell_i^2(x_j) = w_i.$$

2. Since $E_n(1) = 0$, we get
$$\int_a^b 1 \cdot w(x)\,dx = \sum_{j=1}^n w_j \cdot 1.$$

□

Theorem 4.4. *Let $w(x) > 0$ in (a, b). Suppose $\{x_1, ..., x_n\}$ and $\{w_1, ..., w_n\}$ are chosen such that $E_n(p) = 0$ for all $p \in \Pi_{2n-1}$. Then for any $f \in C^{k+\alpha}(a, b)$,*
$$|E_n(f)| \le 2\left(\int_a^b w(x)\,dx\right)\rho_{2n-1}(f)$$
where
$$\rho_{2n-1}(f) := d_\infty(f, \Pi_{2n-1}) \le \frac{C_{k+\alpha}}{(2n-1)^{k+\alpha}}\|f\|_{C^{k+\alpha}}.$$

Proof. 1. By Jackson's theorem, there exists a $q^* \in \Pi_{2n-1}$ such that
$$d_\infty(f, q^*) = d_\infty(f, \Pi_{2n-1}) = \rho_{2n-1}(f).$$

2. Since $q^* \in \Pi_{2n-1}$, we have $E_n(q^*) = 0$. Using this,
$$E_n(f) = E_n(f) - E_n(q^*) = \int_a^b (f - q^*)w(x)\,dx - \sum_{I=1}^n w_i(f(x_i) - q^*(x_i))$$
$$\le \|w\|_1\|f - q^*\|_\infty + \sum_i |w_i|\|f - q^*\|_\infty = 2\|w\|_1\rho_{2n-1}(f).$$

3. From Jackson's theorem
$$\rho_n(f) \le \frac{C_{k+\alpha}}{n^{k+\alpha}}\|f\|_{C^{k+\alpha}},$$
where C is independent of f and n.

□

Derivation II The idea behind is to use Hermite interpolation polynomials. Given $f(x_j)$, $f'(x_j)$, $j = 1, ..., n$, the Hermite interpolation polynomial $H_n(x)$ is of degree $2n - 1$ with
$$H_n(x_j) = f(x_j), \quad H_n'(x_j) = f'(x_j), \quad j = 1, ..., n.$$
Indeed,
$$H_n(x) = \sum_{j=1}^n \left(f_j h_j(x) + f_j'\tilde{h}_j(x)\right),$$
where
$$h_j(x_i) = \delta_{ij}, \quad h_j'(x_i) = 0 \tag{4.1}$$
$$\tilde{h}_j'(x_i) = \delta_{ij}, \quad \tilde{h}_j(x_i) = 0. \tag{4.2}$$

with

$$deg\, h_j = deg\, \tilde{h}_j = 2n - 1.$$

Such functions h_j and $\tilde{h}_j$ can be constructed easily.

Lemma 4.2. *Given $x_1, ..., x_n$. The polynomials h_j and $\tilde{h}_j$, $j = 1, ..., n$ of degree $2n-1$ satisfying (4.1), (4.2) are given by*

$$h_j(x) = \left[1 - 2\ell_j'(x_j)(x - x_j)\right] \ell_j^2(x),$$

$$\tilde{h}_j(x) = (x - x_j)\ell_j^2(x).$$

Proof. 1. The polynomial h_j should contain a factor ℓ_j^2 because x_i, $i \neq j$ are its double roots. The only condition it does not satisfy is $(\ell_j^2)'(x_j) = 2\ell_j'(x_j) \neq 0$. We consider

$$h_j(x) = \big(1 - 2\ell_j'(x_j)(x - x_j)\big) \ell_j^2(x).$$

Then $deg\, h = 2n - 1$, $h_j'(x_i) = 0$ for $i = 1, ..., n$ and $h_j(x_i) = \delta_{ij}$.

2. Similarly, x_i, with $i \neq j$ are double roots of $\tilde{h}_j$. Thus, $\tilde{h}_j$ contains a factor $\ell_j^2(x)$. Since x_j is also a root of $\tilde{h}_j$, the term $(x - x_j)$ is also a factor of $\tilde{h}_j(x)$. The polynomial $(x - x_j)\ell_j^2(x)$ has the same roots and same degree of those of $\tilde{h}_j$. Thus $\tilde{h}_j(x) = c(x - x_j)\ell_j^2(x)$. Since the derivative of $(x - x_j)\ell_j^2(x)$ at x_j is 1, we get that $c = 1$.

□

Now, let us go back to derive Gaussian quadrature formula via the Hermite interpolation function. With the Hermite inyterpolation at $\{x_1, ..., x_n\}$, we define

$$\begin{aligned} I_n(f) &:= \int_a^b H_n(x)\, dx = \int_a^b \sum_{j=1}^n \Big(f_j h_j(x) + f_j' \tilde{h}_j(x) \Big) w(x)\, dx \\ &= \int_a^b \sum_{j=1}^n f_j w_j, \quad w_j = \int_a^b h_j(x) w(x)\, dx. \end{aligned}$$

The last line follows from the following lemma.

Lemma 4.3. *If $x_1, ..., x_n$ are the roots of the orthogonal polynomial ϕ_n in Π_n, then*

$$w_j = \int_a^b h_j(x) w(x)\, dx = \int_a^b \ell_j(x) w(x)\, dx = \int_a^b \ell_j^2(x) w(x)\, dx, \quad j = 1, ..., n, \tag{4.3}$$

$$\int_a^b \tilde{h}_j(x) w(x)\, dx = 0, \quad j = 1, ..., n. \tag{4.4}$$

Proof. 1. From $\psi_n \perp \Pi_{n-1}$, we have

$$\begin{aligned}\int_a^b h_j(x)w(x)\,dx &= \int_a^b \left(1 - 2\ell_j'(x_j)(x-x_j)\right)\ell_j^2(x)w(x)\,dx \\ &= \int_a^b \left(\ell_j^2(x) - c\psi_n(x)\ell_j(x)\right)w(x)\,dx \\ &= \int_a^b \ell_j^2(x)w(x)\,dx.\end{aligned}$$

2. Since $\ell_i^2 \in \Pi_{2n-1}$, we get $E_n(\ell_i^2) = 0$. That is

$$\int_a^b \ell_i^2(x)w(x)\,dx = \sum_{j=1}^n w_j\ell_i^2(x_j) = w_i = \int_a^b \ell_i(x)\,dx.$$

3. We rewrite the second integral as

$$\begin{aligned}&\int_a^b w(x)\tilde{h}_j(x)\,dx = \int_a^b w(x)(x-x_j)\left(\frac{\prod_{i\neq j}(x-x_i)}{\prod_{i\neq j}(x_j-x_i)}\right)^2 \\ &= c_j\int_a^b w(x)\phi_n(x)\ell_j(x)\,dx = 0, \text{ for } j = 1,...,n,\end{aligned}$$

because $\phi_n \perp \Pi_{n-1}$.

□

Theorem 4.5. *For $f \in C^{2n+1}[a,b]$, we have*

$$E_n(f) := I(f) - I_n(f) = \frac{f^{(2n+1)}(\eta)}{(2n+1)!}\int_a^b \left[\psi_n(x)\right]^2 w(x)\,dx,$$

for some $\eta \in [a,b]$.

Proof. This follows directly from

$$f - H_n = \left[\psi_n(x)\right]^2 f[x_1,x_1,...,x_n,x_n,x]$$

and

$$\begin{aligned}I(f) - I_n(f) &= \int_a^b \left[\psi_n(x)\right]^2 f[x_1,x_1,...,x_n,x_n,x]w(x)\,dx \\ &= \frac{f^{(2n+1)}(\eta)}{(2n+1)!}\int_a^b \left[\psi_n(x)\right]^2 w(x)\,dx.\end{aligned}$$

□

As comparing with Newton-Cotes:

$$f - p_n = \psi_n(x) f[x_1, ..., x_n, x],$$

$$\begin{aligned} I(f) - \int_a^b p_n(x)\, dx &= \int_a^b [\psi_n(x)]\, f[x_1, x_2, ..., x_n, x]\, dx \\ &\leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \int_a^b |\psi_n(x)|\ dx. \end{aligned}$$

the error of Gaussian quadrature method is much smaller than the error of Newton-Cotes method.

Chapter 5

Numerical Ordinary Differential Equations

5.1 Motivations

Example 1. Van der Pol oscillator In electric circuit theory, van der Pol proposed a model for electric circuit with vacuum tube, where $I = \phi(V)$ is a cubic function. Consider a circuit system with the resistor replaced by a device which obeys a nonlinear Ohm's law: the potential drop across this device is

$$\Delta V = \alpha \left(\frac{I^3}{3} - I \right), \ \alpha > 0.$$

Such a device does appear in vacuum tubes or semiconductors. The corresponding L-C-R circuit equation becomes

$$L\frac{dI}{dt} + \frac{Q}{C} + \alpha \left(\frac{I^3}{3} - I \right) = V(t). \tag{5.1}$$

Differentiate in t, we obtain the Van der Pole equation:

$$L\frac{d^2I}{dt^2} + \alpha(I^2 - 1)\frac{dI}{dt} + \frac{I}{C} = f(t). \tag{5.2}$$

where $f(t) = \dot{V}(t)$ is the applied electric field. The system is dissipative (damping) when $I^2 > 1$ and self current increasing when $I^2 < 1$.

Let x be the current and let us consider a normalized system:

$$\ddot{x} + \epsilon(x^2 - 1)\dot{x} + x = 0.$$

Through a Liénard transform:

$$y = x - \frac{x^3}{3} - \frac{\dot{x}}{\epsilon}$$

the van der Pol equation can be expressed as

$$\begin{aligned} \dot{x} &= \epsilon(x - \frac{x^3}{3} - y) \\ \dot{y} &= \frac{x}{\epsilon} \end{aligned}$$

We can draw the nullclines: $f = 0$ and $g = 0$. From the direction field of (f, g), we see that the field points inwards for large (x, y) and outward for (x, y) near $(0, 0)$. This means that there will be a limiting circle in between.

As $\epsilon >> 1$, we can observe that the time scale on x variable is fast whereas it is slow on the y-variable. That is,

$$\dot{x}(t) = O(\epsilon),\ \dot{y}(t) = O(1/\epsilon).$$

On the x-y plane, consider the curve

$$y = x - \frac{x^3}{3}.$$

The solution moves fast to the curve $y = x - \frac{x^3}{3}$. Once it is closed to this curve, it move slowly along it until it moves to the critical points $(\pm 1, \pm\frac{2}{3})$. At which it moves away from the curve fast and move to the other side of the curve. The solution then periodically moves in this way.

Example 2. The orbit of a star in a galaxy. [1] Galaxy is a rotating object consisting of gases and stars. The potential induced by stars and gases of a galaxy drives the motion of a galaxy, but is an unknown function. What we can observe is the motion of stars. In galaxy, the rotation curve of stars (i.e. the speed of star at radius r) is the observable object. We want to model the potential such that the Newton's law of motion under this potential gives us the observed motion. Based of Gauss law, the potential Φ and the density distribution of stars ρ are related by

$$\triangle\Phi = \rho.$$

Therefore, we model either the potential, or the mass distribution. Then we study the motion of a star in this potential and compare it with our observation. It is natural to consider a disk in cylindrical coordinate system $\mathbf{r} = (R, \phi, z)$. Suppose the potential is axisymmetric, that is, the potential is $\Phi(R, z)$, independent of ϕ. There are many models. For examples,

- Plummer model
$$\Phi_P(r) = -\frac{GM}{\sqrt{r^2 + b^2}}.$$
- Kuzmin model:
$$\Phi_K(R, z) = -\frac{GM}{\sqrt{R^2 + (a + |z|)^2}}$$

[1] Ref. James Binney and Scott Tremaine, Galactic Dynamics, Princeton University Press

- Toomre model:

$$\Phi_M(R,z) = -\frac{GM}{\sqrt{R^2 + (a + \sqrt{z^2+b^2})^2}}.$$

- Logarithmic potential:

$$\Phi_L(R,z) = \frac{1}{2}v_0^2 \ln\left(R^2 + \frac{z^2}{q^2}\right).$$

The governing equation of motion is

$$\frac{d^2 r}{dt^2} = -\nabla\Phi(R,z).$$

We may express $\mathbf{r} = R\hat{e}_R + \phi\hat{\phi} + z\hat{e}_z$ and $\nabla\Phi = \Phi_R\hat{e}_R + \Phi_z\hat{e}_z$. Using $\mathbf{r} = (R\cos\phi, R\sin\phi, z)$, we obtain

$$\ddot{r} = (\ddot{R} - R\dot{\phi}^2)\hat{e}_R + \frac{1}{R}\frac{d}{dt}(R^2\dot{\phi})\hat{e}_\phi + \ddot{z}\hat{e}_z.$$

Then the equation in each component is

$$\ddot{R} - R\dot{\phi}^2 = -\Phi_R,$$

$$\frac{d}{dt}\left(R^2\dot{\phi}\right) = 0,$$

$$\ddot{z} = -\Phi_z.$$

The force $R\dot{\phi}^2$ is the centrifugal force. The second equation is the conservation of angular momentum. Let $L_z = R^2\dot{\phi}$. The centrifugal force is L_z^2/R^3. Thus, we may write the equation as

$$\begin{cases} \ddot{R} = -\frac{\partial\Phi_{\text{eff}}}{\partial R} \\ \ddot{z} = -\frac{\partial\Phi_{\text{eff}}}{\partial z}, \end{cases}$$

where

$$\Phi_{\text{eff}} = \Phi(R,z) + \frac{L_z^2}{2R^2}$$

The equilibrium occurs at $(R_g, 0)$ where $\nabla\Phi_{\text{eff}}(R_g, 0) = 0$. This equilibrium is a circular motion with angular speed

$$\Omega = \frac{L_z}{R_g^2} = \sqrt{\frac{1}{R_g}\frac{\partial V}{\partial R}|_{R_g}}.$$

The goal is to solve this equation of motion under various potentials.

Example 3. Michaelis-Menten Enzyme Kinetics. A chemical substrate S is converted to a product P through an enzyme catalysis E: The reaction equation are

$$\begin{aligned}
\frac{d}{dt}[E] &= -k_1[E][S] + k_{-1}[ES] + k_2[ES] \\
\frac{d}{dt}[S] &= -k_1[E][S] + k_{-1}[ES] \\
\frac{d}{dt}[ES] &= k_1[E][S] - k_{-1}[ES] - k_2[ES] \\
\frac{d}{dt}[P] &= k_2[ES]
\end{aligned}$$

5.2 Basic Numerical Methods for Ordinary Differential Equations

The basic assumption to design numerical algorithm for ordinary differential equations is the smoothness of the solutions, which is in general valid provided the coefficients are also smooth. Basic designning techniques include numerical interpolation, numerical integration, and finite difference approximation.

Euler method

Euler method is the simplest numerical integrator for ODEs. The ODE

$$y' = f(t, y) \tag{5.3}$$

is discretized by

$$\boxed{\frac{y^{n+1} - y^n}{k} = f(t^n, y^n).} \tag{5.4}$$

Here, k is time step size of the discretization. This method is called the forward Euler method. It simply replace $dy/dt(t^n)$ by the forward finite difference $(y^{n+1} - y^n)/k$. We would like to know the growth of the true error e^n, defined by $e^n := y^n - y(t^n)$. To estimate this error, we need to derive an equation for e^n. To do so, suppose $y(\cdot)$ is a true solution. We plug it into the finite difference equation. It will not satisfy the difference equation. The remainder term is called the truncation error. More precisely,

$$\frac{y(t^{n+1}) - y(t^n)}{k} - f(t^n, y(t^n)) = \tau^n \tag{5.5}$$

where

$$\tau^n := \frac{y(t^{n+1}) - y(t^n)}{k} - y'(t^n) = O(k).$$

Subtracting (5.4) from (5.5), we get

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

Taking the absolute values, we get

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

where

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

This is the inequality for error e^n.

Theorem 5.1. *Assume $f \in C^1$ and suppose the solution $y' = f(t,y)$ with $y(0) = y_0$ exists on $[0,T]$. Then the Euler method converges at any $t \in [0,T]$. In fact, the true error e^n has the following estimate:*

$$|e^n| \le \frac{e^{\lambda t}}{\lambda} O(k) \to 0, \text{ as } n \to \infty. \tag{5.6}$$

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

Proof. The 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

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

Summing in m from $m = 0$ to $n-1$, 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 $(1+\lambda k)^n \le e^{\lambda t}$. □

Remarks.

1. The theorem states that the numerical method converges in $[0,T]$ as long as the solutions of the ODE exists.

2. The error is $O(k)$ if the solution is in $C^2[0,T]$.

3. The proof above relies on the existence and smoothness of the solution. However, one can also use this approach to prove the local existence theorem by showning the approximate solutions generated by the Euler method form a Cauchy sequence.

Backward Euler method

In many applications, the system is relaxed to a stable solution in a very short period of time. For instance, consider

$$y' = \frac{\bar{y} - y}{\tau}.$$

The corresponding solution $y(t) \to \bar{y}$ as $t \sim O(\tau)$. The Lipschitz constant is $\lambda = 1/\tau$. If we use forward Euler method, then we should require a very small $k = t/n$ in order to have $G^n = (1 + t/(n\tau))^n$ remains bounded for any n. This leads to an inefficient computation. In general, forward Euler method becomes inefficient (require small k) when

$$\max \left| \frac{\partial f(t,y)}{\partial y} \right| >> 1.$$

However, in the case of relaxation system where the Jacobian $\partial f/\partial y$ is negative definite and large in magnitude, the backward Euler method is recommended:

$$\boxed{y^{n+1} = y^n + kf(t^{n+1}, y^{n+1}).} \tag{5.7}$$

The true solution $y(\cdot)$ satisfies

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

where

$$\tau^n = \frac{y(t^{n+1}) - y(t^n)}{k} - y'(t^{n+1}) = O(k).$$

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

$$e^{n+1} = e^n + k\left(f(t^{n+1}, y(t^{n+1})) - f(t^{n+1}, y^{n+1})\right) + \tau^n$$

Thus,

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

The corresponding fundamental solution is $G^n := (1 - \lambda k)^{-n}$. Using the fundamental solution, we multiply both sides (with index m) by G^{-m} and get

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

Summing m from 0 to $n-1$, we obtain

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

where $t = nk$. Here, we have assumed $\lambda k < 1$.

Leap frog method

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

$$y^{n+1} - y^{n-1} = 2kf(t^n, y^n). \tag{5.8}$$

This is a two-step explicit method.

Homeworks.

1. Prove the convergence theorem for the backward Euler method.
   Hint: show that $e^{n+1} \le e^n + (1 + k\lambda)e^{n+1} + k\tau^n$, where λ is the Lipschitz constant of f.

2. Prove the convergence theorem for the leap-frog method.
   Hint: consider the system $y_1^n = y^{n-1}$ and $y_2^n = y^n$.

5.3 Runge-Kutta methods

The Runge-Kutta method (RK) is a strategy to integrate $\int_{t^n}^{t^{n+1}} f\, d\tau$ by some quadrature method.

RK2 For instance, 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

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

Next, this integration is approximated by

$$1/2(f(t^n, y^n) + f(t^n, y^{n+1}))k.$$

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

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

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 of RK2 is

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

RK4 The 4th order Runge-Kutta method 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

$$y(t^{n+1}) - y(t^n) = y^{n+1} - y(t^n) + O(k^5).$$

General explicit Runge-Kutta methods The method takes the following general form

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

where

$$\begin{aligned} \xi_1 &= f(t^n, y^n), \\ \xi_2 &= f(t^n + c_2k, y^n + ka_{21}\xi_1), \\ \xi_3 &= f(t^n + c_3k, y^n + ka_{31}\xi_1 + ka_{32}\xi_2), \\ &\vdots \\ \xi_s &= f(t^n + c_sk, y^n + k(a_{s1}\xi_1 + \cdots + a_{s,s-1}\xi_{s-1})). \end{aligned}$$

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 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 linear equations for the coefficients.

$$\begin{array}{c|ccccc} 0 & & & & & \\ c_2 & a_{21} & & & & \\ c_3 & a_{31} & a_{32} & & & \\ \vdots & \vdots & & \ddots & & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array}$$

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

$$y^{n+1} = y^n + k\Psi(y^n, t^n, k) \tag{5.9}$$

where

$$\Psi(y^n, t^n, k) := f(y + \frac{1}{2}kf(y)). \tag{5.10}$$

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

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

Reference:

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

5.4 Multistep methods

The idea of multi-step method is to derive a relation between, for instance, y^{n+1}, y^n, y^{n-1} and y'^n and y'^{n-1} so that the corresponding truncation is small. The simplest multistep method is the 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.

We can also design a method which involves y^{n+1}, y^n, y^{n-1} and y'^n, y'^{n-1}. For instance,

$$y^{n+1} = y^n + \frac{k}{2}(3f(y^n) - f(y^{n-1}))$$

The truncation error

$$\tau^n := \frac{1}{k}\left(y^{n+1} - y^n + \frac{k}{2}(3f(y^n) - f(y^{n-1}))\right) = O(k^2).$$

A general r-step multistep method can be written in the form

$$\sum_{m=0}^{r} a_m y^{n+1-r+m} = k\sum_{m=0}^{r} b_m y'^{n+1-r+m} = k\sum_{m=0}^{r} b_m f^{n+1-r+m}. \tag{5.11}$$

We will always assume $a_r \neq 0$. When $b_r = 0$ the method is explicit; otherwise it is implicit. For a smooth solution of (5.3), 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 5.1. *A multispep method is called of order p if $\tau^n = O(k^p)$ uniformly in n. It is called consistent if $\tau^n(k) \to 0$ uniformly in n as $k \to 0$.*

Remark. When f is smooth, the solution of ODE $y' = f(y)$ 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 $k \to 0$. Let us set $a_m = 0$, $b_m = 0$ for $m > r$ for notational convinience. Taking Taylor expansion about t^{n+1-r}, we

get

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

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

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

To obtain order p scheme, we need to require

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

There are $2(r+1)$ unknowns for the coefficients $\{a_m\}_{m=0}^r$, $\{b_m\}_{m=0}^r$. In principle, we can choose $p = 2r + 1$ to have the same number of equations. Unfortunately, there is some limitation from stability requirement. The order of accuracy p is required to 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. We shall not discuss here. See Trefethen. 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 step size 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 and the step size $r = p$

- 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: $y^{n+1} = y^n + \frac{k}{24}(9f(y^{n+1}) + 19f(y^n) - 5f(y^{n-1}) + f(y^{n-2}))$

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 scheme: 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 problem, we should use implicit scheme instead.

Formal algebra 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. The Taylor expansion

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

can be expressed formally as

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

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,

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

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

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

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

$$\frac{a(e^\kappa)}{b(e^\kappa)} = \kappa + O(\kappa^{p+1}) \text{ as } \kappa \to 0. \tag{5.12}$$

We have the following theorem

Theorem 5.2. *A multistep method with $b(1) \neq 0$ is of order p if and only if*

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

It is consistent if and only if

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

Proof. The first formula can be obtained from (5.12) by writing $e^\kappa = z$. For the second formula, we expand $\log z$ about 1. We can 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

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

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

Homeworks.

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 previous exercise.

5.5 Linear difference equation

Second-order linear difference equation. In the linear case $y' = \lambda y$, the above difference scheme results in a linear difference equation. Let us consider general second order linear difference equation with constant coefficients:

$$ay^{n+1} + by^n + cy^{n-1} = 0, \tag{5.13}$$

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

$$a\rho^2 + b\rho + c = 0.$$

There are two solutions ρ_1 and ρ_2. In case $\rho_1 \neq \rho_2$, these two solutions 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 is

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

where C_1, C_2 are constants. In case of $\rho_1 = \rho_2$, then we can use the two solutions ρ_2^n and ρ_1^n with $\rho_2 \neq \rho_1$, but very closed, to produce another nontrivial solution:

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

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

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

Linear finite difference equation of order r . We consider general linear finite difference equation of order r:

$$a_r y^{n+r} + \cdots + a_0 y^n = 0, \tag{5.14}$$

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

$$a(\rho) := a_r\rho^r + \cdots + a_0 = 0. \tag{5.15}$$

The polynomial $a(\rho)$ is called the characteristic polynomial of (5.14) 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.

In the case of simple roots, we can express general solution as

$$y^n = C_1 \rho_1^n + \cdots + C_r \rho_r^n,$$

where the constants $C_1, ..., C_r$ are determined by

$$y^i = C_1 \rho_1^i + \cdots + C_r \rho_r^i, i = 0, ..., -r+1.$$

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

$$\mathbf{A}_0 \mathbf{y}^{n+1} = \mathbf{A}\mathbf{y}^n \tag{5.16}$$

where

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

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

We may divide (5.16) by $\mathbf{A}_0$ and get

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

We call $\mathbf{G}$ the fundamental matrix of (5.16). 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:

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

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

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

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

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

Definition 5.2. *The fundamental matrix $\mathbf{G}$ is called stable if $\mathbf{G}^n$ remains bounded under certain norm $\|\cdot\|$ for all n.*

Theorem 5.3. *The fundamental matrix $\mathbf{G}$ is stable if and only if its eigenvalues satisfy the following condition:*

$$\begin{array}{l} \textit{either } |\rho| = 1 \textit{ and } \rho \textit{ is a simple root}, \\ \textit{or } |\rho| < 1 \end{array} \tag{5.17}$$

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. □

Nonhomogeneous linear finite difference system In general, we consider the nonhomogeneous linear difference system:

$$\mathbf{y}^{n+1} = \mathbf{G}\mathbf{y}^n + \mathbf{f}^n \tag{5.18}$$

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. 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 $y' = \lambda y$.

5.6 Stability analysis

There are two kinds of stability concepts.

- Fix $t = nk$, the computed solution y^n remains bounded as $n \to \infty$ (or equivalently, $k \to 0$).
- Fix $k > 0$, the computed solution y^n remains bounded as $n \to \infty$.

The first one is refered to zero stability. The second is called absolute stability.

5.6.1 Zero Stability

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

$$\mathcal{L}y^n = \sum_{m=0}^{r} a_m y^{n+1-r+m} - k\sum_{m=0}^{r} b_m f(t^{n+1-r+m}, y^{n+1-r+m}) = 0. \tag{5.19}$$

Definition 5.3. *The truncation error $\tau^n(k)$ for the above finite difference 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 5.4. *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 the multistep method, 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 5.5. *A difference scheme is called zero stable if its solutions at time step n remain bounded as the mesh size $k \to 0$ ($nk = T$ is fixed, according $n \to \infty$).*

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

Theorem 5.4 (Dahlquist)**.** *For finite difference schemes for ODE $y' = f(t, y)$,*

$$\textit{consistency} \ + \ \textit{zero-stability} \ \Leftrightarrow \ \textit{convergence}$$

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

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

to be bounded. We 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

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

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

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:

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

We have seen in the last section that

Theorem 5.5. *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:*

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

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, we choose y^0 such that $\mathbf{y}^0 \cdot \mathbf{y}_i \neq 0$. Then the corresponding $bfy^n := \mathbf{A}^n\mathbf{y}^0$ will be unbounded. Hence it cannot converge to a constant, as $k \to 0$. Here, we use that fact that $\mathbf{y}^0 \cdot \mathbf{y}_i \neq 0$. We generate $bfy^0 = (y_0^{r-1}, \cdots, y_0)^T$ by some explicit scheme starting from y^0. The point is that bfy^0 depends on the mesh size k and $\mathbf{y}^0(k) \to (y_0, \cdots, y_0)^T$ as $k \to 0$. With this, given any $\mathbf{y}^i$, we can always construct $\mathbf{y}^0(k)$ such that $\mathbf{y}^0(k) \cdot \mathbf{y}_i \neq 0$ when $k \neq 0$. □

Convergence ⇒ Consistency

Proof. We need to show that $a(1) = 0$ and $a'(1) = b(1)$. To show the first, we consider $y' = 0$ with $y(0) = 1$. For the second, we consider $y' = 1$ 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
$$y^r = -a_0y^0 - \cdots - a_{r-1}y^{r-1} = -a_0 - \cdots - a_{r-1}.$$
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
$$a(Z)y^n - kb(Z)1 = 0. \tag{5.20}$$
We write
$$a(Z) = a'(1)(Z-1) + O((Z-1)^2), b(Z)1 = b(1).$$
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 (5.20) provided its initial data y^n has the same form for $0 \leq n < r$. Since $nk = t$, the convergence $y^n \to t$ as $n \to \infty$ forces $\frac{b(1)}{a'(1)} = 1$.

□

Stability + Consistency $\Rightarrow$ Convergence

Proof. We recall that we can express the scheme as

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

Let Y be an exact solution, then plug it into the above =scheme, we get

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

We substract these two and call $\mathbf{e}^n := \mathbf{Y}^n - \mathbf{y}^n$. We get

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

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

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

with C independent of n and k. The reason is the follows. First, we assume that f is Lipschitz. Thus, the functions L_{-m} above are uniformly bounded (independent of n). Hence the term $\|\mathbf{BL}\|$ is uniformly bounded. Second we have a lemma

Lemma 5.1. *If $\|\mathbf{A}^n\|$ is bounded and $\|\mathbf{B}_n\|$ are uniformly bounded, then the product*

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

is also uniformly bounded.

We have

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

From the lemma, we get

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

□