Math 6620 : Analysis of Numerical Methods II

Chee Han Tan

Last modified : August 13, 2017

Contents

Abstract: These notes are largely based on **Math 6620: Analysis of Numerical Methods II** course, taught by Yekaterina Epshteyn in Spring 2017, at the University of Utah. Additional examples or remarks or results from other sources are added as I see fit, mainly to facilitate my understanding. These notes are by no means accurate or applicable, and any mistakes here are of course my own. Please report any typographical errors or mathematical fallacy to me by email tan@math.utah.edu

Chapter 1

Solvers For Nonlinear Equations

Root-finding problems are one of the most fundamental problem in Applied Mathematics, due to the fact that almost any problems can be rephrase as solving the equation $f(x) = 0$. In general, explicit solutions are not possible and one must be contented with being able to numerically compute a root to any specified degree of accuracy. In this chapter, we will study three classical root-finding algorithms belonging to the more general class of **iterative methods**; the central idea is one chooses an initial guess to initiate the algorithm, which then generates sequence of successive approximations to a solution. Let us first define the convergence and rate of convergence for iterative methods:

Definition 1.0.1 ([Atk08])**.** Given an iterative method, a sequence of iteratives $(x_n)_{n=0}^{\infty}$ is said to converge with **order $p \geqslant 1$** to a point α if there exists a constant $C > 0$ such that

$$|\alpha - x_{n+1}| \leqslant C|\alpha - x_n|^p \quad \text{for all } n \geqslant 0. \tag{1.0.1}$$

If $p = 1$, the iterative method is said to have linear convergence; if $p = 2$, the iterative method is said to have quadratic convergence.

Remark 1.0.2. This definition of order of convergence is inconvenient for some linearly convergent iterative methods. Indeed, iterating the inequality (1.0.1) with $p = 1$ yields

$$|\alpha - x_n| \leqslant C^n|\alpha - x_0| \quad \text{for all } n \geqslant 0. \tag{1.0.2}$$

Observe that (1.0.2) is a weaker condition compare to (1.0.1) and can be shown directly in some cases. If this were true, the iterative method will still be said to converge linearly provided (1.0.2) holds for $C < 1$.

1.1 Bisection Method

Assume that $f \in C[a,b]$ such that $f(a)f(b) < 0$. From the **Intermediate Value Theorem**, it follows that there exists at least one root $\alpha \in [a,b]$. The idea is to repeatedly halved the interval $[a,b]$ and then selects a subinterval containing α, until we reach the desired accuracy. Below we present a pseudocode for the bisection method:

Bisection Method: Bisect$(f, a, b, \text{root}, \varepsilon)$

1. Define $c_1 := (a+b)/2$.
2. If either $|b - c_1| < \varepsilon$ or $|a - c_1| < \varepsilon$, set root:= c_1 and exit. Otherwise, check the sign of $f(c_1)$.
3. If $f(a)f(c_1) < 0$, then set $b := c_1$ and return to Step 1. Otherwise, set $a := c_1$ and return to Step 1.

On completion of the algorithm, c_n will be an approximation to the root with $|\alpha - c_n| < \varepsilon$ due to Step 2. Moreover, the interval $[a,b]$ is halved in size after every successive iteration. This leads to the following upper bound:

$$|\alpha - c_n| \leqslant \left(\frac{1}{2}\right)^n (b-a) \quad \text{and} \quad \lim_{n\to\infty} c_n = \alpha. \tag{1.1.1}$$

From Remark (1.0.2), we say that the bisection method converges linearly with a rate of $C = 1/2$. Note that the actual error may not decrease by a factor of $1/2$ at each iteration, but the average rate of decrease is $1/2$. (1.1.1) tells us how many iterations are needed to achieve a given accuracy $\varepsilon > 0$. Indeed,

$$\frac{b-a}{2^n} \leqslant \varepsilon \iff n \geqslant \frac{\ln(b-a) - \ln(\varepsilon)}{\ln(2)}.$$

Bisection method is guaranteed to converge provided $f \in C^0[a,b]$ and $a, b \in \mathbb{R}$ are such that $f(a)f(b) < 0$. Moreover, we obtain a robust error estimate (1.1.1) and the rate of convergence $C = 1/2$ is independent of initial guesses a, b. The downside is that it converges very slowly comparing to Newton's method and secant method, as we shall see in Section 1.2 and Section 1.3. In practice, the bisection method is often used to obtain a "good" initial guess for rapidly convergent methods.

1.2 Newton's Method

Albeit some difficulties and limitations, Newton's method is an extremely powerful root-finding algorithm since it converges quadratically in general. Applications of Newton's method include optimisation problems and solving nonlinear equations. In what follows, we first derive Newton's method both geometrically and algebraically. We then prove a convergence theorem of Newton's method and discuss its limitations. Lastly, we provide a pseudocode for the method.

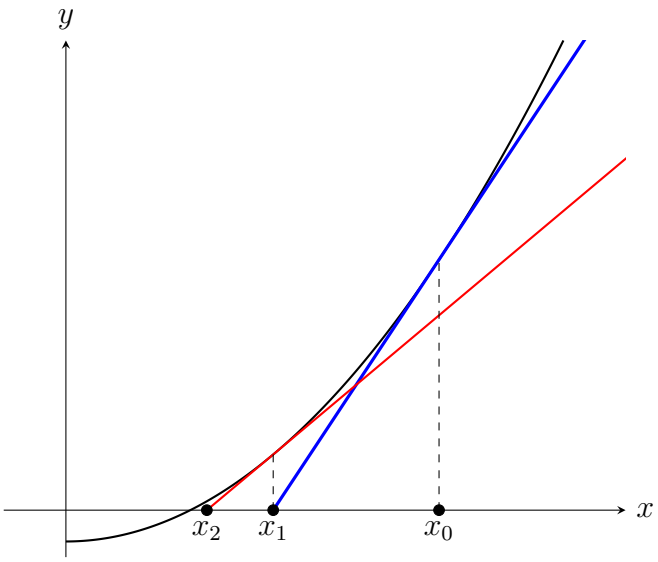


Figure 1.1: Newton's method with $f(x) = x^2 - 1$ and $x_0 = 3$. We obtain $x_2 = \dfrac{34}{30} \approx 1.13$.

Consider an initial guess x_0 of the desired root α of $f(x) = 0$, which we may assumed to be sufficiently close to α. The main idea is to represent $f(x)$ in a neighbourhood of x_0 with its linear approximation, *i.e.*

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0).$$

We then use the root of this tangent line, denoted by x_1, to approximate α. Since the equation of the tangent line is $y - f(x_0) = f'(x_0)(x - x_0)$, we obtain:

$$-f(x_0) = f'(x_0)(x_1 - x_0) \implies x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}, \quad \text{if } f'(x_0) \neq 0.$$

This procedure can be repeated and provided $f'(x_n) \neq 0$ for each $n \geqslant 0$, we obtain the recursive relation for x_n:

$$\boxed{x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n \geqslant 0.} \qquad \text{(Newton's method)}$$

Alternatively, one can derive (1.2.1) using a Taylor series approach. More precisely, assume $f \in C^2$ in some neighbourhood of α. Expanding $f(x)$ about the point x_n yields

$$f(x) = f(x_n) + f'(x_n)(x - x_n) + \frac{f''(\xi_n)}{2!}(x - x_n)^2, \quad \text{for some } \xi_n \text{ between } x \text{ and } x_n.$$

Setting $x = \alpha$ and using $f(\alpha) = 0$ gives

$$\begin{aligned} 0 &= f(x_n) + f'(x_n)(\alpha - x_n) + \frac{f''(\xi_n)}{2}(\alpha - x_n)^2 \\ \implies \alpha &= \underbrace{\left(x_n - \frac{f(x_n)}{f'(x_n)}\right)}_{x_{n+1}} - \underbrace{\frac{f''(\xi_n)}{2f'(x_n)}(\alpha - x_n)^2}_{\textbf{Error term}}, \end{aligned} \tag{1.2.1}$$

where ξ_n is now between α and x_n.

1.2.1 Convergence Analysis

We now prove a convergence result which shows the speed of convergence and also an interval from which initial guesses can be chosen.

Theorem 1.2.1. *Suppose $f \in C^2$ in some neighbourhood of α where $f(\alpha) = 0, f'(\alpha) \neq 0$. If x_0 is chosen sufficiently close to α, the iterates $(x_n)_{n=0}^{\infty}$ of (Newton's method) will converge to α. Moreover, we have the following asymptotic rate:*

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{(\alpha - x_n)^2} = -\frac{f''(\alpha)}{2f'(\alpha)}, \tag{1.2.2}$$

which shows that Newton's method has quadratic convergence.

Proof. Choose a sufficiently small interval $I_\varepsilon = [\alpha - \varepsilon, \alpha + \varepsilon]$ on which $f'(x) \neq 0$ on I_ε; such interval exists by continuity of f'. Set

$$M := \frac{\max\limits_{x\in I_\varepsilon} |f''(x)|}{2 \min\limits_{x\in I_\varepsilon} |f'(x)|} = \max_{x\in I_\varepsilon} \frac{1}{2}\left(\frac{|f''(x)|}{|f'(x)|}\right).$$

Choose $x_0 \in I_\varepsilon$. From (1.2.1) with $n = 0$, we have the following estimate:

$$|\alpha - x_1| \leqslant M|\alpha - x_0|^2 \implies M|\alpha - x_1| \leqslant \Big[M|\alpha - x_0|\Big]^2.$$

If we choose x_0 satisfying $M|\alpha - x_0| < 1$ in addition to $x_0 \in I_\varepsilon$, then

$$M|\alpha - x_1| \leqslant M|\alpha - x_0| < 1 \quad \text{and} \quad |\alpha - x_1| \leqslant |\alpha - x_0| \leqslant \varepsilon.$$

A similar argument shows that

$$|\alpha - x_n| \leqslant \varepsilon \quad \text{and} \quad M|\alpha - x_n| < 1 \quad \text{for all } n \geqslant 1.$$

Convergence is now easy to show. Indeed, iterating (1.2.3) yields

$$M|\alpha - x_n| \leqslant \Big[M|\alpha - x_{n-1}|\Big]^2 \leqslant \Big[M|\alpha - x_{n-2}|\Big]^{2^2} \leqslant \ldots \leqslant \Big[M|\alpha - x_0|\Big]^{2^n}.$$

Since $M|\alpha - x_0| < 1$, we have that

$$|\alpha - x_n| \leqslant \frac{1}{M}\Big[M|\alpha - x_0|\Big]^{2^n} \longrightarrow 0 \text{ as } n \longrightarrow \infty.$$

Moreover, this implies that $\xi_n \longrightarrow \alpha$ as $n \longrightarrow \infty$ since ξ_n lies between x_n and α. Finally, using the assumption that $f \in C^2(I_\varepsilon)$,

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{(\alpha - x_n)^2} = -\lim_{n\to\infty} \frac{f''(\xi_n)}{2f'(x_n)} = -\frac{f''(\alpha)}{2f'(\alpha)}.$$

■

Remark 1.2.2.

1. This is a local result and the interval of convergence is not known a-priori. We stress again that the Newton's method has quadratic convergence if x_0 is chosen in such a way that
$$|\alpha - x_0| \leqslant \min\left\{\varepsilon, \frac{1}{M}\right\}.$$
The constant $\dfrac{1}{M}$ ensures that all the iterates $(x_n)_{n=0}^{\infty}$ remains in the interval I_ε where $f'(x_n) \neq 0$.

2. The assumption $f'(\alpha) \neq 0$ says that α is a root of multiplicity 1. If α has multiplicity $p > 1$, some modifications on the iteration formula is required in order to preserve quadratic convergence. We will see the following proposed modification in Section 1.5:
$$x_{n+1} = x_n - p\left(\frac{f(x_n)}{f'(x_n)}\right).$$
We will also show in Section 1.5 that it is possible to achieve convergence in any order of accuracy if we increase the regularity of f.

3. If f is not C^2, then we would still expect convergence, but the order of convergence might be linear instead of quadratic.

1.2.2 Error Estimate

Using the **Mean Value Theorem**,
$$f(x_n) = f(x_n) - f(\alpha) = f'(\xi_n)(x_n - \alpha) \implies \alpha - x_n = -\frac{f(x_n)}{f'(\xi_n)},$$
with ξ_n between x_n and α. If $f'(x)$ is not changing too rapidly near α, *i.e.* f is not oscillating near α, then we have $f'(\xi_n) \approx f'(x_n)$ and
$$\alpha - x_n \approx -\frac{f(x_n)}{f'(x_n)} = x_{n+1} - x_n, \tag{1.2.3}$$
where the last equality follows from the iterative formula for (Newton's method). Note that (1.2.3) corresponds to absolute error estimate. For relative error estimate,
$$\frac{\alpha - x_n}{\alpha} \approx \frac{x_{n+1} - x_n}{x_{n+1}}.$$
Note that these are emperical estimates under the assumption of Theorem 1.2.1.

Despite the quadratic convergence, Newton's method is very sensitive to initial guess. Usually there are clear choices of an initial guess x_0 that comes from the problem, otherwise one can run a few iterations of bisection method to obtain a reasonable x_0. The other downside is the need to evaluate the derivative explicitly, which could be computationally expensive. One way to overcome this difficulty is to approximate the derivative using suitable finite-differences, but this comes at the cost of a somewhat slower speed of convergence.

Newton's method: Newton(f, df, x_0, ε,root, itmax, ier), where

$$\begin{aligned} df &= \text{derivative } f' \\ \varepsilon &= \text{tolerance} \\ \text{root} &= \text{numerical root} \\ \text{itmax} &= \text{maximum number of iterations} \\ \text{ier} &= \text{error flag.} \end{aligned}$$

1. itnum := 1.
2. Set denom := $df(x_0)$.
3. If denom = 0, then set ier := 2 and exit.
4. $x_1 = x_0 - \dfrac{f(x_0)}{\text{denom}}$.
5. If $|x_1 - x_0| \leqslant \varepsilon$, then set ier := 0, root := x_1 and exit.
6. If itnum = itmax, set ier := 1 and exit.
7. Otherwise, set itnum := itnum + 1, $x_0 = x_1$ and go to Step 2.

1.3 Secant Method

The secant method can be seen as a modification of Newton's method, in which one replaces the derivative f' by the first-order backward finite-difference. More precisely, let x_0, x_1 be two initial guesses of the desired root α of $f(x) = 0$. Recall that in Newton's method, we performed a linear approximation for $f(x)$ in the neighbourhood of x_0. An alternative to approximate the graph of $f(x)$ is using the **secant line** determined by $(x_0, f(x_0))$ and $(x_1, f(x_1))$. The root of this secant line, denoted by x_2, is then used to approximate α. Since x_0, x_1, x_2 are collinear,

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

and solving for x_2 gives:

$$x_2 = x_1 - f(x_1) \underbrace{\left(\frac{(x_1 - x_0)}{f(x_1) - f(x_0)} \right)}_{\approx 1/f'(x_1)}.$$

Repeating this procedure yields the following iteration formula, known as the **secant method**

$$x_{n+1} = x_n - f(x_n) \left[\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \right], \quad n \geqslant 1. \qquad \text{(Secant method)}$$

It does not necessarily converge, but when it does, the speed of convergence is usually greater than the bisection method.

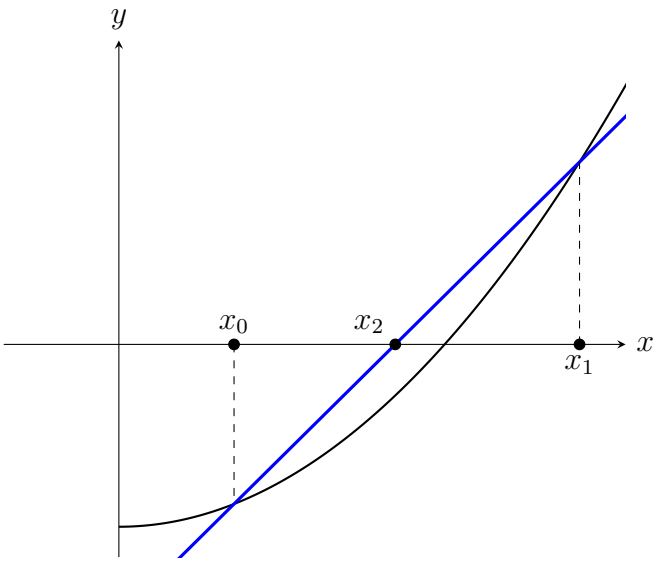


Figure 1.2: Secant method with $f(x) = x^2 - 2$ and $x_0 = 1, x_1 = 2$. We obtain $x_2 = 1.2$.

Theorem 1.3.1. *Assume $f \in C^2$ in some interval containing α, where $f(\alpha) = 0, f'(\alpha) \neq 0$. If x_0, x_1 are chosen sufficiently close to α, the iterates $(x_n)_{n=0}^{\infty}$ of the* (Secant method) *converges to α. Moreover, we have the following asymptotic rate*

$$\lim_{n\to\infty} \frac{|\alpha - x_{n+1}|}{|\alpha - x_n|^{(1+\sqrt{5})/2}} = \left|\frac{f''(\alpha)}{2f'(\alpha)}\right|^{(\sqrt{5}-1)/2}, \tag{1.3.1}$$

which shows that the order of convergence is $p = \dfrac{1+\sqrt{5}}{2} \approx 1.62$.

Proof. Following the proof of Theorem 1.2.1, choose a sufficiently small interval $I_\varepsilon = [\alpha - \varepsilon, \alpha + \varepsilon]$ on which $f'(x) \neq$ on I_ε. (Read proof from [Atk08, pp. 68-69]). ■

The secant method only requires one function evaluation per iterate if we store the previous value, whereas Newton's method requires two function evaluations per iterate. In terms of computation, the secant method is more appealing since it is less costly comparing to Newton's method, but it is generally slower.

1.4 One-Point Iteration Methods

It turns out that root-finding problems belong to a more general class of problems, known as fixed-point problem, which has the form $x = g(x)$. Consider solving $x^2 - \alpha = 0, \alpha > 0$. We can reformulate this problem in terms of $x = g(x)$ several ways:

1. $x = x^2 + x - \alpha$;
2. $x = \alpha/x$;
3. $x = \dfrac{1}{2}\left(x + \dfrac{\alpha}{x}\right)$.

Consider solving for $x = \alpha$ the equation $x = g(x)$ using the iteration method:

$$x_{n+1} = g(x_n), \quad n \geqslant 0. \tag{FPI}$$

Such solution is called a **fixed point** of g. In the case of Newton's method, we have

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Lemma 1.4.1. *Given $g \in C[a,b]$ such that $g\colon [a,b] \longrightarrow [a,b]$. Then g has at least one fixed point $\alpha \in [a,b]$ satisfying $\alpha = g(\alpha)$.*

Proof. Consider the function $f(x) = g(x) - x$ which is continuous on $[a,b]$. The assumption $a \leqslant g(x) \leqslant b$ implies

$$\begin{aligned} f(a) &= g(a) - a \geqslant 0 \\ f(b) &= g(b) - b \leqslant 0. \end{aligned}$$

It follows from the **Intermediate Value Theorem** that there exists $c \in [a,b]$ such that $f(c) = g(c) - c = 0$.

■

Theorem 1.4.2. *Let $g \in C[a,b]$ such that $g\colon [a,b] \longrightarrow [a,b]$. If there exists a constant $\lambda \in (0,1)$ such that*

$$|g(x) - g(y)| \leqslant \lambda |x - y| \quad \textit{for all } x, y \in [a,b],$$

then g has a unique fixed point $\alpha \in [a,b]$. Moreover, the iterates $(x_n)_{n=0}^{\infty}$ of (FPI) converges to the fixed point α for any choice of initial guess $x_0 \in [a,b]$ and we have the following a-priori error estimate:

$$|\alpha - x_n| \leqslant \frac{\lambda^n}{1-\lambda} |x_1 - x_0|. \tag{1.4.1}$$

Proof. Observe that existence of a fixed point of g follows from Lemma 1.4.1. To prove uniqueness, suppose there are two fixed points α, β of g in $[a,b]$. Then

$$|\alpha - \beta| = |g(\alpha) - g(\beta)| \leqslant \lambda |\alpha - \beta| \implies (1-\lambda)|\alpha - \beta| \leqslant 0,$$

and it follows that $\alpha = \beta$ since $(1-\lambda) > 0$. Note that the iterates $(x_n)_{n=0}^{\infty} \in [a,b]$ since g maps $[a,b]$ into itself. Let α be the fixed point of g. Then

$$|\alpha - x_{n+1}| = |g(\alpha) - g(x_n)| \leqslant \lambda |\alpha - x_n|,$$

and iterating this inequality yields the inequality:

$$|\alpha - x_n| \leqslant \lambda^n |\alpha - x_0| \longrightarrow 0 \quad \text{as } n \longrightarrow \infty,$$

since $\lambda \in (0,1)$. Thus, $x_n \longrightarrow \alpha$ as $n \longrightarrow \infty$ for any choice of initial guess $x_0 \in [a,b]$. Finally, to prove (1.4.1), applying triangle-inequality yields

$$\begin{aligned} |\alpha - x_0| \leqslant |\alpha - x_1| + |x_1 - x_0| &= |g(\alpha) - g(x_0)| + |x_1 - x_0| \\ &\leqslant \lambda |\alpha - x_0| + |x_1 - x_0| \end{aligned} \tag{1.4.2}$$

and so

$$|\alpha - x_n| \leqslant \lambda^n |\alpha - x_0| \leqslant \frac{\lambda^n}{1-\lambda} |x_1 - x_0|.$$

■

Remark 1.4.3. The error estimate (1.4.1) shows that the iteration (FPI) converges linearly, with the rate of convergence bounded by λ. From (1.4.2), if we replace the pair (x_0, x_1) by (x_n, x_{n+1}), then we obtain the following *a-posteriori error estimate*:

$$|\alpha - x_{n+1}| \leqslant \lambda|\alpha - x_n| \leqslant \frac{\lambda}{1-\lambda}|x_{n+1} - x_n| \quad \text{for any } \ n \geqslant 0.$$

There is a caveat: this error estimate is not applicable when $\lambda \approx 1$.

Theorem 1.4.4. *Assume $g \in C^1[a,b]$ such that $g\colon [a,b] \longrightarrow [a,b]$ and*

$$\lambda := \max_{x\in[a,b]} |g'(x)| < 1.$$

Then g has a unique fixed point $\alpha \in [a,b]$. Moreover, the iterates $(x_n)_{n=0}^{\infty}$ of (FPI) converges to the fixed point α for any choice of initial guess $x_0 \in [a,b]$ and

$$|\alpha - x_n| \leqslant \lambda^n|\alpha - x_0| \leqslant \frac{\lambda^n}{1-\lambda}|x_1 - x_0|.$$

Moreover, the following asymptotic rate holds:

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{\alpha - x_n} = g'(\alpha),$$

which says that the convergence of (FPI) is linear if $g'(\alpha) \neq 0$.

Proof. Every result is a consequence of Theorem 1.4.2 except for the asymptotic rate. To show the asymptotic rate, applying the **Mean Value Theorem** yields

$$\alpha - x_{n+1} = g(\alpha) - g(x_n) = g'(\xi_n)(\alpha - x_n),$$

for some ξ_n between α and x_n. Since $x_n \longrightarrow \alpha$ as $n \longrightarrow \infty$, we also have $\xi_n \longrightarrow \alpha$ as $n \longrightarrow \infty$. Consequently,

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{\alpha - x_n} = \lim_{n\to\infty} g'(\xi_n) = g'(\alpha),$$

where the last equality follows from the assumption $g \in C^1[a,b]$.

■

Corollary 1.4.5. *Assume α is a fixed point of g, where $g \in C^1$ in some interval around α with $|g'(\alpha)| < 1$. Then Theorem 1.4.4 still holds, provided the initial guess x_0 is chosen sufficiently close to α.*

Proof. Choose a number $\lambda > 0$ satisfying $|g'(\alpha)| < \lambda < 1$. Next, choose a sufficiently small interval $I_\varepsilon = [\alpha - \varepsilon, \alpha + \varepsilon]$ such that

$$\max_{x\in I_\varepsilon} |g'(x)| \leqslant \lambda < 1.$$

We claim that $g(I_\varepsilon) \subset I_\varepsilon$: for any $x \in I_\varepsilon$ we have that for some $\xi \in I_\varepsilon$

$$|\alpha - g(x)| = |g(\alpha) - g(x)| = |g'(\xi)||\alpha - x| \leqslant \lambda|\alpha - x| \leqslant \varepsilon.$$

The desired result follows by applying Theorem 1.4.4 on the interval I_ε.

■

1.5 Higher Order One-Point Methods

We extend Theorem 1.4.2 by considering methods with an order of convergence $p > 1$.

Theorem 1.5.1. *Assume α is a fixed point of g, where $g \in C^p$ in some interval around α for some $p \geqslant 2$ and*

$$g'(\alpha) = g''(\alpha) = \ldots = g^{(p-1)}(\alpha) = 0.$$

If the initial guess x_0 is chosen sufficiently close to α, then the iteration (FPI) will have an order of convergence p. Moreover, we have the following asymptotic rate:

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{(\alpha - x_n)^p} = (-1)^{p-1} \left(\frac{g^{(p)}(\alpha)}{p!} \right).$$

Proof. It follows from Corollary 1.4.5 that the iteration (FPI) converges since $|g'(\alpha)| = 0 < 1$. To obtain the asymptotic rate, we expanding $g(x_n)$ about the point α to get:

$$\begin{aligned} x_{n+1} = g(x_n) = g(\alpha) + g'(\alpha)(x_n - \alpha) + \frac{g''(\alpha)}{2!}(x_n - \alpha)^2 + \ldots \\ \ldots \quad + \frac{g^{(p-1)}(\alpha)}{(p-1)!}(x_n - \alpha)^{p-1} + \frac{g^{(p)}(\xi_n)}{p!}(x_n - \alpha)^p \end{aligned}$$

where ξ_n is some number between α and x_n. Since $g^{(j)}(\alpha) = 0$ for every $j = 1, \ldots, p-1$ by assumption, the above reduces to:

$$\frac{x_{n+1} - \alpha}{(x_n - \alpha)^p} = \frac{1}{p!} g^{(p)}(\xi_n) \implies \frac{\alpha - x_{n+1}}{(\alpha - x_n)^p} = \frac{(-1)^{p-1}}{p!} g^{(p)}(\xi_n).$$

Consequently,

$$\lim_{n\to\infty} \frac{\alpha - x_{n+1}}{(\alpha - x_n)^p} = \lim_{n\to\infty} \frac{(-1)^{p-1}}{p!} g^{(p)}(\xi_n) = (-1)^{p-1} \left(\frac{g^{(p)}(\alpha)}{p!} \right),$$

where the last equality follows from the assumption that $g \in C^p$.

■

Example 1.5.2. As an illustration, we apply Theorem 1.5.1 onto Newton's method. Since $g(x) = x - \dfrac{f(x)}{f'(x)}$,

$$\begin{aligned} g'(x) &= 1 - \left(1 - \frac{f(x)f''(x)}{[f'(x)]^2} \right) = \frac{f(x)f''(x)}{[f'(x)]^2} \\ g''(x) &= \frac{[f'(x)]^2[f(x)f''(x) + f'(x)f''(x)] - f(x)f''(x)[2f'(x)f''(x)]}{[f'(x)]^4} \\ &= \frac{f(x)[f'(x)]^2 f'''(x) + [f'(x)]^3 f''(x) - 2f(x)f'(x)[f''(x)]^2}{[f'(x)]^4} \\ &= \frac{f(x)f'(x)f'''(x) + [f'(x)]^2 f''(x) - 2f(x)[f''(x)]^2}{[f'(x)]^3}. \end{aligned}$$

It follows from $f(\alpha) = 0, f'(\alpha) \neq 0$ that $g'(\alpha) = 0, g''(\alpha) = \dfrac{f''(\alpha)}{f'(\alpha)}$ and we recover the quadratic convergence for Newton's method. Note that if $f''(\alpha) = 0$, then $g''(\alpha) = 0$ and we obtain super-convergence ($p = 3$).

1.6 Newton's Method For Repeated Roots

Considering Newton's method as a fixed point problem, *i.e.*

$$x_{n+1} = g(x_n) := x_n - \frac{f(x_n)}{f'(x_n)}, \quad n \geqslant 0. \tag{1.6.1}$$

Suppose $f(x)$ has a root of multiplicity $p > 1$, we can rewrite $f(x)$ as:

$$f(x) = (x-\alpha)^p h(x), \tag{1.6.2}$$

with $h(\alpha) \neq 0$ and h continuous at $x = \alpha$. Observe that if h is sufficiently differentiable at $x = \alpha$, then (1.6.2) is equivalent to

$$f(\alpha) = f'(\alpha) = \ldots = f^{(p-1)}(\alpha) = 0, \quad f^{(p)}(\alpha) \neq 0. \tag{1.6.3}$$

First, simplify $g(x)$ by computing $f'(x)$:

$$\begin{aligned} f'(x) &= p(x-\alpha)^{p-1}h(x) + (x-\alpha)^p h'(x) \\ \implies g(x) &= x - \frac{(x-\alpha)^p h(x)}{p(x-\alpha)^{p-1}h(x) + (x-\alpha)^p h'(x)} \\ &= x - \frac{(x-\alpha)h(x)}{ph(x) + (x-\alpha)h'(x)}. \end{aligned}$$

Next, computing $g'(x)$ yields

$$g'(x) = 1 - \left(\frac{h(x)}{ph(x) + (x-\alpha)h'(x)} + (x-\alpha)\frac{d}{dx}\left(\frac{h(x)}{ph(x) + (x-\alpha)h'(x)} \right) \right)$$

It follows that $g'(\alpha) = 1 - \dfrac{1}{p} \neq 0$ since $p > 1$. Consequently, Newton's method for repeated root converges linearly with rate of convergence $1 - \dfrac{1}{p} = \dfrac{p-1}{p}$.

To recover quadratic convergence, we need to modify $g(x)$ such that $g'(\alpha) = 0$. From the expression of $g'(x)$, we propose a new function $g_p(x)$ that has the form:

$$g_p(x) = x - p\left(\frac{f(x)}{f'(x)} \right).$$

By construction, $g_p'(\alpha) = 0$. Moreover, for some ξ_n between α and x_n we have:

$$\begin{aligned} \alpha - x_{n+1} = g(\alpha) - g(x_n) &= -[g(x_n) - g(\alpha)] \\ &= -\left[g'(\alpha)(x_n - \alpha) + \frac{g''(\xi_n)}{2!}(x_n - \alpha)^2 \right] \\ &= -\frac{g''(\xi_n)}{2}(x_n - \alpha)^2, \end{aligned}$$

which is consistent with the asymptotic rate from Theorem 1.5.1.

1.7 Problems

1. Consider the iterative method to find a root $\alpha \in [a, b]$ of $f(x)$

$$x_{n+1} = g(x_n) := x_n - \left(\frac{b-a}{f(b)-f(a)}\right) f(x_n). \tag{1.7.1}$$

Show that the sequence of the iterates x_n converges to the root α and state the required conditions on $f(x)$. What is the expected order of the convergence?

Solution: First, if g is sufficiently differentiable near $x = \alpha$, then its pth derivative with respect to x, evaluated at $x = \alpha$ for $p \geqslant 1$ is:

$$g'(\alpha) = 1 - \left(\frac{b-a}{f(b)-f(a)}\right) f'(\alpha)$$
$$g^{(p)}(\alpha) = -\left(\frac{b-a}{f(b)-f(a)}\right) f^{(p)}(\alpha), \quad p \geqslant 2.$$

Below we present three different convergence results, each having different assumptions on f:

(a) Suppose $g'(\alpha) = 0$, *i.e.* $f'(\alpha) = \dfrac{f(b)-f(a)}{b-a}$. Assuming $f \in C^p$ in some interval around α for $p \geqslant 2$ and

$$f'(\alpha) = \frac{f(b)-f(a)}{b-a}, \quad f''(\alpha) = \ldots = f^{(p-1)}(\alpha) = 0,$$

and the initial guess x_0 is chosen sufficiently close to α. It follows from Theorem 1.5.1 that the iteration (1.7.1) converges, with order of convergence p.

(b) Suppose $g'(\alpha) \neq 0$. There are two possible convergence results:

i. Assume $g \in C^1$ in some interval around α, which amounts to assuming $f \in C^1$ in some interval around α, together with

$$|g'(\alpha)| = \left|1 - \left(\frac{b-a}{f(b)-f(a)}\right) f'(\alpha)\right| < 1,$$

which is equivalent to

$$0 < \left(\frac{b-a}{f(b)-f(a)}\right) f'(\alpha) < 2.$$

If the initial guess x_0 is chosen sufficiently close to α, then the iteration (1.7.1) converges from Corollary 1.4.5, with order of convergence $p = 1$.

ii. If $f \in C^1[a,b]$ such that for every $x \in [a,b]$,

$$\begin{cases} a \leqslant x - \left(\dfrac{b-a}{f(b)-f(a)}\right) f(x) \leqslant b, \\ 0 < \left(\dfrac{b-a}{f(b)-f(a)}\right) f'(x) < 2, \end{cases}$$

where each of these corresponds to $g\colon [a,b] \longrightarrow [a,b]$ and $\max\limits_{x\in[a,b]} |g'(x)| < 1$ respectively. Then for any choice of initial guess $x_0 \in [a,b]$, the iteration (1.7.1) converges, with order of convergence $p = 1$.

Chapter 2

Interpolation Theory

Given a finite set of data points $(x_j, y_j), j = 0, 1, \ldots, n$, the **interpolation problem** is to find a function $p(x)$ from a given class of functions that passes through these points, *i.e.* $p(x)$ satisfies

$$p(x_j) = y_j, \quad j = 0, 1, \ldots, n. \tag{2.0.1}$$

The points $x_0, x_1, \ldots, x_n$ are called the **interpolation points/nodes**. Clearly, a trivial solution exists by simply connecting all given points with straight lines. However, one usually seek smooth functions such as polynomials and trigonometric functions. We mention a well-known result, the **Weierstrass approximation theorem**, which states that every continuous function on a closed interval can be uniformly approximated by a polynomial function.

2.1 Polynomial Interpolation Theory

Ideas from polynomial interpolation theory is widely used in developing methods in the areas of approximation theory, numerical integration and the numerical solution of differential equation. Consider the problem of finding a polynomial $p(x)$ that interpolates the given data points (x_j, y_j), where $x_0, x_1, \ldots, x_n$ are assumed to be distinct real or complex numbers, with associated function values $y_0, y_1, \ldots, y_n$. This is a classic existence and uniqueness problem.

Observe that there exists infinitely many polynomials that interpolates the data if there is no restriction on $\deg(p)$. However, we will show in Theorem 2.1.1 that a unique interpolating polynomial p exists, provided $\deg(p) \leqslant n$. Consider a polynomial of degree n which has the form

$$p(x) = \sum_{j=0}^{n} a_j x^j = a_0 + a_1 x + \ldots + a_n x^n. \tag{2.1.1}$$

Substituting (2.0.1) into (2.1.1) yields the following system of $(n+1)$ linear equations

$$\begin{aligned} a_0 + a_1 x_0 + \ldots + a_n x_0^n &= y_0 \\ a_0 + a_1 x_1 + \ldots + a_n x_1^n &= y_1 \\ \vdots \qquad\qquad & \quad \vdots \\ a_0 + a_1 x_n + \ldots + a_n x_n^n &= y_n. \end{aligned}$$

This can be written in the matrix form $V\boldsymbol{a} = \boldsymbol{y}$, where $V \in \mathbb{C}^{(n+1)\times(n+1)}$ is the **Vandermonde matrix**

$$V = \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix}. \tag{2.1.2}$$

Theorem 2.1.1 (Existence and Uniqueness)**.** *Given $(n+1)$ distinct points $x_0, x_1, \dots, x_n$ and $(n+1)$ ordinates $y_0, y_1, \dots, y_n$, there exists a polynomial $p(x)$ of degree at most n such that $p(x_j) = y_j$ for every $j = 0, 1, \dots, n$. Such polynomial is unique among the set of all polynomials of degree at most n.*

Proof. We present three different proofs.

1. The linear system $V\boldsymbol{a} = \boldsymbol{y}$ has a unique solution if and only if $\mathcal{N}(V) = \{\mathbf{0}\}$. Suppose $V\boldsymbol{b} = \mathbf{0}$ for some $\boldsymbol{b}$. This means that the polynomial $p(x) = b_0 + b_1 x + \dots + b_n x^n$ has $(n+1)$ distinct zeros $x_0, x_1, \dots, x_n$. It follows from the **Fundamental Theorem of Algebra** that $p(x) \equiv \mathbf{0}$, which gives $\boldsymbol{b} = \mathbf{0}$.

2. We exploit the structure of the Vandermonde matrix (2.1.2). It can be shown that

$$\det(V) = \prod_{0 \leqslant j < i \leqslant n} (x_i - x_j).$$

   This shows that $\det(V) \neq 0$ since the interpolation nodes are assumed to be distinct. Thus V is nonsingular and there exists a unique solution to the linear system $V\boldsymbol{a} = \boldsymbol{y}$.

3. This last one is a constructive proof. It suffices to solve the following special case of the polynomial interpolation problem: for some fixed $0 \leqslant i \leqslant n$, let $y_j = \delta_{ij}$, where δ_{ij} is the Kronecker delta function. This leads to finding a polynomial of degree $\leqslant n$ having the form

$$p_i(x) = c(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n),$$

   where c is a constant to be determined. Solving for $p_i(x_i) = y_i = \delta_{ii} = 1$ yields

$$c = \Big[(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)\Big]^{-1}.$$

   This can be written in a more compact form

$$p_i(x) = l_i(x) = \prod_{j \neq i} \left(\frac{x - x_j}{x_i - x_j}\right), \quad l_i(x_j) = \delta_{ij}. \tag{2.1.3}$$

   We claim that the unique solution to the general problem is given by

$$p_n(x) = \sum_{i=0}^{n} y_i l_i(x). \tag{2.1.4}$$

It is clear from (2.1.3) that p_n satisfies (2.0.1) and $\deg(p_n) \leqslant n$. To prove uniqueness, suppose q is another polynomial of degree $\leqslant n$ satisfying (2.0.1). Then

$$p(x_j) - q(x_j) = 0 \quad \text{for every } j = 0, 1, \ldots, n.$$

Since $p - q$ is of degree $\leqslant n$ and $p - q$ has $(n + 1)$ distinct zeros, it follows from the **Fundamental Theorem of Algebra** that $p - q \equiv \mathbf{0}$.

■

Remark 2.1.2.

1. The set of functions $l_i(x)$ form the **Lagrange basis polynomials**. The formula (2.1.4) is called **Lagrange's formula** for the interpolating polynomial, where $l_i(x_j) = \delta_{ij}$. Given two pairs of data $(x_0, y_0), (x_1, y_1)$, $p_1(x)$ corresponds to **linear interpolation** having the form

$$p_1(x) = \left(\frac{x - x_1}{x_0 - x_1}\right) y_0 + \left(\frac{x - x_0}{x_1 - x_0}\right) y_1.$$

Given three pairs of data $(x_0, y_0), (x_1, y_1), (x_2, y_2)$, $p_2(x)$ corresponds to **quadratic interpolation** having the form

$$p_2(x) = \left[\frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}\right] y_0 + \left[\frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}\right] y_1 + \left[\frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}\right] y_2.$$

2. Uniqueness is a strong property in the theory of interpolation. Very often, we derive other formulas for the interpolation polynomial and they must coincide with the Lagrange's formula by uniqueness. Without uniqueness, the linear system $V\boldsymbol{a} = \boldsymbol{y}$ is not uniquely solvable; this means that there exists $\boldsymbol{y}^*$ such that there is no interpolating polynomial of degree at most n satisfying (2.0.1).

Example 2.1.3. We compute the linear and quadratic interpolation to $e^{0.826}$ using the following values

	x	e^x	
x_0	0.82	2.270500	y_0
x_1	0.83	2.293319	y_1
x_2	0.84	2.316367	y_2

Observe that $p_2(0.826)$ is unique in this case, but we have three different choices for $p_1(0.826)$. One can show that $p_2(0.826) \approx 2.2841639$. On the other hand, since $0.826 \in [0.82, 0.83]$, we compute $p_1(0.826)$ using the first two pairs of data; this gives $p_1(0.826) \approx 2.2841914$. Comparing them to the exact value $e^{0.826} \approx 2.2841638$, we deduce that $p_2(0.826)$ is a better choice of interpolation, which agrees with our intuition.

If a function $f(x)$ with sufficient regularity is given, we can approximate it using the Lagrange's formula (2.1.4)

$$p_n(x; f) = p_n(x) = \sum_{i=0}^{n} f(x_i) l_i(x),$$

where n is the degree of approximation. The following theorem is a standard result concerning the error of interpolation.

Theorem 2.1.4. *Let $x_0, x_1, \ldots, x_n$ be $(n+1)$ distinct real numbers and let f be a given real-valued function such that $f \in C^{(n+1)}(I_t)$, where I_t is the smallest interval containing $\{t, x_0, x_1, \ldots, x_n\}$ with t some given real number. There exists $\xi \in I_t$ such that*

$$f(t) - \sum_{i=0}^{n} f(x_i) l_i(t) = \frac{(t-x_0)(t-x_1)\ldots(t-x_n)}{(n+1)!} f^{(n+1)}(\xi). \tag{IEF}$$

Proof. Note that the interpolation error formula (IEF) is trivial if t is any node points, so suppose not. For all $x \in I_t$, define the following quantity

$$E(x) = f(x) - \sum_{i=0}^{n} f(x_i) l_i(x) = f(x) - p_n(x)$$
$$G(x) = E(x) - \left[\frac{\phi(x)}{\phi(t)}\right] E(t),$$

where $E(x)$ is the error function and $\phi(x) = \prod_{j=0}^{n}(x - x_j)$. Observe that

1. Since both $E(x)$ and $\phi(x)$ are $C^{(n+1)}$ functions on I_t, so is $G \in C^{(n+1)}(I_t)$.
2. G has $(n+2)$ distinct zeros $\{t, x_0, x_1, \ldots, x_n\}$ on I_t. Applying the **Mean Value Theorem** on the $(n+1)$ subintervals between these zeros shows that G' has at least $(n+1)$ distinct zeros on I_t. An induction argument shows that $G^{(j)}$ has at least $(n+2-j)$ distinct zeros on I_t, $0 \leqslant j \leqslant n+1$.

Let ξ be a zero of $G^{(n+1)}$ on I_t, *i.e.* $G^{(n+1)}(\xi) = 0$ for some $\xi \in I_t$. Computing $G^{(n+1)}$ yields

$$\begin{aligned} G^{(n+1)}(x) &= E^{(n+1)}(x) - \left[\frac{\phi^{(n+1)}(x)}{\phi(t)}\right] E(t) \\ &= f^{(n+1)}(x) - \left[\frac{(n+1)!}{\phi(t)}\right] E(t) \\ \implies 0 = G^{(n+1)}(\xi) &= f^{(n+1)}(\xi) - \left[\frac{(n+1)!}{\phi(t)}\right] E(t). \end{aligned}$$

The desired result follows from rearranging the equation above in terms of $E(t)$. ∎

Example 2.1.5. Consider $f(x) = e^x$ on $[0, 1]$. Choosing the linear interpolation $p_1(x)$ yields the interpolation error

$$e^x - p_1(x) = \frac{(x-x_0)(x-x_1)}{2} e^{\xi},$$

where $\xi \in \Big(\min\{x, x_0, x_1\}, \max\{x, x_0, x_1\}\Big)$. Assume that $x \in (x_0, x_1)$, then

$$e^x - p_1(x) = \frac{-(x - x_0)(x_1 - x)}{2} e^{\xi}.$$

Let $h = x_1 - x_0$ and consider $g(x) = (x - x_0)(x_1 - x)$. Computing g' gives

$$g'(x) = x_1 - x - (x - x_0) = -2x + x_0 + x_1 = 0 \iff x = \frac{x_0 + x_1}{2}.$$

Thus, g has a maximum at $x = (x_0 + x_1)/2$, with value

$$g\left(\frac{x_0 + x_1}{2}\right) = \frac{h^2}{8} \implies |e^x - p_1(x)| \leqslant \left(\frac{h^2}{8}\right) \max_{\xi \in [0,1]} e^{\xi} \leqslant \frac{h^2 e}{8}.$$

2.2 Newton's Divided Difference Interpolation Formula

2.2.1 Divided Difference

We first introducing a discrete version of the derivative of $f(x)$. Let x_0, x_1 be two distinct numbers. Define the **first order divided difference** of $f(x)$ as follows

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}. \tag{2.2.1}$$

Observe that if $f(x)$ is differentiable on an interval containing x_0 and x_1, then it follows from the mean value theorem that $f[x_0, x_1] = f'(c)$ for some $c \in (x_0, x_1)$. If x_0 and x_1 are sufficiently close, then

$$f[x_0, x_1] = f'\left(\frac{x_0 + x_1}{2}\right).$$

A nice feature about **higher order divided differences** is that one could obtain a recursive formula using lower order divided differences. Let x_0, x_1, x_2 be distinct real numbers. Define the **second order divided difference** of $f(x)$ as

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \tag{2.2.2}$$

Let x_0, x_1, x_2, x_3 be distinct real numbers. Define the **third order divided difference** of $f(x)$ as

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0} \tag{2.2.3}$$

In general, for distinct real numbers $x_0, x_1, \ldots, x_n$, define the $(\boldsymbol{n+1})$**th order divided difference** of $f(x)$ as

$$f[x_0, x_1, \ldots, x_n] = \frac{f[x_1, \ldots, x_n] - f[x_0, \ldots, x_{n-1}]}{x_n - x_0} \tag{2.2.4}$$

Theorem 2.2.1 (Mean Value Theorem for Divided Differences)**.** *Let $n \geqslant 1$ and assume that $f \in C^n[\alpha, \beta]$. For any $(n+1)$ distinct numbers $x_0, x_1, \ldots, x_n$ in $[\alpha, \beta]$, we have the following equality*

$$f[x_0, x_1, \ldots, x_n] = \frac{f^{(n)}(c)}{n!}.$$

for some $c \in \Big(\min\{x_0, x_1, \ldots, x_n\}, \max\{x_0, x_1, \ldots, x_n\} \Big)$.

- It relates the divided difference to the classical derivative for some unknown point.

Lemma 2.2.2. *We have the following properties of divided differences:*

(a) The nth divided difference is permutation invariant. More precisely, for any permutation $(i_0, i_1, \ldots, i_n)$ of $(0, 1, \ldots, n)$ we have that

$$f[x_{i_0}, x_{i_1}, \ldots, x_{i_n}] = f[x_0, x_1, \ldots, x_n].$$

(b) One can relaxed the definition of divided difference by removing the requirement that interpolation nodes are distinct. Define

$$f[x_0, x_0, \ldots, x_0] \coloneqq \frac{f^{(n)}(x_0)}{n!}.$$

For the first order divided difference of $f(x)$ on a single node point x_0,

$$f[x_0, x_0] \coloneqq \lim_{x_1 \to x_0} f[x_0, x_1] = \lim_{x_1 \to x_0} \left(\frac{f(x_1) - f(x_0)}{x_1 - x_0} \right).$$

2.2.2 Newton's Divided Difference

In this subsection, we employ the concept of divided difference from Subsection 2.2.1 to define another convenient form of the interpolation polynomial. Given $(n+1)$ distinct interpolation nodes, let $p_n(x)$ be the polynomial interpolating $f(x_j)$ at x_j for every $j = 0, \ldots, n$, where $\deg(p_n) \leqslant n$, *i.e.*

$$p_n(x_j) = f(x_j) \quad \text{for all } j = 0, \ldots, n. \tag{2.2.5}$$

We establish such interpolation polynomials $p_n(x)$ as a recursive relation involving divided difference.

Theorem 2.2.3. *Consider the* ***Newton's divided difference interpolation polynomial*** *p_n defined by*

$$\begin{aligned} p_1(x) &= f(x_0) + (x - x_0) f[x_0, x_1] \\ p_2(x) &= \underbrace{f(x_0) + (x - x_0) f[x_0, x_1]}_{p_1(x)} + (x - x_0)(x - x_1) f[x_0, x_1, x_2] \end{aligned}$$

$$\begin{array}{ccccc} \vdots = & \vdots & \vdots & \vdots & \\ p_n(x) = p_{n-1}(x) + \underbrace{(x-x_0)(x-x_1)\dots(x-x_{n-1})f[x_0,x_1,\dots,x_n]}_{\text{correction term}}, & n \geqslant 1. & & & (2.2.6) \end{array}$$

Then p_n solves the interpolation problem (2.2.5).

Proof. It is clear that for $n \geqslant 1$, $p_n(x)$ is a polynomial of degree at most n by construction. From Theorem 2.1.1, it suffices to show that these polynomials satisfy (2.2.5). Moreover, it suffices to show that $p_j(x_j) = f(x_j)$ for every $j = 0, 1, \dots, n$ by the recursive relation. The case $n = 1$ is clear and here we only prove it for the case $n = 2$:

$$\begin{aligned} p_2(x_2) &= f(x_0) + (x_2 - x_0)f[x_0, x_1] + (x_2 - x_0)(x_2 - x_1)f[x_0, x_1, x_2] \\ &= f(x_0) + (x_2 - x_0)f[x_0, x_1] + (x_2 - x_1)\Big[f[x_1, x_2] - f[x_0, x_1]\Big] \\ &= f(x_0) + (x_1 - x_0)f[x_0, x_1] + (x_2 - x_1)f[x_1, x_2] \\ &= f(x_0) + \Big[f(x_1) - f(x_0)\Big] + \Big[f(x_2) - f(x_1)\Big] = f(x_2). \end{aligned}$$

■

Remark 2.2.4. By Theorem 2.1.1, the Newton's divided difference formula must coincide with the Lagrange's formula (2.1.4). The former has the advantage over computation, in that it can be defined recursively in terms of lower order interpolation polynomials plus some correction terms. Moreover, the coefficients of the polynomial are computed using divided differences.

To evaluate the Newton's interpolation polynomial in an efficient manner, recall the recursive relation (2.2.6)

$$p_n(x) = D_0 + (x - x_0)D_1 + (x - x_0)(x - x_1)D_2 + \dots + (x - x_0)\dots(x - x_{n-1})D_n, \qquad (2.2.7)$$

where $D_j = f[x_0, \dots, x_{j-1}], j = 0, 1, \dots, n$. We can rewrite this in a nested formula

$$p_n(x) = D_0 + (x - x_0)\Big\{D_1 + (x - x_1)\Big[D_2 + \dots + (x - x_{n-2})[D_{n-1} + (x - x_{n-1})D_n]\dots\Big]\Big\} \qquad (2.2.8)$$

For example,

$$p_3(x) = D_0 + (x - x_0)\Big[D_1 + (x - x_1)[D_2 + (x - x_2)D_3]\Big].$$

The nested formula (2.2.8) has the numerical advantage that it only involves n multiplications, while (2.2.7) involves n^2 multiplications. We are now ready to prove the **Mean Value Theorem for Divided Differences**.

Proof of Theorem 2.2.1. Let t be a real number, distinct from the interpolating nodes $x_0, x_1, \dots, x_n$. Recall the Newton's divided difference formula (2.2.6)

$$p_{n+1}(x) = p_n(x) + (x - x_0)(x - x_1)\dots(x - x_n)f[x_0, x_1, \dots, x_n, t].$$

Using $p_{n+1}(t) = f(t)$, we obtain the error formula

$$f(t) - p_n(t) = (t - x_0)(t - x_1)\dots(t - t_n)f[x_0, x_1, \dots, x_n, t]. \qquad (2.2.9)$$

Comparing with the error formula in Theorem 2.1.4, we see that

$$f[x_0, x_1, \ldots, x_n, t] = \frac{f^{(n+1)}(\xi)}{(n+1)!} \quad \text{for some } \xi \in I_t. \tag{2.2.10}$$

■

Remark 2.2.5. Setting $n = m - 1, t = x_{n+1}$, (2.2.10) is equivalent to

$$f[x_0, x_1, \ldots, x_m] = \frac{f^{(m)}(\xi)}{m!}.$$

It turns out that convergence is related to the distribution of nodes.

Example 2.2.6. Suppose $f(x) = \sin(x)$ is approximated by $p_9(x)$ that interpolates f at 10 points in $[0, 1]$. Using the error formula from Theorem 2.1.4,

$$|f(x) - p_9(x)| \leqslant \frac{1}{10!} \left(\prod_{i=0}^{9} (x - x_i) \right) |f^{(10)}(\xi)| \leqslant \frac{1}{10!}.$$

2.3 Further Discussion on Interpolation Error

Assume that $f \in C^{(n+1)}(I_x)$, where I_x is the smallest interval containing the points $x_0, x_1, \ldots, x_n, x$. Define the following quantity

$$C_{n+1} = \max_{t \in I_x} |f^{(n+1)}(t)|.$$

From the interpolation error formula (IEF), one has to estimate the polynomial

$$\phi_n(x) = \prod_{i=0}^{n} (x - x_i) = (x - x_0)(x - x_1) \ldots (x - x_n),$$

independent of the interpolation nodes. A direct estimate on the error yields

$$\max_{x \in I_x} |f(x) - p_n(x)| \leqslant \frac{C_{n+1}}{(n+1)!} \max_{x \in I_x} |\phi_n(x)|.$$

1. **Consider $n = 1$.** Then $\phi_1(x) = (x - x_0)(x - x_1)$ and setting $x_1 - x_0 = h$, one can show that
$$\max_{x_0 \leqslant x \leqslant x_1} |\phi_1(x)| = \frac{h^2}{4} \implies \max_{x_0 \leqslant x \leqslant x_1} |f(x) - p_1(x)| \leqslant \frac{C_2 h^2}{8}.$$
This shows that the error is bounded in terms of distance between two nodes.

2. **Consider $n \geqslant 2$.** Assume we have uniform/equidistant nodes $\{x_0, x_1, \ldots, x_n\}$, where $x_j = x_0 + jh, j = 0, 1, \ldots, n$.

(a) **For the case $n = 2$**, one can show that

$$\max_{x_0 \leqslant x \leqslant x_2} |\phi_2(x)| = \frac{2\sqrt{3}}{9}h^3 \implies \max_{x_0 \leqslant x \leqslant x_2} |f(x) - p_2(x)| \leqslant \frac{C_2\sqrt{3}h^2}{27}.$$

This shows that the error is bounded in terms of (uniform) distance between nodes. More importantly, this implies that the distance between x and x_1 does not affected the estimate, although it will make a difference for higher degree interpolation.

(b) **For the case $n = 3$**, one can show that

$$\max_{x_0 \leqslant x \leqslant x_3} |\phi_3(x)| = h^4.$$

However, choosing $x \in (x_1, x_2)$ yields a different bound

$$\max_{x_1 \leqslant x \leqslant x_2} |\phi_3(x)| = \frac{9}{16}h^4.$$

(c) **For the case $n = 6$**,

$$\max_{x_0 \leqslant x \leqslant x_6} |\phi_6(x)| = 96h^7, \quad \max_{x_2 \leqslant x \leqslant x_4} |\phi_6(x)| = 12h^7.$$

We deduce that under the equidistant interpolating nodes assumption, the nodes should be chosen such that the point of interest x is as close as possible to the midpoint of $[x_0, x_n]$ in order to minimise the interpolation error.

The above discussion leads to a natural question: does the interpolating polynomial converges to the true function as $n \longrightarrow \infty$ if the interpolating nodes are uniformly spaced between them? More precisely, consider approximating a given function f on a given interval $[a, b]$ using interpolating polynomials. Construct an uniformly spaced subdivision of $[a, b]$, *i.e.* for each $n \geqslant 1$, define

$$h = \frac{b-a}{n}, \quad x_j = a + jh, \quad j = 0, 1, \ldots, n.$$

and let p_n be an polynomial interpolating $f(x)$ at the interpolation nodes $x_0, x_1, \ldots, x_n$. Does

$$\max_{x \in [a,b]} |f(x) - p_n(x)| \longrightarrow 0 \quad \text{as } n \longrightarrow \infty?$$

Surprisingly, the answer is no; there exists functions that are well-behaved but convergence does not occur.

Example 2.3.1. Consider the Runge function $f(x) = \dfrac{1}{1+x^2}$ on $[-5, 5]$. It can be shown that for any $3.64 < |x| < 5$,

$$\sup_{n \in \mathbb{N}} |f(x) - p_n(x)| = \infty.$$

Viewing f as a function in the complex plane $\mathbb{C}$, we see that f has simple poles at $x = \pm i$.

2.4 Chebyshev Polynomials of First Kind

Consider the interpolation error over $x \in [-1, 1]$. We are interested on the following quantity

$$\min_{x_0,\dots,x_n} \max_{x\in[-1,1]} \prod_{j=0}^{n} |(x - x_j)|.$$

The **Chebyshev polynomials of the first kind** are defined by the following recurrence relation: Let $T_0(x) = 1, T_1(x) = x$ and

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n \geqslant 1 \tag{2.4.1}$$

Observe that T_n is a polynomial of degree n for each $n \geqslant 0$ and $T_n(x) = 2^{n-1}x^n + \dots$. For example,

$$\begin{aligned} T_2(x) &= 2x^2 - 1 \\ T_3(x) &= 4x^3 - 3x \\ T_4(x) &= 8x^4 - 8x^2 + 1. \end{aligned}$$

Lemma 2.4.1. *For any $x \in [-1, 1]$, the Chebyshev polynomials of the first kind have the following closed form expression*

$$T_n(x) = \cos\big(n \cos^{-1}(x)\big), \quad n \geqslant 0. \tag{2.4.2}$$

Proof. First, the expression (2.4.2) is well-defined under the assumption that $x \in [-1, 1]$. The case $n = 0, 1$ is trivial. For $n \geqslant 2$, we need to show that (2.4.2) coincides with the recursive relation (2.4.1). Using the double-angle formula,

$$\begin{aligned} &\cos((n \pm 1)\theta) = \cos\theta\cos(n\theta) \mp \sin\theta\sin(n\theta) \\ \implies &\cos((n+1)\theta) = 2\cos\theta\cos(n\theta) - \cos((n-1)\theta) \end{aligned}$$

The desired result follows by setting $\theta = \cos^{-1}(x)$.

■

Remark 2.4.2. It follows that $|T_n(x)| \leqslant 1$ and $T_n(\cos x) = \cos(nx)$ for all $x \in [-1, 1]$. Moreover,

$$T_n\left(\cos\left(\frac{j\pi}{n}\right)\right) = (-1)^j, \quad 0 \leqslant j \leqslant n \tag{2.4.3}$$

$$T_n\left(\cos\left(\frac{(2j+1)\pi}{2n}\right)\right) = 0, \qquad 0 \leqslant j \leqslant n-1. \tag{2.4.4}$$

Theorem 2.4.3. *If p is a monic polynomial of degree n, then*

$$\|p\|_\infty = \max_{x\in[-1,1]} |p(x)| \geqslant 2^{1-n}.$$

Proof. We argue by contradiction. Suppose the given estimate fails to hold. This means that

$$|p(x)| < 2^{1-n} \quad \text{for all } x \in [-1,1].$$

Normalising the highest term in Chebyshev polynomials of the first kind, define a monic polynomial $q = 2^{1-n}T_n$ which has degree at most n. Let $x_j = \cos\left(\frac{j\pi}{n}\right), 0 \leqslant j \leqslant n$. It follows from Remark 2.4.2 that

$$|p(x_j)| < 2^{1-n} = (-1)^j q(x_j). \tag{2.4.5}$$

On the other hand, we have the trivial inequality

$$(-1)^j p(x_j) \leqslant |p(x_j)|. \tag{2.4.6}$$

Combining (2.4.5) and (2.4.6) gives

$$(-1)^j [q(x_j) - p(x_j)] > 0, \quad 0 \leqslant j \leqslant n. \tag{2.4.7}$$

Now, (2.4.7) shows that the function $q - p$ oscillates in sign $(n+1)$ times on $[-1,1]$, which implies that $q-p$ has n distinct roots on $(-1,1)$. This contradicts the **Fundamental Theorem of Algebra**, since $q - p$ is a polynomial of degree $\leqslant n-1$ due to the assumption that both q and p are monic polynomials.

■

Theorem 2.4.4 (Interpolation error for Chebyshev nodes)**.** *Given a real-valued function $f \in C^{(n+1)}(I_x)$, suppose the interpolating nodes $\{x_j\}_{j=0}^n$ are chosen to be zeros of the Chebyshev polynomial of the first kind T_{n+1} given by*

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

and I_x is the smallest interval containing $\{x, x_0, x_1, \ldots, x_n\}$. The interpolation error formula satisfies

$$|f(x) - p_n(x)| \leqslant \frac{1}{2^n(n+1)!} \max_{\xi \in [-1,1]} |f^{(n+1)}(\xi)|.$$

Proof. First, a direct estimate on the interpolation error over the interval $x \in [-1,1]$ gives

$$|f(x) - p_n(x)| \leqslant \frac{1}{(n+1)!} \max_{\xi \in [-1,1]} |f^{(n+1)}(\xi)| \max_{x \in [-1,1]} \left| \prod_{j=0}^n x - x_j \right|.$$

Theorem 2.4.3 gives

$$\max_{x \in [-1,1]} \left| \prod_{j=0}^n x - x_j \right| \geqslant 2^{1-(n+1)} = 2^{-n}.$$

One can show that this bound is actually attained if one chooses the interpolation nodes x_j to be zeros of Chebyshev polynomials of the first kind.

■

Remark 2.4.5. Suppose f is now a $C^{(n+1)}$ function on $[a, b]$. One can obtain a function g on $[-1, 1]$ equivalent to f using an affine transformation, given by

$$g(x) = f\left(\frac{(b+a)+x(b-a)}{2}\right), \quad -1 \leqslant x \leqslant 1.$$

This can be found by guessing an ansatz of the form $y = Cx + D$ and solving for C, D given that $y(-1) = a, y(1) = b$. The "transformed" Chebyshev nodes has the form

$$\hat{x}_j = \frac{a+b}{2} + \left(\frac{b-a}{2}\right)\cos\left(\frac{(2j+1)\pi}{2n+2}\right), \quad 0 \leqslant j \leqslant n,$$

Since $g^{(n)}(x) = \left(\frac{b-a}{2}\right)^n f^{(n)}(x)$, it follows from Theorem 2.4.4 that the interpolation error of a function $f \in C^{(n+1)}[a, b]$ satisfies

$$|f(x) - p_n(x)| \leqslant \frac{1}{2^n(n+1)!}\left(\frac{b-a}{2}\right)^n \max_{\xi\in[-1,1]} |f^{(n+1)}(\xi)|.$$

2.5 Approximation Theory

The choice of using polynomials in approximating continuous functions is justified by the following two theorems:

1. **Weierstrass approximation theorem**, which states that every continuous function on a closed interval can be uniformly approximated by a polynomial function to any desired accuracy. If f is only bounded, then we only have pointwise convergence. Note that a constructive proof is given using the **Bernstein polynomials**

$$p_n(x) = \sum_{k=0}^{n} \binom{n}{k} f\left(\frac{k}{n}\right) x^k(1-x)^{n-k}, \quad 0 \leqslant x \leqslant 1.$$

2. **Taylor's theorem**, which states that for any $f \in C^{(n+1)}[a, b]$,

$$f(x) = p_n(x) + R_{n+1}(x),$$

where

$$\begin{cases} p_n(x) & = f(x_0) + (x-x_0)f'(x_0) + \ldots + \dfrac{f^{(n)}(x_0)}{n!}(x-x_0), \\ R_{n+1}(x) & = \dfrac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)^{n+1}, \end{cases}$$

for some ξ between x and x_0.

2.5.1 The Minimax Approximation Problem

Let f be a continuous function on $[a, b]$. Since there are many polynomial approximations $p(x)$ to $f(x)$, it is natural to ask what is the best possible accuracy that can be obtained using polynomials of degree at most $n \geqslant 0$. This raises another question: how do we qualitatively describe accuracy of polynomial approximations? One can measure the difference using the L^2 norm or L^∞ norm. The latter leads to the **minimax error**:

$$\rho_n(f) = \inf_{q \in \mathbb{R}[x]:\, \deg(q) \leqslant n} \|f - q\|_\infty \tag{2.5.1}$$

This is again an existence and uniqueness problem. If such polynomial approximation $q^*(x)$ exists, *i.e.*

$$\rho_n(f) = \|f - q^*\|_\infty,$$

then $q^*(x)$ is called the **minimax approximation** to $f(x)$ on $[a, b]$.

Example 2.5.1. We wish to compute the minimax polynomial approximation $q_1^*(x)$ to the function e^x on the interval $[-1, 1]$. Let $q_1^*(x) = a_0 + a_1 x$, we present a geometrical argument here. Let $\epsilon(x) = e^x - [a_0 + a_1 x]$ and $\rho_1 = \max_{x \in [-1,1]} |\epsilon(x)|$. Observe that

1. q_1^* and e^x must be equal at 2 points in $[-1, 1]$, say at $-1 < x_1 < x_2 < 1$; otherwise the approximation can be improved by moving the graph of q_1^* appropriately.
2. The maximum error ρ_1 is attained at exactly three points, namely
$$\epsilon(-1) = \epsilon(x_3) = \epsilon(1) = \rho_1, \quad \text{where } x_3 \in (x_1, x_2).$$
3. Since $\epsilon(x)$ has a relative minimum at x_3, we have $\epsilon'(x_3) = 0$.

This yields four equations

$$\begin{cases} e^{-1} - [a_0 - a_1] & = \rho_1 \\ e - [a_0 + a_1] & = \rho_1 \\ e^{x_3} - [a_0 + a_1 x_3] & = \rho_1 \\ e^{x_3} - a_1 & = 0, \end{cases}$$

which has solution

$$\begin{aligned} a_1 &= \frac{e - e^{-1}}{2} && \approx 1.1752 \\ x_3 &= \ln(a_1) && \approx 0.1614 \\ \rho_1 &= \frac{e^{-1}}{2} + \frac{x_3}{4}(e - e^{-1}) && \approx 0.2788 \\ a_0 &= \rho_1 + (1 - x_3)a_1 && \approx 1.2643. \end{aligned}$$

Hence, $q_1 * (x) = 1.2643 + 1.1752x$ and $\rho_1 \approx 0.2788$. We note that the error resulting from Taylor's approximation is approximately 0.718.

Remark 2.5.2. In general, one uses Remes algorithm to construct minimax approximation.

Theorem 2.5.3 (Chebyshev Equioscillation Theorem)**.** *Given any $f \in C[a,b]$, for any $n \geqslant 0$ there exists a unique polynomial $q_n^*(x)$ of degree at most n for which*

$$\rho_n(f) = \|f - q_n^*\|_\infty.$$

Such polynomial is uniquely characterised by the following property: there are at least $(n+2)$ points satisfying $a \leqslant x_0 < x_1 < \ldots < x_n < x_{n+1} \leqslant b$ for which

$$f(x_j) - q_n^*(x_j) = \sigma(f,n)(-1)^j \rho_n(f), \quad j = 0, 1, \ldots, n+1,$$

where $\sigma(f,n) = \pm 1$ depending on the function f and n.

2.5.2 The Least Squares Approximation Problem

Due to the difficulty in computing the minimax polynomial approximation, one usually perform an intermediate approximation called the **Least Squares Approximation**. Essentially, one replaces the L^∞ norm in (2.5.1) with the L^2 norm

$$\|g\|_2^2 = \int_a^b |g(x)|^2 \, dx, \quad g \in C[a,b].$$

This leads to define

$$M_n(f) = \inf_{r \in \mathbb{R}[x]:\, \deg(r) \leqslant n} \|f - r\|_2. \tag{2.5.2}$$

Example 2.5.4. Consider $f(x) = e^x$ on $[-1,1]$ and define $r_1(x) = b_0 + b_1 x$. We wish to minimise

$$\|f - r_1\|_2^2 = \int_{-1}^1 [e^x - b_0 - b_1 x]^2 \, dx := F(b_0, b_1),$$

where $F(b_0, b_1)$ is a quadratic polynomial in b_0, b_1. To find a minimum, we set $\dfrac{\partial F}{\partial b_0} = \dfrac{\partial F}{\partial b_1} = 0$. Then

$$\begin{aligned}
\int_{-1}^1 \frac{\partial}{\partial b_0}\Big(e^x - b_0 - b_1 x\Big)^2 \, dx &= -2 \int_{-1}^1 \Big(e^x - b_0 - b_1 x\Big) \, dx \qquad = 0 \\
\int_{-1}^1 \frac{\partial}{\partial b_1}\Big(e^x - b_0 - b_1 x\Big)^2 \, dx &= 2 \int_{-1}^1 \Big(e^x - b_0 - b_1 x\Big)(-x) \, dx = 0.
\end{aligned}$$

Solving for b_0 and b_1 yields

$$\begin{aligned}
b_0 &= \frac{1}{2} \int_{-1}^1 e^x \, dx \quad = \sinh(1) \approx 1.1752 \\
b_1 &= \frac{3}{2} \int_{-1}^1 x e^x \, dx = 3e^{-1} \qquad \approx 1.1036.
\end{aligned}$$

Hence, $r_1^*(x) = 1.1752 + 1.1036x$ and it can be shown that $\|e^x - r_1^*\|_\infty \approx 0.44$.

It is often useful to consider approximation of the form

$$r_n^*(x) = \sum_{j=0}^{n} (f, \phi_j)_w \phi_j,$$

where $(\cdot,\cdot)_w$ is now a weighted inner product defined by

$$(f, g)_w = \int_a^b w(x) f(x) g(x)\, dx$$

for some nonnegative weight function $w(x)$ on (a, b) and (ϕ_j) is an orthonormal (with respect to $(\cdot,\cdot)_w$) set of polynomials. Some examples:

$$w(x) = 1, \qquad P_n(x) = \frac{(-1)^n}{2^n n!}\frac{d^n}{dx^n}[(1-x^2)^n], \quad n \geqslant 1, \quad x \in [-1, 1]. \qquad \text{(Legendre)}$$

$$w(x) = \frac{1}{\sqrt{1-x^2}}, \quad T_n(x) = \cos\big(n \cos^{-1}(x)\big), \qquad n \geqslant 0, \quad x \in [-1, 1]. \qquad \text{(Chebyshev)}$$

$$w(x) = e^{-x}, \qquad L_n(x) = \frac{1}{n! e^{-x}}\frac{d^n}{dx^n}(x^n e^{-x}), \qquad n \geqslant 0, \quad x \in [0, \infty) \qquad \text{(Laguerre)}$$

2.6 Problems

1. Given the interpolation data (points) (0,2), (0.5,5), (1,4),

   (a) Find the function $f(x) = a_0 + a_1 \cos(\pi x) + a_2 \sin(\pi x)$, which interpolates the given data;

   Solution: Substituting the interpolation data into the function $f(x)$ yields the following system of linear equations

   $$2 = a_0 + a_1 \cos(0) \quad + a_2 \sin(0) \quad \Longrightarrow \; 2 = a_0 + a_1 \qquad (2.6.1)$$

   $$5 = a_0 + a_1 \cos\left(\frac{\pi}{2}\right) + a_2 \sin\left(\frac{\pi}{2}\right) \Longrightarrow \; 5 = a_0 + a_2 \qquad (2.6.2)$$

   $$4 = a_0 + a_1 \cos(\pi) \quad + a_2 \sin(\pi) \quad \Longrightarrow \; 4 = a_0 - a_1. \qquad (2.6.3)$$

   Adding (2.6.1) and (2.6.3) yields $6 = 2a_0 \implies a_0 = 3$. We then solve for a_1, a_2

   $$a_1 = 2 - a_0 = 2 - 3 = -1 \quad \text{and} \quad a_2 = 5 - a_0 = 5 - 3 = 2.$$

   Hence, the function $f(x)$ that interpolates the given data is

   $$f(x) = 3 - \cos(\pi x) + 2 \sin(\pi x).$$

   (b) Find the quadratic polynomial interpolating the data.

   Solution: Denote the given interpolation points as

   $$(x_0, y_0) = (0, 2), \quad (x_1, y_1) = (0.5, 5), \quad (x_2, y_2) = (1, 4).$$

We know that the quadratic interpolating polynomial $p_2(x)$ has the form

$$p_2(x) = \sum_{i=0}^{2} l_i(x) y_i, \text{ where } l_i(x) = \prod_{j \neq i}^{2} \left(\frac{x - x_j}{x_i - x_j} \right).$$

We now compute all the $l_i(x)$'s:

$$\begin{aligned}
l_0(x) &= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{(x - 0.5)(x - 1)}{(0 - 0.5)(0 - 1)} &&= (2x - 1)(x - 1) \\
l_1(x) &= \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = \frac{(x - 0)(x - 1)}{(0.5 - 0)(0.5 - 1)} &&= -4x(x - 1) \\
l_2(x) &= \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{(x - 0)(x - 0.5)}{(1 - 0)(1 - 0.5)} &&= x(2x - 1).
\end{aligned}$$

Hence,

$$\begin{aligned}
p_2(x) = \sum_{i=0}^{2} l_i(x) y_i &= (2x - 1)(x - 1)(2) - 4x(x - 1)(5) + x(2x - 1)(4) \\
&= (2x - 1)[2x - 2 + 4x] - 20x(x - 1) \\
&= (2x - 1)(6x - 2) - 20x(x - 1) \\
&= 12x^2 - 10x + 2 - 20x^2 + 20x \\
&= -8x^2 + 10x + 2.
\end{aligned}$$

2. Bound the error in terms of $h > 0$ of the quadratic interpolation to $f(x) = e^x$ on $[0, 1]$ with evenly spaced interpolation points $x_0, x_1 = x_0 + h, x_2 = x_0 + 2h$. Assume that $x_0 < x < x_2$.

Solution: Suppose $x_0 < x < x_2$. First, a direct estimate yields

$$\begin{aligned}
|e^x - p_2(x)| &= \left| \frac{(x - x_0)(x - x_1)(x - x_2)}{3!} e^{\xi} \right| \quad \text{for some } \xi \in [x_0, x_2] \\
&\leqslant \left(\frac{e^{x_2}}{6} \right) \max_{x \in [x_0, x_2]} |(x - x_0)(x - x_1)(x - x_2)| \\
&\leqslant \left(\frac{e}{6} \right) \max_{x \in [x_0, x_2]} |g(x)|.
\end{aligned}$$

Since the interpolation points are evenly spaced, to find the extrema of $g(x)$ over $[x_0, x_2]$, it is equivalent (by shifting along the x-axis) to find the extrema of the function

$$G(x) = (x + h)x(x - h) = x^3 - h^2 x \quad \text{over } [-h, h].$$

Setting $G'(x^*) = 0$ and solving for x^* yields

$$0 = G'(x^*) = 3(x^*)^2 - h^2 \implies (x^*)^2 = \frac{h^2}{3} \implies x^* = \pm \frac{h}{\sqrt{3}}.$$

Since $G(x)$ is a cubic function with three distinct zeros, these critical points x^*'s must correspond to exactly one local minimum and one local maximum. Because $G(x)$ is an odd function, $\left|G\left(\frac{h}{\sqrt{3}}\right)\right| = \left|G\left(-\frac{h}{\sqrt{3}}\right)\right|$. Thus,

$$\begin{aligned}\max_{x\in[x_0,x_2]} |g(x)| = \max_{x\in[-h,h]} |G(x)| &= \left|G\left(-\frac{h}{\sqrt{3}}\right)\right| \\ &= \left(-\frac{h}{\sqrt{3}}+h\right)\left(-\frac{h}{\sqrt{3}}\right)\left(-\frac{h}{\sqrt{3}}-h\right) \\ &= h^3\left(-\frac{1}{\sqrt{3}}+1\right)\left(-\frac{1}{\sqrt{3}}\right)\left(-\frac{1}{\sqrt{3}}-1\right) \\ &= \frac{h^3}{\sqrt{3}}\left(\frac{2}{3}\right) = \frac{2h^3}{3\sqrt{3}}.\end{aligned}$$

Hence, the error of the quadratic interpolation to $f(x) = e^x$ on $[0,1]$ with evenly spaced interpolation points $x_0, x_1, x_2, x_0 < x < x_2$, satisfies the bound

$$|e^x - p_2(x)| \leqslant \left(\frac{e}{6}\right)\left(\frac{2h^3}{3\sqrt{3}}\right) = \frac{eh^3}{9\sqrt{3}} = \frac{\sqrt{3}eh^3}{27}.$$

3. (a) Suppose you are given symmetric data

$$(x_i, y_i), \quad i = -n, -n+1, \ldots, n-1, n,$$

such that

$$x_{-i} = -x_i \quad \text{and} \quad y_{-i} = -y_i, \quad i = 0, 1, \ldots, n. \tag{2.6.4}$$

What is the required degree of the interpolating polynomial p, where x_i's are distinct nodes? Show that the interpolating polynomial is odd, *i.e.* $p(x) = -p(-x)$ for all real numbers x.

Solution: Since we were given $(2n+1)$ distinct points, it follows from Theorem 2.1.1 the unique interpolating polynomial $p(x)$ is of degree at most $2n$. We exploit the uniqueness of $p(x)$ to show that $p(x)$ is an odd function. Consider the polynomial $q(x) := -p(-x)$. Then

$$\begin{aligned} q(x_j) &= -p(-x_j) &&= -p(x_{-j}) = -y_{-j} = y_j \\ q(x_{-j}) &= -p(-x_{-j}) = -p(x_j) &&= -y_j = y_{-j}. \end{aligned}$$

This implies that $q(x)$ also interpolates the given data and it follows from the uniqueness of interpolating polynomial that

$$p(x) \equiv q(x) = -p(-x).$$

Finally, since $p(x)$ is an odd function, it must be the case that $p(x)$ only contains terms with odd degree. Consequently, the required degree of the interpolating polynomial $p(x)$ is $2n-1$.

Remark: We could deduce the same result geometrically. We first find the unique polynomial $p_+(x)$ interpolating $(x_i, y_i), i = 0, 1, \ldots, n$, which is of degree n. Since the given data is symmetric, the unique polynomial interpolating $(x_{-i}, y_{-i}), i = 0, 1, \ldots, n$ is simply the odd extension of $p_+(x)$. Counting the number of critical points leads us to deduce that the required degree of the interpolating polynomial is $2n - 1$.

(b) Let $l_i(x)$ be the Lagrange basis functions with distinct nodes $x_0, x_1, \ldots, x_n$ with $n = 2017$. Prove that

$$\sum_{i=0}^{2017} l_i(x) = 1.$$

for all x.

Solution: We exploit the uniqueness property of the interpolating polynomial. Consider $(n + 1)$ distinct nodes $x_0, x_1, \ldots, x_n$ with corresponding values $y_i = 1$ for every $i = 0, 1, \ldots, n$, where $n = 2017$. The interpolating polynomial in **Lagrange form** is given by

$$p_n(x) = p_{2017}(x) = \sum_{i=0}^{2017} l_i(x) y_i = \sum_{i=0}^{2017} l_i(x).$$

At the same time, the constant function $p(x) \equiv 1$, which is a polynomial of degree 0, also interpolates the data points $(x_i, f(x_i)), i = 0, 1, \ldots, 2017$. We conclude by uniqueness that $1 = \sum_{i=0}^{2017} l_i(x)$ for all x.

Remark: The result holds for any arbitrary but finite $n \geqslant 1$.

4. (a) Consider finding a rational function $p(x) = \dfrac{a + bx}{1 + dx}$ that satisfies

$$p(x_i) = y_i, \quad i = 1, 2, 3,$$

with distinct x_1, x_2, x_3. Does such a function $p(x)$ exists, or are additional conditions needed to ensure existence and uniqueness of $p(x)$?

Solution: Given data points $(x_i, y_i), i = 1, 2, 3$ with distinct x_1, x_2, x_3, consider a rational function p of the form $p(x) = \dfrac{a + bx}{1 + dx}$. Suppose we impose the condition that $p(x_i) = y_i$ for every $i = 1, 2, 3$. First, expanding $p(x_i) = y_i$ gives

$$\frac{a + bx_i}{1 + dx_i} = y_i \implies a + bx_i = y_i(1 + dx_i) \implies a + (x_i)b - (x_i y_i)d = y_i. \quad (2.6.5)$$

where we require $1 + dx_i \neq 0, i = 1, 2, 3$. Since (2.6.5) is true for each $i = 1, 2, 3$, we obtain three linear equations with unknowns a, b, d. In matrix form, this

system of linear equations can be wrritten as

$$M\mathbf{z} = \begin{bmatrix} 1 & x_1 & -x_1y_1 \\ 1 & x_2 & -x_2y_2 \\ 1 & x_3 & -x_3y_3 \end{bmatrix} \begin{bmatrix} a \\ b \\ d \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \boldsymbol{y}. \tag{2.6.6}$$

The linear system $M\mathbf{z} = \boldsymbol{y}$ has a unique solution if and only if the matrix M is nonsingular, or equivalently, $\det(M) \neq 0$. Computing $\det(M)$ by expanding the first column yields

$$\begin{aligned} \det(M) &= \begin{vmatrix} x_2 & -x_2y_2 \\ x_3 & -x_3y_3 \end{vmatrix} - \begin{vmatrix} x_1 & -x_1y_1 \\ x_3 & -x_3y_3 \end{vmatrix} + \begin{vmatrix} x_1 & -x_1y_1 \\ x_2 & -x_2y_2 \end{vmatrix} \\ &= [-x_2x_3y_3 + x_2x_3y_2] - [-x_1x_3y_3 + x_1x_3y_1] + [-x_1x_2y_2 + x_1x_2y_1] \\ &= [x_1y_1x_2 - x_1y_1x_3] + [x_2y_2x_3 - x_2y_2x_1] + [x_3y_3x_1 - x_3y_3x_2] \\ &= x_1y_1(x_2 - x_3) + x_2y_2(x_3 - x_1) + x_3y_3(x_1 - x_2). \end{aligned}$$

Thus, for any given data points $(x_i, y_i), i = 1, 2, 3$, the given interpolating problem has a unique solution if and only if

$$x_1y_1(x_2 - x_3) + x_2y_2(x_3 - x_1) + x_3y_3(x_1 - x_2) \neq 0.$$

Below, we investigate a few special cases.

i. WLOG, suppose $x_1 = 0$. Then $\det(M)$ reduces to

$$\det(M) = x_2y_2x_3 - x_3y_3x_2 = x_2x_3(y_2 - y_3).$$

Since x_2, x_3 are both nonzero, for this particular case, the given interpolating problem has a unique solution if and only if $y_2 \neq y_3$.

ii. Suppose $y_1 = y_2 = y_3 = 0$, then $\det(M) = 0$ and the interpolating problem has infinitely many solutions $\mathbf{z} \in \mathbb{R}^3$ of the form $(0, 0, \lambda)^T$, λ any real numbers.

iii. WLOG, suppose $y_1 = y_2 = 0, y_3 \neq 0$. Then $\det(M)$ reduces to

$$\det(M) = x_3y_3(x_1 - x_2).$$

Since $x_1 \neq x_2$, if $y_1 = y_2 = 0$, the given interpolating problem has a unique solution if and only if $(x_3, y_3) \neq (0, 0)$.

(b) Let $x_0, x_1, \ldots, x_n$ be distinct real points, and consider the following interpolation problem. Choose a function

$$F_n(x) = \sum_{j=0}^{n} c_j e^{jx},$$

such that

$$F_n(x_i) = y_i, \quad i = 0, 1, \ldots, n,$$

with $\{y_i\}_{i=0}^n$ the given data. Show that there is a unique choice of $c_0, \ldots, c_n$.

Solution: Introducing a change of variable $z = e^x$, the function $F_n(x)$ becomes:

$$F_n(x) \equiv g_n(z) = \sum_{j=0}^{n} c_j z^j.$$

The distinct real points $\{x_0, x_1, \ldots, x_n\}$ becomes $\{e^{x_0}, e^{x_1}, \ldots, e^{x_n}\}$ which are also distinct real points since the exponential function is injective. The interpolating problem now takes the following form: find $g_n(x)$ such that $g_n(z_j) = y_j, j = 0, 1, \ldots, n$. Since $g_n(z)$ is a polynomial of degree at most n, it follows from Theorem 2.1.1 that there exists a unique choice of $\{c_0, \ldots, c_n\}$.

5. Consider the function $f(x) = \dfrac{1}{x^2+1}$ on the interval $[-5, 5]$. For each $n \geqslant 1$, define $h = 10/n$ and $x_j = -5 + jh$ for $j = 0, 1, \ldots, n$. Let $p_n(x)$ be the polynomial of degree n which interpolates f at the nodes $x_0, x_1, \ldots, x_n$. Compute p_n for $n = 1, 2, \ldots, 20$. Plot $f(x)$ and $p_n(x)$ for each n. Estimate the maximum error $|f(x) - p_n(x)|$ for $x \in (-5, 5)$. Discuss what you find.

Solution: We choose to find the interpolating polynomial in Newton's form. We first compute all the required higher order divided differences, $p_n(x)$ can then be found by using a nested form of polynomial multiplication, *i.e.*

$$p_n(x) = D_0 + (x - x_0)\Big\{D_1 + (x - x_1)\Big[D_2 + \ldots + (x - x_{n-2})[D_{n-1} + (x - x_{n-1})D_n]\ldots\Big]\Big\},$$

where $D_j = f[x_0, \ldots, x_{j-1}], j = 0, 1, \ldots, n$. The numerical result shows that the interpolating polynomials are all even functions, this is due to f being an even function.

We choose to measure the interpolation error using the L^∞ norm, *i.e.* for each $p_n(x), n = 1, 2, \ldots, 20$, we compute

$$E_n = \max_{x\in[-5,5]} |f(x) - p_n(x)| \quad \text{or equivalently} \quad E_n = -\left[\min_{x\in[-5,5]} -|f(x) - p_n(x)|\right]$$

The command fminbnd is applicable to the second expression of E_n. Note that fminbnd returns the argmin of the function $-|f(x) - p(x)|$ over $[-5, 5]$, but unfortunately this does not correspond to the absolute minimum over $[-5, 5]$. To overcome this, we plot the graph of $-|f(x) - p(x)|$, locate the argmin x^* of its absolute minimum over $[-5, 5]$ from the graph and apply fminbnd on an interval around x^*. We present the interpolation error E_n for different $n = 1, 2, \ldots, 20$ in the following table.

n	Interpolation error E_n
1	0.961538461538462
2	0.646229268183428
3	0.707013574660634
4	0.438357141903084
5	0.432692307692308
6	0.616947968654934
7	0.247358606559315
8	1.045176657474316
9	0.300297936742191
10	1.915658914837769

n	Interpolation error E_n
11	0.556775115226897
12	3.663394060743355
13	1.070105627260649
14	7.194881834955054
15	2.107561131513046
16	14.393854684643465
17	4.224288081812976
18	29.190582028039042
19	8.579090824899694
20	59.822308737051372

The interpolation error is consistent with the result stated in the lecture, namely for any $3.64 < |x| < 5$ we have that

$$\sup_{n\in\mathbb{N}} |f(x) - p_n(x)| = \infty.$$

Chapter 3

Numerical Integration

In this chapter, we derive and analyse various numerical methods for approximating definite integrals of the form

$$I(f) = \int_a^b f(x)\,dx, \tag{3.0.1}$$

with $[a, b]$ some finite interval. The approximation of $I(f)$ is commonly known as **numerical integration** or **quadrature**. There are several motivations for performing numerical integration

1. It is often the case that the integrand $f(x)$ are only known at few points.

2. Not every integrand has an antiderivative that is an elementary function.

3. Even if an explicit antiderivative formula exists, it might not be the most efficient way of computing the definite integral. This is the case when the antiderivative is given as an infinite sum or product.

The simplest method for approximating (3.0.1) is as follows. Given an integrand $f(x)$ on $[a, b]$, construct a family of approximating function $(f_n), n \geqslant 1$, where n refers to the number of subintervals on $[a, b]$. Define

$$I_n(f) \coloneqq I(f_n) = \int_a^b f_n(x)\,dx. \tag{3.0.2}$$

and the error function

$$E_n(f) \coloneqq I(f) - I_n(f) = \int_a^b \Big[f(x) - f_n(x)\Big]\,dx. \tag{3.0.3}$$

In a relatively simple case, one usually requires that $\|f - f_n\|_\infty \longrightarrow 0$ as $n \longrightarrow \infty$, since

$$|E_n(f)| \leqslant \int_a^b |f(x) - f_n(x)|\,dx \leqslant (b-a)\|f - f_n\|_\infty \longrightarrow 0 \text{ as } n \longrightarrow \infty.$$

3.1 The Trapezoidal and Simpson's Rule

Most quadrature formulas are based on approximating $f(x)$ with polynomial or piecewise polynomial interpolation. We consider such cases with evenly spaced node points for the remaining section.

3.1.1 Simple Trapezoidal Rule

We approximate the integrand $f(x)$ using linear interpolation $p_1(x)$, this simply refers to the straight line joining the points $(a, f(a))$ and $(b, f(b))$. Referring to the Lagrange's formula (2.1.4), we have the following approximation

$$f(x) \approx p_1(x) = \left(\frac{x-b}{a-b}\right) f(a) + \left(\frac{x-a}{b-a}\right) f(b) = \frac{(b-x)f(a) + (x-a)f(b)}{b-a}.$$

This gives rise to the **Simple Trapezoidal Rule**

$$I_1(f) = \int_a^b \frac{(b-x)f(a) + (x-a)f(b)}{b-a}\, dx = \left(\frac{b-a}{2}\right)[f(a) + f(b)], \qquad \text{(Simple Trapezoidal)}$$

which is simply the area of trapezoid.

Error Analysis

To analyse the error, assume $f \in C^2[a, b]$. For a linear interpolant, the interpolation error formula (IEF) from Theorem 2.1.4 gives

$$\begin{aligned} f(x) - \frac{(b-x)f(a) + (x-a)f(b)}{b-a} = f(x) - p_1(x) &= (x-a)(x-b)\frac{f''(\xi)}{2} \\ &= (x-a)(x-b)f[a, b, x], \end{aligned}$$

where $f[a, b, x]$ is the second order divided difference. Since $g(x) = (x-a)(x-b) \leqslant 0$ on $[a, b]$, it follows from the **Mean Value Theorem for Integrals** that there exists an $\xi \in [a, b]$ such that

$$\begin{aligned} E_1(f) = \int_a^b f(x) - p_1(x)\, dx &= \int_a^b (x-a)(x-b)f[a, b, x]\, dx \\ &= f[a, b, \xi] \int_a^b (x-a)(x-b)\, dx \\ &= \left[\frac{f''(\eta)}{2}\right]\left[-\frac{1}{6}(b-a)^3\right], \quad \text{for some } \eta \in [a, b]. \end{aligned}$$

Writing $b - a$ as h, we have

$$E_1(f) = -\left[\frac{f''(\eta)}{12}\right] h^3 \quad \text{for some } \eta \in [a, b]. \tag{3.1.1}$$

Observe that the simple trapezoidal rule is exact for polynomials of degree at most 1. A bad feature of the simple trapezoidal rule is that it does not serve as a good approximation to oscillatory functions. For completeness, we include the proof of the mean value theorem for integrals.

Theorem 3.1.1 (Mean Value Theorem For Integrals)**.** *Suppose f, g are continuous on $[a, b]$ and g does not change sign on $[a, b]$. There exists $c \in [a, b]$ such that*

$$\int_a^b f(x)g(x)\,dx = f(c)\int_a^b g(x)\,dx.$$

Proof. By Extreme Value Theorem, f attains its extremum, *i.e.* there exists $x_0 < x_1$ (WLOG) such that

$$f(x_0) = m = \min_{x\in[a,b]} f(x), \quad f(x_1) = M = \min_{x\in[a,b]} f(x).$$

Consider the continuous function $h(x) = f(x)\displaystyle\int_a^b g(s)\,ds$ on $[a, b]$. WLOG, suppose g is non-negative on $[a, b]$. We have that

$$\begin{aligned} mg(x) \leqslant \quad & f(x)g(x) \quad \leqslant Mg(x) && \text{for all } x \in [a,b] \\ \implies h(x_0) = m\int_a^b g(x)\,dx \leqslant & \int_a^b f(x)g(x)\,dx \leqslant M\int_a^b g(x)\,dx = h(x_1) && \text{for all } x \in [a,b]. \end{aligned}$$

It follows from the Intermediate Value Theorem that there exists an $c \in [a, b]$ such that

$$h(c) = \int_a^b f(x)g(x)\,dx = f(c)\int_a^b g(x)\,dx.$$

■

3.1.2 Composite Trapezoidal Rule

If $(b - a)$ is not sufficiently small, it follows from the previous error analysis that the simple trapezoidal rule is not very useful. For such an integral, we subdivide the interval $[a, b]$ into n evenly spaced subintervals and apply (Simple Trapezoidal) onto each of these subintervals. More precisely, for any $n \geqslant 1$, define

$$h = \frac{b-a}{n}, \quad x_j = a + jh \quad \text{for } j = 0, 1, \ldots, n.$$

Then

$$\begin{aligned} I(f) = \int_a^b f(x)\,dx &= \sum_{j=1}^n \int_{x_{j-1}}^{x_j} f(x)\,dx \\ &= \sum_{j=1}^n \left(\underbrace{\frac{h}{2}[f(x_{j-1}) + f(x_j)]}_{\text{simple trapezoidal rule}} - \underbrace{\frac{h^3}{12}f''(\eta_j)}_{\text{error}} \right) \end{aligned}$$

$$= \sum_{j=1}^{n} \left(\frac{h}{2}[f(x_{j-1}) + f(x_j)] \right) - \sum_{j=1}^{n} \frac{h^3}{12} f''(\eta_j),$$

where $\eta_j \in [x_{j-1}, x_j], j = 1, \ldots, n$. Denoting $f_j \colon f(x_j), j = 0, \ldots, n$, the first sum is defined as the **Composite Trapezoidal Rule**, given by

$$I_n(f) = h\left(\frac{1}{2}f_0 + f_1 + f_2 + \ldots + f_{n-1} + \frac{1}{2}f_n\right), \; n \geqslant 1. \qquad \text{(Composite Trapezoidal)}$$

Error Analysis

Assume $f \in C^2[a, b]$. For the error in $I_n(f)$,

$$E_n(f) = I(f) - I_n(f) = -\sum_{j=1}^{n} \frac{h^3}{12} f''(\eta_j), \tag{3.1.2}$$

i.e. the error consists of sum of local errors. One can obtain an average estimate for the sum of local errors by looking at the global behaviour of the integrand $f(x)$. Since $f \in C^2[a, b]$, $f'' \in C[a, b]$ and it follows from the **Extreme Value Theorem** that

$$\min_{x \in [a,b]} f''(x) \leqslant f''(\eta_j) \leqslant \max_{x \in [a,b]} f''(x) \quad \text{for each } 1 \leqslant j \leqslant n. \tag{3.1.3}$$

Summing (3.1.3) from $j = 1$ to $j = n$ and dividing by n yields

$$\min_{a \leqslant x \leqslant b} f''(x) \leqslant \frac{1}{n} \sum_{j=1}^{n} f''(\eta_j) \leqslant \max_{a \leqslant x \leqslant b} f''(x).$$

It follows from the **Intermediate Value Theorem** that there exists $\eta \in [a, b]$ such that

$$f''(\eta) = M = \frac{1}{n} \sum_{j=1}^{n} f''(\eta_j).$$

Thus, (3.1.2) reduces to

$$E_n(f) = -\frac{h^3}{12} \sum_{j=1}^{n} f''(\eta_j) = -\frac{h^3 n}{12} f''(\eta) = -\left[\frac{(b-a)f''(\eta)}{12}\right] h^2 \quad \text{for some } \eta \in [a, b]. \tag{3.1.4}$$

We see that $E_n(f) = \mathcal{O}(h^2)$ as $h \longrightarrow 0$. Observe that the composite trapezoidal rule is exact if f is a polynomial of degree at most 1.

Remark 3.1.2. A similar argument produces another error estimate for the composite trapezoidal rule. Indeed,

$$\lim_{n \to \infty} \frac{E_n(f)}{h^2} = -\frac{1}{12} \lim_{n \to \infty} \underbrace{\left(\sum_{j=1}^{n} f''(\eta_j) h \right)}_{\text{Riemann sum}} = -\frac{1}{12} \int_a^b f''(x)\, dx = -\frac{1}{12}[f'(b) - f'(a)],$$

3.1.3 Simpson's Rule

To improve upon the simple trapezoidal rule, we approximate the integrand $f(x)$ using quadratic interpolant $p_2(x)$ instead. Since we require at least three interpolating nodes for quadratic interpolation, we choose the third node to be the midpoint $c = \dfrac{a+b}{2}$. Thus,

$$\begin{aligned} I_2(f) &= \int_a^b p_2(x)\,dx \\ &= \int_a^b \left[\frac{(x-c)(x-b)}{(a-c)(a-b)} f(a) + \frac{(x-a)(x-c)}{(b-a)(b-c)} f(b) + \frac{(x-a)(x-b)}{(c-a)(c-b)} f(c) \right] dx. \end{aligned}$$

Let $h = \dfrac{b-a}{2}$, performing a change of variable $x = y + a$,

$$\begin{aligned} \int_a^b \frac{(x-c)(x-b)}{(a-c)(a-b)}\,dx &= \frac{1}{2h^2} \int_a^{a+2h} (x-c)(x-b)\,dx \\ &= \frac{1}{2h^2} \int_0^{2h} (y+a-c)(y+a-b)\,dy \\ &= \frac{1}{2h^2} \int_0^{2h} (y-h)(y-2h)\,dy \\ &= \frac{1}{2h^2} \int_0^{2h} [y^2 - 3hy + 2h^2]\,dy \\ &= \left(\frac{1}{2h^2}\right)\left(\frac{2h^3}{3}\right) = \frac{h}{3}. \end{aligned}$$

Repeating the same procedure, we obtain

$$I_2(f) = S_2(f) = \frac{h}{3}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right], \quad h = \frac{b-a}{2}. \tag{Simpson}$$

This is called the **Simpson's Rule**.

Error Analysis

Assume $f \in C^4[a,b]$. The interpolation error formula (IEF) from Theorem 2.1.4 gives

$$\begin{aligned} E_2(f) = \int_a^b \Big[f(x) - p_2(x) \Big]\,dx &= \int_a^b (x-a)(x-c)(x-b) \frac{f^{(3)}(\xi)}{3!}\,dx \\ &= \int_a^b (x-a)(x-c)(x-b) f[a,b,c,x], \end{aligned}$$

where $f[a,b,c,x]$ is the third order divided difference. Ideally, we would like to use the mean value theorem for integrals again, but it is not applicable here since $g(x) = (x-a)(x-c)(x-b)$ changes sign at $x = c = \dfrac{a+b}{2}$. Define

$$w(x) = \int_a^x (t-a)(t-c)(t-b)\,dt.$$

Clearly, $w(a) = 0$ by construction. A symmetry argument (since c is the midpoint between a and b) shows that $w(b) = 0$, while a critical point argument shows that $w(x) > 0$ for all $x \in (a, b)$. Now, integrating by parts gives

$$\begin{aligned}\int_a^b (x-a)(x-c)(x-b)f[a,b,c,x]\,dx &= \int_a^b w'(x)f[a,b,c,x]\,dx \\ &= \Big[w(x)f[a,b,c,x]\Big]\Big|_a^b - \int_a^b w(x)\left(\frac{d}{dx}f[a,b,c,x]\right)dx \\ &= -\int_a^b w(x)f[a,b,c,x,x]\,dx,\end{aligned}$$

where it can be shown using Lemma 2.2.2 that

$$\begin{aligned}\frac{d}{dx}f[x_0,x_1,\ldots,x_n,x] &= \lim_{h\to 0}\frac{f[x_0,x_1,\ldots,x_n,x+h]-f[x_0,x_1,\ldots,x_n,x]}{h} \\ &= \lim_{h\to 0}\frac{f[x_0,x_1,\ldots,x_n,x+h]-f[x,x_0,x_1,\ldots,x_n]}{h} \\ &= \lim_{h\to 0} f[x,x_0,x_1,\ldots,x_n,x+h] \\ &= f[x,x_0,x_1,\ldots,x_n,x].\end{aligned}$$

With $g(x) = w(x) \geqslant 0$ on $[a, b]$, it follows from the **Mean Value Theorem for Integrals** that there exists an $\xi \in [a, b]$ such that

$$\begin{aligned}E_2(f) &= -\int_a^b w(x)f[a,b,c,x,x]\,dx \\ &= -f[a,b,c,\xi,\xi]\int_a^b w(x)\,dx \\ &= -\frac{f^{(4)}(\eta)}{24}\left(\frac{4}{15}h^5\right), \quad \text{for some } \eta\in[a,b], h=\frac{b-a}{2}\end{aligned}$$

Hence,

$$\boxed{E_2(f) = -\left(\frac{f^{(4)}(\eta)}{90}\right)h^5 \quad \text{for some } \eta\in[a,b].} \tag{3.1.5}$$

Observe that Simpson's rule is exact for polynomials of degree at most 3, even though quadratic interpolation is exact for polynomial of degree at most 2. In essence, Simpson's rule is much more accurate than the trapezoidal rule.

3.1.4 Composite Simpson's Rule

As before, the error analysis shows that Simpson's rule is not so useful if $h = b - a$ is not sufficiently small. Following a similar idea to that used to derive the composite trapezoidal rule (Composite Trapezoidal), we subdivide the interval [a,b] onto n evenly spaced subintervals and apply (Simpson) onto each of these two successive subintervals. More precisely, for any even integer $n \geqslant 2$, define

$$h = \frac{b-a}{n}, \quad x_j = a + jh \quad \text{for } j = 0, 1, \ldots, n.$$

Applying Simpson's rule onto the subintervals $[x_{2j}, x_{2(j+1)}], j = 0, 1, \ldots, \dfrac{n-2}{2}$ gives

$$\begin{aligned} I(f) &= \int_{x_0}^{x_2} f(x)\,dx + \int_{x_2}^{x_4} f(x)\,dx + \ldots + \int_{x_{n-2}}^{x_n} f(x)\,dx \\ &= \sum_{j=0}^{\frac{n-2}{2}} \int_{x_{2j}}^{x_{2j+2}} f(x)\,dx \\ &= \sum_{j=0}^{\frac{n-2}{2}} \left(\underbrace{\frac{h}{3}\Big[f(x_{2j}) + 4f(x_{2j+1}) + f(x_{2j+2})\Big]}_{\text{simple trapezoidal rule}} - \underbrace{\frac{h^5}{90} f^{(4)}(\eta_j)}_{\text{error}} \right). \end{aligned}$$

This yields the **Composite Simpson's Rule**

$$\begin{aligned} I_n(f) = S_n(f) &= \sum_{j=0}^{\frac{n-2}{2}} \frac{h}{3}\Big[f(x_{2j}) + 4f(x_{2j+1}) + 2f(x_{2j+2})\Big] \\ &= \frac{h}{3}\Big[f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \ldots + 2f_{n-2} + 4f_{n-1} + f_n\Big] \end{aligned}$$

(Composite Simpson)

Error Analysis

Similar to the composite trapezoidal rule, one can show that

$$\begin{aligned} E_n(f) = I(f) - I_n(f) &= -\sum_{j=0}^{\frac{n-2}{2}} \frac{h^5}{90} f^{(4)}(\eta_j) && \text{for some } \eta_j \in [x_{2j}, x_{2j+2}] \\ &= -\frac{h^5}{90}\left(\frac{n}{2}\right)\left(\frac{2}{n}\right) \sum_{j=0}^{\frac{n-2}{2}} f^{(4)}(\eta_j) \\ &= -\frac{h^5 n}{180} f^{(4)}(\eta). && \text{for some } \eta \in [a, b] \end{aligned}$$

Substituting $n = (b-a)/h$ gives

$$E_n(f) = -\left[\frac{(b-a) f^{(4)}(\eta)}{180}\right] h^4 \quad \text{for some } \eta \in [a, b]. \tag{3.1.7}$$

Observe that the composite Simpson's rule is exact for polynomials of degree at most 3.

3.2 Note on Newton-Cotes Formulas

Given $(n+1)$ interpolation nodes, consider the interpolating polynomial $p_n(x)$ in Lagrange's form

$$p_n(x) = \sum_{i=0}^{n} l_i(x) f(x_i), \text{ where } \quad l_i(x) = \prod_{j \neq i}^{n} \frac{x - x_j}{x_i - x_j}.$$

Approximating the integrand $f(x)$ using $p_n(x)$ yields

$$I(f) = \int_a^b f(x)\,dx \approx \int_a^b p_n(x)\,dx = \int_a^b \left(\sum_{i=0}^{n} l_i(x) f(x_i) \right) dx = \sum_{i=0}^{n} A_i f(x_i), \tag{3.2.1}$$

where A_i's are called **weights**, having the form

$$A_i = \int_a^b l_i(x)\,dx.$$

For **equally spaced nodes**, (3.2.1) is called the **Newton-Cotes formula** or the **Newton-Cotes quadrature rules**. The simple trapezoidal rule (Simple Trapezoidal) $(n = 1)$ and Simpson's rule (Simpson) $(n = 2)$ are two examples of the Newton-Cotes formula.

To estimate the error, we employ a similar argument as in Section 2.3. More precisely, if $|f^{(n+1)}(x)| \leqslant M$ on $[a, b]$, one has

$$\begin{aligned}
\left| \int_a^b f(x)\,dx - \sum_{i=0}^{n} A_i f(x_i) \right| &= \left| \int_a^b \Big[f(x) - p(x) \Big]\,dx \right| \\
&= \left| \int_a^b \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \left(\prod_{i=0}^{n} (x - x_i) \right) dx \right| \\
&\leqslant \frac{M}{(n+1)!} \left(\int_a^b \prod_{i=0}^{n} |x - x_i|\,dx \right) = \frac{M}{(n+1)!} \phi_n(x).
\end{aligned}$$

3.2.1 Chebyshev Polynomials of Second Kind

Following Section 2.4, we can minimise the error bound for Newton-Cotes formula by choosing the interpolation nodes to be roots of **Chebyshev polynomials of the second kind**. Let $U_0(x) = 1, U_1(x) = 2x$ and

$$U_{n+1}(x) = 2xU_n(x) - U_{n-1}(x), \quad n \geqslant 1. \tag{3.2.2}$$

Observe that U_n is a polynomial of degree n for each $n \geqslant 0$ and $U_n(x) = 2^n x^n + \ldots$. For example,

$$\begin{aligned}
U_2(x) &= 4x^2 - 1 \\
U_3(x) &= 8x^3 - 4x \\
U_4(x) &= 16x^4 - 12x^2 + 1.
\end{aligned}$$

Lemma 3.2.1. *For any $x \in [-1,1]$, the Chebyshev polynomials of the second kind admits the closed form expression*

$$U_n(x) = \frac{\sin((n+1)\cos^{-1}(x))}{\sin(\cos^{-1} x)}, n \geqslant 0. \tag{3.2.3}$$

It follows that $U_n(\cos(x)) = \dfrac{\sin((n+1)x)}{\sin(x)}, n \geqslant 0.$

Proof. The case $n = 0, 1$ is trivial. For $n \geqslant 2$, we need to show that (3.2.3) coincides with the recursive relation (3.2.2). ■

It is clear from Lemma 3.2.1 that $U_n(x)$ has zeros at

$$x_j = \cos\left(\frac{j\pi}{n+1}\right), \quad 1 \leqslant j \leqslant n. \tag{3.2.4}$$

One can show a similar result to Theorem 2.4.4, namely $\phi_n(x)$ is minimised if the interpolation nodes $\{x_j\}_{j=0}^n$ are chosen to be zeros of the Chebyshev polynomial of the second kind U_{n+1} given by

$$x_j = \cos\left(\frac{(j+1)\pi}{n+2}\right), \quad 0 \leqslant j \leqslant n. \tag{3.2.5}$$

This leads to

$$\min \phi_n(x) = \int_a^b \left(\prod_{i=0}^n |x - x_i|\right) dx = \frac{1}{2^n}.$$

3.2.2 Midpoint Rule

There are Newton-Cotes formulas in which one of both of the endpoints of integration are removed from the interpolation nodes. We approximate the integrand f using the constant function $f\left(\dfrac{a+b}{2}\right)$ and this leads to the **Simple Midpoint Rule**

$$I_1(f) = \int_a^b f\left(\frac{a+b}{2}\right) dx = (b-a) f\left(\frac{a+b}{2}\right). \tag{Simple Midpoint}$$

Error Analysis

Assume $f \in C^2[a,b]$. Denote $c = \dfrac{a+b}{2}$. It follows from Taylor's theorem that:

$$E_0(f) = \int_a^b f(x) - f(c)\, dx = \int_a^b \left[f'(c)(x-c) + \frac{f''(\xi_x)}{2}(x-c)^2\right] dx,$$

for some $\xi_x \in [a,b]$. The first integral vanishes since the function $x - c$ is symmetric about the point c which is the midpoint of the interval $[a,b]$. For the second integral, since $(x-c)^2 \geqslant 0$ for all $x \in [a,b]$, it follows from the **Mean Value Theorem for Integrals** that there exists an $\eta \in [a,b]$ such that:

$$E_0(f) = \int_a^b (x-c)^2 \left(\frac{f''(\xi_x)}{2}\right) dx = \left(\frac{f''(\eta)}{2}\right) \int_a^b (x-c)^2\, dx$$

$$= \left(\frac{f''(\eta)}{2}\right)\int_0^{2h}(y-h)^2\,dx$$
$$= \left(\frac{f''(\eta)}{2}\right)\left(\frac{2h^3}{3}\right), \quad \text{where } h=(c-a).$$

Consequently, the error is given by:

$$E_0(f) = \frac{(b-a)^3}{24}f''(\eta) \quad \text{for some } \eta \in [a,b]. \tag{3.2.6}$$

3.2.3 Composite Midpoint Rule

We subdivide the interval into n evenly spaced subintervals and apply the (Simple Midpoint) onto each of these subintervals. More precisely, for any given $n \geqslant 1$, define

$$h = \frac{b-a}{n}, \quad x_j = a+jh, \quad j=0,1,\ldots,n, \quad c_j = \frac{x_{j-1}+x_j}{2}, \quad j=1,\ldots,n.$$

where c_j's are the midpoints of the subintervals $[x_{j-1}, x_j]$. Then

$$I(f) = \int_a^b f(x)\,dx = \sum_{j=1}^{n}\int_{x_{j-1}}^{x_j} f(x)\,dx$$
$$= \sum_{j=1}^{n}\left(hf(c_j) + \frac{h^3}{24}f''(\eta_j)\right)$$

where $\eta_j \in [x_{j-1}, x_j], j = 1,\ldots,n$. The first sum is defined as the **Composite Midpoint Rule**, given by

$$I_n(f) = h[f(c_1)+f(c_2)+\ldots+f(c_n)], n \geqslant 1, \quad \text{where } c_j := \frac{x_{j-1}+x_j}{2}.$$

(Composite Midpoint)

A similar argument for the error analysis of composite trapezoidal rule shows that

$$E_n(f) = \frac{h^2(b-a)}{24}f''(\eta) \quad \text{for some } \eta \in [a,b]. \tag{3.2.7}$$

Observe that the composite midpoint rule is exact for polynomials of degree at most 1.

Remark 3.2.2. Newton-Cotes formula has limited application due to failure of convergence for some functions, where the problem arises from the assumption that the interpolation nodes are uniformly spaced.

3.3 Gaussian Quadrature

The composite trapezoidal and Simpson's rule are based on using a lower-order polynomial approximation of the integrand on subintervals of decreasing size. Instead of approximating

the integrand, we approximate the integral directly. Consider

$$I_n(f) = \sum_{j=1}^{n} \omega_j f(x_j) \approx \int_a^b w(x) f(x)\, dx = I(f), \tag{3.3.1}$$

where $w(x)$ is the weight function that is nonnegative, integrable on $[a, b]$, and satisfying the following two properties:

1. $\int_a^b |x|^n w(x)\, dx$ is integrable and finite for all $n \geqslant 0$,
2. If $\int_a^b w(x)g(x)\, dx = 0$ for some nonnegative, continuous function $g(x)$, then $g(x) \equiv \mathbf{0}$ on (a, b).

$\{x_j\}_{j=1}^n$ and $\{w_j\}_{j=1}^n$ are called the **Gauss nodes** and **Gauss weight** respectively. Note that the Gauss weight w_j should not be viewed as an approximation of the weight function $w(x)$.

To gain some intuition behind the construction of Gaussian quadrature, let us choose $\omega(x) \equiv 1$ and consider the integral over the interval $[-1, 1]$. More precisely, we consider the particular case

$$I(f) = \int_{-1}^{1} f(x)\, dx \approx \sum_{j=1}^{n} w_j f(x_j) = I_n(f). \tag{3.3.2}$$

Define the error function

$$E_n(f) := I(f) - I_n(f) = \int_{-1}^{1} f(x)\, dx - \sum_{j=1}^{n} w_j f(x_j). \tag{3.3.3}$$

The aim is to choose the Gauss nodes and weights such that $E_n(p) = 0$ for all polynomials $p(x)$ of as large degree as possible. Suppose $p(x)$ has the form

$$p(x) = \sum_{k=0}^{m} a_k x^k = a_0 + a_1 x + \ldots + a_{m-1} x^{m-1} + a_m x^m. \tag{3.3.4}$$

Expanding $E_n(p)$ gives:

$$\begin{aligned} 0 = E_n(p) &= \int_{-1}^{1} \left(\sum_{k=0}^{m} a_k x^k \right) dx - \sum_{j=1}^{n} w_j \left(\sum_{k=0}^{m} a_k x_j^k \right) \\ &= \sum_{k=0}^{m} a_k \left(\int_{-1}^{1} x^k\, dx \right) - \sum_{k=0}^{m} a_k \left(\sum_{j=1}^{n} w_j x_j^k \right) \\ &= \sum_{k=0}^{m} a_k E_n(x^k). \end{aligned}$$

Since this must holds for any choice of $a_0, a_1, \ldots, a_m$, $E_n(p) = 0$ for every polynomials of degree $\leqslant m$ if and only if

$$E_n(1) = E_n(x) = \ldots = E_n(x^m) = 0, \tag{3.3.5}$$

and nodes and weights are found by solving (3.3.5). We present below how to find nodes and weights for the case $n = 1, 2$, and from there, deduce the equations for nodes and weights for a general $n \geqslant 1$.

1. **Suppose $n = 1$.**
$$\int_{-1}^{1} f(x)\,dx \approx w_1 f(x_1).$$
Since we have two unknowns w_1, x_1, we require that $E_1(1) = E_1(x) = 0$. This gives
$$\begin{aligned} \int_{-1}^{1} 1\,dx = w_1 &\qquad \implies \omega_1 = 2 \\ \int_{-1}^{1} x\,dx = w_1 x_1 = 2x_1 &\implies x_1 = 0. \end{aligned}$$
The Gaussian quadrature in this case corresponds to the simple midpoint rule (Simple Midpoint), *i.e.*
$$\int_{-1}^{1} f(x)\,dx \approx 2f(0),$$
This is exact for polynomials of degree at most 1.

2. **Suppose $n = 2$.** Then
$$\int_{-1}^{1} f(x)\,dx \approx w_1 f(x_1) + w_2 f(x_2).$$
Since we have four unknowns w_1, w_2, x_1, x_2, we require that $E_2(1) = E_2(x) = E_2(x^2) = E_2(x^3) = 0$. This yields
$$E_n(x^j) = \int_{-1}^{1} x^j\,dx - (w_1 x_1^j + w_2 x_2^j) = 0, \quad j = 0, 1, 2, 3,$$
which gives four nonlinear equations
$$w_1 + w_2 = 2 \tag{3.3.6a}$$
$$w_1 x_1 + w_2 x_2 = 0 \tag{3.3.6b}$$
$$w_1 x_1^2 + w_2 x_2^2 = \frac{2}{3} \tag{3.3.6c}$$
$$w_1 x_1^3 + w_2 x_2^3 = 0. \tag{3.3.6d}$$
Taking advantage of the symmetry structure of (3.3.6), we look for solutions of the form
$$w_1 = w_2, \quad x_1 = -x_2.$$
It follows from (3.3.6a) that $w_1 = w_2 = 1$, and (3.3.6c) gives
$$2x_1^2 = \frac{2}{3} \implies x_1 = \pm\frac{\sqrt{3}}{3}.$$
The Gaussian quadrature in this case has the form
$$\int_{-1}^{1} f(x)\,dx \approx f\left(-\frac{\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right),$$
and this is exact for polynomials of degree at most 3. This is better compared to Simpson's rule which uses three interpolation nodes.

3. **For a general $n \geqslant 1$**, there are $2n$ unknowns and we require $2n$ equations given by

$$E_n(x^k) = \int_{-1}^{1} x^k \, dx - \sum_{j=1}^{n} w_j x_j^k = 0, \quad k = 0, 1, \ldots, 2n-1.$$

Exploiting the integral structure of monomials, the following $2n$ nonlinear equations are obtained

$$\sum_{j=1}^{n} w_j x_j^k = \begin{cases} 0 & \text{if } k = 1, 3, \ldots, 2n-1, \\ \dfrac{2}{k+1} & \text{if } k = 0, 2, \ldots, 2n-2. \end{cases} \tag{3.3.7}$$

Theorem 3.3.1. *For each $n \geqslant 1$, there is a unique quadrature formula $I_n(f) = \sum_{j=1}^{n} w_j f(x_j)$ of degree of precision $(2n-1)$. Assuming $f \in C^{2n}[a,b]$, we have*

$$\int_a^b w(x) f(x) \, dx = \sum_{j=1}^{n} w_j f(x_j) + E_n(f), \tag{3.3.8}$$

where

$$\begin{cases} E_n(f) &= \left[\dfrac{f^{(2n)}(\xi)}{(2n)!}\right] \underbrace{\displaystyle\int_a^b q^2(x) w(x) \, dx}_{\text{only depends on } n} \quad \text{for some } \xi \in (a,b), \\ q(x) &= \displaystyle\prod_{j=1}^{n} (x - x_j). \end{cases} \tag{3.3.9}$$

The Gauss nodes $\{x_j\}_{j=1}^n$ are zeros of $\phi_n(x)$, where $\{\phi_n(x)\}$ are orthogonal polynomials on $[a,b]$ with respect to the weighted inner product $(\cdot,\cdot)_w$, i.e.

$$(\phi_j, \phi_k)_w := \int_a^b w(x) \phi_j(x) \phi_k(x) \, dx = 0 \quad \text{for any } j \neq k.$$

Remark 3.3.2.

1. Examples of weight function with their respective orthogonal polynomials can be found at the end of Section 2.5.

2. One can show that the Gauss weight has the form

$$w_j = \int_a^b w(x) [l_j(x)]^2 \, dx > 0,$$

where $l_j(x)$ are the Lagrange basis polynomials, $j = 1, \ldots, n$ [See (2.1.3)]. Thus, Gauss weights are positive for all $n \geqslant 1$. Note also that

$$\sum_{i=j}^{n} w_j = \int_a^b w(x) \, dx,$$

since the Gauss quadrature is exact for $f(x) \equiv 1$.

3. For integrals on an arbitrary but finite interval $[a, b]$ with weight function $w(x) \equiv 1$, one can reduce the integral over $[a, b]$ to the integral over $[-1, 1]$ using the affine transformation
$$\int_a^b f(t)\,dt = \left(\frac{b-a}{2}\right)\int_{-1}^1 f\left(\frac{a+b+x(b-a)}{2}\right)\,dx.$$

Theorem 3.3.3. *If $f \in C[a, b]$, then*
$$\sum_{j=1}^n w_j f(x_j) \longrightarrow \int_a^b w(x)f(x)\,dx \quad \text{as } n \longrightarrow \infty.$$

Proof. WLOG, we may choose the interval to be $[-1, 1]$; otherwise we can rescale the function using an affine transformation. Given $\varepsilon > 0$, it follows from the **Weierstrass Approximation Theorem** that there exists a polynomial $p(x)$ such that
$$|f(x) - p(x)| < \varepsilon \quad \text{for all } x \in [-1, 1].$$
Choose $n \geqslant 1$ sufficiently large such that $2n$ exceeds the degree of p. Then
$$\sum_{j=1}^n w_j p(x_j) = \int_{-1}^1 w(x)p(x)\,dx,$$
which leads to
$$\begin{aligned}\left|\int_{-1}^1 w(x)f(x)\,dx - \sum_{j=1}^n w_j f(x_j)\right| &\leqslant \left|\int_{-1}^1 w(x)f(x)\,dx - \int_{-1}^1 w(x)p(x)\,dx\right| + \left|\sum_{j=1}^n w_j p(x_j) - \sum_{j=1}^n w_j f(x_j)\right| \\ &\leqslant \int_{-1}^1 w(x)|f(x) - p(x)|\,dx + \sum_{j=1}^n w_j |p(x_j) - f(x_j)| \\ &\leqslant \left(\int_{-1}^1 w(x)\,dx\right)\varepsilon + \left(\sum_{j=1}^n w_j\right)\varepsilon \\ &= 2\varepsilon\left(\int_{-1}^1 w(x)\,dx\right) \longrightarrow 0 \quad \text{as } n \longrightarrow \infty,\end{aligned}$$
since the weight function $w(x)$ is assumed to be integrable. ∎

3.3.1 2/14/2017

$$f(\theta, \phi) = \sum_{l=0}^\infty f_l^m P_l^m(\cos\theta)e^{-im\phi}\sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}$$
$$f(\theta, \phi) = \sum_{l=0}^\infty f_l^m P_l^m(\cos\theta)e^{-im\phi}\sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}}$$

$$\begin{aligned}\begin{pmatrix} C_l^m \\ S_l^m \end{pmatrix} &= \int_0^{2\pi}\int_0^{\pi} f(\theta,\phi)P_l^m(\cos\theta)\begin{pmatrix}\cos m\phi \\ \sin m\phi\end{pmatrix}\sin\theta\, d\theta d\phi \\ &= \int_{-1}^{1}\left[\int_0^{2\pi} f(x,\theta)\begin{pmatrix}\cos m\phi \\ \sin m\phi\end{pmatrix} d\phi\right] P_l^m(x)\, dx\end{aligned}$$

3.4 Adaptive Quadrature

The idea of **adaptive quadrature** is largely motivated by the following question: Can we minimise the computational cost/function evaluation such that the approximation is within a given tolerance?

1. An important feature of composite quadrature rules is that of evenly spaced nodes, where the intervals are subdivided uniformly until a desired accuracy is achieved. However, this does not take into account the behaviour of the integrand, one example being the magnitude of functional variations.

2. When the integrand is badly behaved at some point α over the interval of integration, one requires sufficiently many nodes to compensate this. Consequently, this forces the need to perform unnecessary computation over parts of $[a,b]$ in which the function is well behaved.

It is thus useful to introduce a quadrature rule that adjust its placement of nodes to reflect the local behaviour of the integrand.

We illustrate the idea behind adaptive quadrature with an **adaptive Simpson's rule**. Consider an interval of integration $[a,b]$ and denote $h = (b-a)/2$, it follows from (Simpson) and (Composite Simpson) that

$$\int_a^b f(x)\,dx = S_{[a,a+2h]} - \frac{h^5}{90}f^{(4)}(\eta) \tag{3.4.1}$$

$$\begin{aligned}\int_a^b f(x)\,dx &= S_{[a,a+h]} + S_{[a+h,a+2h]} - 2\left(\frac{h}{2}\right)^5\frac{1}{90}f^{(4)}(\tilde{\eta}) \\ &= S_{[a,a+h]} + S_{[a+h,a+2h]} - \frac{1}{16}\left(\frac{h^5}{90}f^{(4)}(\tilde{\eta})\right)\end{aligned} \tag{3.4.2}$$

Assuming $f^{(4)}(\eta) \approx f^{(4)}(\tilde{\eta})$, subtracting (3.4.2) from (3.4.1) gives

$$0 = S_{[a,a+2h]} - S_{[a,a+h]} - S_{[a+h,a+2h]} - \underbrace{\frac{15}{16}\left(\frac{h^5}{90}f^{(4)}(\eta)\right)}_{15E^{(2)}_{[a,a+2h]}}$$

$$\implies E^{(2)}_{[a,a+2h]} = \frac{1}{15}[S_{[a,a+2h]} - S_{[a,a+h]} - S_{[a+h,a+2h]}].$$

Given a tolerance $\tau > 0$, if $|E^{(2)}_{[a,a+2h]}| < \tau$, we apply the composite Simpson's rule to approximate the integral. Otherwise, the subinterval is subdivided evenly and the composite Simpson's rule is applied on both halves separately. The new approximation is now tested against the tolerance $\dfrac{\hat{h}\tau}{(b-a)}$, where $\hat{h}$ is the length of the subinterval.

Let's combine

3.5 Singular Integrals

We are interested with two problems in this section:

1. Integrals whose integrands contain a singularity in the interval of integration,
2. Integrals with an infinite interval of integration.

By examining the nature of the singular behaviour of the integrand, it is usually possible to derive a more rapidly convergent approximations.

3.5.1 Change of Variables

The idea is to transform the interval of integration in such a way that the new integrand is sufficiently smooth near the singularity. For a finite interval of integration $[a, b]$ containing singularities, consider for example the following integral

$$I = \int_0^b \frac{f(x)}{\sqrt{x}}\, dx,$$

where $f \in C^n[a, b]$ for some $n \geqslant 1$. Let $x = u^2$, then $dx = 2udu$ and I becomes

$$I = 2\int_0^{\sqrt{b}} f(u^2)\, du.$$

The new integrands is now smooth and standard numerical quadrature can be applied to it.

For an infinite interval of integration, consider the following example

$$I = \int_1^\infty \frac{f(x)}{x^p}\, dx, \quad p > 1, \quad \text{with } \lim_{x\to\infty} f(x) \text{ exists .}$$

Assume f is smooth on $[1, \infty)$. Performing a change of variable

$$x = \frac{1}{u^\alpha} \implies dx = \frac{-\alpha}{u^{1+\alpha}} du, \quad \alpha > 0.$$

This transforms the interval $[1, \infty)$ to the interval $[0, 1]$. This leads to

$$I = \alpha \int_0^1 u^{p\alpha} f\left(\frac{1}{u^\alpha}\right) \frac{du}{u^{1+\alpha}}$$

$$= \alpha \int_0^1 u^{(p-1)\alpha-1} f\left(\frac{1}{u^\alpha}\right) du.$$

The goal now is to maximising the smoothness of the new integrand at $u = 0$, which can be done by choosing $\alpha > 0$ sufficiently large.

Example 3.5.1. Consider the integral

$$I = \int_1^\infty \frac{f(x)}{x\sqrt{x}}\, dx.$$

Performing a change of variable $x = 1/u^4$, one can show that

$$I = 4 \int_0^1 u f\left(\frac{1}{u^4}\right) du.$$

Assuming a behaviour of $f(x)$ near $x = \infty$

$$f(x) = c_0 + \frac{c_1}{x} + \frac{c_2}{x^2} + \ldots,$$

then

$$u f\left(\frac{1}{u^4}\right) = c_0 u + c_1 u^5 + c_2 u^9 + \ldots,$$

i.e. the new integrand is smooth at $u = 0$.

3.5.2 Analytic Treatment of Singularity

This method is common in PDEs and Harmonic Analysis, where the idea is to isolate the singularity. As an example, consider the following integral:

$$I = \int_0^b f(x)\ln(x)\, dx = \int_0^\varepsilon f(x)\ln(x)\, dx + \int_\varepsilon^b f(x)\ln(x)\, dx = I_1 + I_2.$$

Assuming $f(x)$ is smooth on $[\varepsilon, b]$, one can apply a standard numerical quadrature to approximate I_2. Near $x = 0$, assume f has a convergent power series on $[0, \varepsilon]$, *i.e.*

$$f(x) = \sum_{j=0}^\infty a_j x^j.$$

Substituting this into I_1 and integrating by parts gives

$$\begin{aligned} I_1 = \int_0^\varepsilon f(x)\ln(x)\, dx &= \int_0^\varepsilon \left(\sum_{j=0}^\infty a_j x^j\right) \ln(x)\, dx \\ &= \sum_{j=0}^\infty a_j \left(\frac{\varepsilon^{j+1}}{j+1}\right)\left[\ln(\varepsilon) - \frac{1}{j+1}\right], \end{aligned}$$

where we use the following important fact to remove one of the boundary term:

$$\lim_{t\to 0^+} x^\beta \ln(x) = 0 \quad \text{for} \quad \beta > 0.$$

Example 3.5.2. Consider the integral

$$I = \int_0^{4\pi} \cos(x) \ln(x)\, dx.$$

Choosing $\varepsilon = 0.1$, we obtain

$$I_1 = \int_0^{0.1} \cos(x) \ln(x)\, dx, I_2 = \int_{0.1}^{4\pi} \cos(x) \ln(x)\, dx.$$

One can show that I_1 is an alternating series, having the form

$$I_1 = \varepsilon[\ln(\varepsilon) - 1] - \frac{\varepsilon^3}{6}\left(\ln(\varepsilon) - \frac{1}{3}\right) + \frac{\varepsilon^5}{600}\left(\ln(\varepsilon) - \frac{1}{5}\right).$$

Since I_1 is a convergent power series, one can truncate I_1 to obtain an approximation to any desired accuracy. For this particular case, since the terms in I_1 decays fast enough, the first three terms are sufficient to give an accurate value of I_1.

3.6 Problems

1. Derive error estimate for the composite midpoint rule in the form

$$|E_n^M| \leqslant \frac{(b-a)h^2}{24} \max_{x\in[a,b]} |f''(x)|. \tag{3.6.1}$$

The composite midpoint rule is given by

$$M_n(f) = h[f(x_1) + f(x_2) + \ldots + f(x_n)],$$

where $h = (b-a)/n$ and

$$x_j = a + \left(j - \frac{1}{2}\right) h, \quad j = 1, \ldots, n.$$

Solution: Assume $f \in C^2[a,b]$. Consider the case $n = 1$, *i.e.* the simple midpoint rule. For this particular case, $h = b - a$ and

$$M_1(f) = (b-a) f\left(\frac{a+b}{2}\right) = (b-a) f\left(a + \frac{h}{2}\right).$$

Let $c = a + \dfrac{h}{2}$, Taylor expand $f(x)$ around $x = c$ gives

$$\begin{aligned} E_1^M &= \int_a^b f(x)\,dx - (b-a) f\left(a + \frac{h}{2}\right) \\ &= \int_a^b [f(x) - f(c)]\,dx \\ &= \int_a^b \left[f'(c)(x-c) + \frac{f''(\xi)}{2}(x-c)^2 \right] dx \quad \text{for some } \xi \in (a,b). \end{aligned}$$

Observe that

$$\int_a^b (x-c)\,dx = \int_{c-\frac{h}{2}}^{c+\frac{h}{2}} (x-c)\,dx = 0,$$

since the function $x - c$ is odd about the point $x = c$. So the first definite integral is zero and we are left with

$$\begin{aligned} |E_1^M| &\leqslant \frac{1}{2} \int_a^b (x-c)^2 |f''(\xi)|\,dx \\ &\leqslant \frac{1}{2} \left(\max_{x\in[a,b]} |f''(x)| \right) \int_a^b (x-c)^2\,dx \end{aligned}$$

Performing a change of variable $y = x - a$ gives

$$\int_a^b (x-c)^2\,dx = \int_a^{a+h} \left(x - a - \frac{h}{2}\right)^2 dx = \int_0^h \left(y - \frac{h}{2}\right)^2 dy$$

$$= \frac{1}{3}\left(y - \frac{h}{2}\right)^3 \Big|_0^h$$
$$= \frac{1}{3}\left[\left(\frac{h}{2}\right)^3 - \left(-\frac{h}{2}\right)^3\right]$$
$$= \frac{h^3}{12}.$$

Thus,

$$|E_1^M| \leqslant \frac{1}{2}\left(\max_{x\in[a,b]} |f''(x)|\right)\left(\frac{h^3}{12}\right) = \frac{(b-a)^3}{24}\max_{x\in[a,b]} |f''(x)|. \tag{3.6.2}$$

which matches with (3.6.1) since $h = (b-a)$ in this case.

Denote the quadrature error for the simple midpoint rule over an interval $[a,b]$ by $(E_1^M)_{[a,b]}$. For a general $n \geqslant 1$, $h = (b-a)/n$ and

$$\begin{aligned}
|E_n^M| &= \left|\int_a^b f(x)\,dx - M_n(f)\right| \\
&= \left|\int_a^b f(x)\,dx - h[f(x_1) + f(x_2) + \ldots + f(x_n)]\right| \\
&= \left|\int_a^{a+h} f(x)\,dx - hf(x_1) + \int_{a+h}^{a+2h} f(x)\,dx - hf(x_2)\right. \\
&\qquad\qquad \left. + \ldots + \int_{a+(n-1)h}^{a+nh} f(x)\,dx - hf(x_n)\right| \\
&= \left|\sum_{j=1}^n (E_1^M)_{[a+(j-1)h,a+jh]}\right| \\
&\leqslant \sum_{j=1}^n \left|(E_1^M)_{[a+(j-1)h,a+jh]}\right| \\
&\leqslant \sum_{j=1}^n \frac{h^3}{24}\left(\max_{x\in[a+(j-1)h,a+jh]} |f''(x)|\right) \quad \left[\text{ from (3.6.2)}\right] \\
&\leqslant \frac{h^3 n}{24}\max_{x\in[a,b]} |f''(x)|,
\end{aligned}$$

since $[a+(j-1)h, a+jh] \subset [a,b]$ for every $j = 1,\ldots,n$. Substituting $n = (b-a)/h$ thus yields the desired error estimate (3.6.1).

2. (a) Derive the two-point Gaussian quadrature formula for

$$I(f) = \int_0^1 f(x) \ln\left(\frac{1}{x}\right) dx,$$

in which the weight function is $w(x) = \ln\left(\frac{1}{x}\right)$.

Solution: Two-point Gaussian quadrature formula means we want to approximate $I(f)$ as

$$I(f) \approx w_1 f(x_1) + w_2 f(x_2).$$

Define the error function $E_2(f) = I(f) - w_1 f(x_1) - w_2 f(x_2)$. Since we have four parameters w_1, w_2, x_1, x_2, we impose $E_2(1) = E_2(x) = E_2(x^2) = E_2(x^3) = 0$. This yields

$$E_n(x^j) = \int_0^1 x^j \ln\left(\frac{1}{x}\right) dx - w_1 x_1^j - w_2 x_2^j = 0 \quad \text{for every } j = 0, 1, 2, 3. \quad (3.6.3)$$

For any $n \geqslant 0$, consider the following integral

$$G_n = \int_0^1 x^n \ln\left(\frac{1}{x}\right) dx = \lim_{b \to 0^+} \int_b^1 x^n \ln\left(\frac{1}{x}\right) dx.$$

Chain rule gives

$$\frac{d}{dx}\left(\ln\left(\frac{1}{x}\right)\right) = \left(\frac{1}{1/x}\right) \frac{d}{dx}\left(\frac{1}{x}\right) = x\left(-\frac{1}{x^2}\right) = -\frac{1}{x}.$$

Integrating by parts gives

$$\begin{aligned}
\int_b^1 x^n \ln\left(\frac{1}{x}\right) dx &= \left[\frac{x^{n+1}}{n+1} \ln\left(\frac{1}{x}\right)\right]\Bigg|_b^1 - \int_b^1 \left(\frac{x^{n+1}}{n+1}\right)\left(-\frac{1}{x}\right) dx \\
&= -\left[\frac{b^{n+1}}{n+1} \ln\left(\frac{1}{b}\right)\right] + \frac{1}{n+1} \int_b^1 x^n \, dx \\
&= -\left[\frac{b^{n+1}}{n+1} \ln\left(\frac{1}{b}\right)\right] + \left[\frac{1 - b^{n+1}}{(n+1)^2}\right]
\end{aligned}$$

Using L'Hôpital rule,

$$\begin{aligned}
\lim_{b \to 0^+} b^{n+1} \ln\left(\frac{1}{b}\right) = \lim_{b \to 0^+} \frac{\ln\left(\frac{1}{b}\right)}{\frac{1}{b^{n+1}}} &= \lim_{b \to 0^+} \frac{\left(-\frac{1}{b}\right)}{\left(-\frac{n+1}{b^{n+2}}\right)} \\
&= \lim_{b \to 0^+} \left(\frac{1}{b}\right)\left(\frac{b^{n+2}}{n+1}\right) = 0, \quad \text{since } n \geqslant 0.
\end{aligned}$$

On the other hand,

$$\lim_{b\to 0^+} \frac{1-b^{n+1}}{(n+1)^2} = \frac{1}{(n+1)^2}.$$

Thus,

$$G_n = \int_0^1 x^n \ln\left(\frac{1}{x}\right)\, dx = \frac{1}{(n+1)^2}, \quad n \geqslant 0. \tag{3.6.4}$$

Now, using the identity (3.6.4) to expand (3.6.3) for each $j = 0, 1, 2, 3$, we obtain four nonlinear equations

$$w_1 + w_2 = 1 \tag{3.6.5a}$$

$$w_1x_1 + w_2x_2 = \frac{1}{4} \tag{3.6.5b}$$

$$w_1x_1^2 + w_2x_2^2 = \frac{1}{9} \tag{3.6.5c}$$

$$w_1x_1^3 + w_2x_2^3 = \frac{1}{16} \tag{3.6.5d}$$

We decouple the system of nonlinear equations into two equations involving w_1, x_1 and solve them using WolframAlpha. Let

$$w_2 = 1 - w_1, \quad x_2 = \frac{1}{w_2}\left(\frac{1}{4} - w_1x_1\right) = \frac{1}{1-w_1}\left(\frac{1}{4} - w_1x_1\right). \tag{3.6.6}$$

Substituting (3.6.6) into (3.6.5c) and (3.6.5d) yields

$$\begin{aligned} w_1x_1^2 + \frac{1}{1-w_1}\left(\frac{1}{4} - w_1x_1\right)^2 &= \frac{1}{9} \\ w_1x_1^3 + \frac{1}{(1-w_1)^2}\left(\frac{1}{4} - w_1x_1\right)^3 &= \frac{1}{16}, \end{aligned}$$

which has two pairs of solutions (w_1, x_1) given by

$$w_1 = \frac{1}{2} \mp \frac{9}{4\sqrt{106}}, \quad x_1 = \frac{5}{14} \pm \frac{\sqrt{106}}{42}.$$

Choosing the first set of solution $w_1 = \frac{1}{2} - \frac{9}{4\sqrt{106}}, x_1 = \frac{5}{14} + \frac{\sqrt{106}}{42}$, we obtain that

$$w_2 = 1 - w_1 = \frac{1}{2} + \frac{9}{4\sqrt{106}}, \quad x_2 = \frac{5}{14} - \frac{\sqrt{106}}{42}.$$

Hence,

$$I(f) \approx 0.2815 \ln\left(\frac{1}{0.6023}\right) + 0.7815 \ln\left(\frac{1}{0.1120}\right)$$

(b) Show that there is no set of nodes $x_1, x_2, \ldots, x_n$ and coefficients $\alpha_1, \alpha_2, \ldots, \alpha_n$ such that the quadrature rule

$$\sum_{j=1}^{n} \alpha_j f(x_j),$$

exactly equals to the integral $\int_a^b f(x)w(x)\,dx$ for all polynomials of degree less than or equal to $2n$. Here, $w(x)$ is the weight function.

Solution: Suppose, by contradiction, that there exists a set of nodes $x_1, x_2, \ldots, x_n$ and Gauss weight $\alpha_1, \alpha_2, \ldots, \alpha_n$ such that the Gauss quadrature is exact for all polynomials of degree less than or equal to $2n$, *i.e.*

$$\int_a^b P(x)w(x)\,dx = \sum_{j=1}^{n} \alpha_j P(x_j) \quad \text{for all } P(x) \text{ with } \deg(P) \leqslant 2n. \tag{3.6.7}$$

Recall that the weight function $w(x)$ satisfies the property that if $\int_a^b g(x)w(x)\,dx = 0$ for some nonegative, continuous function $g(x)$, then $g(x) \equiv \mathbf{0}$ on (a, b).

Consider the following polynomial

$$p(x) = \prod_{j=1}^{n}(x - x_j)^2 = (x - x_1)^2(x - x_2)^2 \ldots (x - x_n)^2.$$

On one hand,

$$\sum_{j=1}^{n} \alpha_j p(x_j) = 0, \tag{3.6.8}$$

since $\{x_0, x_1, \ldots, x_n\}$ are zeros of $p(x)$ by construction. On the other hand,

$$\int_a^b p(x)w(x)dx \neq 0, \tag{3.6.9}$$

since p is a nonnegative, continuous function and p is not identically equal to zero on (a, b); this follows from the assumption we impose on the weight function $w(x)$ (See above). Comparing (3.6.8) and (3.6.9), we see that this contradicts (3.6.7) since p is of degree $2n$.

Remark: The fact that $p(x)$ does not change sign is crucial here in order to use the assumption about $w(x)$. Otherwise, we can lower the degree of $p(x)$ and conclude that the Gauss quadrature with n nodes and n weight cannot be exact for all polynomials of degree less than $2n - 1$, contradicting Theorem 3.3.1.

3. Consider the following three methods

(a) the trapezoidal rule with n subdivisions;

(b) Simpson's rule with n subdivisions;

(c) the "corrected trapezoidal rule" with n subdivisions,

$$I_n = h\left(\frac{1}{2}f_0 + f_1 + \ldots + f_{n-1} + \frac{1}{2}f_n\right) - \frac{h^2}{12}[f'(b) - f'(a)].$$

Using these methods to compute the following integrals:

$$I = \int_0^1 e^{-x^2}\,dx, \quad I = \int_0^{2\pi} \frac{dx}{2+\cos(x)}.$$

Analyse empirically the rate of convergence of I_n to I by calculating

$$\frac{I_{2n} - I_n}{I_{4n} - I_{2n}}. \tag{3.6.10}$$

Solution: We compute the derivative of the integrand since this is required in the "corrected trapezoidal rule".

$$\frac{d}{dx}(e^{-x^2}) = -2xe^{-x^2}$$
$$\frac{d}{dx}\left(\frac{1}{2+\cos(x)}\right) = \frac{\sin(x)}{(2+\cos(x))^2}.$$

The remaining discussion concerning (3.6.10) is adopted from Atkinson's book. For simplicity, let us denote the ratio (3.6.10) by J_n. Suppose the numerical quadrature has an asymptotic error formula of the form

$$I - I_n = \frac{C}{n^p}, \quad \text{for some constant } C \text{ and } p > 0. \tag{3.6.11}$$

Substituting this into J_n yields

$$J_n = \frac{(I - I_n) - (I - I_n)}{(I - I_{2n}) - (I - I_{4n})} = \frac{\frac{1}{n^p} - \frac{1}{(2n)^p}}{\frac{1}{(2n)^p} - \frac{1}{(4n)^p}} = \frac{\frac{(4n)^p}{n^p} - \frac{(4n)^p}{(2n)^p}}{\frac{(4n)^p}{(2n)^p} - \frac{(4n)^p}{(4n)^p}}$$
$$= \frac{4^p - 2^p}{2^p - 1}$$
$$= \frac{2^p(2^p - 1)}{2^p - 1} = 2^p.$$

If the asymptotic error formula (3.6.11) is actually valid, then the rate of convergence of the numerical quadrature I_n is directly proportional to J_n since p increases as J_n increases, *i.e.* large magnitude of J_n corresponds to fast convergence of I_n.

We present, in tables, the numerical results J_n and the number of steps required to converge to the numerical solution for each quadrature rule. We measure the difference between succesive numerical solutions and say that the quadrature

rule converges if the difference is less than 10^{-16} in absolute value. Note that for a given n, I_{4n} has to be evaluated for trapezoidal and corrected trapezoidal rule, while I_{8n} has to evaluated for Simpson's rule since Simpson's rule can only be applied for even number of subintervals.

n	Trapezoidal	Corrected trapezoidal	$n/2$	Simpson
10	4.001249076550956	15.993888726363565	10	15.992355020559598
20	4.000312442895004	15.998500426033873	20	15.998120214869884
30	4.000138877400072	15.999262560132889	30	15.999315115814516
40	4.000078121145502	15.999468365365903	40	15.999117119340783
50	4.000049998387922	15.999611778748653	50	15.998586261240382
60	4.000034723577809	16.002523613814983	60	15.995723620987892
70	4.000025510370252	16.000756867459152	70	16.001157613535174
80	4.000019531834193	16.001139298192314	80	16.013529948312556

Table 3.1: J_n for the first integral, with integrand e^{-x^2}.

n	Trapezoidal	Corrected trapezoidal	$n/2$	Simpson
10	5.2418663×10^5	5.2418663×10^5	10	5.2422671×10^5
20	2.97295×10^4	2.97295×10^4	20	∞
30	2	2	30	-2
40	1	1	40	0
50	-3	-3	50	-0.6
60	-0.2	-0.2	60	0.6666667

Table 3.2: J_n for the second integral, with integrand $(2+\cos(x))^{-1}$.

	Trapezoidal	Corrected trapezoidal	Simpson
1st integral	23844	356	307
Numerical solution	0.7468241327	0.7468241328	0.7468241328
2nd integral	33	33	28
Numerical solution	3.6275987285	3.6275987285	3.6275987285

Table 3.3: Number of steps required for convergence and numerical solutions.

We see that for the first integral, the asymptotic error of trapezoidal rule behaves like $\mathcal{O}(h^2)$, while corrected trapezoidal and Simpson's rule both behave like $\mathcal{O}(h^4)$. For the second integral, all three quadrature rules converges rapidly, with error behaving like $\mathcal{O}(\log_2(5.24 \times 10^5))$.

(d) Derive the "corrected trapezoidal rule" formula given in part (c).

Solution: Assume $f \in C^2[a,b]$ and denote $f_j = f(x_j), j = 0, 1, \ldots, n$. The **Mean Value Theorem** asserts that for every $j = 1, \ldots, n$, there exists an $\xi_j \in (x_{j-1}, x_j)$ such that

$$\frac{f'(x_j) - f'(x_{j-1})}{x_j - x_{j-1}} = f''(\xi_j) \implies hf''(\xi_j) = f'(x_j) - f'(x_{j-1}).$$

Hence,

$$\begin{aligned}
I(f) &= \underbrace{\sum_{j=1}^{n} \left(\frac{h}{2}[f_{j-1} + f_j] \right)}_{\text{Composite trapezoidal rule}} - \underbrace{\sum_{j=1}^{n} \frac{h^3}{12} f''(\eta_j)}_{\text{Sum of local errors}} \quad \text{for some } \eta_j \in [x_{j-1}, x_j] \\
&= \sum_{j=1}^{n} \left(\frac{h}{2}[f_{j-1} + f_j] \right) - \sum_{j=1}^{n} \frac{h^3}{12}[f''(\eta_j) - f''(\xi_j) + f''(\xi_j)] \\
&= \sum_{j=1}^{n} \left(\frac{h}{2}[f_{j-1} + f_j] \right) - \sum_{j=1}^{n} \frac{h^2}{12}[hf''(\xi_j)] - \sum_{j=1}^{n} \frac{h^3}{12}[f''(\eta_j) - f''(\xi_j)] \\
&= \sum_{j=1}^{n} \left(\frac{h}{2}[f_{j-1} + f_j] \right) - \underbrace{\sum_{j=1}^{n} \frac{h^2}{12}[f'(x_j) - f'(x_{j-1})]}_{\text{Telescopic sum}} - \sum_{j=1}^{n} \frac{h^3}{12}[f''(\eta_j) - f''(\xi_j)] \\
&= \sum_{j=1}^{n} \left(\frac{h}{2}[f_{j-1} + f_j] \right) - \frac{h^2}{12}[f'(x_n) - f'(x_0)] - \sum_{j=1}^{n} \frac{h^3}{12}[f''(\eta_j) - f''(\xi_j)] \\
&= \underbrace{h\left(\frac{1}{2}f_0 + f_1 + \ldots + f_{n-1} + \frac{1}{2}f_n \right) - \frac{h^2}{12}[f'(b) - f'(a)]}_{\text{Corrected trapezoidal rule}} - \sum_{j=1}^{n} \frac{h^3}{12}[f''(\eta_j) - f''(\xi_j)]
\end{aligned}$$

Truncating the second sum yields the "corrected trapezoidal rule".

Chapter 4

Numerical Methods for ODEs

We are interested in numerically solving general **initial value problems (IVP)**, having the form

$$\begin{cases} y' & = f(x,y), \\ y(x_0) & = Y_0. \end{cases} \tag{IVP}$$

The function $f(x,y)$ is assumed to be continuous in some domain $\mathcal{D}$ of the xy-plane and $(x_0, Y_0) \in \mathcal{D}$.

Definition 4.0.1. We say that a function $Y(x)$ is a solution on $\mathcal{D}$ of (IVP) if for all $x \in \mathcal{D}$,

1. $(x, Y(x)) \in \mathcal{D}$,
2. $Y(x_0) = Y_0$,
3. $Y'(x)$ exists and $Y'(x) = f(x, Y(x))$.

Example 4.0.2. Consider the first-order ODE of the form

$$y' = \lambda y + g(x), \quad y(0) = Y_0,$$

where $g \in C[0,\infty)$. Using the method of integrating factors, one can show that the solution has the form

$$Y(x) = Y_0 e^{\lambda x} + \int_0^x e^{\lambda(x-t)} g(t)\, dt, \quad x \in [0,\infty).$$

4.1 Existence, Uniqueness and Stability Theory

One should always ensure that the problem is **well-posed** before even attempting to solve (IVP) numerically, *i.e.* a unique solution exists and the solution is stable with respect to small perturbation of initial data.

Theorem 4.1.1 (Local Existence). *Assume $f \in C(\mathcal{D}), (x_0, Y_0) \in \mathcal{D}$, and $f(x,y)$ satisfies the Lipschitz condition, i.e. there exists an $K \geqslant 0$ such that*

$$|f(x,y_1) - f(x,y_2)| \leqslant K|y_1 - y_2| \quad \text{for all } (x,y_1),(x,y_2) \in \mathcal{D}.$$

Then for a suitably chosen interval $I = [x_0 - \alpha, x_0 + \alpha]$, there is a unique solution $Y(x)$ on I of (IVP).

Remark 4.1.2. If $\dfrac{\partial f(x,y)}{\partial y}$ exists and is bounded on $\mathcal{D}$, it follows from the mean value theorem that $f(x,y)$ satisfies the Lipschitz condition with Lipschitz constant

$$K = \max_{(x,y)\in\mathcal{D}} \left| \frac{\partial f(x,y)}{\partial y} \right|.$$

Note that this is a stronger assumption since Lipschitz functions might not be differentiable (although Lipschitz functions are differentiable almost everywhere by Rademacher's theorem).

Example 4.1.3. Consider the initial value problem $y' = 1 + \sin(xy)$ on

$$\mathcal{D} = \{(x,y) \in \mathbb{R}^2 \colon x \in [0,1], y \in \mathbb{R}\}.$$

Computing the partial derivative of f with respect to y gives

$$\frac{\partial f(x,y)}{\partial y} = x\cos(xy) \implies K = \max_{(x,y)\in\mathcal{D}} \left| \frac{\partial f(x,y)}{\partial y} \right| = 1.$$

Thus for any initial data (x_0, Y_0) with $x_0 \in (0,1)$, there exists a unique solution $Y(x)$ on some interval $[x_0 - \alpha, x_0 + \alpha] \subset [0,1]$.

We now turn to stability of (IVP), in which we want to see how the solution changes when we perturbs the initial data or the function $f(x,y)$ (with respect to x).

Theorem 4.1.4 (Stability). *Consider the perturbed problem of the original problem* (IVP)

$$\begin{cases} y' & = f(x,y) + \delta(x), \\ y(x_0) & = Y_0 + \varepsilon. \end{cases} \tag{IVP$_\varepsilon$}$$

Assume that

1. *$f(x,y)$ satisfies the condition of Theorem 4.1.1,*
2. *$\delta(x)$ is continuous for all x such that $(x,y) \in \mathcal{D}$.*

Then there exists a unique solution $Y(x;\delta,\varepsilon)$ of (IVP$_\varepsilon$) *on an interval $[x_0 - \alpha, x_0 + \alpha]$ for some $\alpha > 0$, uniformly for all perturbations ε and $\delta(x)$ satisfying*

$$|\varepsilon| \leqslant \varepsilon_0, \quad \|\delta\|_\infty \leqslant \varepsilon_0,$$

for some sufficiently small $\varepsilon_0 > 0$. Moreover, if $Y(x)$ is the solution of the unperturbed problem (IVP)*, then we have the following stability estimate:*

$$\max_{|x-x_0|\leqslant\alpha} |Y(x) - Y(x;\delta,\varepsilon)| \leqslant C[|\varepsilon| + \alpha\|\delta\|_\infty], \tag{4.1.1}$$

where $C = \dfrac{1}{1-\alpha K}$*, K the Lipschitz constant of $f(x,y)$.*

Remark 4.1.5. We say that (IVP) is well-posed or stable in the sense of Theorem 4.1.4, *i.e.* the solution $Y(x)$ depends continuously on the data of the problem, namely the function f and initial condition Y_0. Note that it is possible for (IVP) to be stable but **ill-conditioned** with respect to numerical computation, since the constant C can be really large!

For the sake of illustration, we consider only perturbations ε in the initial condition Y_0. Let $Y(x;\varepsilon)$ be the solution to this particular perturbed problem. It satisfies

$$\begin{cases} Y'(x;\varepsilon) = f(x, Y(x;\varepsilon)) \quad \text{on } x \in [x_0-\alpha, x_0+\alpha], \\ Y(x_0;\varepsilon) = Y_0 + \varepsilon. \end{cases} \tag{4.1.2}$$

Denote $Z(x;\varepsilon) = Y(x;\varepsilon) - Y(x)$, subtracting (IVP) from (4.1.2) gives

$$\begin{cases} Z'(x;\varepsilon) &= f(x, Y(x;\varepsilon)) - f(x, Y(x)), \\ Z(x_0;\varepsilon) &= \varepsilon. \end{cases} \tag{4.1.3}$$

If $Y(x;\varepsilon)$ is sufficiently close to $Y(x)$ for small values of ε, then we can approximate the RHS of (4.1.3) by its first order Taylor expansion of f with respect to the second variable, which gives

$$f(x, Y(x;\varepsilon)) - f(x, Y(x)) \approx \frac{\partial f(x, Y(x))}{\partial y} Z(x;\varepsilon).$$

Thus, (4.1.3) reduces to a separable ODE and it has an explicit solution of the form

$$Z(x;\varepsilon) \approx \varepsilon \exp\left(\int_{x_0}^{x} \frac{\partial f(t, Y(t))}{\partial y}\, dt\right).$$

Clearly, the behaviour of the perturbed problem depends on the sign and magnitude of $\dfrac{\partial f(t, Y(t))}{\partial y}$. For a problem to be **well-conditioned**, we require the integral

$$\int_{x_0}^{x} \frac{\partial f(t, Y(t))}{\partial y}\, dt \quad \text{on } |x - x_0| \leqslant \alpha,$$

to be bounded from above by zero or a small positive number, as x increases. Consequently, the perturbation $Z(x;\varepsilon)$ will be bounded by some small constant times ε.

Example 4.1.6. The initial value problem

$$\begin{cases} y' &= 100y - 101e^{-x}, \\ y(0) &= 1, \end{cases}$$

has solution $Y(x) = e^{-x}$. The perturbed problem

$$\begin{cases} y' & = 100y - 101e^{-x}, \\ y(0) & = 1 + \varepsilon, \end{cases}$$

has solution $Y(x;\varepsilon) = e^{-x} + \varepsilon e^{100x}$. It is clear that $Y(x;\varepsilon)$ diverges rapidly from $Y(x)$ as x increases. Such problem is said to be ill-conditioned.

In the special case where

$$\frac{\partial f(t, Y(t))}{\partial y} \leqslant 0 \quad \text{on } |t - x_0| \leqslant \alpha,$$

the perturbation $Z(x;\varepsilon)$ probably remains bounded by ε as x increases; the unperturbed problem (IVP) is said to be well-conditioned. If, in addition, the partial derivative has large magnitude, we see that $Z(x;\varepsilon) \longrightarrow 0$ rapidly as x increases. Unfortunately, **numerical methods might fail to capture the rapid decaying behaviour of $Z(x;\varepsilon)$**, it is as if we are solving the unperturbed problem since $Z(x;\varepsilon)$ is almost negligible. Such problems are still said to be well-conditioned, but it can be a challenging task for many numerical methods; they are known as **stiff DEs**.

4.2 Euler's Method

In practice, we construct simplified models to obtain qualitative approximations of real-world models that are difficult to solve analytically, but in many cases even these simplified models are difficult or impossible to solve either explicitly or implicitly. Thus, it is important in having numerical methods to numerically approximate the true solution.

We begin with Euler's method which is a first-order numerical scheme for solving (IVP), and it often serves as the basis for constructing complex numerical methods. Consider solving the initial value problem (IVP) on the finite interval $[a, b]$. For simplicity, suppose we have a uniform spaced grid nodes

$$a = x_0 < x_1 < x_2 < \ldots < x_N = b,$$

where the nodes are given by $x_n = x_0 + jh, j = 0, 1, \ldots$. Denote $N(h)$ to be the largest index $N > 0$ such that

$$x_N \leqslant b \quad \text{and} \quad x_{N+1} > b.$$

As before, $Y(x_n)$ is the true solution at x_n and $y_h(x_n) = y_n$ the approximate solution at x_n.

4.2.1 Derivation

Euler's method is defined as follows

$$\boxed{y_{n+1} = y_n + hf(x_n, y_n), \quad n = 0, 1, 2, \ldots, \text{ with } y_0 = Y_0.} \tag{Euler}$$

Below are four possible derivations of Euler's method.

1. **Finite difference approximation**. Using definition of a derivative,

$$Y'(x) = \lim_{h\to 0} \frac{Y(x+h) - Y(x)}{h} \approx \frac{Y(x+h) - Y(x)}{h}.$$

Since $Y'(x_n) = f(x_n, Y(x_n))$, we have

$$\frac{Y(x_{n+1}) - Y(x_n)}{h} \approx f(x_n, Y(x_n)),$$

and rearranging gives $Y(x_{n+1}) \approx Y(x_n) + hf(x_n, Y(x_n))$.

2. **Tangent line approximation**. Geometrically, we approximate $Y(x_{n+1})$ by extending tangent line at x_n. In point-slope form, we have

$$\frac{Y(x_{n+1}) - Y(x_n)}{h} \approx Y'(x_n) = f(x_n, Y(x_n)).$$

Iterating this over each interval $[x_0, x_1], [x_1, x_2], \ldots$ gives (Euler)

3. **Taylor series expansion**. Expanding $Y(x_{n+1})$ about x_n gives

$$Y(x_{n+1}) = Y(x_n) + hY'(x_n) + \frac{h^2}{2}Y''(\xi_n) \quad \text{for some } \xi_n \in [x_n, x_{n+1}].$$

(Euler) is obtained by dropping the error term $T_n = \dfrac{h^2}{2}Y''(\xi_n)$ which is also called the **local truncation error** at x_{n+1}.

4. **Numerical integration**. Integrating the ODE over $[x_n, x_{n+1}]$ gives

$$\int_{x_n}^{x_{n+1}} Y'(t)\,dt = \int_{x_n}^{x_{n+1}} f(t, Y(t))\,dt$$
$$\implies Y(x_{n+1}) = Y(x_n) + \int_{x_n}^{x_{n+1}} f(t, Y(t))\,dt.$$

Different choices of quadrature rules on the RHS yields different numerical method, and (Euler) is obtained by approximating the integral using the **left-hand rectangular rule**, *i.e.*

$$\int_{x_n}^{x_{n+1}} f(t, Y(t))\,dt = (x_{n+1} - x_n) f(x_n, Y(x_n)) = hf(x_n, Y(x_n)).$$

Remark 4.2.1. One obtain the **midpoint method** if we use the simple midpoint rule (Simple Midpoint).

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n), \quad n \geqslant 1.$$

Example 4.2.2. Consider the initial value problem

$$\begin{cases} y'(x) &= -y(x), \\ y(0) &= 1. \end{cases}$$

Its unique solution is $Y(x) = e^{-x}$. Applying (Euler) to the IVP with the choice of step size $h = 0.1$, we get

$$y_0 = 1,\ y_1 = y_0 - hy_0 = 0.9$$
$$y_2 = y_1 - hy_1 = 0.81.$$

We can compute the error explicitly since we have an explicit solution.

$$Y(x_1) - y_1 = e^{-0.1} - 0.9 \approx 0.004837$$
$$Y(x_2) - y_2 = e^{-0.2} - 0.81 \approx 0.001873.$$

4.2.2 Convergence Analysis

Despite its simplicity, does Euler's method in any meaningful sense provide approximations to the true solution of (IVP)? Intuitively, the local truncation error in Euler's method (error at any given steps) is of order $\mathcal{O}(h^2)$, but since we need to take $\mathcal{O}(1/h)$ steps to reach some finite x, we expect the global error (error at a given time step) in Euler's method is of order $\mathcal{O}(h)$.

Remark 4.2.3. The **local truncation error** is defined to be the error at any given steps, assuming there is no error in the previous step. The **global (truncation) error** is defined to be the error at any given time step; it consists of error propagated from all the previous steps along with error generated in the current step. We can think of global error as cumulative error produced by approximate solutions.

Example 4.2.4. Consider the initial value problem

$$\begin{cases} y' & = 2x, \\ y(0) & = 0. \end{cases}$$

Its unique solution is $Y(x) = x^2$. The global error of Euler's method can be analysed directly here. Since $y_0 = 0$ and

$$y_{n+1} = y_n + 2hx_n, \quad x_n = nh,$$

we have

$$\begin{aligned} y_1 &= 0 + 2h(0) &&= 0 \\ y_2 &= 0 + 2h(x_1) &&= x_1x_2 \\ y_3 &= x_1x_2 + 2h(x_2) &&= x_2x_3 \\ y_4 &= x_2x_3 + 2h(x_3) &&= x_3x_4. \end{aligned}$$

We now show that $y_n = x_{n-1}x_n, n \geqslant 1$ using method of induction. The base case $n = 1$ is trivial. Suppose $y_n = x_{n-1}x_n$ holds, then

$$y_{n+1} = x_{n-1}x_n + 2h(x_n) = x_n(x_{n-1} + 2h) = x_nx_{n+1}.$$

Thus,

$$Y(x_n) - y_n = x_n^2 - x_nx_{n-1} = (x_n)(x_n - x_{n-1}) = hx_n,$$

and we see that the global error at each fixed value x_n is proportional to h. This is not surprising, since the global error is the sum of local truncation error and n is proportional to h itself.

The standard approach behind almost every convergence proof of a numerical scheme is to write an equation for the error and then use a Gronwall-type argument to estimate the magnitude of the solution. We state a relatively simple but extremely useful lemma in the analysis of finite difference methods.

Lemma 4.2.5. *For any $x \in \mathbb{R}$,*

$$e^x \geqslant 1 + x,$$

and for any $x \geqslant -1$,

$$0 \leqslant (1+x)^m \leqslant e^{mx}.$$

Proof. The first inequality can be easily seen by applying Taylor's theorem on e^x. Indeed,

$$e^x = 1 + x + \frac{x^2}{2} e^{\xi},$$

for some ξ in between 0 and x. ■

Unless stated otherwise, we will now assume that the function $f(x, y)$ satisfies the **stronger Lipschitz condition**, *i.e.* the following holds for any $y_1, y_2 \in \mathbb{R}, x \in [x_0, b]$:

$$|f(x, y_1) - f(x, y_2)| \leqslant K|y_1 - y_2|. \tag{4.2.1}$$

This will simplify the remaining discussion. If the function f satisfies the *Lipschitz condition*, f can be modified in such a way that the intrinsic property of (IVP) and its true solution $Y(x)$ remain unchanged, but f now satisfies the *stronger Lipschitz condition*.

Theorem 4.2.6. *Assume that the true solution $Y(x)$ of (IVP) has a bounded second derivative on $[x_0, b]$. Then the approximate solution $\{y_h(x_n) \colon x_n \in [x_0, b]\}$ obtained by Euler's method satisfies the following pointwise error estimate:*

$$\max_{x_n \in [x_0, b]} |Y(x_n) - y_h(x_n)| \leqslant e^{(b-x_0)K}|e_0| + \underbrace{\left[\frac{e^{(b-x_0)K} - 1}{K}\right] \tau(h)}_{\text{error generated by Euler's method}}, \tag{4.2.2}$$

where

$$\tau(h) = \frac{h}{2}\|Y''\|_\infty \quad \text{and} \quad e_0 = Y_0 - y_h(x_0).$$

If in addition,

$$|e_0| \leqslant C_1 h \quad \text{as} \quad h \longrightarrow 0,$$

for some constant $C_1 \geqslant 0$, then there exists a constant $B \geqslant 0$ for which

$$\max_{x_n \in [x_0, b]} |Y(x_n) - y_h(x_n)| \leqslant Bh. \tag{4.2.3}$$

Proof. Define $e_n := Y(x_n) - y_h(x_n), n \geqslant 0$ and

$$\tau_n = \frac{h}{2}Y''(\xi_n), \quad 0 \leqslant n \leqslant N(h) - 1,$$

Clearly,

$$\max_{0\leqslant n\leqslant N-1} |\tau_n| \leqslant \tau(h) := \frac{h}{2}\|Y''\|_\infty.$$

Denote $Y_n \equiv Y(x_n)$. To derive the error equation, we first expand Y_{n+1} about point x_n which yields

$$Y_{n+1} = Y_n + hf(x_n, Y_n) + \frac{h^2}{2}Y''(\xi_n) = Y_n + hf(x_n, Y_n) + h\tau_n. \tag{4.2.4}$$

Subtracting the Euler's method from (4.2.4), we have that

$$e_{n+1} = e_n + h[f(x_n, Y_n) - f(x_n, y_n)] + h\tau_n.$$

Using the strong Lipschitz condition of $f(x, y)$, for every $0 \leqslant n \leqslant N(h) - 1$ we have

$$\begin{aligned} |e_{n+1}| &\leqslant |e_n| + h|f(x_n, Y_n - f(x_n, y_n)| + h|\tau_n| \\ &\leqslant |e_n| + hK|Y_n - y_n| + h|\tau_n| \\ &\leqslant (1 + hK)|e_n| + h\tau(h). \end{aligned}$$

Iterating this inequality gives,

$$\begin{aligned} |e_n| &\leqslant (1 + hK)[(1 + hK)|e_{n-1}| + h\tau(h)] + h\tau(h) \\ &\leqslant \cdots\cdots \\ &\leqslant (1 + hK)^n|e_0| + \underbrace{\left[1 + (1 + hK) + (1 + hK)^2 + \ldots + (1 + hK)^{n-1}\right]}_{\text{geometric series}} h\tau(h). \end{aligned}$$

Summing the geometric series and using Lemma 4.2.5, we have

$$\begin{aligned} |e_n| &\leqslant (1 + hK)^n|e_0| + \left[\frac{(1 + hK)^n - 1}{hK}\right] h\tau(h) \\ &\leqslant e^{nhK}|e_0| + \left(\frac{e^{nhK} - 1}{K}\right)\tau(h) \\ &= e^{(x_n - x_0)K}|e_0| + \left(\frac{e^{(x_n - x_0)K} - 1}{K}\right)\tau(h) \\ &\leqslant e^{(b - x_0)K}|e_0| + \left(\frac{e^{(b - x_0)K} - 1}{K}\right)\tau(h). \end{aligned}$$

Taking the maximum over all $x_n \in [x_0, b]$ yields the error estimate (4.2.2). To obtain (4.2.3), simply set

$$B = C_1 e^{(b - x_0)K} + \left(\frac{e^{(b - x_0)K} - 1}{K}\right)\frac{\|Y''\|_\infty}{2}.$$

■

4.2.3 Stability Analysis

Following a similar idea from the stability analysis of (IVP), we consider the following perturbed numerical scheme

$$\begin{cases} z_{n+1} &= z_n + h[f(x_n, z_n) + \delta(x_n)] \\ z_0 &= y_0 + \varepsilon. \end{cases} \tag{4.2.5}$$

for $0 \leqslant n \leqslant N(h) - 1$. We want to compare these numerical solutions $\{y_n\}, \{z_n\}$ as $h \longrightarrow 0$. Let $e_n := z_n - y_n$, then $e_0 = \varepsilon$ and

$$e_{n+1} = e_n + h[f(x_n, z_n) - f(x_n, y_n)] + h\delta(x_n).$$

It follows from Theorem 4.2.6 that

$$\begin{aligned} \max_{0 \leqslant n \leqslant N(h)} |z_n - y_n| &\leqslant e^{(b-x_0)K}|\varepsilon| + \left(\frac{e^{(b-x_0)K} - 1}{K}\right) \|\delta\|_\infty \\ &\leqslant K_1|\varepsilon| + K_2\|\delta\|_\infty, \end{aligned} \tag{4.2.6}$$

where $K_1, K_2 > 0$ are constants independent of the step size h. Observe that (4.2.6) is analogous to the stability estimate (4.1.1) for the continuous IVP, and it says that Euler's method is a stable approximation scheme for the true solution of (IVP). Note that the imitation of such stability estimate is something we want from all numerical methods.

4.3 Taylor's Method

Euler's method belongs to the more general class of single-step methods, where the numerical scheme only requires knowledge of the numerical solution y_n to determine y_{n+1}. To improve the order of convergence, we seek for approximations that are more accurate and one choice is simply extending Euler's method by considering higher-order Taylor expansions. We present the mechanism behind Taylor's method with the following example.

Example 4.3.1. Consider the initial value problem

$$\begin{cases} y'(x) &= -y(x) + 2\cos(x), \\ y(0) &= 1. \end{cases}$$

Its unique solution is given by $Y(x) = \sin(x) + \cos(x) \in C^\infty$. Consider the second-order Taylor expansion of $Y(x_{n+1})$ around x_n:

$$Y(x_{n+1}) = Y(x_n) + hY'(x_n) + \frac{h^2}{2}Y''(x_n) + \frac{h^3}{3!}Y^{(3)}(\xi_n), \quad \xi_n \in [x_n, x_{n+1}]. \tag{4.3.1}$$

Computing $Y'(x_n)$ and $Y''(x_n)$ gives

$$\begin{aligned} Y'(x_n) &= -Y(x_n) + 2\cos(x_n) \\ Y''(x_n) &= -Y'(x_n) - 2\sin(x_n) = Y(x_n) - 2\cos(x_n) - 2\sin(x_n). \end{aligned}$$

Setting $Y(x_n) = y_n$, substituting these expressions into (4.3.1) and dropping the truncation error, we obtain

$$y_{n+1} = y_n + h[-y_n + 2\cos(x_n)] + \frac{h^2}{2}[y_n - 2\cos(x_n) - 2\sin(x_n)].$$

Since the local truncation error is of order h^3, we expect this numerical method to be more accurate than Euler's method.

To solve (IVP) on the interval $[x_0, b]$ using Taylor's method, we choose an order $p \geqslant 1$ where p is such that the true solution $Y(x) \in C^{(p+1)}[x_0, b]$ and approximate $Y(x_{n+1})$ with its pth-order Taylor expansion around x_n:

$$Y(x_{n+1}) \approx Y(x_n) + hY'(x_n) + \ldots + \frac{h^p}{p!}Y^{(p)}(x_n), \tag{Taylor}$$

with the local truncation error

$$T_n = \frac{h^{p+1}}{(p+1)!}Y^{(p+1)}(\xi_n), \quad \text{for some } \xi_n \in [x_n, x_{n+1}].$$

The remaining task is to find higher order derivatives of Y and express them in terms of higher order derivatives of $f(x_n, Y(x_n))$.

Remark 4.3.2. If the solution $Y(x)$ and $f(x, Y(x))$ are sufficiently differentiable, then we can show that Taylor's method satisfies the error estimate

$$\max_{x_0 \leqslant x_n \leqslant b} |Y(x_n) - y_h(x_n)| \leqslant Ch^p \max_{x_0 \leqslant x \leqslant b} |Y^{(p+1)}(x)| \tag{4.3.2}$$

Taylor's method can be viewed as a simple and accurate numerical method, but it requires computing derivatives of $f(x, y)$ which can be very difficult and time-consuming, not to mention the regularity assumption of $Y(x)$ and $f(x, y)$.

4.4 Runge-Kutta Method

Using symbolic manipulation on a computer, the Taylor's method can be easily produced. Nonetheless, the derivatives are still likely to be quite time-consuming to evaluate. We would like to develop higher order numerical methods that avoid the need to compute higher order derivatives, while mantaining the accuracy of Taylor's method. This is precisely the main idea behind the Runge-Kutta (RK) methods, where it evaluates $f(x, y)$ at more points to "recover" the desired accuracy. All RK methods can be written in the general form:

$$y_{n+1} = y_n + hF(x_n, y_n; h), \quad n \geqslant 0, \quad y_0 = Y_0. \tag{4.4.1}$$

Intuitively, $F(x_n, y_n; h)$ is interpreted as some kind of "average slope" on the interval $[x_n, x_{n+1}]$. As pointed out above, $F(x_n, y_n; h)$ is constructed so that (4.4.1) behaves like a higher order

Taylor's method.

To gain some insights on how to derive higher order RK methods, we first illustrate the derivation of a family of RK methods of order 2 (RK2). We suppose F has the general form

$$F(x, y; h) = \gamma_1 f(x, y) + \gamma_2 f(x + \alpha h, y + \beta h f(x, y)), \tag{4.4.2}$$

where $\gamma_1, \gamma_2, \alpha, \beta$ are chosen such that when we substitute $Y(x)$ into (4.4.1), the local truncation error

$$T_n(h) := Y(x_{n+1}) - [Y(x_n) + hF(x_n, Y(x_n); h)] = \mathcal{O}(h^3). \tag{4.4.3}$$

To achieve (4.4.3), we consider the third-order Taylor expansion of $Y(x_{n+1})$ around x_n and the second-order Taylor expansion of $f(x_n + \alpha h, y_n + \beta f(x_n, y_n))$ around (x_n, y_n). Denote $Y(x_n) = Y_n$. Computing these expansion yields

$$Y_{n+1} = Y_n + hY_n' + \frac{h^2}{2}Y_n'' + \frac{h^3}{6}Y_n^{(3)} + \mathcal{O}(h^4). \tag{4.4.4}$$

$$\begin{aligned}
&f(x_n + \alpha h, y_n + \beta f(x_n, y_n)) \\
&= f + [\alpha h f_x + \beta h f f_y] + \frac{1}{2}\left[(\alpha h)^2 f_{xx} + (\alpha h)(\beta h f) f_{xy} + (\beta h f)(\alpha h) f_{yx} + (\beta h f)^2 f_{yy}\right] + \mathcal{O}(h^3) \\
&= f + h[\alpha f_x + \beta f f_y] + h^2\left(\frac{1}{2}\alpha^2 f_{xx} + \alpha\beta f f_{xy} + \frac{1}{2}\beta^2 f^2 f_{yy}\right) + \mathcal{O}(h^3).
\end{aligned} \tag{4.4.5}$$

Using $Y_n' = f(x_n, Y_n)$, we have

$$Y_n'' = f_x + f_y Y_n' = f_x + f_y f \tag{4.4.6}$$

$$\begin{aligned}
Y_n^{(3)} &= f_{xx} + f_{xy}Y_n' + (f_y)' f + f_y f' \\
&= f_{xx} + f_{xy} f + [f_{yx} + f_{yy}Y_n']f + f_y[f_x + f_y Y_n'] \\
&= f_{xx} + f_{xy} f + [f_{yx} + f_{yy} f]f + f_y[f_x + f_y f] \\
&= f_{xx} + 2f_{xy} f + f_{yy} f^2 + f_y f_x + f_y^2 f.
\end{aligned} \tag{4.4.7}$$

Substituting (4.4.4), (4.4.5), (4.4.6), (4.4.7) into (4.4.3) and collecting common powers of h gives:

$$\begin{aligned}
T_n(h) &= hY_n' + \frac{h^2}{2}Y_n'' + \frac{h^3}{6}Y_n''' + \mathcal{O}(h^4) - h\Big[\gamma_1 f(x_n, Y_n) + \gamma_2 f(x_n + \alpha h, Y_n + \beta h f(x_n, Y_n))\Big] \\
&= h\Big[1 - \gamma_1 - \gamma_2\Big] f + h^2\left[\left(\frac{1}{2} - \gamma_2\alpha\right) f_x + \left(\frac{1}{2} - \gamma_2\beta\right) f_y f\right] \\
&\qquad + h^3\left[\left(\frac{1}{6} - \frac{1}{2}\gamma_2\alpha^2\right) f_{xx} + \left(\frac{1}{3} - \gamma_2\alpha\beta\right) f_{xy} f\right) \\
&\qquad\quad + \left(\frac{1}{6} - \frac{1}{2}\gamma_2\beta^2\right) f_{yy} f^2 + \frac{1}{6} f_y f_x + \frac{1}{6} f_y^2 f\Big] + \mathcal{O}(h^4),
\end{aligned}$$

where f and all its partial derivatives are evaluated at (x_n, Y_n). The coefficient of h^3 cannot be zero in general, if f is allowed to vary arbitrarily. Setting the coefficients of h and h^2 to be zero gives

$$\gamma_1 + \gamma_2 = 1, \quad \gamma_2\alpha = \frac{1}{2}, \quad \gamma_2\beta = \frac{1}{2}. \tag{4.4.8}$$

The nonlinear system (4.4.8) is underdetermined, and its general solution is given by

$$\gamma_1 = 1 - \gamma_2, \quad \alpha = \beta = \frac{1}{2\gamma_2}. \tag{4.4.9}$$

where γ_2 is a free parameter and $\gamma_2 \neq 0$, since otherwise we would recover Euler's method. We require $\alpha \in [0, 1]$ so that $x_n + \alpha h \in [x_n, x_{n+1}]$. Three typical choices are $\gamma_2 = \frac{1}{2}, \frac{3}{4}, 1$. With $\gamma_2 = \frac{1}{2}$,

$$F(x_n, y_n; h) = \frac{1}{2}\Big[f(x_n, y_n) + f(x_n + h, y_n + hf(x_n, y_n))\Big],$$

and we obtain one of the possible RK2 method:

$$\boxed{y_{n+1} = y_n + \frac{h}{2}\Big[f(x_n, y_n) + f(x_{n+1}, y_n + hf(x_n, y_n))\Big].} \tag{RK2}$$

(RK2) is sometimes called a **two-stage method**.

Higher-order RK methods can be generated in a similar fashion, but the algebra becomes very tedious. Let $p \geqslant 1$ be the number of evaluations of $f(x, y)$. We assume F has the general formula

$$F(x_n, y_n; h) = \sum_{j=1}^{p} \gamma_j v_j, \quad \text{where } v_1 = f(x_n, y_n)$$

$$v_j = f\left(x_n + \alpha_j h, y_n + h\sum_{i=1}^{j-1} \beta_{ji} v_i\right), \quad j = 2, \ldots, p.$$

A popular classical method is the fourth order RK method

$$\boxed{\begin{cases} v_1 & = f(x_n, y_n) \\ v_2 & = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}v_1\right) \\ v_3 & = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}v_2\right) \\ v_4 & = f(x_n + h, y_n + hv_3) \\ y_{n+1} & = y_n + \frac{h}{6}(v_1 + 2v_2 + 2v_3 + v_4). \end{cases}} \tag{RK4}$$

It can be shown that the local truncation error for (RK4) is of order $\mathcal{O}(h^5)$. If $y' = f(x)$, then (RK4) simplifies to

$$y_{n+1} = y_n + \frac{h}{6}\left[f(x_n) + 4f\left(x_n + \frac{h}{2}\right) + f(x_n + h)\right],$$

i.e. (RK4) reduces to (Simpson) rule for the integral on $[x_n, x_{n+1}]$.

Remark 4.4.1. If the true solution $Y(x)$ of (IVP) and $f(x, Y(x))$ are sufficiently differentiable, then the RK method has local truncation error of order $\mathcal{O}(h^{p+1})$ and

$$\max_{x_0 \leqslant x_n \leqslant b} |Y(x_n) - y_h(x_n)| \leqslant Ch^p.$$

There is a relationship between p and the maximum possible order for the local truncation error $T_n(h)$:

Number of function evaluations (stages)	1	2	3	4	5	6	7	8
Maximum order of method	1	2	3	4	4	5	6	6

4.5 Multistep Methods

All the numerical methods we encountered so far are single-step methods. RK methods only requires y_n to compute y_{n+1}, it does however invoke a series of intermediate values during the computation. Multistep methods use the previous p step values to approximate the solution at the next step and in the case of **linear multistep methods**, a linear combination of the previous points and derivative values is used.

For the sake of convenience, denote $f(x_n, y_n) = y'(x_n) = y'_n$. Consider solving (IVP) on the interval $[x_0, b]$ and assume that the problem is well-posed. The general form of linear multistep methods is:

$$y_{n+1} = \sum_{j=0}^{p} a_j y_{n-j} + h \sum_{j=-1}^{p} b_j f(x_{n-j}, y_{n-j}), \quad x_{p+1} \leqslant x_{n+1} \leqslant b, \quad n \geqslant p \geqslant 0, \tag{LMM}$$

where the coefficients $a_0, \ldots, a_p$ and $b_{-1}, b_0, \ldots, b_p$ are constants. This is the $(p+1)$-step method if $a_p \neq 0$ or $b_p \neq 0$ and $y_1, \ldots, y_p$ must be obtained separately, usually using lower-order methods. If $b_{-1} = 0$, we have an explicit method, such as the *forward Euler method*; if $b_{-1} \neq 0$, we have an implicit method, such the *backward Euler method*.

Denote $Y(x_n) = Y_n$. Integrating $Y' = f(x, Y)$ over the interval $[x_n, x_{n+1}]$ gives

$$\begin{aligned} \int_{x_n}^{x_{n+1}} Y'(x)\,dx &= \int_{x_n}^{x_{n+1}} f(x, Y(x))\,dx \\ \implies Y_{n+1} - Y_n &= \int_{x_n}^{x_{n+1}} f(x, Y(x))\,dx. \end{aligned}$$

Adams methods arise when we approximate the integral by replacing the integrand $g(x) := f(x, Y(x))$ with interpolating polynomials.

4.5.1 Adams-Bashforth (AB) Methods

Fix an integer $q \geqslant 0$ corresponding to the degree of interpolating polynomial, and consider the set of interpolation nodes $\{x_{n-q}, x_{n-q+1}, \ldots, x_{n-1}, x_n\}$. From Theorem 2.1.4, it follows that for some $\xi_n \in [x_{n-q}, x_{n+1}]$ we have

$$g(x) = p_q(x) + \frac{(x - x_{n-q})(x - x_{n-q+1}) \ldots (x - x_n)}{(q+1)!} f^{(q+1)}(\xi_n)$$

$$= p_q(x) + \underbrace{\frac{Y^{(q+2)}(\xi_n)}{(q+1)!}\left(\prod_{j=0}^{q}(x - x_{n-q+j})\right)}_{\text{local truncation error}},$$

provided $Y \in C^{(q+2)}[x_{n-q}, x_{n+1}]$. Recall that $g(x_n) = f(x_n, Y(x_n)) = Y'(x_n) = Y'_n$.

1. **Suppose $q = 0$**. The interpolating node is x_n and we have that

$$p_0(x) = g(x_n).$$

Computing the integral gives

$$\int_{x_n}^{x_{n+1}} g(x)\,dx \approx \int_{x_n}^{x_{n+1}} g(x_n)\,dx = hg(x_n) = hY'_n,$$

with local truncation error

$$T_n(h) = Y''(\xi_n)\int_{x_n}^{x_{n+1}}(x - x_n)\,dx = \frac{h^2}{2}Y''(\xi_n).$$

Dropping $T_n(h)$, we obtain the 1-step AB method of order 1 which is just the **forward/explicit Euler method**:

$$\boxed{y_{n+1} = y_n + hy'_n, \quad n \geqslant 0.} \tag{AB1}$$

It turns out that forward Euler method is stable if the step size h is sufficiently small.

2. **Suppose $q = 1$**. The interpolating nodes are $\{x_{n-1}, x_n\}$ and we have that

$$\begin{aligned} p_1(x) &= \frac{(x - x_n)}{(x_{n-1} - x_n)}g(x_{n-1}) + \frac{(x - x_{n-1})}{(x_n - x_{n-1})}g(x_n) \\ &= \frac{1}{h}\Big[(x_n - x)g(x_{n-1}) + (x - x_{n-1})g(x_n)\Big]. \end{aligned}$$

Computing the integral gives

$$\begin{aligned} \int_{x_n}^{x_{n+1}} g(x)\,dx &\approx \frac{g(x_{n-1})}{h}\int_{x_n}^{x_{n+1}}(x_n - x)\,dx + \frac{g(x_n)}{h}\int_{x_n}^{x_{n+1}}(x - x_{n-1})\,dx \\ &= \left(\frac{g(x_{n-1})}{h}\right)\left(-\frac{h^2}{2}\right) + \left(\frac{g(x_n)}{h}\right)\left(\frac{4h^2 - h^2}{2}\right) \\ &= \frac{3h}{2}g(x_n) - \frac{h}{2}g(x_{n-1}) \\ &= \frac{3h}{2}Y'_n - \frac{h}{2}Y'_{n-1}, \end{aligned}$$

with the local truncation error

$$T_n(h) = \frac{Y^{(3)}(\xi_n)}{2}\int_{x_n}^{x_{n+1}}(x - x_{n-1})(x - x_n)\,dx = \frac{5}{12}h^3Y^{(3)}(\xi_n).$$

Dropping $T_n(h)$, we obtain the 2-step AB method:

$$y_{n+1} = y_n + \frac{h}{2}\Big[3y'_n - y'_{n-1}\Big], \quad n \geqslant 1. \tag{AB2}$$

Observe that (AB2) requires two initial values y_0, y_1 to begin with. It can be shown that the global error of (AB2) is $\mathcal{O}(h^2)$, assuming that y_1 is chosen appropriately.

3. **Suppose $q = 2$.** The interpolating nodes are $\{x_{n-2}, x_{n-1}, x_n\}$ and we have that

$$\begin{aligned} p_2(x) &= \left[\frac{(x-x_{n-1})(x-x_n)}{(x_{n-2}-x_{n-1})(x_{n-2}-x_n)}\right] g(x_{n-2}) + \left[\frac{(x-x_{n-2})(x-x_n)}{(x_{n-1}-x_{n-2})(x_{n-1}-x_n)}\right] g(x_{n-1}) \\ &\qquad + \left[\frac{(x-x_{n-2})(x-x_{n-1})}{(x_n-x_{n-2})(x_n-x_{n-1})}\right] g(x_n) \\ &= \frac{1}{2h^2}\Big[(x-x_{n-1})(x-x_n)g(x_{n-2}) + 2(x-x_{n-2})(x-x_n)g(x_{n-1}) \\ &\qquad + (x-x_{n-2})(x-x_{n-1})g(x_n)\Big] \end{aligned}$$

The local truncation error is

$$T_n(h) = \frac{Y^{(4)}(\xi_n)}{6}\int_{x_n}^{x_{n+1}} (x-x_{n-2})(x-x_{n-1})(x-x_n)\,dx = \frac{3}{8}h^4 Y^{(4)}(\xi_n).$$

Dropping $T_n(h)$, we obtain the 3-step AB method:

$$y_{n+1} = y_n + \frac{h}{12}\Big[23y'_n - 16y'_{n-1} + 5y'_{n-2}\Big], \quad n \geqslant 2. \tag{AB3}$$

Observe that (AB3) requires three initial values y_0, y_1, y_2 to begin with. It can be shown that the global error of (AB3) is $\mathcal{O}(h^3)$, assuming that y_1, y_2 are chosen appropriately.

The $(q+1)$-step AB methods are based on interpolation of degree q. It can be shown that the local truncation error satisfies

$$T_n = Ch^{q+2}Y^{(q+2)}(\xi_n) \quad \text{for some } \xi_n \in [x_{n-q}, x_{n+1}],$$

and $y_1, y_2, \ldots, y_q$ must be approximated using another method. If these approximations satisfy

$$Y(x_k) - y_h(x_k) = \mathcal{O}(h^{q+1}), \quad k = 1, \ldots, q,$$

then the $(q+1)$-step AB method is also of order $\mathcal{O}(h^{q+1})$.

Remark 4.5.1. In the case of (AB2), since we require $|Y(x_n) - y_h(x_n)| = \mathcal{O}(h^2)$, this must be true for $n = 1$ as well. There are many choices of achieving this:

1. The simplest choice is to use Euler's method, in which y_1 is given by

$$y_1 = y_0 + hf(x_0, y_0) = Y_0 + hf(x_0, Y_0) \implies Y_1 - y_1 = \frac{h^2}{2}Y''(\xi_1),$$

for some $\xi_1 \in [x_0, x_1]$. Globally, Euler's method is of order $\mathcal{O}(h)$, but for a single step, it is of order $\mathcal{O}(h^2)$.

2. If we use the RK method of order 2 (RK2), then for a single step, $|Y(x_1) - y_h(x_1)| = \mathcal{O}(h^3)$, which is more than required.

4.5.2 Adams-Moulton (AM) Methods

The idea is similar to Adams-Bashforth methods, except that we consider the set of interpolation nodes $\{x_{n-q+1}, x_{n-q+2}, \ldots, x_n, x_{n+1}\}$ instead. From Theorem 2.1.4, it follows that for some $\xi_n \in [x_{n-q+1}, x_{n+1}]$ we have

$$\begin{aligned} g(x) &= p_q(x) + \frac{(x - x_{n-q+1})(x - x_{n-q+2}) \ldots (x - x_{n+1})}{(q+1)!} f^{(q+1)}(\xi_n) \\ &= p_q(x) + \underbrace{\frac{Y^{(q+2)}(\xi_n)}{(q+1)!} \left(\prod_{j=0}^{q} (x - x_{n-q+(j+1)}) \right)}_{\text{local truncation error}}, \end{aligned}$$

assuming $Y \in C^{(q+2)}[x_{n-q+1}, x_{n+1}]$.

1. **Suppose $q = 0$**. The interpolating node is x_{n+1} and we have that

   $$p_0(x) = g(x_{n+1}).$$

   Computing the integral gives

   $$\int_{x_n}^{x_{n+1}} g(x)\, dx \approx \int_{x_n}^{x_{n+1}} g(x_{n+1})\, dx = h g(x_{n+1}) = h Y'_{n+1},$$

   with local truncation error

   $$T_n(h) = Y''(\xi_n) \int_{x_n}^{x_{n+1}} (x - x_{n+1})\, dx = -\frac{h^2}{2} Y''(\xi_n).$$

   Dropping $T_n(h)$, we obtain the 1-step AM method of order 1 which is just the **backward/implicit Euler method**:

   $$\boxed{y_{n+1} = y_n + h y'_{n+1}, \quad n \geqslant 0.} \tag{AM1}$$

   As we shall see later, the implicit Euler method is unconditional stable, *i.e.* the stability does not depend on the step size h.

2. **Suppose $q = 1$**. The interpolating nodes are $\{x_n, x_{n+1}\}$ and we have that

   $$\begin{aligned} p_1(x) &= \frac{(x - x_{n+1})}{(x_n - x_{n+1})} g(x_n) + \frac{(x - x_n)}{(x_{n+1} - x_n)} g(x_{n+1}) \\ &= \frac{1}{h} \Big[(x_{n+1} - x) g(x_n) + (x - x_n) g(x_{n+1}) \Big]. \end{aligned}$$

   Computing the integral gives

   $$\begin{aligned} \int_{x_n}^{x_{n+1}} g(x)\, dx &\approx \frac{g(x_n)}{h} \int_{x_n}^{x_{n+1}} (x_{n+1} - x)\, dx + \frac{g(x_{n+1})}{h} \int_{x_n}^{x_{n+1}} (x - x_n)\, dx \\ &= \left(\frac{g(x_n)}{h} \right) \left(\frac{h^2}{2} \right) + \left(\frac{g(x_{n+1})}{h} \right) \left(\frac{h^2}{2} \right) \end{aligned}$$

$$= \frac{h}{2}g(x_n) + \frac{h}{2}g(x_{n+1})$$
$$= \frac{h}{2}Y'_n + \frac{h}{2}Y'_{n+1},$$

with the local truncation error

$$T_n(h) = \frac{Y'''(\xi_n)}{2}\int_{x_n}^{x_{n+1}} (x - x_n)(x - x_{n+1})\,dx = -\frac{h^3}{12}Y^{(3)}(\xi_n).$$

Dropping $T_n(h)$, we obtain the 1-step AM method of order 2 which is the (Simple Trapezoidal) rule for approximating integral:

$$\boxed{y_{n+1} = y_n + \frac{h}{2}\Big[y'_n + y'_{n+1}\Big], \quad n \geqslant 0.} \tag{AM2}$$

This is often the choice for solving diffusion problem or parabolic PDEs, where one discretizes in space and solve in time. However, this is not the method of choice for wave problems since it lacks some notion of stability.

3. **Suppose $q = 2$**. The interpolating nodes are $\{x_{n-1}, x_n, x_{n+2}\}$ and we have that

$$p_2(x) = \left[\frac{(x - x_n)(x - x_{n+1})}{(x_{n-1} - x_n)(x_{n-1} - x_{n+1})}\right] g(x_{n-1}) + \left[\frac{(x - x_{n-1})(x - x_{n+1})}{(x_n - x_{n-1})(x_n - x_{n+1})}\right] g(x_n)$$
$$+ \left[\frac{(x - x_{n-1})(x - x_n)}{(x_{n+1} - x_{n-1})(x_{n+1} - x_n)}\right] g(x_{n+1})$$
$$= \frac{1}{2h^2}\Big[(x - x_n)(x - x_{n+1})g(x_{n-1}) + 2(x - x_{n-1})(x - x_{n+1})g(x_n)$$
$$+ (x - x_{n-1})(x - x_n)g(x_{n+1})\Big]$$

The local truncation error is

$$T_n(h) = \frac{Y^{(4)}(\xi_n)}{6}\int_{x_n}^{x_{n+1}} (x - x_{n-1})(x - x_n)(x - x_{n+1})\,dx = -\frac{h^4}{24}Y^{(4)}(\xi_n).$$

Dropping $T_n(h)$, we obtain the 2-step AM method:

$$\boxed{y_{n+1} = y_n + \frac{h}{12}\Big[5y'_{n+1} + 8y'_n - y'_{n-1}\Big], \quad n \geqslant 1.} \tag{AM3}$$

If $f(x, y)$ is linear with respect to y, then the Adams-Moulton methods reduce to explicit numerical methods. However, finding y_{n+1} requires solving a nonlinear equation in general. In the case of (AM2), we have

$$y_{n+1} - \frac{h}{2}f(x_{n+1}, y_{n+1}) = y_n + \frac{h}{2}f(x_n, y_n). \tag{4.5.1}$$

One way is to view this as a root-finding problem and solve it using Newton's method, but this is practical only for a small system. For large system, we employ a fixed-point iteration method. Choosing $y_{n+1}^{(0)}$ appropriately, we solve the following iteration equation

$$y_{n+1}^{(j+1)} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(j)})], \quad j = 0, 1, \ldots. \tag{4.5.2}$$

To determine conditions under which (4.5.2) converges, subtract (4.5.2) from (4.5.1) to obtain

$$\begin{aligned} y_{n+1} - y_{n+1}^{(j+1)} &= \frac{h}{2}[f(x_{n+1}, y_{n+1}) - f(x_{n+1}, y_{n+1}^{(j)})] \\ &= \frac{h}{2}\frac{\partial f(x_{n+1}, \xi_{n+1})}{\partial y}(y_{n+1} - y_{n+1}^{(j)}), \end{aligned}$$

for some ξ_{n+1} in between y_{n+1} and $y_{n+1}^{(j)}$, where we apply the *mean value theorem* on the last line. Consequently, we see that $y_{n+1}^{(j)} \longrightarrow y_{n+1}$ if $\left|\frac{h}{2}\frac{\partial f(x_{n+1}, \xi_{n+1})}{\partial y}\right| < 1$.

Remark 4.5.2. In practice, the stepsize h and initial guess $y_{n+1}^{(0)}$ are chosen to ensure that only one iterate need be computed, and then we can take $y_{n+1} \approx y_{n+1}^{(1)}$. The initial guess $y_{n+1}^{(0)}$ can be computed using (AB1) or (AB2). For instance, using (AB1) gives:

$$y_{n+1}^{(0)} = y_n + hy_n'.$$

This has a local truncation error of order $\mathcal{O}(h^2)$ and it matches with the global error for (AM2) .

q	Global error	Adams-Bashforth formula	Truncation error	
0	$\mathcal{O}(h)$	$y_{n+1} = y_n + hy_n'$	$\frac{h^2}{2}Y''(\xi_n)$	1-step
1	$\mathcal{O}(h^2)$	$y_{n+1} = y_n + \frac{h}{2}\left(3y_n' - y_{n-1}'\right)$	$\frac{5}{12}h^3Y^{(3)}(\xi_n)$	2-step
2	$\mathcal{O}(h^3)$	$y_{n+1} = y_n + \frac{h}{12}\left(23y_n' - 16y_{n-1}' + 5y_{n-2}'\right)$	$\frac{3}{8}h^4Y^{(4)}(\xi_n)$	3-step

Table 4.1: Adams-Bashforth methods (explicit).

q	Global error	Adams-Moulton formula	Truncation error	
0	$\mathcal{O}(h)$	$y_{n+1} = y_n + hy_{n+1}'$	$-\frac{h^2}{2}Y''(\xi_n)$	1-step
1	$\mathcal{O}(h^2)$	$y_{n+1} = y_n + \frac{h}{2}\left(y_n' + y_{n+1}'\right)$	$-\frac{h^3}{12}Y^{(3)}(\xi_n)$	1-step
2	$\mathcal{O}(h^3)$	$y_{n+1} = y_n + \frac{h}{12}\left(5y_{n+1}' + 8y_n' - y_{n-1}'\right)$	$-\frac{h^4}{24}h^4Y^{(4)}(\xi_n)$	2-step

Table 4.2: Adams-Moulton methods (implicit).

4.6 Consistency and Convergence of Multistep Methods

Definition 4.6.1.

1. For any differentiable function $Y(x)$, the **local truncation error** for integrating $Y'(x)$ is given by

$$T_n(Y) = Y(x_{n+1}) - \left(\sum_{j=0}^{p} a_j Y(x_{n-j}) + h \sum_{j=-1}^{p} b_j f(x_{n-j}, y_{n-j}) \right), \quad n \geqslant p \geqslant 0. \tag{4.6.1}$$

2. Define the function $\tau_n(Y) = T_n(Y)/h$. We say that (LMM) is **consistent** if

$$\tau(h) = \max_{x_p \leqslant x_n \leqslant b} |\tau_h(Y)| \longrightarrow 0 \quad \text{as} \quad h \longrightarrow 0. \tag{4.6.2}$$

for all $Y(x) \in C^1[x_0, b]$. Formally, a numerical method is consistent if its discrete operator converges to the continuous operator of the ODE as $h \longrightarrow 0$, *i.e.* the true solution almost satisfies the discrete equation.

The speed of convergence of the approximate solution $\{y_n\}$ to the exact solution $Y(x)$ is related to the speed of convergence in (4.6.2). The following theorem provides conditions under which $\tau(h) = \mathcal{O}(h^m)$ for some $m \geqslant 1$, which does not involve Taylor expansion.

Theorem 4.6.2. *Let $m \geqslant 1$ be a given integer. The consistency condition (4.6.2) holds for any $Y(x) \in C^1[x_0, b]$ if and only if*

$$\sum_{j=0}^{p} a_j = 1 \quad \text{and} \quad -\sum_{j=0}^{p} j a_j + \sum_{j=-1}^{p} b_j = 1. \tag{4.6.3}$$

Moreover, we have that $\tau(h) = \mathcal{O}(h^m)$ for any $Y(x) \in C^{(m+1)}[x_0, b]$ if and only if (4.6.3) holds and

$$\sum_{j=0}^{p} (-j)^k a_j + k \sum_{j=-1}^{p} (-j)^{k-1} b_j = 1 \quad \text{for any } k = 2, \ldots, m. \tag{4.6.4}$$

*The largest value of $m \geqslant 1$ such that (4.6.4) holds is called the **order of convergence** of (LMM).*

Proof. Assuming $Y(x) \in C^{(m+1)}[x_0, b]$, expanding $Y(x)$ about the point x_n yields:

$$Y(x) = \sum_{k=0}^{m} \frac{Y^{(k)}(x_n)}{k!}(x - x_n)^k + \underbrace{\frac{Y^{(m+1)}(\xi)}{(m+1)!}(x - x_n)^{m+1}}_{R_{m+1}(x)},$$

for some $\xi \in [x_0, b]$. Using the linearity of the local truncation error T_n (as a function of Y), we obtain:

$$T_n(Y) = \sum_{k=0}^{m} \frac{Y^{(k)}(x_n)}{k!} T_n\Big((x - x_n)^k\Big) + T_n(R_{m+1}).$$

For $k = 0$,

$$T_n\Big((x - x_n)^0\Big) = T_n(1) = 1 - \sum_{j=0}^{p} a_j = c_0.$$

For $k \geqslant 1$,

$$\begin{aligned} T_n\Big((x-x_n)^k\Big) &= (x_{n+1}-x_n)^k - \left(\sum_{j=0}^{p} a_j(x_{n-j}-x_n)^k + h\sum_{j=-1}^{p} b_j k(x_{n-j}-x_n)^{k-1}\right) \\ &= h^k - \left(\sum_{j=0}^{p} a_j(-jh)^k + k\sum_{j=-1}^{p} b_j h(-jh)^{k-1}\right) \\ &= \left(1 - \left[\sum_{j=0}^{p}(-j)^k a_j + k\sum_{j=-1}^{p}(-j)^{k-1}b_j\right]\right)h^k \\ &= c_k h^k. \end{aligned}$$

A similar argument shows that:

$$T_n(R_{m+1}) = \frac{Y^{(m+1)}(\xi)}{(m+1)!}T_n\Big((x-x_n)^{m+1}\Big) = \frac{Y^{(m+1)}(\xi)}{(m+1)!}c_{m+1}h^{m+1}.$$

It follows that

$$T_n(Y) = \sum_{k=0}^{m}\frac{Y^{(k)}(x_n)}{k!}c_k h^k + \frac{Y^{(m+1)}(\xi)}{(m+1)!}c_{m+1}h^{m+1}.$$

For (LMM) to be consistent, we require $\tau(h) = \mathcal{O}(h)$ and this requires $T_n(Y) = \mathcal{O}(h^2)$. With $m = 1$, we must have $c_0 = c_1 = 0$ and this gives (4.6.3). To obtain $\tau(h) = \mathcal{O}(h^m)$, we require $T_n(Y) = \mathcal{O}(h^{m+1})$ and this is true if and only if

$$c_0 = c_1 = \ldots = c_m = 0.$$

This gives the condition (4.6.4).

■

Formally, a numerical method converges to the true solution if decreasing the step size h leads to decreased error, in such a way that the error go to zero in the limit as h goes to zero. The following theorem gives sufficient conditions for (LMM) to be convergent.

Theorem 4.6.3. *Consider solving the (IVP) on $[x_0, b]$ using the linear multistep method (LMM). Assume that*

1. *the initial error satisfy*

$$\eta(h) = \max_{0\leqslant i\leqslant p}|Y(x_i) - y_h(x_i)| \longrightarrow 0 \quad \text{as} \quad h \longrightarrow 0.$$

2. *(LMM) is consistent.*

3. *The coefficients $a_j, j = 0, 1, \ldots, p$ in (LMM) are all nonnegative.*

Then (LMM) is convergent and

$$\max_{x_0\leqslant x_n\leqslant b}|Y(x_n) - y_h(x_n)| \leqslant C_1\eta(h) + C_2\tau(h), \tag{4.6.5}$$

for some constants C_1, C_2 independent of h. Moreover, if $\eta(h) = \mathcal{O}(h^m)$ and $\tau(h) = \mathcal{O}(h^m)$, then (LMM) is convergent with order m.

Remark 4.6.4. The same convergence result can be obtained if we weaken assumption (3) in the Theorem 4.6.3. Observe that to obtain (LMM) with a rate of convergence of $\mathcal{O}(h^m)$, it is necessary that the local truncation error $T_n(Y)$ is of order $\mathcal{O}(h^{m+1})$, but the initial values $y_0, y_1, \ldots, y_p$ need to be computed only with an accuracy of $\mathcal{O}(h^m)$, since $\eta(h) = \mathcal{O}(h^m)$ is sufficient in the error estimate (4.6.5).

Example 4.6.5. Consider (AB2), the 2-step Adams-Bashforth method of order 2:

$$y_{n+1} = y_n + \frac{h}{2}(3y'_n - y'_{n-1}).$$

In this case, $p = 1$ and

$$a_0 = 1, \quad a_1 = 0, \quad b_{-1} = 0, \quad b_0 = \frac{3}{2}, \quad b_1 = -\frac{1}{2}.$$

We verify using Theorem 4.6.2 that (AB2) is consistent and $\tau(h) = \mathcal{O}(h^2)$:

$$\begin{aligned}
\sum_{j=0}^{1} a_j &= a_0 + a_1 = 1 \\
-\sum_{j=0}^{1} j a_j + \sum_{j=-1}^{1} b_j &= 0 + \left(0 + \frac{3}{2} - \frac{1}{2}\right) = 1 \\
\sum_{j=0}^{1} (-j)^2 a_j + 2\sum_{j=-1}^{1} (-j)^1 b_j &= 0 + 2\left(0 + 0 - \left(-\frac{1}{2}\right)\right) = 1.
\end{aligned}$$

It follows from Theorem 4.6.3 that (AB2) is convergent with order 2, provided $\eta(h) = \mathcal{O}(h^2)$.

4.7 Stability of Multistep Methods

Before we define what it means for the linear multistep method (LMM) to be stable, we analyse one example of (LMM) in which it is an unstable method.

Example 4.7.1. Consider the following 2-step explicit method

$$y_{n+1} = 3y_n - 2y_{n-1} + \frac{h}{2}[f(x_n, y_n) - 3f(x_{n-1}, y_{n-1})], \quad n \geqslant 1. \tag{4.7.1}$$

If the true solution $Y(x) \in C^3[x_0, b]$, it can be shown using Taylor expansion that

$$Y(x_{n+1}) = 3Y(x_n) - 2Y(x_{n-1}) + \frac{h}{2}[Y'(x_n) - 3Y'(x_{n-1})] + \frac{7}{12}h^3 Y'''(\xi_n).$$

This means that $T_n = \mathcal{O}(h^3)$ and $\tau(h) = \mathcal{O}(h^2)$. Comparing with (LMM), we see that

$$a_0 = 3, \quad a_1 = -2, \quad b_{-1} = 0, \quad b_0 = \frac{1}{2}, \quad b_1 = -\frac{3}{2}.$$

We first verify that it is a consistent numerical method and $\tau(h) = \mathcal{O}(h^2)$.

$$\sum_{j=0}^{1} a_j = a_0 + a_1 = 1$$
$$-\sum_{j=0}^{1} j a_j + \sum_{j=-1}^{1} b_j = -(0-2) + \left(0 + \frac{1}{2} - \frac{3}{2}\right) = 1$$
$$\sum_{j=0}^{1} (-j)^2 a_j + 2\sum_{j=-1}^{1} (-j)^1 b_j = -2 + 2\left(0 + 0 - \left(-\frac{3}{2}\right)\right) = 1.$$

Now, consider the following initial value problem

$$\begin{cases} y'(x) &= 0 \\ y(0) &= 1. \end{cases}$$

Its unique solution is $Y(x) \equiv 1$. Applying the 2-step explicit method (4.7.1) to IVP, we have

$$y_{n+1} = 3y_n - 2y_{n-1}, \quad n \geqslant 1, \quad y_0 = 1.$$

If we choose $y_1 = 1$, then $y_n = 1$ for all $n \geqslant 0$. Suppose we perturb the initial values to

$$y_{\varepsilon,0} = 1 + \varepsilon, \quad y_{\varepsilon,1} = 1 + 2\varepsilon,$$

then

$$y_{\varepsilon,2} = 3(1 + 2\varepsilon) - 2(1 + \varepsilon) = 1 + 4\varepsilon = 1 + 2^2\varepsilon.$$

We claim that $y_{\varepsilon,n} = 1 + 2^n\varepsilon$ for each $n \geqslant 0$. Using method of strong induction,

$$y_{\varepsilon,n+1} = 3y_{\varepsilon,n} - 2y_{\varepsilon,n-1} = 3(1 + 2^n\varepsilon) - 2(1 + 2^{n-1}\varepsilon) = 1 + 2^{n+1}\varepsilon.$$

For example, take $x_n = 1, n = 1/h, x_n = nh$, the perturbation on the original approximate solutions satisfies

$$y_{\varepsilon,n} - y_n = \varepsilon 2^n = \varepsilon 2^{1/h} \longrightarrow \infty \quad \text{as} \quad h \longrightarrow 0,$$

i.e. the numerical method is not convergent.

Definition 4.7.2.

1. Let $\{y_n \colon 0 \leqslant n \leqslant N(h)\}$ be the solution of (LMM) for some differential equation $y' = f(x, y)$ for all sufficiently small $h \leqslant h_0$. For every $h \leqslant h_0$, perturb initial values $y_0, \ldots, y_p \mapsto z_0, \ldots, z_p$ with

   $$\max_{0 \leqslant n \leqslant p} |y_n - z_n| \leqslant \varepsilon, \quad 0 < h \leqslant h_0.$$

   The family of solution $\{y_n\}$ is **stable** if there exists a constant C, independent of $h \leqslant h_0$ and valid for all sufficiently small $\varepsilon > 0$ such that

   $$\max_{0 \leqslant n \leqslant N(h)} |y_n - z_n| \leqslant C\varepsilon, \quad 0 < h \leqslant h_0.$$

2. Consider all the (IVP) with $f(x, y)$ continuous and satisfy the strong Lipschitz condition (4.2.1). Assume that $\{y_n\}$ produced from (LMM) are all stable. We say that (LMM) is a **stable numerical method** (This is a global property of a numerical scheme, true for any IVPs of admissible class).

3. Consider (IVP) with $f(x, y)$ continuous and satisfy the strong Lipschitz condition (4.2.1). Assume that all initial values $y_0, \ldots, y_p$ satisfy

$$\eta(h) := \max_{0 \leqslant n \leqslant p} |Y(x_n) - y_h(x_n)| \longrightarrow 0 \quad \text{as} \quad h \longrightarrow 0.$$

Then the numerical solution $\{y_n\}$ is said to converge to the true solution $Y(x)$ of (IVP) if

$$\max_{x_0 \leqslant x_n \leqslant b} |Y(x_n) - y_h(x_n)| \longrightarrow 0 \quad \text{as} \quad h \longrightarrow 0.$$

If (LMM) is convergent for all initial value problems, then it is called a **convergent numerical method**.

Remark 4.7.3. It can be shown that convergence of (LMM) implies consistency of (LMM). As an example, consider the following initial value problem:

$$\begin{cases} y'(x) &= 0 \\ y(0) &= 1. \end{cases}$$

Since (LMM) is assumed to be convergent, we must have $y_{p+1} \longrightarrow Y(x_{p+1}) \equiv 1$. Choosing the initial values $y_0, \ldots, y_p = 1$, it follows that $1 = \sum_{j=0}^{p} a_j$.

It turns out that the convergence and stability of (LMM) are linked to the roots of the polynomial

$$\rho(r) = r^{p+1} - \sum_{j=0}^{p} a_j r^{p-j}. \tag{4.7.2}$$

This can be obtained by considering (LMM) in the limit as $h \longrightarrow 0$ and setting y_p as r^p. Observe that $\rho(1) = 0$ from the consistency condition, so $r = 1$ is called the *principal leading root*.

Definition 4.7.4. Let $r_0, \ldots, r_p$ be the roots of (4.7.2), repeated according to their multiplicity and set $r_0 = 1$. We say that the linear multistep method (LMM) satisfies the **root condition** if

1. Each roots lies in the (closed) unit disk in $\mathbb{C}$, *i.e.* $|r_j| \leqslant 1$ for every $j = 0, 1, \ldots, p$;

2. Roots on the boundary of unit disk are simple, *i.e.* $|r_j| = 1 \implies \rho'(r_j) \neq 0$.

Theorem 4.7.5. *Assume* (LMM) *satisfies the consistency condition* (4.6.3).

(a) (LMM) *is stable if and only if the root condition is satisfied.*

(b) (LMM) *is convergent if and only if the root condition is satisfied.*

Corollary 4.7.6. *Let* (LMM) *be a consistent linear multistep method. Then it is convergent if and only if it is stable.*

Example 4.7.7. Recall the 2-step explicit method

$$y_{n+1} = 3y_n - 2y_{n-1} + \frac{h}{2}[f(x_n, y_n) - 3f(x_{n-1}, y_{n-1})], \quad n \geqslant 1.$$

We already shown that this numerical scheme is consistent. With $p + 1 = 2$, we have

$$\rho(r) = r^2 - 3r + 2 = (r - 2)(r - 1).$$

Since one of the roots is $r = 2$ which lies outside the unit disk, it violates the root condition and thus it is unstable.

In Theorem 4.7.5, the step size h has to be sufficiently small, but exactly how small should h be? Clearly, h cannot be extremely small, otherwise (LMM) is impractical for most problems. We investigate the stability of (LMM) by considering only the following model problem:

$$\begin{cases} y'(x) &= \lambda y(x) \\ y(0) &= 1. \end{cases}$$

An intuitive reason is as follows: Expanding $f(x, y)$ about the point (x_0, Y_0) yields the approximation:

$$Y'(x) \approx f(x_0, Y_0) + f_x(x_0, Y_0)(x - x_0) + f_Y(x_0, Y_0)(Y - Y_0),$$

which is valid if $x \approx x_0$. Define $V(x) = Y(x) - Y_0$, the approximation above becomes:

$$V'(x) \approx \lambda V(x) + g(x),$$

where $\lambda = f_y(x_0, Y_0)$ and $g(x) = f(x_0, Y_0) + f_x(x_0, Y_0)(x - x_0)$. Note that the inhomogeneous term $g(x)$ will drop out from the error equation because we are concerned with differences of solutions when investigating numerical stability.

As a motivating example, consider the stability of Euler's method. Applying it to the model equation with $y(0) = Y_0$ gives:

$$y_{n+1} = y_n + h\lambda y_n, \quad n \geqslant 0, \quad y_0 = Y_0. \tag{4.7.3}$$

Consider the perturbed problem where we only perturb the initial condition. Applying Euler's method to the perturbed problem gives:

$$z_{n+1} = z_n + h\lambda z_n, \quad n \geqslant 0, \quad z_0 = Y_0 + \varepsilon. \tag{4.7.4}$$

We are interested in the case where $\mathrm{Re}(\lambda) < 0$, so that for any sufficiently small $\varepsilon > 0$ we have:

$$Z(x) - Y(x) = \varepsilon e^{\lambda x} \longrightarrow 0 \quad \text{as } x \longrightarrow \infty.$$

For such λ, we want to find values of h such that the numerical solutions of (4.7.3) and (4.7.4) will retain the behaviour associated with $Y(x)$ and $Z(x)$. Define $e_n := z_n - y_n$, taking the difference yields:

$$e_{n+1} = e_n + h\lambda e_n = (1 + h\lambda)e_n \implies e_n = (1 + h\lambda)^n e_0 = (1 + h\lambda)^n \varepsilon, \quad n \geqslant 0.$$

Consequently, $e_n \longrightarrow 0$ as $x_n \longrightarrow \infty$ if and only if $|1 + h\lambda| < 1$, *i.e.* $h\lambda \in B_1(-1) \subset \mathbb{C}$.

Remark 4.7.8. From the convergence point of view, we can view the original differential equation as perturbation of the numerical method. In the case of Euler's method,

$$Y(x_{n+1}) = Y(x_n) + h\lambda Y(x_n) + \frac{h}{2} Y''(\xi_n),$$

which is a perturbation of (4.7.3) at every step $x_0, x_1, \ldots$. Nonetheless, the preceeding argument can be modified to show that the dependence of the error $Y(x_n) - y_n$ on the bound $|1 + h\lambda| < 1$.

Definition 4.7.9. For all λ satisfying $\mathrm{Re}(\lambda) < 0$, the set of all complex $h\lambda$ such that the numerical solution $\{y_n\}$ of (LMM), when applied to the model equation $y' = \lambda y$, tend to 0 as $x_n \longrightarrow \infty$ for all choices of initial values $\{y_0, y_1, \ldots, y_p\}$, is called the **region of absolute stability** of (LMM).

Applying (LMM) to the model equation, we obtain the following linear $(p + 1)$ order homogeneous difference equation:

$$y_{n+1} = \sum_{j=0}^{p} a_j y_{n-j} + h\lambda \sum_{j=-1}^{p} b_j y_{n-j}, \quad n \geqslant p \geqslant 0. \tag{4.7.5}$$

The method for solving (4.7.5) is analogous to that of linear homogeneous ODE, but instead of exponential solutions we look for polynomial solutions of (4.7.5). Setting $y_k = r^k$ in (4.7.5) yields:

$$0 = r^{n+1} - \sum_{j=0}^{p} a_j r^{n-j} - h\lambda \sum_{j=-1}^{p} b_j r^{n-j}.$$

Dividing by r^{n-p}, we obtain the **characteristic polynomial**:

$$0 = p(r) = r^{p+1} - \sum_{j=0}^{p} a_j r^{p-j} - h\lambda \sum_{j=-1}^{p} b_j r^{p-j} = \rho(r) - h\lambda\sigma(r),$$

where

$$\sigma(r) = b_{-1} r^{p+1} + \sum_{j=0}^{p} b_j r^{p-j}. \tag{4.7.6}$$

Denote the roots of $p(r) = \rho(r) - h\lambda\sigma(r)$ as $r_0(h\lambda), r_1(h\lambda), \ldots, r_p(h\lambda)$, counting multiplicity. Observe that as $h \longrightarrow 0$, $p(r) = 0$ reduces to $\rho(r) = 0$ and

$$r_j(h\lambda) = r_j(0) \quad \text{for } j = 0, 1, \ldots, p.$$

Because $r = 1$ is a root of $\rho(r)$ from the consistency condition (4.6.3), we let $r_0(h\lambda)$ to be the root of $p(r)$ such that $r_0(0) = 1$, and we call this the *principal root* of the characteristic equation.

We now construct the general solution to (4.7.5). Suppose the characteristic equation has distinct roots, then the general solution of (4.7.5) is given by:

$$y_n = \sum_{j=0}^{p} \gamma_j [r_j(h\lambda)]^n, \quad n \geqslant 0, \tag{4.7.7}$$

where γ_j are constants determined by initial conditions. In the case of repeated roots, say $r_j(h\lambda)$ with multiplicity $\nu > 1$, it can be shown that

$$\Big\{[r_j(h\lambda)]^n, n[r_j(h\lambda)]^n, \ldots, n^{\nu-1}[r_j(h\lambda)]^n\Big\} \tag{4.7.8}$$

form a linearly independent set of solutions of (4.7.5). Consequently, a necessary and sufficient condition for $y_n \longrightarrow 0$ as $x_n \longrightarrow \infty$ for all choices of $y_0, y_1, \ldots, y_p$ is that

$$|r_j(h\lambda)| < 1 \quad \text{for all } j = 0, 1, \ldots, p. \tag{4.7.9}$$

Definition 4.7.10. The set of all complex $h\lambda$ for $\mathrm{Re}(\lambda) < 0$ that satisfies condition (4.7.9) is also called the region of absolute stability. Note that this region is contained in the set defined in the preceeding definition, and they usually coincide. *The second definition is more flexible, in the sense that the characteristic equation is easier to write down and root-finding algorithm can be used if necessary to find the characteristic roots. It is difficult in general to express y_n in terms of initial values even for 2-step methods!*

Example 4.7.11. Consider the Backward Euler method

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}), \quad n \geqslant 0,$$

which is a 1-step implicit method. With $a_0 = 1, b_{-1} = 1, b_0 = 0$, the characteristic polynomial has the form:

$$p(r) = \rho(r) - h\lambda\sigma(r) = r - 1 - h\lambda r,$$

which has a simple root $r = \dfrac{1}{1 - h\lambda}$ assuming $1 - h\lambda \neq 0$. The region of absolute stability with respect to the second definition is the set of all complex $h\lambda$ with $\mathrm{Re}(\lambda) < 0$, and we say that the Backward Euler method is an **A-stable method**.

On the other hand, applying the Backward Euler method to the model equation and solving for y_{n+1} yields:

$$\begin{aligned} y_{n+1} = y_n + h\lambda y_{n+1} &\implies y_{n+1} = \left(\frac{1}{1 - h\lambda}\right) y_n \\ &\implies y_n = \left(\frac{1}{1 - h\lambda}\right)^n y_0 = [r_0(h\lambda)]^n y_0, \quad n \geqslant 0. \end{aligned}$$

We must have $|r_0(h\lambda)| < 1$ in order to have $y_n \longrightarrow 0$ as $x_n \longrightarrow \infty$ and recover the same region of absolute stability.

Relative Stability and Weak Stability

Consider again the model problem:

$$\begin{cases} y'(x) &= \lambda y(x), \\ y(0) &= 1. \end{cases}$$

Assuming distinct roots of the characteristic equation, the general solution using (LMM) is given by

$$y_n = \sum_{j=0}^{p} \gamma_j [r_j(h\lambda)]^n, \quad n \geqslant 0.$$

It can be shown that

$$\gamma_0 [r_0(h\lambda)]^n \longrightarrow Y(x) = e^{\lambda x} \quad \text{on } [0, b] \quad \text{as } h \longrightarrow 0.$$

The remaining terms $\gamma_j [r_j(h\lambda)]^n, j = 1, 2, \ldots, p$ are called **parasitic solutions** and they can be shown to converge to 0 as $h \longrightarrow 0$. However, for fixed h with increasing x_n, we would like to keep the parasitic solutions relatively small compared to the principal part $\gamma_0 [r_0(h\lambda)]^n$.

Definition 4.7.12.

1. We say that (LMM) is **relatively stable** if

   $$|r_j(h\lambda)| \leqslant r_0(h\lambda), \quad j = 1, 2, \ldots, p,$$

   for all sufficiently small nonzero values of $|h\lambda|$.

2. (LMM) is said to satisfy the **strong root condition** if

   $$|r_j(0)| < 1, \quad j = 1, 2, \ldots, p.$$

3. If (LMM) is stable but not relatively stable, then it is called **weakly stable**.

Remark 4.7.13. Using continuity of the characteristic roots $r_j(h\lambda)$ with respect to $h\lambda$, it can be shown that the strong root condition implies relative stability. The converse does not necessarily hold, although they are equivalent for most practical methods.

Example 4.7.14. The characteristic polynomial for the Adams-Bashforth and Adams-Moulton methods in the case $h = 0$ coincides and has the form

$$p(r) = r^{p+1} - r^p,$$

which has roots $r_0(0) = 1, r_j(0) = 0, j = 1, 2, \ldots, p$. Thus the strong root condition is satisfied and the Adams methods are relatively stable.

Example 4.7.15. Integrating $Y' = f(x, Y)$ over the interval $[x_{n-1}, x_{n+1}]$ yields:

$$Y(x_{n+1}) = Y(x_{n-1}) + \int_{x_{n-1}}^{x_{n+1}} f(t, Y(t))\, dt.$$

Applying the (Simple Midpoint) rule onto the integral gives the **midpoint method**:

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n), \quad n \geqslant 1, \tag{Midpoint}$$

with local truncation error

$$T_n(Y) = \frac{h^3}{3} f''(\xi_n) = \frac{h^3}{8} Y^{(3)}(\xi_n) \quad \text{for some } \xi_n \in [x_{n-1}, x_{n+1}].$$

It is an explicit 2-step method and the order of convergence is 2. Now, applying to the model problem, the method reduces to

$$y_{n+1} = y_{n-1} + 2h\lambda y_n, \quad n \geqslant 1,$$

and the characteristic polynomial ($p = 1$) is

$$p(r) = r^2 - 1 - 2h\lambda r.$$

This has roots

$$r = \frac{2h\lambda \pm \sqrt{4h^2\lambda^2 + 4}}{2} = h\lambda \pm \sqrt{1 + h^2\lambda^2}.$$

We see that $r_0(h\lambda) = r_+, r_1(h\lambda) = r_-$ and they satisfy:

$$\begin{aligned} r_0(h\lambda) &= 1 + h\lambda + \mathcal{O}(h^2) \\ r_1(h\lambda) &= -1 + h\lambda + \mathcal{O}(h^2). \end{aligned}$$

Consequently, the midpoint method is weakly stable when $\lambda < 0$. We justify this in the case of real λ. When $\lambda > 0$, we have

$$r_0 > |r_1| > 0 \quad \text{for all } h > 0,$$

and the principal part $\gamma_0 r_0^n$ will dominate the parasitic solution $\gamma_1 r_1^n$. When $\lambda < 0$, we have that for all $h > 0$:

$$0 < r_0 < 1, \quad r_1 < -1.$$

This means that the parasitic solution will eventually dominate the principal part as n increases, for fixed h, no matter how small h is chosen initially. Indeed, $\gamma_0 r_0^n \longrightarrow 0$ as $n \longrightarrow \infty$ whereas $\gamma_1 r_1^n$ oscillates while increases in its magnitude.

4.8 Problems

1. Derive a Taylor method of order 3 for the following initial value problem

$$y'(x) = -y(x) + 2\sin(x), \quad y(0) = -1. \tag{4.8.1}$$

What is the truncation error of the method? What is the exact solution for the IVP?

Solution: We first solve the given initial value problem (4.8.1). Solving the homogeneous part of (4.8.1) gives the complimentary solution $Y_c(x) = Ce^{-x}$. We guess a particular solution of the form

$$Y_p(x) = A\cos(x) + B\sin(x).$$

Computing its first derivative and substituting into (4.8.1), we obtain

$$\begin{aligned} Y_p' + Y_p &= -A\sin(x) + B\cos(x) + A\cos(x) + B\sin(x) \\ &= (B + A)\cos(x) + (B - A)\sin(x) \\ &= 2\sin(x). \end{aligned}$$

This yields two linear equations

$$\begin{cases} B + A &= 0 \\ B - A &= 2, \end{cases}$$

and solving these yields $A = -1, B = 1$, *i.e.* $y_p(x) = \sin(x) - \cos(x)$. Thus, the general solution of (4.8.1) has the form

$$\begin{aligned} Y(x) = Y_c(x) + Y_p(x) &= Ce^{-x} + \sin(x) - \cos(x) \\ &= \sin(x) - \cos(x), \end{aligned}$$

where $C = 0$ is found using the initial condition $y(0) = -1$.

Denote $Y(x_n) = Y_n, n \geqslant 0$. To find the Taylor's method of order 3 for (4.8.1), consider the third order Taylor expansion of $Y(x_{n+1})$ around x_n:

$$Y_{n+1} = Y_n + hY_n' + \frac{h^2}{2}Y_n'' + \frac{h^3}{6}Y_n^{(3)} + \underbrace{\frac{h^4}{24}Y^{(4)}(\xi_n)}_{\text{local truncation error}}, \tag{4.8.2}$$

for some $\xi_n \in [x_n, x_{n+1}]$. Next, we compute derivatives of $Y(x)$ using (4.8.1):

$$\begin{aligned} Y'(x) &= -Y(x) + 2\sin(x) \\ Y''(x) &= -Y'(x) + 2\cos(x) \\ &= -[-Y(x) + 2\sin(x)] + 2\cos(x) \\ &= Y(x) - 2\sin(x) + 2\cos(x) \\ Y^{(3)}(x) &= Y'(x) - 2\cos(x) - 2\sin(x) \end{aligned}$$

$$\begin{aligned} &= [-Y(x) + 2\sin(x)] - 2\cos(x) - 2\sin(x) \\ &= -Y(x) - 2\cos(x) \\ Y^{(4)}(x) &= -Y'(x) + 2\sin(x) \\ &= -[-Y(x) + 2\sin(x)] + 2\sin(x) \\ &= Y(x). \end{aligned}$$

Substituting these into (4.8.2) and dropping the local truncation error, we obtain the Taylor's method of order 3 for (4.8.1):

$$\begin{aligned} y_{n+1} = y_n + h\Big[- y_n + 2\sin(x_n)\Big] + \frac{h^2}{2}\Big[y_n - 2\sin(x_n) + 2\cos(x_n)\Big] \\ + \frac{h^3}{6}\Big[- y_n - 2\cos(x_n)\Big] \\ = \left(1 - h + \frac{h^2}{2} - \frac{h^3}{6}\right) y_n + \left(2h - h^2\right)\sin(x_n) + \left(h^2 - \frac{h^3}{3}\right)\cos(x_n). \end{aligned}$$

for every $n \geqslant 0$, with $y_0 = -1$. Since $Y^{(4)}(x) = Y(x)$, the local truncation error for each step is

$$T_n(Y) = \frac{h^4}{24} Y(\xi_n) \quad \text{for some } \xi_n \in [x_n, x_{n+1}],\ n \geqslant 1.$$

2. Construct an example of (using definitions and theory discussed in class, such as root conditions, consistency condition, etc):

   (a) a consistent but not stable linear multistep method;

Solution: Consider the following 2-step explicit linear multistep method:

$$y_{n+1} = 3y_n - 2y_{n-1} - hf(x_{n-1}, y_{n-1}), \quad n \geqslant 1.$$

We first show that it is consistent by checking the consistency condition in Theorem 4.6.2:

$$\sum_{j=0}^{1} a_j = a_0 + a_1 = 3 - 2 = 1$$

$$-\sum_{j=0}^{1} j a_j + \sum_{j=-1}^{1} b_j = -a_1 + b_{-1} + b_0 + b_1 = -(-2) - 1 = 1.$$

To check the root condition, solving $\rho(r) = 0$ yields:

$$\rho(r) = r^2 - 3r + 2 = (r-2)(r-1) = 0 \implies r = 1 \quad \text{or} \quad r = 2.$$

Thus, the root condition is violated and the proposed scheme is not stable. For this scheme, the numerical solution will grow unboundedly as $h \longrightarrow 0, n \longrightarrow \infty$ with $nh = t$ fixed.

   (b) a stable but not consistent linear multistep method.

Solution: Consider the general form of 1-step explicit linear multistep method:

$$y_{n+1} = a_0 y_n + h b_0 f(x_n, y_n). \tag{4.8.3}$$

Clearly, (4.8.3) is stable if and only if $|a_0| \leqslant 1$ (from the root condition), and not consistent if and only if $a_0 \neq 1$. Choosing $a_0 = 1/2$ and $b_0 = 0$ thus yields a stable but not consistent linear multistep method:

$$y_{n+1} = \frac{1}{2} y_n = \left(\frac{1}{2}\right)^{n+1} y_0, \quad n \geqslant 0.$$

The numerical solution does not blow up, but it does not converge to the true solution of the initial value problem. Indeed, we see that $y_n \longrightarrow 0$ as $n \longrightarrow \infty$, independent of the initial value y_0.

3. Find the range of $a \in \mathbb{R}$ for which the method

$$y_{n+2} + (a-1)y_{n+1} - a y_n = \frac{h}{4}\Big[(a+3) f(t_{n+2}, y_{n+2}) + (3a+1) f(t_n, y_n)\Big], \tag{4.8.4}$$

is consistent and stable.

Solution: First, rearranging (4.8.4) into the general form of linear multistep method:

$$y_{n+2} = (1-a)y_{n+1} + a y_n + \frac{h}{4}\Big[(a+3) f(t_{n+2}, y_{n+2}) + (3a+1) f(t_n, y_n)\Big].$$

Clearly,

$$a_0 = (1-a), \quad a_1 = a, \quad b_{-1} = \frac{a+3}{4}, \quad b_0 = 0, \quad b_1 = \frac{3a+1}{4}. \tag{4.8.5}$$

In order for (4.8.4) to be consistent, we require that

$$\sum_{j=0}^{1} a_j = a_0 + a_1 = 1 \tag{4.8.6a}$$

$$-\sum_{j=0}^{1} j a_j + \sum_{j=-1}^{1} b_j = -a_1 + b_{-1} + b_0 + b_1 = 1. \tag{4.8.6b}$$

Upon substituting (4.8.5) into (4.8.6), we obtain that the numerical method (4.8.4) is consistent for any $a \in \mathbb{R}$. Indeed,

$$a_0 + a_1 = (1-a) + a = 1$$

$$-a_1 + b_{-1} + b_0 + b_1 = -a + \frac{a+3}{4} + 0 + \frac{3a+1}{4} = \frac{-4a + a + 3 + 3a + 1}{4} = 1.$$

To determine the range of $a \in \mathbb{R}$ for stability, we first write out the polynomial

$$\rho(r) = r^{1+1} - \sum_{j=0}^{1} a_j r^{1-j} = r^2 - (1-a)r - a.$$

Solving $\rho(r) = 0$ using quadratic formula gives:

$$r_\pm = \frac{(1-a) \pm \sqrt{(1-a)^2 + 4a}}{2} = \frac{(1-a) \pm \sqrt{(a+1)^2}}{2} = \frac{(1-a) \pm (a+1)}{2}.$$

The roots are

$$r_+ = \frac{1-a+a+1}{2} = 1, \quad r_- = \frac{1-a-(a+1)}{2} = -a.$$

Now, in order for (4.8.4) to be stable, both r_+, r_- must satisfy the root condition. The first one states that r_+, r_- must lie in the closed unit disk, which imposes $|a| \leqslant 1$. The second one states that roots on the boundary must be simple. For r_+, we require

$$\rho'(r_+) = 2r_+ - (1-a) = a + 1 \neq 0 \iff a \neq -1.$$

The root r_- lies on the boundary for $a = \pm 1$, but since we already exclude $a = -1$, we simply need to check whether r_- for $a = 1$, *i.e.* $r_- = -1$, is simple:

$$\rho'(-1) = 2(-1) - (1-1) = -2 \neq 0.$$

Thus, the numerical method (4.8.4) is stable for any $a \in (-1, 1]$. Consequently, the numerical method (4.8.4) is consistent and stable if and only if $a \in (-1, 1]$.

4. Show that the region of absolute stability for the trapezoidal method is the set of all complex $h\lambda$ with $\mathrm{Re}(\lambda) < 0$.

Solution: Applying the trapezoidal method to the model equation $y' = \lambda y$ yields

$$y_{n+1} = y_n + \frac{h}{2}[\lambda y_n + \lambda y_{n+1}], \quad n \geqslant 0,$$

and rearranging this yields

$$\left(1 - \frac{h\lambda}{2}\right) y_{n+1} = \left(1 + \frac{h\lambda}{2}\right) y_n \implies y_{n+1} = \left(\frac{1 + \frac{h\lambda}{2}}{1 - \frac{h\lambda}{2}}\right) y_n = \left(\frac{2 + h\lambda}{2 - h\lambda}\right) y_n. \tag{4.8.7}$$

Upon iterating the recursive equation (4.8.7), we obtain

$$y_n = \left(\frac{2 + h\lambda}{2 - h\lambda}\right)^n y_0, \quad n \geqslant 0,$$

and $y_n \longrightarrow 0$ as $x_n \longrightarrow \infty$ if and only if $\left|\dfrac{2+h\lambda}{2-h\lambda}\right| < 1$. To find the region of absolute stability, we simply expand this inequality:

$$\begin{aligned} |2+h\lambda|^2 &< |2-h\lambda|^2 \\ [2+h\,\mathrm{Re}(\lambda)]^2 + \cancel{[h\,\mathrm{Im}(\lambda)]^2} &< [2-h\,\mathrm{Re}(\lambda)]^2 + \cancel{[-h\,\mathrm{Im}(\lambda)]^2} \\ \cancel{4} + 4h\,\mathrm{Re}(\lambda) + \cancel{[h\,\mathrm{Re}(\lambda)]^2} &< \cancel{4} - 4h\,\mathrm{Re}(\lambda) + \cancel{[h\,\mathrm{Re}(\lambda)]^2} \\ 8h\,\mathrm{Re}(\lambda) &< 0. \end{aligned}$$

Since $\mathrm{Re}(\lambda)$ is assumed to be negative, we see that this inequality holds for all $h > 0$. Hence, the region of absolute stability for the trapezoidal method is the set of all complex $h\lambda$ with $\mathrm{Re}(\lambda) < 0$.

Chapter 5

Numerical Methods for PDEs

The main focus of this chapter is finite difference methods for solving partial differential equations (PDEs) on a bounded domain, coupled with initial and boundary conditions. The basic idea behind finite difference methods is to seek approximations to solutions of PDEs on a lattice (grid) by replacing derivatives with finite difference approximations, *i.e.* derivatives are approximated with suitably weighted differences of lattice values at neighbouring points called *difference formulas*, see Section 5.1.1. Different choices of finite difference approximations lead to numerical schemes with different properties, this is demonstrated using three classical PDEs:

1. Heat equation, which is of *parabolic* type,
2. Advection equation, which is of *hyperbolic* type,
3. Poisson equation, which is of *elliptic* type.

Solutions to a given PDEs often possess certain properties which are intrinsic to the PDEs itself. As such, these desirable properties should carry over when designing and testing numerical methods.

Among the important step in setting up finite difference methods is discretising the domain of interest into a mesh grid, the hope being that the numerical solution converges to the true solution of the PDE as the mesh spacing, typically Δx and Δt, both go to zero at arbitrary independent rates. It turns out that convergence is expected only if Δx and Δt go to zero at some specific rate, depending on both the scheme and the PDEs. Similar to numerical method of ODEs, we will discuss consistency (accuracy) and stability of finite difference methods. This leads to the *fundamental theorem of finite difference methods*, which says that under suitable definition of stability, we may expect that stability is a sufficient condition for convergence of consistent finite difference methods.

5.1 Background

This section forms the basis for many of the development of finite difference methods later. We begin by deriving relevant difference formulas that we would use throughout this chapter. For linear PDEs, applying finite difference method reduces the problem to solving a large but finite matrix system. For reasons that will become apparent later, we study the spectrum of these related finite difference matrices. Because we are solving difference equations, these numerical

solutions are functions defined on a discrete set of points; as such, some discrete version of norms are needed to correctly measure the magnitude of the error and this is done in Subsection 5.1.3.

5.1.1 Difference Formulas

Difference formulas are obtained using Taylor series and higher-order approximations are obtained by retaining higher order terms in the Taylor series. For simplicity, we will work with functions of one variable, say, $u : \mathcal{D} \longrightarrow \mathbb{R}$ for some open subset $\mathcal{D} \subset \mathbb{R}$ but the same idea extends to multivariable functions. Define discrete points $x_j = jh \in \mathcal{D}, j = 0, 1, \ldots$ for some constant $h > 0$.

1. First-order forward difference: Expanding $u(x_{j+1})$ around the point x_j gives:

$$\begin{aligned} & u(x_{j+1}) = u(x_j) + hu'(x_j) + \frac{h^2}{2}u''(\xi_j) \\ \implies & u'(x_j) = \frac{u(x_{j+1}) - u(x_j)}{h} - \frac{h}{2}u''(\xi_j) := \frac{\delta_+ u(x_j)}{h} + \mathcal{O}(h), \end{aligned} \tag{1stFD}$$

   for some $\xi_j \in [x_j, x_{j+1}]$.

2. First-order backward difference: Expanding $u(x_{j-1})$ around the point x_j gives:

$$\begin{aligned} & u(x_{j-1}) = u(x_j) - hu'(x_j) + \frac{h^2}{2}u''(\xi_j) \\ \implies & u'(x_j) = \frac{u(x_j) - u(x_{j-1})}{h} - \frac{h}{2}u''(\xi_j) := \frac{\delta_- u(x_j)}{h} + \mathcal{O}(h), \end{aligned} \tag{1stBD}$$

   for some $\xi_j \in [x_{j-1}, x_j]$.

These two approximations give *first order accurate* approximations to $u'(x_j)$, *i.e.* the size of the error is $\mathcal{O}(h)$ as $h \longrightarrow 0$. To obtain a *second order accurate* approximation to $u'(x_j)$, expanding $u(x_{j+1})$ and $u(x_{j-1})$ around x_j gives:

$$u(x_{j+1}) = u(x_j) + hu'(x_j) + \frac{h^2}{2}u''(x_j) + \frac{h^3}{6}u^{(3)}(x_j) + \mathcal{O}(h^4) \tag{5.1.1a}$$

$$u(x_{j-1}) = u(x_j) - hu'(x_j) + \frac{h^2}{2}u''(x_j) - \frac{h^3}{6}u^{(3)}(x_j) + \mathcal{O}(h^4) \tag{5.1.1b}$$

Substracting (5.1.1b) from (5.1.1a) and rearranging yields:

$$\begin{aligned} u'(x_j) &= \frac{u(x_{j+1}) - u(x_{j-1})}{2h} - \frac{h^2}{6}u^{(3)}(x_j) + \mathcal{O}(h^3) \\ &:= \frac{\delta_c u(x_j)}{2h} + \mathcal{O}(h^2), \end{aligned} \tag{1stCD}$$

Now, suppose we expand $u(x_{j+1})$ and $u(x_{j-1})$ around x_j to higher order:

$$u(x_{j+1}) = u(x_j) + hu'(x_j) + \frac{h^2}{2}u''(x_j) + \frac{h^3}{6}u^{(3)}(x_j) + \frac{h^4}{24}u^{(4)}(x_j) + \mathcal{O}(h^6) \tag{5.1.2a}$$

$$u(x_{j-1}) = u(x_j) - hu'(x_j) + \frac{h^2}{2}u''(x_j) - \frac{h^3}{6}u^{(3)}(x_j) + \frac{h^4}{24}u^{(4)}(x_j) + \mathcal{O}(h^6) \tag{5.1.2b}$$

Adding (5.1.2a) and (5.1.2b) and rearranging yields the standard second-order central difference approximation for $u''(x_j)$:

$$\begin{aligned} u''(x_j) &= \frac{u(x_{j+1}) - 2u(x_j) + u(x_{j-1})}{h^2} - \frac{h^2}{12}u^{(4)}(x_j) + \mathcal{O}(h^4) \\ &:= \frac{\delta_c^2 u(x_j)}{h^2} + \mathcal{O}(h^2), \end{aligned} \tag{2ndCD}$$

Observe that either odd or even order terms will cancel out for symmetric centred approximations and typically leads to higher order approximations. All the terms with Big-O notations are the local truncation errors of the corresponding finite difference approximations, which are caused from truncating the Taylor series.

Remark 5.1.1. If we expect the error to behave like powers of h, then we can plot the error against h on a log-log scale, since if the error behaves like

$$\text{Error} \approx Ch^p,$$

then

$$\log|\text{Error}| \approx \log|C| + p\log h,$$

i.e. on a log-log scale the error behaves linearly with slope p, where p is the order of accuracy.

5.1.2 Tridiagonal Matrix

Matrices with constant on diagonals, called **Toeplitz matrices**, arise frequently not only in finite difference approximations, but also finite element and spectral approximations of PDEs. Examples include tridiagonal and circulant matrices. Here, we only state and prove results about the spectrum of tridiagonal matrix, which can be extremely useful in estimating the matrix 2-norm under special circumstances.

Theorem 5.1.2. *Consider the eigenvalue problem $Au^{(p)} = \lambda_p u^{(p)}$, where $A \in \mathbb{R}^{n\times n}$ is the tridiagonal matrix*

$$\begin{bmatrix} a & b & & & \\ b & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & b \\ & & & b & a \end{bmatrix},$$

with $a, b \in \mathbb{R}$. Let $h = 1/(n+1)$. For each $p = 1, 2, \ldots, n$, the eigenpairs $(\lambda_p, u^{(p)})$ are defined by:

$$\begin{aligned} \lambda_p &= a + 2b\cos(p\pi h) \\ u_j^{(p)} &= \sin(p\pi jh), \quad j = 1, \ldots, n \end{aligned}$$

Proof. Examining the eigenvalue equation $Au^{(p)} = \lambda_p u^{(p)}$ component-wise, we obtain:

$$bu_{j-1}^{(p)} + au_j^{(p)} + bu_{j+1}^{(p)} = \lambda_p u_j^{(p)}, \quad j = 1, 2, \ldots, n,$$

with the convention $u_0^{(p)} = u_{n+1}^{(p)} = 0$. We verify directly that the given expression for $(\lambda_p, u^{(p)})$ are eigenpairs of A. For $j = 2, \ldots, n-1$,

$$\begin{aligned}
bu_{j-1}^{(p)} + au_j^{(p)} + bu_{j+1}^{(p)} &= b\sin(p\pi(j-1)h) + a\sin(p\pi jh) + b\sin(p\pi(j+1)h) \\
&= b\sin(p\pi jh - p\pi h) + a\sin(p\pi jh) + b\sin(p\pi jh + p\pi h) \\
&= a\sin(p\pi jh) + b\Big[\sin(p\pi jh + p\pi h) + \sin(p\pi jh - p\pi h)\Big] \\
&= a\sin(p\pi jh) + b\Big[2\sin(p\pi jh)\cos(p\pi h)\Big] \\
&= [a + 2b\cos(p\pi h)]\sin(p\pi jh) \\
&= \lambda_p u_j^{(p)}.
\end{aligned}$$

For $j = 1$, we apply the double-angle formula:

$$\begin{aligned}
bu_0^{(p)} + au_1^{(p)} + bu_2^{(p)} &= au_1^{(p)} + bu_2^{(p)} \\
&= a\sin(p\pi h) + b\sin(p\pi 2h) \\
&= a\sin(p\pi h) + b\Big[2\sin(p\pi h)\cos(p\pi h)\Big] \\
&= [a + 2b\cos(p\pi h)]\sin(p\pi h) \\
&= \lambda_p u_1^{(p)}
\end{aligned}$$

For $j = n$, we apply the difference formula for sine:

$$\begin{aligned}
bu_{n-1}^{(p)} + au_n^{(p)} + bu_{n+1}^{(p)} &= bu_{n-1}^{(p)} + au_n^{(p)} \\
&= b\sin(p\pi(n-1)h) + a\sin(p\pi nh) \\
&= b\Big[\sin(p\pi nh)\cos(p\pi h) - \cos(p\pi nh)\sin(p\pi h)\Big] + a\sin(p\pi nh) \\
&= [a + 2b\cos(p\pi h)]\sin(p\pi nh),
\end{aligned}$$

where the last equality follows provided we can show that $-\cos(p\pi nh)\sin(p\pi h) = \sin(p\pi nh)\cos(p\pi h)$. The trick is to rewrite $\cos(p\pi nh)$ and $\sin(p\pi h)$ in terms of complex exponential:

$$\begin{aligned}
-\cos(p\pi nh)\sin(p\pi h) &= -\frac{1}{4i}(e^{ip\pi nh} + e^{-ip\pi nh})(e^{ip\pi h} - e^{-ip\pi h}) \\
&= -\frac{1}{4i}\left(e^{ip\pi(n+1)h} - e^{ip\pi nh}e^{-ip\pi h} + e^{-ip\pi nh}e^{ip\pi h} - e^{-ip\pi(n+1)h}\right) \\
&= \frac{1}{4i}\left(e^{-ip\pi(n+1)h} + e^{ip\pi nh}e^{-ip\pi h} - e^{-ip\pi nh}e^{ip\pi h} - e^{ip\pi(n+1)h}\right) \\
&= \frac{1}{4i}\left(e^{ip\pi(n+1)h} + e^{ip\pi nh}e^{-ip\pi h} - e^{-ip\pi nh}e^{ip\pi h} - e^{-ip\pi(n+1)h}\right) \\
&= \frac{1}{4i}\Big[e^{ip\pi nh}(e^{ip\pi h} + e^{-ip\pi h}) - e^{-ip\pi nh}(e^{ip\pi h} + e^{-ip\pi h})\Big] \\
&= \frac{1}{4i}\left(e^{ip\pi nh} - e^{-ip\pi nh}\right)(e^{ip\pi h} + e^{ip\pi h})
\end{aligned}$$

$$= \sin(p\pi nh)\cos(p\pi h).$$

The desired result follows since choosing different $p \in \{1, \dots, n\}$ gives n distinct eigenvalues. ■

Very often, tridiagonal matrices arised from finite-difference methods are strictly diagonally dominant. The next theorem tells us that these tridiagonal matrices are in fact non-singular, *i.e.* they are invertible.

Theorem 5.1.3. *Strictly diagonally dominant matrices are invertible.*

Proof. Suppose by contradiction that $A \in \mathbb{R}^{n\times n}$ is a strictly diagonally dominant matrix that is singular. There exists an $x \in \mathbb{R}^n, x \neq \mathbf{0}$ such that $Ax = \mathbf{0}$. Let $J \in \{1, \dots, n\}$ be such that

$$|x_J| = \max_{j=1,\dots,n} |x_j|.$$

Expanding the Jth component of Ax yields:

$$0 = (Ax)_J = \sum_{j=1}^{n} a_{Jj}x_j \implies a_{JJ} = -\sum_{j\neq J}^{n} a_{Jj}\frac{x_j}{x_J}$$

$$|a_{JJ}| \leqslant \sum_{j\neq J}^{n} |a_{Jj}| \left|\frac{x_j}{x_J}\right| \leqslant \sum_{j\neq J}^{n} |a_{Jj}|.$$

This contradicts the assumption that A is strictly diagonally dominant and the statement follows. ■

5.1.3 Discrete Norms

The following discussion is based from [LeV07, pp. 251] and is in the one-dimensional setting but it can be generalised to higher dimension $d > 1$. Choose $\Omega = (0, 1) \subset \mathbb{R}$ and subdivide the interval $[0, 1]$ uniformly into $N + 1$ subintervals. This gives $N + 2$ grid points $x_0, x_1, \dots, x_{N+1}$, with N interior grid points and 2 boundary grid points. Finite difference methods produce a set of discrete values $(u_i)_{i\in\Lambda}$, where Λ is the index set depending on how we incorporate the boundary data into the finite difference methods; typically $\Lambda = 1, \dots, N$. Most of the time, (u_i) approximates the true value of the solution u at the point (x_i), *i.e.* $u_i \approx u(x_i)$ for all $i \in \Lambda$.

In discussing the convergence of numerical methods for ODEs, we measure the magnitude of the error function using certain norms. Because we have a discrete set of approximate solutions here, let us define the **error vector** $E = U - U_{\text{exact}}$, where

$$U = (u_1, \dots, u_N) \quad \text{and} \quad U_{\text{exact}} = (u(x_1), \dots, u(x_N)).$$

Choosing the standard 1-norm:

$$\|E\|_1 = \sum_{j=1}^{N} |e_j|,$$

turns out to provide a wrong interpretation of the magnitude of the error. Indeed, $\|E\|_1$ can be expected to be roughly N times as large as the error at any single grid point and as N increases, $\|E\|_1$ might diverge even if the error at each grid point decreases, which is clearly not the correct behaviour. Since we started by discretising the domain, we should discretise the L^1-norm instead:

$$\|\cdot\|_1 = \int_0^1 (\cdot)(x)\,dx,$$

which suggests the following function:

$$\|E\|_1 = h\sum_{j=1}^{N} |e_j|.$$

Since $h = \dfrac{1}{N+1}$, this discrete-grid-definition of $\|\cdot\|_1$ is just the average value of the error vector E over the interval $[0,1]$. One can show that this function defines a norm. Moreover, $\|E\|_1$ corresponds exactly to a Riemann sum and we recover the standard L^1-norm as $h \longrightarrow 0$. A similar argument shows that the discrete-grid-definition of $\|\cdot\|_p$ for $p \in [1,\infty)$ should be:

$$\|e\|_p = \left(h\sum_{j=1}^{N} |e_j|^p\right)^{1/p}.$$

For the case $p = \infty$, the ∞-norm does not need to be scaled since $h^{1/p} \longrightarrow 1$ as $p \longrightarrow \infty$. Unless stated otherwise, we assume that $\|\cdot\|_p$ takes the discrete-grid-definition when we analyse finite difference methods.

5.1.4 Von Neumann Stability Analysis

The von Neumannn stability analysis (also known as Fourier stability analysis) provides necessary condition for the stability of finite difference schemes as applied to constant coefficient linear PDEs. Due to its simplicity, it is often used in practice to provide a good guess for the time step restriction (if any) used in the scheme. Motivated by the fact that the general solution to linear homogeneous PDEs is a Fourier series, the main idea of the von Neumann stability analysis is to study the effects of finite difference schemes on each of these Fourier modes.

Let $u(x,t)$ be the true solution of some given homogeneous PDEs. Assume an ansatz of the form

$$u(x,t) = w(t)e^{rxI}, \quad \text{where } I^2 = -1 \text{ and } r = \text{ Fourier frequencies.}$$

Evaluating $u(x,t)$ at a discrete grid point (x_i, t_j) yields

$$u(x_i,t_j) \coloneqq u_i^j = w_j e^{rx_iI}. \tag{5.1.3}$$

Substituting (5.1.3) into the given finite difference method yields

$$w_{j+1} = Kw_j = K^{j+1}w_0, \quad j = 0,1,\ldots, \tag{5.1.4}$$

where K is called the **amplification factor** for the method. One might wonder why the same Fourier-type approach works for finite difference operator, the reasons is that the function

e^{rx_iI} is an eigenfunction for any translation-invariant finite difference operator. We see that to obtain a stable method, w_j must remain bounded as $j \longrightarrow \infty$ for any Fourier frequencies r. This means that a necessary condition for the stability of the finite difference scheme is $|K| \leqslant 1$.

Remark 5.1.4. It is important to keep in mind that the von Neumann stability analysis is usually applied to infinite domain or bounded domain with periodic boundary conditions (which is equivalent to a Cauchy problem with periodic initial data). Moreover, it only addresses the issue of stability of the PDE discretisation but does not take into account the discretisation of the boundary conditions (if any), such as Neumann and Robin boundary conditions.

5.1.5 Local Truncation Error and Finite Precision Arithmetic Error

The *local truncation error* (LTE) is the error caused from discretising the differential operator $\mathcal{L}$ of a given PDE, *i.e.* it is the error coming from dropping the higher order terms in the finite difference approximations. It can also be obtained as follows: Suppose $\mathcal{L}u = 0$, then the local truncation error is obtained as $\tau_{ij} = \mathcal{L}_{\text{diff}}u$, where $\mathcal{L}_{\text{diff}}$ is the finite difference operator (discrete differential operator). In other words, it is obtained by replacing the numerical solution with the true solution in the finite difference scheme.

Definition 5.1.5. A finite difference method is said to be **consistent** if the discrete problem approximates the continuous problem, *i.e.* the local truncation error τ_{ij} converges to 0 as the mesh spacing converges to 0.

Convergence is related to the magnitude of the difference between the true solution and the numerical solution from the finite difference scheme. Denote the following quantity:

$$\begin{aligned} u(x_i, t_j) &= \text{exact solution of PDE at } (x_i, t_j) \\ u_i^j &= \text{exact solution of finite difference scheme at } (x_i, t_j) \\ \overline{u_i^j} &= \text{solution of difference equation from finite difference scheme at } (x_i, t_j) \end{aligned}$$

Calculating the error by component, we obtain:

$$l_{ij} = |u(x_i, t_j) - \overline{u_i^j}| \leqslant |u(x_i, t_j) - u_i^j| + \underbrace{|u_i^j - \overline{u_i^j}|}_{\sim 10^{-16}},$$

where the second difference accounts for finite precision arithmetic error.

5.2 Heat Equations and Parabolic Problems

A simple example of a parabolic PDE is the one-dimensional heat equation:

$$u_t - Du_{xx} = 0 \quad \text{in } 0 < x < L,\, t > 0,$$

which models the heat conduction in a rod, with D the thermal diffusivity of the rod. In practice, the solution is obtained only for a finite time $T > 0$. To solve the heat equation, we need to specify initial condition at $t = 0$ and boundary conditions at $x = 0$ and $x = L$:

$$\begin{aligned} u(x,0) &= g(x), \quad 0 \leqslant x \leqslant L \\ u(0,t) &= u_l(t), \quad t > 0 \\ u(L,t) &= u_r(t), \quad t > 0. \end{aligned}$$

For the remaining section, we only deal with Dirichlet boundary conditions, but there are other boundary conditions such as Neumann boundary condition which specify the heat flux across the boundary and Robin boundary condition which are combinations of both Dirichlet and Neumann.

Example 5.2.1. Take $D = 1, L = 1$ and $u_l(t) = u_r(t) = 0$. One can obtain an explicit solution to the heat equation using the separation of variables method:

$$u(x,t) = \sum_{n=1}^{\infty} A_n e^{-n^2\pi^2 t} \sin(n\pi x), \text{ where } A_n = 2\int_0^1 g(x)\sin(n\pi x)\,dx.$$

We list three essential features of solutions to the heat equation:

1. **Smoothing property**, which says that the solution to the heat equation is smooth on the interior even if the initial data are discontinuous.

2. **Maximum principles**, which states that the extremum of the solution occurs on the boundary of the space-time cylinder. An important consequence is that the solution of the heat equation is stable with respect to small perturbation in the initial data.

3. **Instant messaging**, which means that the heat equation transmits information instantly.

Consider the initial-boundary value problem for the inhomogeneous heat equation:

$$\begin{aligned} u_t - u_{xx} &= f(x,t), \quad 0 < x < L,\ t > 0 \\ u(x,0) &= g(x), \quad\quad 0 \leqslant x \leqslant L \\ u(0,t) &= h_l(t), \quad\quad t > 0 \\ u(L,t) &= h_r(t), \quad\quad t > 0. \end{aligned} \tag{Heat}$$

We discretise both the spatial and time domain as follows, which gives a set of discrete grid points (x_i, t_j):

$$\begin{aligned} x_i &= ih, \quad i = 0, 1, \ldots, N+1 \\ t_j &= j\Delta t, \quad j = 0, 1, \ldots, M. \end{aligned}$$

with $h > 0$ the spatial mesh spacing with $(N + 1)$ spatial subintervals and Δt the time step with M time subintervals. For a uniform grid, we have

$$h = \frac{L}{N+1}, \quad \Delta t = \frac{T}{M}.$$

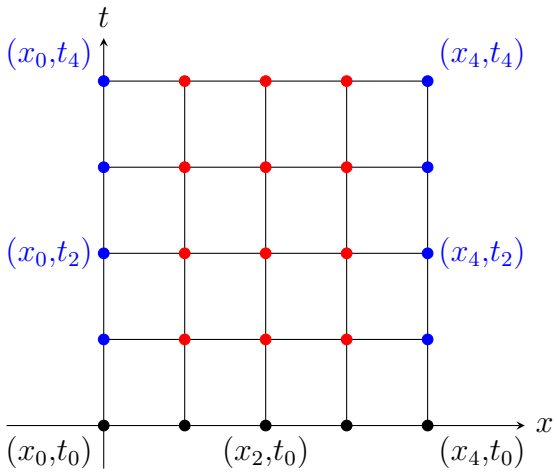


Figure 5.1: An example of a mesh grid with $L = T = 4$ and $N = 3, M = 4$. The black and blue dots represent the initial and boundary data respectively. The red dots are the interior nodes where the approximation solution is computed.

5.2.1 Forward-Time Central-Space (FTCS) Scheme

Let $u_i^j \approx u(x_i, t_j)$ represent the numerical approximation at a grid point (x_i, t_j). At each discrete point (x_i, t_j), we have from (Heat)

$$u_{xx}(x_i, t_j) = u_t(x_i, t_j) - f(x_i, t_j). \tag{5.2.1}$$

The **Forward-Time Central-Space (FTCS) scheme** is obtained by replacing the time derivative u_t by the first-order forward difference approximation (1stFD) and the spatial derivative u_{xx} by the second-order central difference approximation (2ndCD). This gives:

$$\frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j)}{h^2} + \tau_{ij} = \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t} - f(x_i, t_j), \tag{5.2.2}$$

where the local truncation error τ_{ij} has the form

$$\tau_{ij} = \frac{\Delta t}{2} u_{tt}(x_i, \eta_j) - \frac{h^2}{12} u_{xxxx}(x_i, t_j) + \mathcal{O}(h^4) = \mathcal{O}(\Delta t) + \mathcal{O}(h^2).$$

Denote $\lambda = \dfrac{\Delta t}{h^2}$, multiplying each side of (5.2.2) by Δt and rearranging gives:

$$u(x_i, t_{j+1}) = u(x_i, t_j) + \lambda\Big[u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j)\Big] + \Delta t f(x_i, t_j) + \Delta t \tau_{ij}. \tag{5.2.3}$$

Dropping the local truncation error τ_{ij} and setting $u(x_i, t_j) \approx u_i^j, f(x_i, t_j) \approx f_i^j$ in (5.2.3), we obtain:

$$\boxed{u_i^{j+1} = \lambda u_{i+1}^j + (1 - 2\lambda) u_i^j + \lambda u_{i-1}^j + \Delta t f_i^j,} \tag{FTCS}$$

for $i = 1, 2, \ldots, N$ and $j = 0, 1, \ldots, M-1$, together with initial and boundary conditions:

$$\boxed{\begin{aligned} u_i^0 &= g(x_i) = g_i, && i = 0, 1, \ldots, N+1 \\ u_0^j &= h_l(t_j) = h_0^j, && j = 1, \ldots, M \\ u_{N+1}^j &= h_r(t_j) = h_{N+1}^j, && j = 1, \ldots, M \end{aligned}}$$

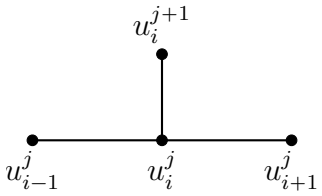


Figure 5.2: Four-point stencil for the FTCS scheme.

The (FTCS) scheme is said to be **first order accurate in time** and **second order accurate in space** since the truncation error is $\mathcal{O}(\Delta t) + \mathcal{O}(h^2)$.

Stencil and Matrix Formulation

Note that the (FTCS) scheme is an explicit numerical method, since u_i^{j+1} is uniquely determined from three values $u_{i-1}^j, u_i^j, u_{i+1}^j$. This yields the four-point stencil for the (FTCS) scheme, as shown in Figure (5.4). Now, define:

$$U_j := \begin{bmatrix} u_1^j \\ u_2^j \\ \vdots \\ u_N^j \end{bmatrix}, \quad F_j = \begin{bmatrix} f_1^j + u_0^j/h^2 \\ f_2^j \\ \vdots \\ f_N^j + u_{N+1}^j/h^2 \end{bmatrix}, \quad G = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_N \end{bmatrix},$$

where we choose to incorporate the boundary conditions into the vector F_j. We can then rewrite the (FTCS) scheme as a matrix system:

$$U_{j+1} = AU_j + \Delta t F_j, \quad j = 0, 1, \ldots, M-1, \quad U_0 = G,$$

where $A \in \mathbb{R}^{N \times N}$ is the symmetric tridiagonal matrix:

$$A = \begin{bmatrix} 1-2\lambda & \lambda & & & \\ \lambda & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \lambda \\ & & & \lambda & 1-2\lambda \end{bmatrix}.$$

Von Neumann Stability Analysis

Assume an ansatz of the form

$$u_i^j = w_j e^{r x_i I}. \tag{5.2.4}$$

Substituting (5.2.4) into the (FTCS) scheme (assuming $f \equiv \mathbf{0}$), we obtain

$$w_{j+1} e^{r x_i I} = \lambda w_j e^{r x_{i+1} I} + (1-2\lambda) w_j e^{r x_i I} + \lambda w_j e^{r x_{i-1} I}$$

$$
\begin{aligned}
&= w_j e^{rx_iI}\Big(\lambda e^{rhI} + (1-2\lambda) + \lambda e^{-rhI}\Big) \\
&= w_j e^{rx_iI}\Big(2\lambda\cos(rh) + (1-2\lambda)\Big) \\
&= w_j e^{rx_iI}\Big(2\lambda[1-2\sin^2(rh/2)] + (1-2\lambda)\Big) \\
&= w_j e^{rx_iI}\Big(1-4\lambda\sin^2(rh/2)\Big),
\end{aligned}
$$

where we use the following formula:

$$2\cos\theta = e^{I\theta} + e^{-I\theta} \quad \text{and} \quad \cos(2\theta) = 1 - 2\sin^2\theta.$$

Cancelling the exponential term, we obtain

$$w_{j+1} = (1-4\lambda\sin^2(rh/2))w_j = Kw_j = K^{j+1}w_0, \quad j = 0, 1, \ldots. \tag{5.2.5}$$

A necessary condition for the stability of the (FTCS) scheme is:

$$|K| = |1-4\lambda\sin^2(rh/2)| \leqslant 1.$$

The upper inequality is trivially satisfied since $\lambda > 0$. For the lower inequality,

$$1-4\lambda\sin^2(rh/2) \geqslant -1 \implies 2\lambda\sin^2(rh/2) \leqslant 1,$$

and this holds for any r provided $\lambda \leqslant 1/2$. Hence, the (FTCS) scheme is conditionally stable only if:

$$\lambda = \frac{\Delta t}{h^2} \leqslant \frac{1}{2}.$$

For diffusion coefficient $D \neq 1$, a similar analysis shows that the (FTCS) scheme is conditionally stable only if:

$$\lambda \leqslant \frac{1}{2D}.$$

This is expected due to the instant messaging property of the solution to the heat equation.

Remark 5.2.2. This stability condition says that the time step must be much smaller than the mesh size to control the growth of the approximate solution. Moreover, it agrees with the intuition that one requires the finite-difference approximation to have similar orders of accuracy, *i.e.* $\Delta t \approx h^2$.

Global Error

Define the error value $E_i^j = u(x_i, t_j) - u_i^j$. Subtracting the (FTCS) scheme from (5.2.3) yields the error equation:

$$E_i^{j+1} = \lambda E_{i+1}^j + (1-2\lambda)E_i^j + \lambda E_{i-1}^j + \Delta t\tau_{ij}, \tag{5.2.6}$$

for $i = 1, 2, \ldots, N$ and $j = 0, 1, \ldots, M-1$, with initial and boundary conditions:

$$E_i^0 = 0, \quad i = 0, 1, \ldots, N+1$$

$$E_0^j = 0, \quad j = 1, \dots, M$$
$$E_{N+1}^j = 0, \quad j = 1, \dots, M.$$

Assuming $1 - 2\lambda \geqslant 0$, applying triangle inequality gives:

$$|E_i^{j+1}| \leqslant \lambda |E_{i+1}^j| + (1 - 2\lambda)|E_i^j| + \lambda |E_{i-1}^j| + \Delta t |\tau_{ij}|.$$

Next, taking the supremum over all $i = 0, 1, \dots, N+1$ yields:

$$\begin{aligned}
\max_i |E_i^{j+1}| &\leqslant (\lambda + 1 - 2\lambda + \lambda) \max_i |E_i^j| + \Delta t \max_i |\tau_{ij}| \\
&= \max_i |E_i^j| + \Delta t \max_i |\tau_{ij}| \\
&\leqslant \max_i |E_i^0| + \Delta t \left(\sum_{k=0}^{j} \max_i |\tau_{ik}| \right) \\
&= \Delta t \left(\sum_{k=0}^{j} \max_i |\tau_{ik}| \right)
\end{aligned}$$

Finally, taking the supremum over all $j = 0, 1, \dots, M$ yields:

$$\max_{i,j} |E_i^j| \leqslant \Delta t \left(\sum_{k=0}^{M} \max_i |\tau_{ik}| \right) \leqslant M \Delta t \max_{i,j} |\tau_{ij}| = T \max_{i,j} |\tau_{ij}| = \mathcal{O}(\Delta t) + \mathcal{O}(h^2).$$

5.2.2 Backward-Time Central-Space (BTCS) Scheme

Compared to the (FTCS) scheme, the Backward-Time Central-Space (BTCS) scheme is obtained by replacing the time derivative u_t with the first-order backward difference approximation (1stBD). We obtain:

$$\frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j)}{h^2} + \tau_{ij} = \frac{u(x_i, t_j) - u(x_i, t_{j-1})}{\Delta t} - f(x_i, t_j), \tag{5.2.7}$$

where the local truncation error τ_{ij} has the form

$$\tau_{ij} = \frac{\Delta t}{2} u_{tt}(x_i, \eta_j) - \frac{h^2}{12} u_{xxxx}(x_i, t_j) + \mathcal{O}(h^4) = \mathcal{O}(\Delta t) + \mathcal{O}(h^2),$$

for some $\eta_j \in [t_{j-1}, t_j]$. Denote $\lambda = \dfrac{\Delta t}{h^2}$, multiplying each side of (5.2.7) by Δt and rearranging gives:

$$\lambda u(x_{i+1}, t_j) - 2\lambda u(x_i, t_j) + \lambda u(x_{i-1}, t_j) = u(x_i, t_j) - u(x_i, t_{j-1}) - \Delta t f(x_i, t_j) - \Delta t \tau_{ij}. \tag{5.2.8}$$

Dropping the local truncation error τ_{ij} and setting $u(x_i, t_j) \approx u_i^j$, $f(x_i, t_j) \approx f_i^j$ in (5.2.8), we obtain:

$$\boxed{\lambda u_{i+1}^j - (1 + 2\lambda) u_i^j + \lambda u_{i-1}^j = -u_i^{j-1} - \Delta t f_i^j,} \tag{BTCS}$$

for $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$, with initial and boundary conditions:

$$\begin{aligned} u_i^0 &= g(x_i) = g_i, && i = 0, 1, \ldots, N+1 \\ u_0^j &= h_l(t_j) = h_0^j, && j = 0, 1, \ldots, M \\ u_{N+1}^j &= h_r(t_j) = h_{N+1}^j, && j = 0, 1, \ldots, M \end{aligned}$$

The (BTCS) scheme is **first order accurate in time** and **second order accurate in space** since the truncation error is $\mathcal{O}(\Delta t) + \mathcal{O}(h^2)$.

Stencil and Matrix Formulation

Note that the (BTCS) scheme is an implicit numerical method, since one needs to solve a tridiagonal system of equations for all values $\{u_i^j\}_{j=1}^N$ for a particular time step j. The stencil for the (BTCS) scheme is a four-point stencil, as shown in Figure (5.3). Now, define:

$$U_j := \begin{bmatrix} u_1^j \\ u_2^j \\ \vdots \\ u_N^j \end{bmatrix}, \quad F_j := \begin{bmatrix} f_1^j + (u_0^j/h^2) \\ f_2^j \\ \cdots \\ f_N^j + (u_{N+1}^j/h^2) \end{bmatrix}, \quad G := \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_N \end{bmatrix},$$

where we again incorporate the boundary conditions into the vector F_j. We can then rewrite the (BTCS) scheme as a matrix system:

$$BU_j = U_{j-1} + \Delta t F_j, \quad j = 1, 2, \ldots, M, \quad U_0 = G,$$

where B is the symmetric tridiagonal matrix:

$$B = \begin{bmatrix} 1+2\lambda & -\lambda & & & \\ -\lambda & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -\lambda \\ & & & -\lambda & 1+2\lambda \end{bmatrix} \in \mathbb{R}^{N \times N},$$

Note that the matrix B is invertible since B is strictly diagonally dominant. One can show that U_j depends on every component of U_{j-1}, this phenomena is called instant messaging.

Von Neumann Stability Analysis

Assume an ansatz of the form

$$u_i^j = w_j e^{r x_i I}. \tag{5.2.9}$$

Assuming $f \equiv \mathbf{0}$, substituting (5.2.9) into the (BTCS) scheme yields:

$$\lambda w_j e^{r(x_i+h)I} - (1+2\lambda) w_j e^{r x_i I} + \lambda w_j e^{r(x_i-h)I} = -w_{j-1} e^{r x_i I}$$

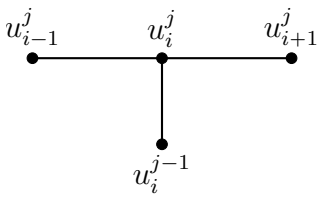


Figure 5.3: Four-point stencil for the BTCS scheme.

$$
\begin{aligned}
w_j \cancel{e^{rx_i I}}\Big(\lambda e^{rhI} - (1+2\lambda) + \lambda e^{-rhI}\Big) &= -w_{j-1}\cancel{e^{rx_i I}} \\
w_j\Big(2\lambda\cos(rh) - (1+2\lambda)\Big) &= -w_{j-1} \\
w_j\Big(2\lambda[1-2\sin^2(rh/2)] - 1 - 2\lambda\Big) &= -w_{j-1} \\
w_j\Big(4\lambda\sin^2(rh/2) + 1\Big) &= w_{j-1} \\
\implies w_j = \left(\frac{1}{1+4\lambda\sin^2(rh/2)}\right) w_{j-1} &= K w_{j-1}, \quad (5.2.10)
\end{aligned}
$$

and iterating (5.2.10) gives $w_j = K^j w_0, j = 0, 1, \ldots$. Since $|K| \leqslant 1$ for any $\Delta t > 0$, the (BTCS) scheme is unconditionally stable and so it has better stability property compare to the (FTCS) scheme. However, it is still advantageous to choose $\Delta t = \mathcal{O}(h^2)$, since we lose the spatial accuracy if the time step is chosen to be $\Delta t = \mathcal{O}(h)$ instead.

5.2.3 Theta Method

The **theta method** could be viewed as an elegant solution to the following problem: can we combine both the (FTCS) and (BTCS) schemes in such a way that it generates a more accurate method with improved stability property? First, in a more convenient notation we have:

$$
\begin{aligned}
\text{(FTCS)}&: u_i^{j+1} - u_i^j = H_i^j \\
\text{(BTCS)}&: u_i^{j+1} - u_i^j = H_i^{j+1},
\end{aligned}
$$

where

$$
H_i^j = \lambda(u_{i+1}^j - 2u_i^j + u_{i-1}^j) + \Delta t f_i^j.
$$

The idea is to take convex combination of the (FTCS) scheme at time step j and the (BTCS) scheme at time step $j+1$, *i.e.* for any $\theta \in [0,1]$,

$$
\begin{aligned}
u_i^{j+1} - u_i^j &= \theta(u_i^{j+1} - u_i^j) + (1-\theta)(u_i^{j+1} - u_i^j) \\
&= \theta H_i^j + (1-\theta) H_i^{j+1}.
\end{aligned} \qquad \text{(Theta)}
$$

Observe that

$$
\begin{aligned}
&\text{for } \theta = 0\text{: recover the (BTCS) scheme.} \\
&\text{for } \theta = 1\text{: recover the (FTCS) scheme.}
\end{aligned}
$$

Von Neumann Stability Analysis

We assume an ansatz of the form

$$u_i^j = w_j e^{rx_i I}, \quad \text{where } I^2 = -1 \text{ and } r\text{=Fourier frequencies.}$$

Assume $f \equiv \mathbf{0}$, since we expect (Theta) to have the form $w_{j+1} = Kw_j$ for some K depending on $\Delta t, h, \theta, r$, we rearrange it in such a way that the LHS and RHS have terms at time step $j+1$ and j respectively:

$$u_i^{j+1} - \lambda(1-\theta)\Big[u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}\Big] = u_i^j + \lambda\theta\Big[u_{i+1}^j - 2u_i^j + u_{i-1}^j\Big]. \tag{5.2.11}$$

We now substitute the ansatz into (5.2.11) and simplify terms using these two trigonometric identity:

$$2\cos\phi = e^{\phi I} + e^{-\phi I} \quad \text{and} \quad \cos(2\phi) = 1 - 2\sin^2(\phi).$$

For the LHS expression,

$$\begin{aligned}
u_i^{j+1} - \lambda(1-\theta)\Big[u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}\Big] &= w_{j+1}\Big[e^{rx_iI} - \lambda(1-\theta)\Big(e^{rx_iI}e^{rhI} - 2e^{rx_i} + e^{rx_iI}e^{-rhI}\Big)\Big] \\
&= w_{j+1}e^{rx_iI}\Big[1 - \lambda(1-\theta)\Big(e^{rhI} - 2 + e^{-rhI}\Big)\Big] \\
&= w_{j+1}e^{rx_iI}\Big[1 - \lambda(1-\theta)\Big(2\cos(rh) - 2\Big)\Big] \\
&= w_{j+1}e^{rx_iI}\Big[1 - 2\lambda(1-\theta)\Big(\cos(rh) - 1\Big)\Big] \\
&= w_{j+1}e^{rx_iI}\left[1 - 2\lambda(1-\theta)\left(-2\sin^2\left(\frac{rh}{2}\right)\right)\right] \\
&= w_{j+1}e^{rx_iI}\left[1 + 4\lambda(1-\theta)\sin^2\left(\frac{rh}{2}\right)\right],
\end{aligned}$$

and for the RHS expression,

$$\begin{aligned}
u_i^j + \lambda\theta\Big[u_{i+1}^j - 2u_i^j + u_{i-1}^j\Big] &= w_j\Big[e^{rx_iI} + \lambda\theta\Big(e^{rx_iI}e^{rhI} - 2e^{rx_i} + e^{rx_iI}e^{-rhI}\Big)\Big] \\
&= w_j e^{rx_iI}\Big[1 + \lambda\theta\Big(e^{rhI} - 2 + e^{-rhI}\Big)\Big] \\
&= w_j e^{rx_iI}\Big[1 + \lambda\theta\Big(2\cos(rh) - 2\Big)\Big] \\
&= w_j e^{rx_iI}\Big[1 + 2\lambda\theta\Big(\cos(rh) - 1\Big)\Big] \\
&= w_j e^{rx_iI}\left[1 + 2\lambda\theta\left(-2\sin^2\left(\frac{rh}{2}\right)\right)\right] \\
&= w_j e^{rx_iI}\left[1 - 4\lambda\theta\sin^2\left(\frac{rh}{2}\right)\right]
\end{aligned}$$

Equating these two simplified expressions and cancelling the exponential term yields:

$$w_{j+1} = \left(\frac{1 - 4\lambda\theta\sin^2\left(\frac{rh}{2}\right)}{1 + 4\lambda(1-\theta)\sin^2\left(\frac{rh}{2}\right)}\right) w_j = Kw_j.$$

To conclude the stability analysis, we need to determine if there are restrictions on λ such that $|K| \leqslant 1$, keeping in mind that such condition necessarily depends on the value of θ. Since $\lambda > 0$, $1 - \theta \geqslant 0$ for any $\theta \in [0,1]$ and $\sin^2\left(\frac{rh}{2}\right) \geqslant 0$, we see that $K \leqslant 1$ for any $\lambda > 0$, regardless of the choice of $\theta \in [0,1]$. For notational convenience, denote $A = \sin^2\left(\frac{rh}{2}\right)$. Expanding the inequality $K \geqslant -1$ yields:

$$\begin{aligned}
\frac{1 - 4\lambda\theta A}{1 + 4\lambda(1-\theta)A} &\geqslant -1 \\
1 - 4\lambda\theta A &\geqslant -1 - 4\lambda(1-\theta)A \\
1 - 4\lambda\theta A &\geqslant -1 - 4\lambda A + 4\lambda\theta A \\
2 - 8\lambda\theta A &\geqslant -4\lambda A \\
1 - 4\lambda\theta A &\geqslant -2\lambda A \\
1 &\geqslant 4\lambda\theta A - 2\lambda A \\
1 &\geqslant 2\lambda A(2\theta - 1). \qquad (5.2.12)
\end{aligned}$$

If $2\theta - 1 \leqslant 0$, *i.e.* $0 \leqslant \theta \leqslant \frac{1}{2}$, then (5.2.12) is satisfied for any $\lambda > 0$ since both λ and A are nonnegative. On the other hand, if $2\theta - 1 \geqslant 0$, *i.e.* $\frac{1}{2} \leqslant \theta \leqslant 1$,

$$2\lambda A(2\theta - 1) \leqslant 1 \implies A \leqslant \frac{1}{2\lambda(2\theta - 1)} \qquad (5.2.13)$$

Since $A = \sin^2\left(\frac{rh}{2}\right) \leqslant 1$ for any r, h, we see that (5.2.13) is satisfied for any choice of r if

$$\frac{1}{2\lambda(2\theta - 1)} \geqslant 1, \quad i.e.\, \lambda \leqslant \frac{1}{2(2\theta - 1)}.$$

Hence, we obtain the following stability condition for the (Theta) method:

1. If $\theta \in [0, 1/2]$, then it is unconditionally stable, *i.e.* the scheme is stable for any choice of time step.
2. If $\theta \in [1/2, 1]$, then it is stable if λ satisfies $0 \leqslant \lambda \leqslant \frac{1}{2(2\theta - 1)}$.

5.2.4 Crank-Nicolson Method

For the special case $\theta = \frac{1}{2}$ of the (Theta) method, we obtain the **Crank-Nicolson method**:

$$u_i^{j+1} - u_i^j = \Delta t[f_i^j + f_i^{j+1}] + \frac{\lambda}{2}\Big[(u_{i+1}^j - 2u_i^j + u_{i-1}^j) + (u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1})\Big] \qquad \text{(Crank-Nicolson)}$$

which coincides with applying trapezoidal rule on H_i^j. The (Crank-Nicolson) method is **second order accurate in time** and **second order accurate in space** since the local truncation

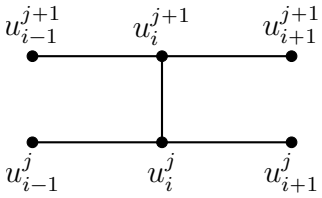


Figure 5.4: Six-point stencil for the Crank-Nicolson method, as well as the Theta method.

error is $\mathcal{O}(\Delta t^2)+\mathcal{O}(h^2)$, as such we can choose the time step $\Delta t = \mathcal{O}(h)$ without losing accuracy in space. Moreover, it is unconditionally stable and is the only second-order theta method in both space and time. The global error satisfies:

$$|u(x_i, t_j) - u_{ij}| = \mathcal{O}(\Delta t^2) + \mathcal{O}(h^2).$$

Rearranging the (Crank-Nicolson) method as follows:

$$\begin{aligned}
u_i^{j+1} - \frac{\lambda}{2}\Big[u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j}\Big] &= u_i^j + \frac{\lambda}{2}\Big[u_{i+1}^{j} - 2u_i^{j} + u_{i-1}^{j}\Big] + \Delta t[f_i^j + f_i^{j+1}] \\
2u_i^{j+1} - \lambda\Big[u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j}\Big] &= 2u_i^j + \lambda\Big[u_{i+1}^{j} - 2u_i^{j} + u_{i-1}^{j}\Big] + 2\Delta t[f_i^j + f_i^{j+1}] \\
-\lambda u_{i+1}^{j+1} + (2+2\lambda)u_i^{j+1} - \lambda u_{i-1}^{j+1} &= \lambda u_{i+1}^{j} + (2-2\lambda)u_i^j + \lambda u_{i-1}^{j} + 2\Delta t[f_i^j + f_i^{j+1}].
\end{aligned}$$

Define the following vectors:

$$U_j := \begin{bmatrix} u_1^j \\ u_2^j \\ \vdots \\ u_N^j \end{bmatrix}, \quad F_j := \begin{bmatrix} f_1^j + (u_0^j/h^2) \\ f_2^j \\ \cdots \\ f_N^j + (u_{N+1}^j/h^2) \end{bmatrix}, \quad G := \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_N \end{bmatrix},$$

We can then rewrite the (Crank-Nicolson) scheme as the matrix system:

$$(B+I)U_{j+1} = (A+I)U_j + 2\Delta t(F_{j+1} + F_j), \quad j = 0, 1, \ldots, M-1, \quad U_0 = G,$$

where $B \in \mathbb{R}^{N\times N}$ is the symmetric tridiagonal matrix arises in the (BTCS) scheme and $A \in \mathbb{R}^{N\times N}$ is the symmetric tridiagonal matrix arises in the (FTCS) scheme.

5.3 Advection Equations and Hyperbolic Systems

Acoustic waves, electromagnetic waves, shock waves and many other types of waves can be modelled by hyperbolic PDEs. The prototype for all hyperbolic PDEs is the **advection equation** which arises when a substance is carried along with a flow; it is the simplest mathematical equation that produces travelling waves. For the Cauchy problem, we need the initial data at time $t = 0$.

$$\begin{cases} u_t + au_x = 0, & -\infty < x < \infty,\ t > 0 \\ u(x,0) = g(x), & -\infty < x < \infty \end{cases} \tag{5.3.1}$$

where a is a constant. Application of advection equation includes gas dynamics and traffic flows, where u is the density of cars and a is the speed of cars.

We claim that the exact solution is of the form

$$u(x,t) = g(x-at). \tag{5.3.2}$$

Indeed,

$$u_t + au_x = -ag'(x-at) + ag'(x-at) = 0.$$

Example 5.3.1. Consider the advection equation with $a > 0$ and with initial condition

$$g(x) = \begin{cases} 1 & \text{if } 0 \leqslant x \leqslant 1, \\ 0 & \text{otherwise.} \end{cases}$$

The solution is given by

$$u(x,t) = g(x-at) = \begin{cases} 1 & \text{if } 0 \leqslant x-at \leqslant 1, \\ 0 & \text{otherwise.} \end{cases}$$

or

$$u(x,t) = \begin{cases} 1 & \text{if } at \leqslant x \leqslant 1+at, \\ 0 & \text{otherwise.} \end{cases}$$

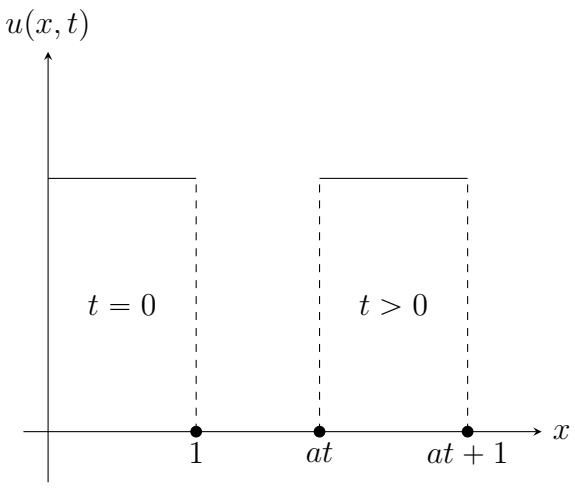


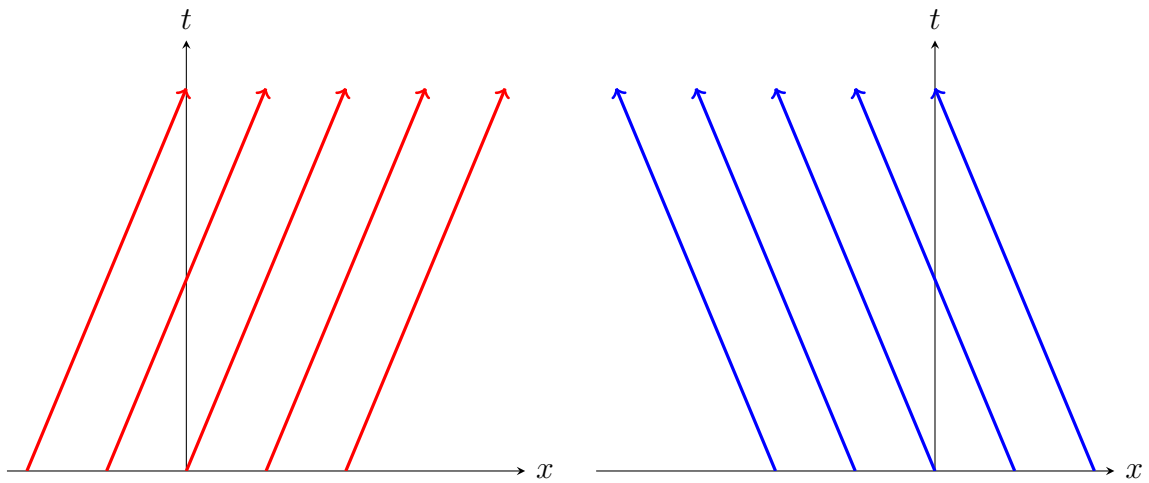
Figure 5.5: The solution travels to the right with speed $a > 0$ without change of shape.

We summarise some of the key observations about the solution (5.3.2):

1. The solution is constant along the lines of the form $x - at =$ constant. These lines are called **characteristics** and information propogates along the characteristics at speed $a > 0$.

2. The solution at a given position $(\bar{x}, \bar{t})$ is determined entirely by the value $x_0 = \bar{x} - a\bar{t}$.

3. The initial shape is preserved. Moreover, the solution formula (5.3.2) requires no differentiability of g. In general, we allow for discontinuous solutions for hyperbolic problems such as shock waves.



(a) $a > 0$ (b) $a < 0$

Figure 5.6: Characteristics of the advection equation for positive and negative a. Observe that the solution at $(\bar{x}, \bar{t})$ will not feel it if we introduce a perturbation around $x_0 = \bar{x} - a\bar{t}$.

5.3.1 Boundary Conditions: Inflow and Outflow Characteristics

On the real line, the initial condition $u(x, 0) = g(x)$ itself uniquely determines the solution so the Cauchy problem (5.3.1) is well-posed. On a finite interval, boundary data are needed and these must be imposed at the correct boundary part, otherwise the problem is overdetermined.

For simplicity, consider the advection equation with $a > 0$ on the domain $\Omega = [0, 1] \times [0, \infty)$. Referring to Figure 5.6, the characteristics propagate from left to right. In particular,

1. The characteristics leaving the left boundary $x = 0$ are travelling towards Ω, *i.e.* these are **inflow characteristics**.

2. The characteristics leaving the right boundary $x = 1$ are travelling away from Ω, *i.e.* these are **outflow characteristics**.

Thus, the problem is well-posed if we impose boundary data $u(0, t) = u_0(t)$ on the left boundary $x = 0$, in addition to the initial condition $u(x, 0) = g(x)$. The resulting solution has the form:

$$u(x,t) = \begin{cases} g(x - at) & \text{if } x - at > 0, \\ u_0\left(t - \dfrac{x}{a}\right) & \text{if } x - at < 0. \end{cases}$$

Note that u will have a jump discontinuity along the characteristic $x - at = 0$ if $u_0(0) \neq g(0)$.

Remark 5.3.2. If $a < 0$, then we have outflow characteristics from the left boundary $x = 0$ and inflow characteristics from the right boundary $x = 1$. Consequently, we must only impose boundary data $u(1, t) = u_1(t)$ on the right boundary $x = 1$. The resulting solution has the form:

$$u(x,t) = \begin{cases} g(x - at) & \text{if } x - at < 1, \\ u_1\left(t - \dfrac{x - 1}{a}\right) & \text{if } x - at > 1. \end{cases}$$

As before, u will have a jump discontinuity along the characteristics $x - at = 1$ if $u_1(1) \neq g(1)$.

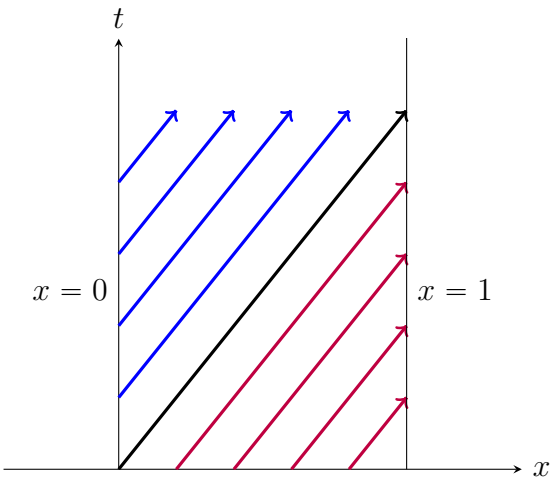


Figure 5.7: With $a > 0$, the purple characteristics carry information from $g(x_0) = g(x - at)$, while the blue characteristics carry information from $u_0\left(t - \frac{x}{a}\right)$.

5.3.2 First-Order Methods

Consider the advection equation on a bounded domain:

$$\begin{aligned} u_t + au_x &= 0 \qquad \text{in } (x_L, x_R) \times (0, T] \\ u(x, 0) &= g(x) \quad \text{on } [x_L, x_R], \end{aligned}$$

where the boundary condition for $t > 0$ is

$$\begin{aligned} u(x_L, t) &= u_L(t) \quad \text{if } a > 0, \\ u(x_R, t) &= u_R(t) \quad \text{if } a < 0. \end{aligned}$$

We discretise this on a uniform grid with spatial mesh spacing $h = \dfrac{x_R - x_L}{N}$ and time step $\Delta t = \dfrac{T}{M}$. This produces a set of discrete grid points (x_i, t_j) defined by:

$$\begin{aligned} x_i &= x_L + ih, \quad i = 0, 1, \ldots, N \\ t_j &= j\Delta t, \qquad j = 0, 1, \ldots, M. \end{aligned}$$

Upwind and Downwind Scheme

We now derive the upwind scheme for $a > 0$ and $a < 0$. We approximate the time derivative u_t with the first-order forward difference (1stFD) which yields explicit method. For the spatial derivative u_x, we could either approximate it by forward or backward differencing in space:

1. For $a > 0$, the upwind scheme is obtained by approximating u_x with the first-order backward difference (1stBD) yields:

$$\frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t} + a\left(\frac{u(x_i, t_j) - u(x_{i-1}, t_j)}{h}\right) - \tau_{ij} = 0, \tag{5.3.3}$$

where the local truncation error τ_{ij} takes the form

$$\tau_{ij} = \frac{\Delta t}{2} u_{tt}(x_i, \eta_j) + \frac{ah}{2} u_{xx}(\xi_i, t_j) = \mathcal{O}(\Delta t) + \mathcal{O}(h). \tag{5.3.4}$$

Denote the dimensionless quantity $\lambda = \dfrac{a\Delta t}{h}$ which is known as the **Courant number**, multiplying each side of (5.3.3) by Δt and rearranging gives:

$$u(x_i, t_{j+1}) = u(x_i, t_j) - \lambda[u(x_i, t_j) - u(x_{i-1}, t_j)] - \Delta t \tau_{ij}. \tag{5.3.5}$$

Dropping τ_{ij} and replacing $u_i^j \approx u(x_i, t_j)$ in (5.3.5), we obtain:

$$\boxed{u_i^{j+1} = (1-\lambda)u_i^j + \lambda u_{i-1}^j,} \qquad \text{(Upwind: } a > 0\text{)}$$

for $i = 1, 2, \ldots, N$ and $j = 0, 1, \ldots, M-1$, together with initial and boundary conditions:

$$\boxed{\begin{aligned} u_i^0 &= g(x_i) = g_i, \quad i = 0, 1, \ldots, N \\ u_0^j &= u_L(t_j), \qquad\quad j = 1, \ldots, M. \end{aligned}}$$

2. For $a < 0$, the upwind scheme is obtained by approximating u_x with the first-order forward difference (1stFD) yields:

$$\frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t} + a\left(\frac{u(x_{i+1}, t_j) - u(x_i, t_j)}{h}\right) + \tau_{ij} = 0, \tag{5.3.6}$$

where the local truncation error τ_{ij} is the same as (5.3.4). Multiplying each side of (5.3.6) by Δt and rearranging gives:

$$u(x_i, t_{j+1}) = u(x_i, t_j) - \lambda[u(x_{i+1}, t_j) - u(x_i, t_j)] - \Delta t \tau_{ij}. \tag{5.3.7}$$

Dropping τ_{ij} and replacing $u_i^j \approx u(x_i, t_j)$ in (5.3.7), we obtain:

$$\boxed{u_i^{j+1} = (1+\lambda)u_i^j - \lambda u_{i+1}^j,} \qquad \text{(Upwind: } a < 0\text{)}$$

for $i = 0, \ldots, N-1$ and $j = 0, 1, \ldots, M-1$, together with initial and boundary conditions:

$$\boxed{\begin{aligned} u_i^0 &= g(x_i) = g_i, \quad i = 0, 1, \ldots, N \\ u_N^j &= u_R(t_j), \qquad\quad j = 1, \ldots, M. \end{aligned}}$$

These two methods are **first order accurate in both time and space**. Moreover, they inherent the asymmetry property of the advection equation, *i.e.* its solution profile translates across time; this is evident by looking at the stencil of these two methods. The stencil also tells us that the choice between these two methods should be dictated by the sign of a. These schemes are called upwind schemes because they retain the important property that the information propogates from left to right (right to left) for $a > 0$ ($a < 0$). The downwind scheme, which is the opposite of upwind scheme, is obtained by forward (backward) differencing for $a > 0$ ($a < 0$).

Figure 5.8: Three-point stencil for the (Upwind: $a > 0$) and (Upwind: $a < 0$) scheme.

Stability via von Neumann Method

Substituting the ansatz $u_i^j = w_j e^{rx_i I}$ into (Upwind: $a > 0$) yields:

$$\begin{aligned} u_i^{j+1} &= (1-\lambda)u_i^j + \lambda u_{i-1}^j \\ w_{j+1}e^{rx_i I} &= (1-\lambda)w_j e^{rx_i I} + \lambda w_j e^{rx_i I}e^{-rhI}. \end{aligned}$$

Cancelling out the exponential term, we obtain:

$$w_{j+1} = w_j\Big[1 - \lambda + \lambda e^{-rhI}\Big] = Kw_j,$$

and we require the amplication factor $|K| \leqslant 1$, *i.e.*

$$\begin{aligned} |1-\lambda+\lambda e^{-irh}|^2 &= \Big[1-\lambda+\lambda\cos(rh)\Big]^2 + \lambda^2\sin^2(rh) \\ &= (1-\lambda)^2 + 2\lambda(1-\lambda)\cos(rh) + \lambda^2\cos^2(rh) + \lambda^2\sin^2(rh) \\ &= 1 - 2\lambda + \lambda^2 + 2\lambda(1-\lambda)\Big[1 - 2\sin^2(rh/2)\Big] + \lambda^2 \\ &= 1 - 2\lambda + 2\lambda^2 + 2\lambda(1-\lambda) - 4\lambda(1-\lambda)\sin^2(rh/2) \\ &= 1 - 4\lambda(1-\lambda)\sin^2(rh/2) \leqslant 1 \\ \implies 0 &\leqslant 4\lambda(1-\lambda)\sin^2(rh/2). \end{aligned}$$

This holds for any r provided $\lambda(1-\lambda) \geqslant 0$, *i.e.* $0 \leqslant \lambda \leqslant 1$. For (Upwind: $a < 0$),

$$\begin{aligned} u_i^{j+1} &= (1+\lambda)u_i^j - \lambda u_{i+1}^j \\ w_{j+1}e^{rx_i I} &= (1+\lambda)w_j e^{rx_i I} - \lambda w_j e^{rx_i I}e^{rhI} \\ w_{j+1} &= w_j\Big[1 + \lambda - \lambda e^{rhI}\Big] = Kw_j. \end{aligned}$$

Expanding $|K|^2$ yields:

$$\begin{aligned} |1+\lambda-\lambda e^{rhI}|^2 &= \Big[1+\lambda-\lambda\cos(rh)\Big]^2 + \lambda^2\sin^2(rh) \\ &= (1+\lambda)^2 - 2\lambda(1+\lambda)\cos(rh) + \lambda^2\cos^2(rh) + \lambda^2\sin^2(rh) \\ &= 1 + 2\lambda + 2\lambda^2 - 2\lambda(1+\lambda)\Big[1 - 2\sin^2(rh/2)\Big] \\ &= 1 + 2\lambda + 2\lambda^2 - 2\lambda(1+\lambda) + 4\lambda(1+\lambda)\sin^2(rh/2) \\ &= 1 + 4\lambda(1+\lambda)\sin^2(rh/2) \leqslant 1 \\ \implies 0 &\leqslant -4\lambda(1+\lambda)\sin^2(rh/2). \end{aligned}$$

This holds for any r provided $\lambda(1+\lambda) \leqslant 0$, *i.e.* $-1 \leqslant \lambda \leqslant 0$.

5.3.3 The Courant-Friedrichs-Lewy (CFL) Condition

The CFL condition is a necessary condition for convergence of finite difference methods, typically time-explicit methods for hyperbolic problems. Roughly speaking, it states that a necessary condition for convergence is that the numerical method must be able to access the information required such as initial data to form the correct solution. Otherwise, we can change such data and hence change the true solution without having any effect on the numerical solution, so the method cannot be convergent for general initial data. To make this rigorous, we introduce the concept of **domain of dependence** which is especially important in the study of hyperbolic system.

Definition 5.3.3. For wave problems, the **continuous domain of dependence** for the solution at $(\bar{x}, \bar{t})$ consists of all the points on the x-axis ($t = 0$) that contributes to the solution at $(\bar{x}, \bar{t})$. For the advection equation, the domain of dependence for $(\bar{x}, \bar{t})$ is the single point $(\bar{x} - a\bar{t}, 0)$. Consequently, the solution at $(\bar{x}, \bar{t})$ changes if and only if $g(\bar{x} - a\bar{t})$ changes.

Definition 5.3.4. On a particular fixed grid, we define the **numerical domain of dependence** of a discrete grid point (x_i, t_j) to be the set of grid points at the initial time t_0 such that their corresponding data contributes to the solution u_i^j.

Consider the (Upwind: $a > 0$) scheme. The value u_i^j depends on u_{i-1}^{j-1} and u_i^{j-1}, which in turn depends on u_{i-2}^{j-2} to u_i^{j-2}. Tracing back to the initial time $t_0 = 0$, we obtain a right triangular array of grid points and we see that u_i^j depends on the initial data at the grid points $\{x_{i-j}, x_{i-j+1}, \ldots, x_i\}$. A similar argument shows that the domain of dependence of u_i^j for the (Upwind: $a < 0$) scheme is the set of grid points $\{x_i, x_{i+1}, \ldots, x_{i+j}\}$. Keeping the ratio $\dfrac{\Delta t}{h}$ fixed, as we refine the grid, we see that the value u_i^j for the (Upwind: $a > 0$) scheme depends on more values of the initial data, but these values all lie within the same interval $[x_{i-j}, x_i]$. Moreover, the numerical domain of dependence will fill in the interval $[x_{i-j}, x_j]$ in the limit as the time step Δt and spatial width h go to zero.

Recall that we want the numerical solution u_i^j converges to the true solution $u(x_i, t_j)$ as the mesh width goes to 0. This gives rise to the **CFL condition**:

> A numerical method can be convergent only if its numerical domain of dependence contains the true domain of dependence of the PDE, at least in the limit as h and Δt go to zero.

Remark 5.3.5. We stress again that in general the CFL condition is only a *necessary* condition for convergence of finite difference methods for hyperbolic PDEs. In addition to this and the consistency requirement, a proper stability analysis is required to determine the stability restriction on Δt and h.

Referring to Figure 5.9, for the (Upwind: $a > 0$) scheme the CFL condition requires that $x_{i-j} \leqslant x_i - at_j$, *i.e.*

$$x_i - jh \leqslant x_i - aj\Delta t$$

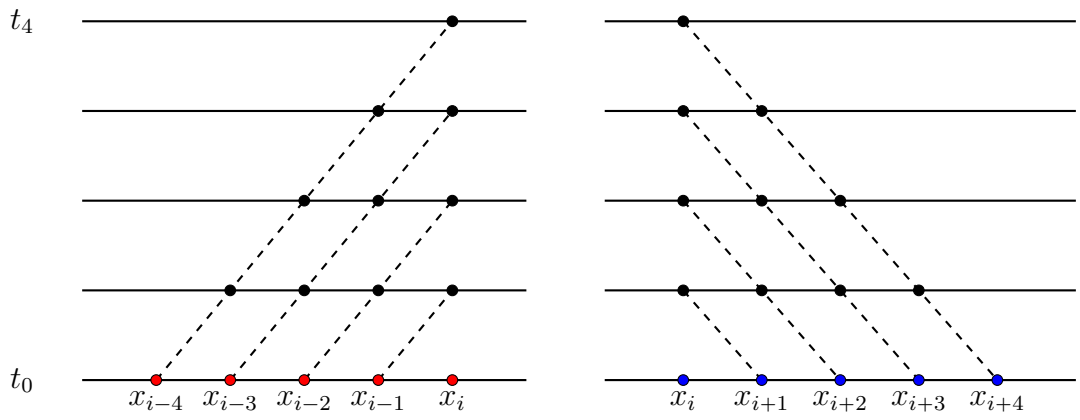


Figure 5.9: Numerical domain of dependence of (x_i, t_4) for the (Upwind: $a > 0$) and (Upwind: $a < 0$) scheme.

$$\begin{aligned} -h &\leqslant -a\Delta t \\ 0 &\leqslant \frac{a\Delta t}{h} \leqslant 1, \end{aligned}$$

while for the (Upwind: $a < 0$) scheme:

$$\begin{aligned} x_i - at_j &\leqslant x_{i+j} \\ x_i - aj\Delta t &\leqslant x_i + jh \\ -a\Delta t &\leqslant h \\ -1 &\leqslant \frac{a\Delta t}{h} \leqslant 0. \end{aligned}$$

Observe that for the (Upwind: $a > 0$) ((Upwind: $a < 0$)) scheme, the positive-definiteness (negative-definiteness) of λ comes from the physical intuition that the method should be used to solve (5.3.1) when $a > 0$ ($a < 0$).

5.3.4 Lax-Friedrichs Method

Another time-explicit method can be obtained by approximating the spatial derivative u_x with the first-order centred difference (1stCD). Dropping the local truncation error and replacing $u_i^j \approx u(x_i, t_j)$ yields

$$u_i^{j+1} = u_i^j - \frac{\lambda}{2}\left(u_{i+1}^j - u_{i-1}^j\right).$$

Performing a von Neumann stability analysis shows that this method is not useful. Indeed, after cancelling the common factor we obtain:

$$w_{j+1} = w_j\left(1 - \frac{\lambda}{2}[e^{rhI} - e^{-rhI}]\right) = w_j\Big[1 - I\lambda\sin(rh)\Big],$$

and the magnitude of the amplication factor satisfies:

$$|K|^2 = 1 + \lambda^2\sin^2(rh) \geqslant 1.$$

By replacing u_i^j with the average of its nearest neighbour, we obtain the **Lax-Friedrichs method**:

$$u_i^{j+1} = \frac{1}{2}\left(u_{i-1}^j + u_{i+1}^j\right) - \frac{\lambda}{2}\left(u_{i+1}^j - u_{i-1}^j\right) \qquad \text{(Lax-Friedrichs)}$$

Let us examine the consistency condition, if any. By replacing u_i^j with the true solution $u(x_i, t_j)$ in (Lax-Friedrichs), we obtain

$$\begin{aligned}
\tau_{ij} &= \frac{u(x_i, t_{j+1}) - \frac{1}{2}\left[u(x_{i-1}, t_j) + u(x_{i+1}, t_j)\right]}{\Delta t} + a\left[\frac{u(x_{i+1}, t_j) - u(x_{i-1}, t_j)}{2h}\right] \\
&= \underbrace{\frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t}}_{\approx u_t} + a\underbrace{\left[\frac{u(x_{i+1}, t_j) - u(x_{i-1}, t_j)}{2h}\right]}_{\approx u_x} \\
&\qquad - \frac{1}{2\Delta t}\underbrace{\left[u(x_{i-1}, t_j) + u(x_{i+1}, t_j) - 2u(x_i, t_j)\right]}_{h^2 u_{xx}}
\end{aligned}$$

It follows from difference formulas (1stFD) for the first term, (1stCD) for the second term and (2ndCD) for the third term that:

$$\begin{aligned}
\tau_{ij} &= u_t(x_i, t_j) + \mathcal{O}(\Delta t) + a\Big[u_x(x_i, t_j) + \mathcal{O}(h^2)\Big] - \frac{h^2}{2\Delta t}\Big[u_{xx}(x_i, t_j) + \mathcal{O}(h^2)\Big] \\
&= \mathcal{O}(\Delta t) + \mathcal{O}(h) + \mathcal{O}\left(\frac{h^2}{\Delta t}\right).
\end{aligned}$$

For the scheme to be consistent, we require that $\dfrac{h^2}{\Delta t} \longrightarrow 0$ as $\Delta t, h \longrightarrow 0$, which suggest choosing the time step $\Delta t = \mathcal{O}(h)$ as $\Delta t, h \longrightarrow 0$. We see that the Lax-Friedrichs method is first order accurate in both time and space.

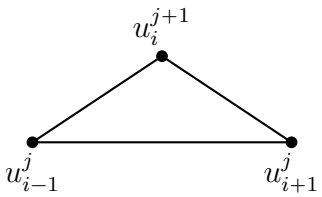


Figure 5.10: Three-point stencil for the Lax-Friedrichs method.

We now turn from consistency to stability. From von Neumann stability analysis,

$$w_{j+1}e^{rx_iI} = \frac{w_j}{2}\left(e^{rx_iI}e^{-rhI} + e^{rx_iI}e^{rhI}\right) - \frac{\lambda w_j}{2}\left(e^{rx_iI}e^{rhI} - e^{rx_iI}e^{-rhI}\right).$$

Cancelling out exponential term, we obtain:

$$w_{j+1} = \frac{w_j}{2}\left(e^{-rhI} + e^{rhI} - \lambda(e^{rhI} - e^{-rhI})\right)$$

$$= w_j\Big[\cos(rh) - I\lambda\sin(rh)\Big].$$

Expanding the amplication factor yields:

$$|K|^2 = \cos^2(rh) + \lambda^2\sin^2(rh) = 1 + (\lambda^2 - 1)\sin^2(rh),$$

and this is bounded by 1 for all r only if $\lambda^2 - 1 \leqslant 0$, *i.e.* $|\lambda| \leqslant 1$.

5.3.5 Lax-Wendroff Method

So far all the methods we discussed are first-order. One possible way to achieve second-order accuracy is to use a second-order finite difference for the time discretisation. For various reasons, it is much more convenient to use a two-level methods for PDEs whenever possible– in more than one dimension the need to store several levels of data may be restrictive and boundary conditions can be harder to impose, to name a few reasons. We guess an ansatz of the form

$$u_i^{j+1} = Au_{i+1}^j + Bu_i^j + Cu_{i-1}^j. \tag{5.3.8}$$

The idea is to use Taylor expansions directly on the advection equation $u_t + au_x = 0$, replacing the time derivatives arising in the Taylor series expansion with spatial derivatives, using expressions obtained by differentiating the given PDE.

Expanding $u(x_i, t_{j+1})$ around t_j gives:

$$u(x_i, t_{j+1}) = u(x_i, t_j) + \Delta t u_t(x_i, t_j) + \frac{\Delta t^2}{2}u_{tt}(x_i, t_j) + \frac{\Delta t^3}{6}u_{ttt}(x_i, t_j) + \mathcal{O}(\Delta t^4). \tag{5.3.9}$$

Assuming u has sufficient regularity, since $u_t = -au_x$, we have that:

$$\begin{aligned} u_{tt} &= -au_{xt} = -au_{tx} = -a(-au_{xx}) = a^2u_{xx} \\ u_{ttt} &= a^2u_{xxt} = a^2u_{txx} = a^2(-au_{xxx}) = -a^3u_{xxx}, \end{aligned}$$

and substituting this into (5.3.9) yields:

$$u(x_i, t_{j+1}) = u(x_i, t_j) - a\Delta t u_x(x_i, t_j) + \frac{a^2\Delta t^2}{2}u_{xx}(x_i, t_j) - \frac{a^3\Delta t^3}{6}u_{xxx}(x_i, t_j) + \mathcal{O}(\Delta t^4). \tag{5.3.10}$$

Next, expanding $u(x_{i\pm1}, t_j)$ around x_i gives:

$$u(x_{i\pm1}, t_j) = u(x_i, t_j) \pm hu_x(x_i, t_j) + \frac{h^2}{2}u_{xx}(x_i, t_j) \pm \frac{h^3}{6}u_{xxx}(x_i, t_j) + \mathcal{O}(h^4). \tag{5.3.11}$$

Substituting (5.3.10) and (5.3.11) into (5.3.8) yields:

$$\begin{aligned} &u(x_i, t_j) - a\Delta t u_x(x_i, t_j) + \frac{a^2\Delta t^2}{2}u_{xx}(x_i, t_j) - \frac{a^3\Delta t^3}{6}u_{xxx}(x_i, t_j) \\ &= A\Big[u(x_i, t_j) + hu_x(x_i, t_j) + \frac{h^2}{2}u_{xx}(x_i, t_j) + \frac{h^3}{6}u_{xxx}(x_i, t_j)\Big] \\ &\quad + Bu(x_i, t_j) + \mathcal{O}(\Delta t^4) + (A + C)\mathcal{O}(h^4) \end{aligned}$$

$$+ C\Big[u(x_i,t_j) - hu_x(x_i,t_j) + \frac{h^2}{2}u_{xx}(x_i,t_j) - \frac{h^3}{6}u_{xxx}(x_i,t_j)\Big]$$

Equating coefficients of $u(x_i,t_j), u_x(x_i,t_j)$ and $u_{xx}(x_i,t_j)$, we obtain the following system of linear equations:

$$\begin{aligned} A + B + C &= 1 \\ A - C &= -\lambda \\ A + C &= \lambda^2, \end{aligned}$$

which has solution

$$A = \frac{\lambda^2 - \lambda}{2}, \quad C = \frac{\lambda^2 + \lambda}{2}, \quad B = 1 - \lambda^2.$$

This yields the **Lax-Wendroff method**:

$$\begin{aligned} u_i^{j+1} &= \left(\frac{\lambda^2 - \lambda}{2}\right) u_{i+1}^j + (1-\lambda^2)u_i^j + \left(\frac{\lambda^2 + \lambda}{2}\right) u_{i-1}^j \qquad \text{(Lax-Wendroff)} \\ &= u_i^j - \frac{\lambda}{2}\left(u_{i+1}^j - u_{i-1}^j\right) + \frac{\lambda^2}{2}\left(u_{i+1}^j - 2u_i^j + u_{i-1}^j\right) \end{aligned}$$

The local truncation error is

$$\begin{aligned} \tau_{ij} &= \frac{u(x_i,t_{j+1}) - \Big[Au(x_{i+1},t_j) + Bu(x_i,t_j) + Cu(x_{i-1},t_j)\Big]}{\Delta t} \\ &= \frac{-a^3\Delta t^2}{6}u_{xxx}(x_i,t_j) - \frac{(A-C)h^3}{6}u_{xxx}(x_i,t_j) + \mathcal{O}(\Delta t^3) - (A+C)\mathcal{O}\left(\frac{h^4}{\Delta t}\right) \\ &= \frac{-a^3\Delta t^2}{6}u_{xxx}(x_i,t_j) + \frac{\lambda h^3}{6\Delta t}u_{xxx}(x_i,t_j) + \mathcal{O}(\Delta t^3) + \mathcal{O}(\Delta t h^2) \\ &= \mathcal{O}(\Delta t^2) + \mathcal{O}(h^2). \end{aligned}$$

Note that division by Δt is because we want to recover the time derivative u_t. The local truncation error can also be obtained directly. Using $u_{tt} = a^2u_{xx}$ and $u_{ttt} = -a^3u_{xxx}$, we have:

$$\begin{aligned} \tau_{ij} &= \frac{u(x_i,t_{j+1}) - u(x_i,t_j)}{\Delta t} + \frac{\lambda}{2\Delta t}\left(u(x_{i+1},t_j) - u(x_{i-1},t_j)\right) \\ &\qquad - \frac{\lambda^2}{2\Delta t}\left(u(x_{i+1},j) - 2u(x_i,t_j) + u(x_{i-1},t_j)\right) \\ &= \frac{u(x_i,t_{j+1}) - u(x_i,t_j)}{\Delta t} + a\left(\frac{u(x_{i+1},t_j) - u(x_{i-1},t_j)}{2h}\right) \\ &\qquad - \frac{a^2\Delta t}{2}\left(\frac{u(x_{i+1},j) - 2u(x_i,t_j) + u(x_{i-1},t_j)}{h^2}\right) \\ &= u_t(x_i,t_j) + \frac{\Delta t}{2}u_{tt}(x_i,t_j) + \frac{\Delta t^2}{6}u_{ttt}(x_i,t_j) + \mathcal{O}(\Delta t^3) \\ &\qquad + a\left(u_x(x_i,t_j) + \frac{h^2}{6}u_{xxx}(x_i,t_j) + \mathcal{O}(h^3)\right) - \frac{a^2\Delta t}{2}\left(u_{xx}(x_i,t_j) + \mathcal{O}(h^2)\right) \\ &= \frac{a^2\Delta t}{2}u_{xx}(x_i,t_j) - \frac{a^3\Delta t^2}{6}u_{xxx}(x_i,t_j) + \mathcal{O}(\Delta t^3) \end{aligned}$$

$$+\frac{ah^2}{6}u_{xxx}(x_i,t_j)+\mathcal{O}(h^3)-\frac{a^2\Delta t}{2}u_{xx}(x_i,t_j)+\mathcal{O}(\Delta t h^2)$$
$$=\mathcal{O}(\Delta t^2)+\mathcal{O}(h^2).$$

We now investigate stability of the scheme using the von Neumann stability analysis. The amplification factor for the (Lax-Wendroff) method is

$$\begin{aligned}K&=\left(\frac{\lambda^2-\lambda}{2}e^{rhI}\right)+(1-\lambda^2)+\left(\frac{\lambda^2+\lambda}{2}e^{-rhI}\right)\\&=1-\lambda^2+\lambda^2\cos(rh)-I\lambda\sin(rh)\\&=1+\lambda^2\Big[\cos(rh)-1\Big]-I\lambda\sin(rh)\\&=1-2\lambda^2\sin^2(rh/2)-2\lambda I\sin(rh/2)\cos(rh/2),\end{aligned}$$

and computing $|K|^2$ yields:

$$\begin{aligned}|K|^2&=\Big[1-2\lambda^2\sin^2(rh/2)\Big]^2+4\lambda^2\sin^2(rh/2)\cos^2(rh/2)\\&=1-4\lambda^2\sin^2(rh/2)+4\lambda^4\sin^4(rh/2)+4\lambda^2\sin^2(rh/2)\cos^2(rh/2)\\&=1-4\lambda^2\sin^2(rh/2)(1-\cos^2(rh/2))+4\lambda^4\sin^4(rh/2)\\&=1-4\lambda^2\sin^4(rh/2)+4\lambda^4\sin^4(rh/2)\\&=1-4\lambda^2(1-\lambda^2)\sin^4(rh/2).\end{aligned}$$

We see that $|K|^2\leqslant 1$ for all r as long as $1-\lambda^2\geqslant 0$, *i.e.* $|\lambda|\leqslant 1$.

5.4 Elliptic Equation

5.4.1 1D Poisson Equation

Consider the one-dimensional Poisson equation (inhomogeneous Laplace's equation) with Dirichlet boundary condition:

$$\begin{cases}u''(x)=f(x), & 0<x<1,\\ u(0)=\alpha,\ u(1)=\beta.\end{cases}\tag{Poisson}$$

This problem is called a **2-point boundary value problem (BVP)** since boundary conditions are specified at two distinct points. For sufficiently nice $f(x)$, the (Poisson) problem can be solved explicitly but studying finite-difference methods for this simple problem will reveal some of the essential features of all such analysis, particularly the relation of the global error to the local truncation error and the use of stability in making this connection.

Subdividing the interval $[0,1]$ uniformly into $m+2$ subintervals gives the set of discrete grid points $(x_j)_{j=0}^{m+1}$ defined by $x_j=jh$, where $h=\dfrac{1}{m+1}$ is the mesh size. At each discrete point x_j, replacing $u''(x)$ with a second-order central difference (2ndCD) gives:

$$\frac{u(x_{j+1})-2u(x_j)+u(x_{j-1})}{h^2}=f(x_j)+\underbrace{\frac{h^2}{12}u^{(4)}(x_j)+\mathcal{O}(h^4)}_{\textbf{local truncation error}},$$

Dropping τ_j and denoting $u_j \approx u(x_j), f_j \approx f(x_j)$ as the numerical approximation at x_j, we obtain the **three-point stencil scheme**:

$$\boxed{\frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} = f_j, \quad j = 1, \ldots, m,} \tag{5.4.1}$$

with boundary conditions $u_0 = \alpha, u_{m+1} = \beta$. Define the following vectors:

$$U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_M \end{bmatrix}, \quad F = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_M \end{bmatrix} - \frac{1}{h^2} \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \\ \beta \end{bmatrix},$$

where we incorporate the boundary conditions into the vector F. We can then rewrite (5.4.1) as the matrix system $A^h U = F$, where $A^h \in \mathbb{R}^{m \times m}$ is the symmetric tridiagonal matrix:

$$A^h = \frac{1}{h^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & -2 \end{bmatrix}.$$

5.4.2 Stability, Consistency and Convergence

We already knew that the local truncation error τ_j of the scheme (5.4.1) is of $\mathcal{O}(h^2)$ from the derivation. In general, if we were given a finite-difference method, the local truncation error is obtained by replacing u_j with the true solution $u(x_j)$ in the scheme. As an illustration, from (5.4.1), using (2ndCD) we obtain:

$$\begin{aligned} \tau_j = \frac{u(x_{j+1}) - 2u(x_j) + u(x_{j-1})}{h^2} - f(x_j) &= u''(x_j) + \frac{h^2}{12} u^{(4)}(x_j) + \mathcal{O}(h^4) - f(x_j) \\ &= \mathcal{O}(h^2) \quad \text{as } h \longrightarrow 0. \end{aligned}$$

If we define $\tau^h = (\tau_j) \in \mathbb{R}^m$ and $U_{\text{exact}} \in \mathbb{R}^m$ is the vector of true solution at grid points, then

$$\tau^h = A^h U_{\text{exact}} - F \implies A^h U_{\text{exact}} = \tau^h + F.$$

To analyse the global error $E^h = (E_j) = U - U_{\text{exact}} \in \mathbb{R}^m$, subtracting $A^h U_{\text{exact}} = F + \tau^h$ from $A^h U = F$ gives

$$A^h E^h = -\tau^h, \tag{5.4.2}$$

which corresponds to the system of equations:

$$\frac{E_{j+1} - 2E_j + E_{j-1}}{h^2} = -\tau_j, \quad j = 1, \ldots, m,$$

with boundary conditions $E_0 = E_{m+1} = 0$. From Theorem 5.1.3, the tridiagonal matrix A^h is invertible since it is strictly diagonally dominant. Solving (5.4.2) for E^h gives:

$$E^h = -(A^h)^{-1}\tau^h,$$

and taking norms gives:

$$\|E^h\| \leqslant \|(A^h)^{-1}\|\|\tau^h\|.$$

We see that the global error will have roughly the same magnitude as the local truncation error if we impose the following condition: there exists a constant $C > 0$ independent of h such that:

$$\|(A^h)^{-1}\| \leqslant C \quad \text{for all } h \text{ sufficiently small.}$$

This leads to

$$\|E^h\| \leqslant C\|\tau^h\|,$$

and so $\|E^h\|$ goes to zero at least as fast as $\|\tau^h\|$ as $h \longrightarrow 0$. This motivates the following definition of stability for linear BVPs:

Definition 5.4.1. Suppose a finite-difference method for a linear BVP gives a sequence of matrix equations of the form $A^hU^h = F^h$, where h is the mesh width. We say that the method is **stable** if:

1. $(A^h)^{-1}$ exists for all h sufficiently small, say $0 < h < h_0$,
2. there exists a constant $C > 0$, independent of h, such that

$$\|(A^h)^{-1}\| \leqslant C \quad \text{for all } h < h_0.$$

Definition 5.4.2. We say that a finite-difference method is **consistent** with the given BVP if

$$\|\tau^h\| \longrightarrow 0 \quad \text{as } h \longrightarrow 0.$$

We say that a finite-difference method is **convergent** if $\|E^h\| \longrightarrow 0$ as $h \longrightarrow 0$.

Theorem 5.4.3. *If a finite-difference method is consistent and stable, then it is convergent.*

Proof. Using the definition of stability and consistency in order, we obtain:

$$\|E^h\| \leqslant \|(A^h)^{-1}\|\|\tau_h\| \leqslant C\|\tau_h\| \longrightarrow 0 \quad \text{as } h \longrightarrow 0.$$

■

The above analysis leads to the following conclusion:

$$\mathcal{O}(h^p) \text{ local truncation error} \quad + \quad \text{stablity} \quad \Longrightarrow \quad \mathcal{O}(h^p) \text{ global error.}$$

Consistency is usually easy to verify, but the difficulty lies in showing stability since the matrix A^h grows in size as the mesh width $h \longrightarrow 0$. Note that for other classes of problems, it may not even be clear how to define stability in an appropriate way that allows one to prove convergence using Theorem 5.4.3. Nonetheless, this forms the basic of proving convergence of finite-difference methods: first compute the local truncation error of the method and then use some form of stability to show that the global error can be bounded in terms of the local truncation error.

5.4.3 Stability in $\|\cdot\|_2$

For the (Poisson) problem, the main goal now is to obtain a bound for $\|(A^h)^{-1}\|$ that is independent of the mesh-width h. Clearly, this depends strongly on the choice of norms, usually dictated by what order of accuracy we want for the given method.

The fact that the matrix A^h is symmetric (and so is $(A^h)^{-1}$) suggest the choice of the matrix 2-norm, because then

$$\|(A^h)^{-1}\|_2 = \rho((A^h)^{-1}) = \max_{1\leqslant p\leqslant m} |\lambda_p^{-1}| = \left(\min_{1\leqslant p\leqslant m} |\lambda_p|\right)^{-1},$$

where $\{\lambda_1, \ldots, \lambda_m\}$ are eigenvalues of A^h. From Theorem 5.1.2 with

$$a = -\frac{2}{h^2}, \quad b = \frac{1}{h^2},$$

the eigenvalues of A^h are:

$$\lambda_p = -\frac{2}{h^2} + \frac{2}{h^2}\cos(p\pi h) = \frac{2}{h^2}\Big(\cos(p\pi h) - 1\Big), \quad p = 1, \ldots, m,$$

with its corresponding eigenvectors $u^{(p)} = (u_j^{(p)})$:

$$u_j^{(p)} = \sin(p\pi jh), \quad j = 1, \ldots, m.$$

The matrix A^h is negative definite since all its eigenvalues $\lambda_p < 0$. The smallest eigenvalue of A^h in magnitude is

$$\begin{aligned}\lambda_1 &= \frac{2}{h^2}\Big(\cos(\pi h) - 1\Big) \\ &= \frac{2}{h^2}\Big(-\frac{\pi^2 h^2}{2} + \frac{\pi^4 h^4}{4!} + \mathcal{O}(h^6)\Big) \\ &= -\pi^2 + \mathcal{O}(h^2) \quad \text{as } h \longrightarrow 0.\end{aligned}$$

This shows that $|\lambda_1| \approx \pi^2$ for sufficiently small $h > 0$ and it is bounded away from 0 as $h \longrightarrow 0$, indicating that the method (5.4.1) is stable in the 2-norm. Moreover, we obtain the global error bound:

$$\|E^h\|_2 \leqslant \|(A^h)^{-1}\|_2 \|\tau^h\|_2 \approx \frac{1}{\pi^2}\|\tau^h\|_2 = \mathcal{O}(h^2) \quad \text{as } h \longrightarrow 0.$$

We also have the following approximation for the local truncation error:

$$\|\tau^h\|_2 \approx \frac{h^2}{12}\|u^{(4)}\|_2 = \frac{h^2}{12}\|f''\|_2,$$

where $\|f''\|_2$ is the discrete-grid-norm of f'' evaluated at $\{x_1, \ldots, x_m\}$.

Remark 5.4.4. The eigenvalues for one-dimensional Laplacian is $\mu_p = -p^2\pi^2, p = 1, 2, \ldots$, with corresponding eigenvector $u^p = \sin(p\pi x)$; p is known as the wave number. The eigenvectors of A^h are simply the first m eigenvectors of the Laplacian evaluated at the discrete grid points $x_j = jh, j = 1, \ldots, m$. For a fixed p, $\lambda_p \approx \mu_p$ for sufficiently small $h > 0$, since

$$\begin{aligned}\lambda_p &= \frac{2}{h^2}\Big(\cos(p\pi h) - 1\Big)\\ &= \frac{2}{h^2}\Big(-\frac{p^2\pi^2h^2}{2} + \frac{p^4\pi^4h^4}{4!} + \mathcal{O}(h^6)\Big)\\ &= -p^2\pi^2 + \mathcal{O}(h^2) \quad \text{as } h \longrightarrow 0.\end{aligned}$$

In the limit as $h \longrightarrow 0$, we recover the eigenvalues for the continuous Laplacian operator.

Since all norms are equivalent in finite-dimensional vector space, one might think that we recover the full error estimate up to a multiplicative constant; this is false because the constant of equivalence actually depends on the mesh width h. For the ∞-norm,

$$h|E_j| \leqslant h|E_j|^2 \leqslant h\sum_{j=1}^{m}|E_j|^2 = \|E^h\|_2^2,$$

and taking the supremum over all $j = 1, \ldots, m$ yields:

$$\|E^h\|_\infty \leqslant \frac{1}{\sqrt{h}}\|E^h\|_2 = \mathcal{O}(h^{3/2}) \quad \text{as } h \longrightarrow 0,$$

This does not show the second order accuracy that we would like to have. Nonetheless, by explicitly calculating the inverse of A^h and then showing that $\|(A^h)^{-1}\|_\infty = \mathcal{O}(1)$ using the idea of Green's function, we obtain the same error estimate in the ∞-norm:

$$\|E\|_\infty \leqslant \|(A^h)^{-1}\|_\infty\|\tau\|_\infty = \mathcal{O}(h^2),$$

since $\|\tau^h\|_\infty = \mathcal{O}(h^2)$.

5.4.4 2D Extension of Poisson Equation

Consider the Poisson equation $u_{xx} + u_{yy} = f$ on the unit square $\mathcal{D}$, with Dirichlet boundary condition $u|_{\partial\mathcal{D}} = u_{\mathcal{D}}$. We discretise the square uniformly, with grid nodes (x_i, y_j) defined by

$$x_i = i\Delta x, \quad y_j = j\Delta y.$$

Let $u_{i,j}, f_{ij}$ represent the finite-difference approximation of $u(x_i, y_j), f(x_i, y_j)$ respectively. Using the second-order central difference (2ndCD) for both u_{xx} and u_{yy} and dropping the local truncation error gives the following:

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

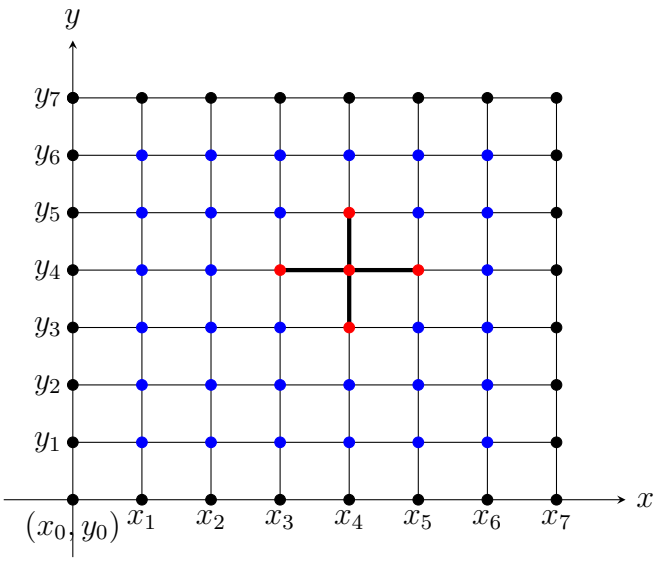


Figure 5.11: An example of a square mesh grid with $m = 6$. The black dots are the boundary data. The red dots are the five-point stencil for the Laplacian about the point (x_4, y_4).

Assuming $\Delta x = \Delta y = h = \frac{1}{m+1}$, we obtain the **five-point stencil scheme**:

$$\boxed{\frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{ij}}{h^2} = f_{ij}, \quad i,j = 1,\dots,m,} \tag{5.4.3}$$

with boundary conditions:

$$\boxed{\begin{aligned} u_{i,0} &= u_{\mathcal{D}}(x_i, 0), \quad i = 1,2,\dots,m \\ u_{i,m+1} &= u_{\mathcal{D}}(x_i, 1), \quad i = 1,2,\dots,m \\ u_{0,j} &= u_{\mathcal{D}}(0, y_j), \quad j = 1,2,\dots,m \\ u_{m+1,j} &= u_{\mathcal{D}}(1, y_j), \quad j = 1,2,\dots,m. \end{aligned}}$$

Observe that the five-point stencil scheme (5.4.3) is a system of m^2 linear equations with m^2 variables, and we can rewrite this as a matrix system $A^h U = F$, where the matrix $A^h \in \mathbb{R}^{m^2 \times m^2}$ is now very **sparse**, *i.e.* most of the elements are 0. There exists different orderings for the grid points, but all such matrices are equivalent up to permutations. In constrast to the one-dimensional problem, there exists more choices in terms of ordering the grid points. Although all such matrices are equivalent up to permutations, a clever choice of ordering will have a significant impact when it comes to numerically solving the matrix system. Unfortunately, in two-dimensions the structure of the matrix is not as compact as in one-dimension, in that the nonzeros cannot be as nicely clustered near the main diagonal.

A natural choice is the **rowwise ordering**, where we take the unknowns along the bottom

row, followed by unknowns in the second row, and so on. As such, define the following vectors:

$$U = \begin{bmatrix} u^{(1)} \\ u^{(2)} \\ \vdots \\ u^{(m)} \end{bmatrix}, \quad F = \begin{bmatrix} f^{(1)} \\ f^{(2)} \\ \vdots \\ f^{(m)} \end{bmatrix} - \frac{1}{h^2} \begin{bmatrix} \text{side}^{(1)} + \text{bottom} \\ \text{side}^{(2)} \\ \vdots \\ \text{side}^{(m-1)} \\ \text{side}^{(m)} + \text{top} \end{bmatrix},$$

where

$$u^{(j)} = \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ \vdots \\ u_{m,j} \end{bmatrix}, \quad f^{(j)} = \begin{bmatrix} f_{1j} \\ f_{2j} \\ \vdots \\ f_{mj} \end{bmatrix}, \quad j = 1, 2, \ldots, m,$$

and

$$\text{bottom} = \begin{bmatrix} u_{1,0} \\ u_{2,0} \\ \vdots \\ u_{m,0} \end{bmatrix}, \quad \text{top} = \begin{bmatrix} u_{1,m+1} \\ u_{2,m+1} \\ \vdots \\ u_{m,m+1} \end{bmatrix}, \quad \text{side}^{(j)} = \begin{bmatrix} u_{0,j} \\ 0 \\ \vdots \\ 0 \\ u_{m+1,j} \end{bmatrix} \in \mathbb{R}^m, \quad j = 1, 2, \ldots, m.$$

The corresponding matrix $A^h \in \mathbb{R}^{m^2 \times m^2}$ of the scheme (5.4.3) has the form:

$$A^h = \frac{1}{h^2} \begin{bmatrix} T & I_m & & & \\ I_m & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & I_m \\ & & & I_m & T \end{bmatrix},$$

where $T \in \mathbb{R}^{m \times m}$ is the symmetric tridiagonal matrix:

$$T = \begin{bmatrix} -4 & 1 & & & \\ 1 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & -4 \end{bmatrix}.$$

Remark 5.4.5.

5.4.5 Accuracy and Stability in 2D

The local truncation error τ_{ij} is obtained by replacing $u_{i,j}$ with the true solution $u(x_i, y_j)$ in the scheme (5.4.3):

$$\begin{aligned}\tau_{ij} &= \frac{1}{h^2}\Big(u(x_{i-1}, y_j) + u(x_{i+1}, y_j) + u(x_i, y_{j-1}) + u(x_i, y_{j+1}) - 4u(x_i, y_j)\Big) - f(x_i, y_j)\\ &= u_{xx}(x_i, y_j) + u_{yy}(x_i, y_j) + \frac{h^2}{12}\Big(u_{xxxx}(x_i, y_j) + u_{yyyy}(x_i, y_j)\Big) + \mathcal{O}(h^4) - f(x_i, y_j)\\ &= \mathcal{O}(h^2) \quad \text{as } h \longrightarrow 0.\end{aligned}$$

This is just a consequence of the one-dimensional local truncation error on both x and y direction. Define $E_{ij} = u_{ij} - u(x_i, y_j)$. The error equation is $A^h E^h = -\tau^h$, and the method will be globally second order accurate provided it is stable in some chosen norm, *i.e.* there exists a constant $C > 0$, independent of h as $h \longrightarrow 0$ such that $\|(A^h)^{-1}\| \leqslant C$.

For rowwise ordering, we show that $(A^h)^{-1}$ is uniformly bounded in the 2-norm as $h \longrightarrow 0$. The eigenvalues of A^h are:

$$\lambda_{pq} = \frac{2}{h^2}\Big[\Big(\cos(p\pi h) - 1\Big) + \Big(\cos(q\pi h) - 1\Big)\Big],$$

with corresponding eigenvectors $u^{(pq)} = \Big(u_{i,j}^{(pq)}\Big) \in \mathbb{R}^{m^2}$:

$$u_{ij}^{pq} = \sin(p\pi i h)\sin(q\pi j h), \quad i, j = 1, 2, \ldots, m,$$

where the parameter $p, q = 1, 2, \ldots, m$ are the wave numbers in the x, y direction respectively. The matrixA^h is again negative definite since all its eigenvalues $\lambda_{pq} < 0$. The smallest eigenvalue of A^h in magnitude is

$$\begin{aligned}\lambda_{11} &= \frac{4}{h^2}\Big(\cos(\pi h) - 1\Big)\\ &= \frac{2}{h^2}\left(-\frac{pi^2 h^2}{2} + \frac{\pi^4 h^4}{4!} + \mathcal{O}(h^6)\right)\\ &= -2\pi^2 + \mathcal{O}(h^2) \quad \text{as } h \longrightarrow 0.\end{aligned}$$

Consequently,

$$\|(A^h)^{-1}\|_2 \approx \frac{1}{2\pi^2} \quad \text{as } h \longrightarrow 0,$$

and the global error satisfies $\|E^h\|_2 = \mathcal{O}(h^2)$ as $h \longrightarrow 0$. Since

$$\|A^h\|_2 = \rho(A^h) = |\lambda_{mm}| = \frac{4}{h^2}\left[1 - \cos\left(\frac{m\pi}{m+1}\right)\right] \approx \frac{8}{h^2},$$

the condition number of A^h with respect to the 2-norm is:

$$\kappa_2(A^h) = \|A^h\|_2\|(A^h)^{-1}\|_2 \approx \left(\frac{8}{h^2}\right)\left(\frac{1}{2\pi^2}\right) = \frac{4}{\pi^2 h^2} = \mathcal{O}\left(\frac{1}{h^2}\right) \quad \text{as } h \longrightarrow 0.$$

This means that the matrix becomes very ill-conditioned as the mesh width h gets smaller.

5.4.6 Variational Formulation

We end the section with some theoretical approach of solving elliptic PDEs. We will study the variational approach which play a prominent role in elliptic PDEs over the last few decades. It is based on the notion of **weak solutions**, which proved to be one of the most elegant yet powerful concept in the field of PDEs.

Consider the one-dimensional Poisson equation with homogeneous Dirichlet boundary conditions:

$$\begin{cases} -u'' = f(x), & 0 < x < 1, \\ u(0) = u(1) = 0. \end{cases} \tag{Dirichlet}$$

Consider the linear space:

$$V = \Big\{ v \in C[0,1] \colon v' \text{ is bounded, piecewise continuous on } [0,1] \text{ and } v(0) = v(1) = 0 \Big\}.$$

Consider the linear functional $F \colon V \longrightarrow \mathbb{R}$ defined by

$$F(v) = \frac{1}{2}\langle v', v' \rangle - \langle f, v \rangle,$$

where $\langle \cdot, \cdot \rangle$ is the standard L^2-inner product on $[0,1]$:

$$\langle f, g \rangle = \int_0^1 f(x) g(x)\, dx.$$

Define the following two problems:

$$\text{Find } u \in V \text{ such that } F(u) \leqslant F(v) \text{ for all } v \in V. \tag{Min}$$

$$\text{Find } u \in V \text{ such that } \langle u', v' \rangle = \langle f, v \rangle. \tag{Var}$$

Theorem 5.4.6. *If u is a solution to the (Dirichlet) problem, then u is also a solution to the problem (Var).*

Proof. Integrating the Poisson equation in (Dirichlet) against a test function $v \in V$ over $[0,1]$:

$$\begin{aligned} \langle f, v \rangle = \int_0^1 f v\, dx &= -\int_0^1 u'' v\, dx \\ &= -\Big[u'v \Big]\Big|_0^1 + \int_0^1 u'v'\, dx \\ &= \int_0^1 u'v'\, dx = \langle u', v' \rangle, \end{aligned}$$

where the boundary term vanishes due to $v(0) = v(1) = 0$.

■

Theorem 5.4.7. *The variational problem (Var) is equivalent to the minimisation problem (Min).*

Proof. Assume u is a solution to the variational problem (Var). We need to show that $F(u) \leqslant F(v)$ for all $v \in V$. Decomposing $v = u + w$, where $w \in V$, then

$$\begin{aligned} F(v) = F(u+w) &= \frac{1}{2}\langle u' + w', u' + w' \rangle - \langle f, u + w \rangle \\ &= \left(\frac{1}{2}\langle u', u' \rangle - \langle f, u \rangle \right) + \left[\langle u', w' \rangle - \langle f, w \rangle \right] + \frac{1}{2}\langle w', w' \rangle \\ &= F(u) + \frac{1}{2}\langle w', w' \rangle \\ &\geqslant F(u), \end{aligned}$$

where the second term vanishes since u is a solution to the variational problem (Var) by assumption. Hence, $F(u) \leqslant F(v)$ for all $v \in V$.

Conversely, suppose u is a solution to the minimisation problem (Min). For any $v \in V$ and $\varepsilon > 0$, we have $u + \varepsilon v \in V$ and so $F(u) \leqslant F(u + \varepsilon v)$. Treating $g(\varepsilon) := F(u + \varepsilon v)$ as a function of ε, we deduce that $g(\varepsilon)$ has a minimum at $\varepsilon = 0$ for any $v \in V$. We now compute the Gâteaux derivative of $F(\cdot)$. Expanding $g(\varepsilon)$ gives:

$$\begin{aligned} g(\varepsilon) = F(u + \varepsilon v) &= \frac{1}{2}\langle u' + \varepsilon v', u' + \varepsilon v' \rangle - \langle f, u + \varepsilon v \rangle \\ &= \frac{1}{2}\langle u', u' \rangle + \varepsilon \langle u', v' \rangle + \frac{\varepsilon^2}{2}\langle v', v' \rangle - \langle f, u \rangle - \varepsilon \langle f, v \rangle \\ \implies g'(\varepsilon) &= \langle u', v' \rangle + \varepsilon \langle v', v' \rangle - \langle f, v \rangle \end{aligned}$$

The desired conclusion follows from $g'(\varepsilon)\Big|_{\varepsilon=0} = 0$.

■

Theorem 5.4.8. *The solution to the variational problem* (Var) *is unique.*

Proof. Suppose $u_1, u_2 \in V$ are any two solutions of the variational problem (Var). For all $v \in V$, u_1, u_2 satisfy the following equation:

$$\begin{aligned} \langle u_1', v' \rangle &= \langle f, v \rangle \\ \langle u_2', v' \rangle &= \langle f, v \rangle \end{aligned}$$

Subtracting both equations then gives us:

$$\langle u_1' - u_2', v' \rangle = 0.$$

We choose the particular $v' = u_1' - u_2' \in V$, then:

$$\langle u_1' - u_2', u_1' - u_2' \rangle = 0 \implies u_1' - u_2' = \mathbf{0} \implies u_1 - u_2 \equiv \mathbf{0}.$$

■

Summarising everything, we have the following relation:

$$\text{(Dirichlet)} \implies \text{(Var)} \iff \text{(Min)}.$$

Is it possible to show that (Var) $\implies$ (Dirichlet)? Assume $u \in C^2[0,1]$ and $f \in C[0,1]$, integrating by parts gives

$$\int_0^1 u''v\,dx + \int_0^1 fv\,dx = 0 \quad \text{for all } v \in V,$$

where again the boundary term vanishes since $v(0) = v(1) = 0$. After some justification, we arrive at

$$u'' + f = 0 \quad \text{for all } x \in (0,1).$$

Bibliography

[Atk08] K. E. Atkinson. *An introduction to numerical analysis*. John Wiley & Sons, 2008.

[LeV07] R. J. LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. Other Titles in Applied Mathematics. SIAM, 2007. DOI: 10.1137/1.9780898717839.