Numerical Analysis

DIWEN XU, University of Washington, USA

These notes are primarily based on the textbook by Gautschi [1].

1 Approximation and Interpolation

1.1 Least Squares Approximation

Definition 1.1 (Approximation in Linear Spaces). Approximate a function f by simpler functions in a linear space Φ. Find $\widehat{\varphi} \in \Phi$ s.t.

$$\|f - \widehat{\varphi}\| \le \|f - \varphi\| \quad \text{for all } \varphi \in \Phi,$$

where $\|\cdot\|$ denotes a chosen norm. The function $\widehat{\varphi}$ is called the *best approximation* to f in Φ.

Definition 1.2 (Approximation Space). Given n basis functions $\{\Pi_j\}_{j=1}^n \subset \Phi$, define

$$\Phi = \Phi_n = \left\{ \varphi : \varphi(t) = \sum_{j=1}^n c_j \Pi_j(t),\ c_j \in \mathbb{R} \right\},$$

which is a linear space of dimension n.

Definition 1.3 (Norms and Measures). Let $w(t) \ge 0$ be a weight function.

$$\|u\|_{p,w} = \left(\int_a^b |u(t)|^p w(t)\, dt \right)^{1/p} \quad \text{or} \quad \left(\sum_{i=1}^N w_i |u(t_i)|^p \right)^{1/p}.$$

To unify notation, introduce a measure $d\lambda$ and define

$$\int_{\mathbb{R}} u(t)\, d\lambda(t) = \int_a^b u(t) w(t)\, dt \quad \text{or} \quad \sum_{i=1}^N w_i u(t_i).$$

Definition 1.4 (Least Squares Approximation). Choose $L^2(d\lambda)$ norm

$$\|u\|_{2,d\lambda} = \left(\int_{\mathbb{R}} |u(t)|^2\, d\lambda(t) \right)^{1/2}.$$

Definition 1.5 (Inner Product). Define the inner product

$$(u, v) = \int_{\mathbb{R}} u(t) v(t)\, d\lambda(t).$$

$\|u\|_{2,d\lambda}^2 = (u, u)$, $(u, v) = 0$ means u and v are orthogonal.

Definition 1.6 (Normal Equations). The least squares objective is

$$\mathcal{E}(\mathbf{c}) = \int_{\mathbb{R}} \left(\sum_{j=1}^n c_j \Pi_j(t) - f(t) \right)^2 d\lambda(t).$$

$$\frac{\partial \mathcal{E}}{\partial c_i} = 0 \Rightarrow \int \left(\sum_{j=1}^n c_j \Pi_j \right) \Pi_i\, d\lambda = \int \Pi_i f\, d\lambda, \quad i = 1, \dots, n.$$

$$\Rightarrow \sum_{j=1}^n (\Pi_i, \Pi_j) c_j = (\Pi_i, f), \quad i = 1, \dots, n.$$

$$A\mathbf{c} = \mathbf{b}, \quad A = (a_{ij})_{n\times n}, \quad a_{ij} = (\Pi_i, \Pi_j), \quad b_i = (\Pi_i, f).$$

Author's Contact Information: Diwen Xu, rwbyaloupeep@gmail.com, University of Washington, Seattle, Washington, USA.

The matrix A is symmetric and positive definite, hence nonsingular. Therefore, the normal equations admit a unique solution. The Hessian of $\mathcal{E}$ is given by

$$(\nabla^2 \mathcal{E})_{ij} = \frac{\partial^2 E}{\partial c_i \partial c_j} = 2(\pi_i, \pi_j) = 2A_{ij},$$

which is positive definite. Therefore,

$$\mathcal{E}(\hat{\mathbf{c}} + \mathbf{c}) \ge \mathcal{E}(\hat{\mathbf{c}}),$$

for all vectors $\mathbf{c}$. The residual $\hat{r} = \sum_{j=1}^n c_j \Pi_j(t) - f(t)$ is orthogonal to Π_i, $i = 1, ..., n$. If $\{\pi_j\}$ is chosen to be orthogonal, i.e.

$$(\pi_i, \pi_j) = 0 \quad (i \ne j),$$

then A is diagonal and

$$\hat{c}_j = \frac{(\pi_j, f)}{(\pi_j, \pi_j)}, \quad j = 1, \dots, n.$$

1.2 Polynomial Interpolation

Definition 1.7 (Vandermonde matrix). Given distinct nodes $\{x_0, \dots, x_n\}$ and data $\{f_0, \dots, f_n\}$, seek a polynomial

$$p(x) = c_0 + c_1 x + \cdots + c_n x^n$$

such that

$$p(x_i) = f_i, \quad i = 0, \dots, n.$$

This leads to the linear system

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ 1 & x_1 & x_1^2 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix}.$$

The coefficient matrix is *Vandermonde matrix*, whose determinant is

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

If all x_i are distinct, then $\det(A) \ne 0$, and the interpolation problem has a unique solution.

Definition 1.8 (Lagrange Interpolation Formula). Given distinct nodes $x_0, x_1, \dots, x_n$ and data values

$$f_i = f(x_i), \quad i = 0, 1, \dots, n,$$

the Lagrange interpolation polynomial is

$$p(x) = \sum_{i=0}^n f_i\, \ell_i(x),$$

where the Lagrange basis polynomials are defined by

$$\ell_i(x) = \prod_{\substack{j=0 \\ j \ne i}}^n \frac{x - x_j}{x_i - x_j}, \quad \ell_i(x_k) = \begin{cases} 1, & k = i, \\ 0, & k \ne i. \end{cases}$$

$$p(x_k) = \sum_{i=0}^n f_i \ell_i(x_k) = f_k,$$

so p interpolates f at all nodes. We denote the interpolant by

$$p_n(f; x_0, \dots, x_n; x) \text{ or simply } p_n(f; x).$$

Theorem 1.9 (Interpolation Error). *The interpolation error is*

$$e_n(x) = f(x) - p_n(f; x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{i=0}^{n} (x - x_i),$$

where ξ_x lies in the interior of the smallest closed interval containing $x_0, \dots, x_n$ and x.

Proof. For $x = x_i$, the formula is true. For fixed $x \neq x_i$, define

$$F(t) = f(t) - p_n(f; t) - \frac{f(x) - p_n(f; x)}{\prod_{i=0}^{n}(x - x_i)} \prod_{i=0}^{n} (t - x_i).$$

Then $F \in C^{n+1}[a, b]$ and

$$F(x_i) = 0, \quad i = 0, \dots, n, \quad F(x) = 0.$$

Hence F has $n + 2$ distinct zeros in $[a, b]$. By repeated application of Rolle's Theorem, there exists $\xi_x \in (a, b)$ such that

$$F^{(n+1)}(\xi_x) = 0.$$

□

Theorem 1.10 (Uniform Error Bound).

$$\|f - p_n\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \left\| \prod_{i=0}^{n} (x - x_i) \right\|_\infty.$$

Let

$$x_i = x_0 + ih, \quad i = 0, 1, \dots, n.$$

$$\|f - p_1(f)\|_\infty \leq \frac{h^2}{8} \|f''\|_\infty.$$

$$\|f - p_2(f)\|_\infty \leq \frac{h^3}{9\sqrt{3}} \|f^{(3)}\|_\infty.$$

If the interpolation points x_i are equally spaced, *then*

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

can become very large near the boundary, leading to poor interpolation behavior when n is large (Runge phenomenon).

Definition 1.11 (Convergence of Interpolation). Let $\{x_0^{(n)}, \dots, x_n^{(n)}\} \subset [a, b]$. We say that Lagrange interpolation *converges uniformly* if

$$p_n(x) \to f(x) \text{ uniformly on } [a, b].$$

$$|f(x) - p_n(x)| \leq \frac{M_{n+1}}{(n+1)!} (b - a)^{n+1}, \quad M_{n+1} = \max_{x \in [a,b]} |f^{(n+1)}(x)|.$$

Definition 1.12 (Chebyshev Polynomials of the First Kind). The Chebyshev polynomial of the first kind T_n is defined by

$$T_n(x) = \cos(n\theta), \quad x = \cos\theta, \quad x \in [-1, 1].$$

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x),$$

with

$$T_0(x) = 1, \quad T_1(x) = x.$$

Chebyshev polynomials satisfy the orthogonality relation

$$\int_{-1}^{1} \frac{T_i(x) T_j(x)}{\sqrt{1 - x^2}} \, dx = \begin{cases} 0, & i \neq j, \\ \pi, & i = j = 0, \\ \frac{\pi}{2}, & i = j \neq 0. \end{cases}$$

The zeros of T_n satisfy

$$\cos(n\theta) = 0 \quad \Rightarrow \quad n\theta = \frac{(2k-1)\pi}{2}, \quad k = 1, \dots, n.$$

Hence the zeros are

$$x_k^{(n)} = \cos\left(\frac{(2k-1)\pi}{2n} \right), \quad k = 1, \dots, n.$$

The monic Chebyshev polynomial $\tilde{T}_n$ satisfies

$$\tilde{T}_n(x) = \frac{1}{2^{n-1}} T_n(x) = \prod_{k=1}^{n} \left(x - x_k^{(n)} \right), \quad \tilde{T}_0 = T_0 = 1.$$

T_n oscillates between -1 and 1 on $[-1, 1]$ with $n + 1$ extreme points

$$y_k^{(n)} = \cos\left(\frac{k\pi}{n} \right), \quad T_n\big(y_k^{(n)}\big) = (-1)^k, \quad k = 0, \dots, n.$$

Theorem 1.13 (Minimax Property). *Among all monic polynomials of degree n, the monic Chebyshev polynomial minimizes the maximum norm on* $[-1, 1]$

$$\max_{x \in [-1,1]} |\tilde{T}_n(x)| = \frac{1}{2^{n-1}}.$$

Proof. Suppose there exists a monic polynomial $\hat{p}_n$ such that

$$\max_{x \in [-1,1]} |\hat{p}_n(x)| < \frac{1}{2^{n-1}}.$$

Define $q(x) = \tilde{T}_n(x) - \hat{p}_n(x)$, which is a polynomial of degree at most $n - 1$. At the $n + 1$ extrema $y_k^{(n)}$, we have alternating signs

$$q(y_0^{(n)}) > 0, \quad q(y_1^{(n)}) < 0, \quad \dots$$

so $q(x)$ changes sign n times on $[-1, 1]$, which is impossible for a polynomial of degree $\leq n - 1$. □

Theorem 1.14 (Chebyshev Interpolation Error). *Let p_n be the interpolant of f at the $n + 1$ Chebyshev nodes*

$$x_k^{(n+1)} = \cos\left(\frac{(2k+1)\pi}{2(n+1)} \right), \quad k = 0, \dots, n.$$

Then,

$$\|f - p_n\|_\infty \leq \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \left\| \prod_{k=0}^{n} (x - x_k^{(n+1)}) \right\|_\infty = \frac{\|f^{(n+1)}\|_\infty}{(n+1)!} \frac{1}{2^n}.$$

For interpolation on $[a, b]$, the Chebyshev nodes are mapped via

$$\hat{x}_k^{(n+1)} = \frac{b - a}{2} x_k^{(n+1)} + \frac{a + b}{2}.$$

Definition 1.15 (Interpolation Operator). Define

$$I_n f := P_n(f; x) = \sum_{i=0}^{n} f(x_i) \, \ell_i(x).$$

For any $p \in \mathcal{P}_n$, we have $I_n p = p$.

Definition 1.16 (Lebesgue Constant). Operator norm of I_n satisfies

$$\|I_n\| = \sup_{f\neq 0} \frac{\|I_n f\|_\infty}{\|f\|_\infty} = \max_{x\in[-1,1]} \sum_{i=0}^{n} |\ell_i(x)| =: \Lambda_n.$$

For Chebyshev points,

$$\Lambda_n = O(\log n).$$

For many other interpolation nodes, Λ_n grows exponentially.

$$\begin{aligned}\|f - I_n f\|_\infty &= \|f - p - I_n(f-p)\|_\infty \\ &\le \|f-p\|_\infty + \|I_n(f-p)\|_\infty \\ &\le (1 + \|I_n\|)\,\|f-p\|_\infty.\end{aligned}$$

Then,

$$\|f - I_n f\|_\infty \le (1+\Lambda_n)\, E_n(f),$$

where

$$E_n(f) := \min_{p\in\mathcal{P}_n} \|f-p\|_\infty.$$

If $f \in C^1[-1,1]$, then

$$E_n(f) = O\left(\frac{1}{n}\right),$$

and therefore

$$\|f - I_n f\|_\infty = O\left(\frac{\log n}{n}\right) \longrightarrow 0 \quad \text{as } n\to\infty.$$

Definition 1.17 (Barycentric Formula). Define

$$w_n(x) := \prod_{j=0}^{n}(x-x_j), \quad \lambda_i^{(n)} := \frac{1}{\prod_{\substack{j=0\\ j\neq i}}^{n}(x_i - x_j)}.$$

Then,

$$\ell_i(x) = \frac{\lambda_i^{(n)}}{x-x_i}\, w_n(x).$$

Since $\sum_{i=0}^n \ell_i(x) \equiv 1$, we obtain

$$P_n(f;x) = \frac{\displaystyle\sum_{i=0}^{n} \frac{\lambda_i^{(n)}}{x-x_i} f(x_i)}{\displaystyle\sum_{i=0}^{n} \frac{\lambda_i^{(n)}}{x-x_i}}, \quad x \neq x_i.$$

Initialize

$$\lambda_0^{(0)} = 1.$$

For $k = 1,\dots,n$, define

$$\lambda_i^{(k)} = \frac{\lambda_i^{(k-1)}}{x_i - x_k}, \quad i = 0,\dots,k-1, \quad \lambda_k^{(k)} = \frac{1}{\prod_{j=0}^{k-1}(x_k - x_j)}.$$

The full set $\{\lambda_i^{(n)}\}_{i=0}^n$ can be computed in $O(n^2)$ time. Adding one more node costs $O(n)$, and evaluating $P_n(x)$ at a point costs $O(n)$.

Definition 1.18 (Divided Differences). Given distinct points $x_0, x_1, \dots, x_n$, define

$$[x_i]f = f(x_i),$$

and recursively for $i < j$,

$$[x_i,\dots,x_j]f = \frac{[x_{i+1},\dots,x_j]f - [x_i,\dots,x_{j-1}]f}{x_j - x_i}.$$

Theorem 1.19 (Newton Interpolation Polynomial). *Let $x_0,\dots,x_n$ be distinct. The interpolation polynomial of degree at most n is*

$$p_n(x) = \sum_{i=0}^{n} [x_0,\dots,x_i]f \prod_{j=0}^{i-1}(x-x_j),$$

where the empty product (for $i=0$) is defined as 1.

Proof. By induction. For $i=0$, we have

$$a_0 = f(x_0) = [x_0]f.$$

Assume

$$a_i = [x_0,\dots,x_i]f.$$

Then,

$$p_{i+1}(x) = \sum_{j=0}^{i}[x_0,\dots,x_j]f\prod_{k=0}^{j-1}(x-x_k) + a_{i+1}\prod_{k=0}^{i}(x-x_k)$$

$$= \sum_{j=1}^{i+1}[x_1,\dots,x_{j+1}]f\prod_{k=1}^{j}(x-x_k) + a_{i+1}\prod_{k=1}^{i+1}(x-x_k).$$

Take i-th derivative, so

$$a_{i+1} = \frac{[x_1,\dots,x_{i+1}]f - [x_0,\dots,x_i]f}{x_{i+1}-x_0} = [x_0,\dots,x_{i+1}]f.$$

□

Theorem 1.20 (Interpolation Error Formula). *Let $x_0,\dots,x_n \in [a,b]$ be distinct and $f \in C^n[a,b]$. Then,*

$$[x_0,\dots,x_n]f = \frac{f^{(n)}(\xi)}{n!}, \quad \xi\in(a,b).$$

If all nodes coalesce, i.e. $x_1,\dots,x_n \to x_0$, then

$$[x_0,\dots,x_0]f = \frac{f^{(n)}(x_0)}{n!}, \quad p_n(x) = \sum_{i=0}^{n}\frac{f^{(i)}(x_0)}{i!}(x-x_0)^i.$$

Definition 1.21 (Hermite Interpolation Problem). Given $k+1$ distinct points $x_0,\dots,x_k$, and integers $m_i \ge 0$, find p_n such that

$$p_n^{(j)}(x_i) = f^{(j)}(x_i), \quad j = 0,\dots,m_i, \quad i = 0,\dots,k.$$

The total number of conditions is

$$n+1 = \sum_{i=0}^{k}(m_i+1), \quad p_n(x) = \sum_{i=0}^{n}[x_0,\dots,x_i]f\prod_{j=0}^{i-1}(x-x_j)$$

Definition 1.22 (Generalized Divided Differences). Define

$$[x_i,\dots,x_j]f = \begin{cases} \dfrac{[x_{i+1},\dots,x_j]f - [x_i,\dots,x_{j-1}]f}{x_j - x_i}, & x_i \neq x_j, \\ \dfrac{f^{(j-i)}(x_i)}{(j-i)!}, & x_i = x_j. \end{cases}$$

1.3 Approximation and Interpolation by Spline Functions

Definition 1.23 (Partition and Spline Spaces). Let

$$\Delta = \{a = x_1 < x_2 < \cdots < x_n = b\}$$

be a partition of $[a, b]$. Define

$$S_m^k(\Delta) = \left\{ s \in C^k[a,b] \;\middle|\; s|_{[x_i, x_{i+1}]} \in \mathbb{P}_m,\ i = 1, \ldots, n-1 \right\}.$$

$\mathbb{P}_m$ denotes polynomials of degree $\leq m$, k controls smoothness across subintervals.

Definition 1.24 (Piecewise Linear Interpolation). Find $s \in S_1^0(\Delta)$ s.t.

$$s(x_i) = f_i = f(x_i), \quad i = 1, \ldots, n.$$

On each interval $[x_i, x_{i+1}]$, the interpolant is

$$s_1(f; x) = f_i + \frac{f_{i+1} - f_i}{x_{i+1} - x_i}(x - x_i), \quad x_i \leq x \leq x_{i+1}.$$

If $f \in C^2[a, b]$, then for $x \in [x_i, x_{i+1}]$,

$$f(x) - s_1(f; x) = \frac{f''(\xi)}{2!}(x - x_i)(x - x_{i+1}), \quad \xi \in (x_i, x_{i+1}).$$

Thus,

$$|f(x) - s_1(f; x)| \leq \frac{\max_{x \in [a,b]} |f''(x)|}{2} \cdot \frac{(x_{i+1} - x_i)^2}{4}.$$

Each subinterval contributes 2 degrees of freedom. Total $2(n-1)$ parameters. $(n-2)$ continuity constraints at interior nodes. Hence,

$$\dim S_1^0(\Delta) = 2(n-1) - (n-2) = n.$$

Definition 1.25 (Hat Function Basis). Let $\{B_i(x)\}_{i=1}^n$ be the hat functions satisfying

$$B_i(x_j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

Then the interpolant can be written as

$$s(x) = \sum_{i=1}^{n} f(x_i) B_i(x).$$

Definition 1.26 (Cubic Splines). Consider the space

$$S_3^2(\Delta),$$

the space of *cubic splines*. At each interior node x_i $(i = 2, \ldots, n-1)$,

$$p'_{i-1}(x_i) = p'_i(x_i), \quad p''_{i-1}(x_i) = p''_i(x_i).$$

Unknowns: $4(n-1)$. Interpolation: $2(n-1)$. C^2 continuity: $2(n-2)$. So we still need two additional boundary conditions.

- **Complete spline**. $p'_1(a) = f'(a),\ p'_{n-1}(b) = f'(b)$.
- **Natural spline**. $p''_1(a) = 0,\ p''_{n-1}(b) = 0$.
- **Not-a-knot spline**. $p_1(x) \equiv p_2(x),\ p_{n-2}(x) \equiv p_{n-1}(x)$.

For the complete spline,

$$\|f - s_3(f; \cdot)\|_\infty \leq \frac{5}{384} \|f^{(4)}\|_\infty h^4.$$

2 Numerical Differentiation and Integration

2.1 Numerical Differentiation

Definition 2.1 (Interpolation Framework).

$$f(x) = p_n(f; x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^{n} (x - x_i),$$

for some $\xi(x)$ in the convex hull of $\{x_0, \ldots, x_n\}$.

$$f'(x) = p'_n(f; x) + \frac{d}{dx}\left[\frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^{n} (x - x_i) \right].$$

$$f'(x_0) = p'_n(f; x_0) + \frac{f^{(n+1)}(\xi(x_0))}{(n+1)!} \prod_{i=1}^{n} (x_0 - x_i),$$

where the second term is the *truncation error*.

Theorem 2.2 (Effect of Noisy Data). *Assume data are perturbed*

$$\tilde{f}(x_0 + h) = f(x_0 + h) + \varepsilon_1, \quad \tilde{f}(x_0 - h) = f(x_0 - h) + \varepsilon_2,$$

with

$$|\varepsilon_1|, |\varepsilon_2| \leq \varepsilon.$$

Then the computed derivative satisfies

$$\tilde{f}'(x_0) = \frac{\tilde{f}(x_0 + h) - \tilde{f}(x_0 - h)}{2h} - \frac{f^{(3)}(\xi(x_0))}{6} h^2.$$

Hence,

$$\left| f'(x_0) - \frac{\tilde{f}(x_0 + h) - \tilde{f}(x_0 - h)}{2h} \right| \leq \frac{\varepsilon}{h} + \frac{M_3}{6} h^2,$$

where

$$M_3 = \max_{x \in [x_0 - h, x_0 + h]} |f^{(3)}(x)|.$$

Balancing truncation and roundoff errors

$$\frac{M_3}{6} h^2 \approx \frac{\varepsilon}{h} \quad \Rightarrow \quad h \sim \varepsilon^{1/3}.$$

2.2 Numerical Integration

Definition 2.3 (Interpolation Approach). Let $p_n(f; x)$ be the interpolating polynomial of degree n at nodes $x_0, x_1, \ldots, x_n$. Then

$$\int_a^b f(x)\, dx = \int_a^b p_n(f; x)\, dx + \int_a^b \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^{n} (x - x_i)\, dx.$$

$$\int_a^b f(x)\, dx = \sum_{i=0}^{n} f(x_i) \underbrace{\int_a^b \ell_i(x)\, dx}_{a_i} + \text{error},$$

where the weights a_i depend only on the nodes x_i, not on f.

Theorem 2.4 (Mean Value Theorem for Integrals). *If $f \in C[a, b]$ and g is integrable and does not change sign on $[a, b]$, then*

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

Definition 2.5 (Newton–Cotes Formulas). Using equi-spaced points leads to the Newton–Cotes formulas. However, for $n \geq 8$, these formulas become unstable.

Definition 2.6 (Composite Quadrature Rules). Partition $[a,b]$ into n subintervals

$$a = x_0 < x_1 < \cdots < x_n = b, \quad x_i = a + ih, \quad h = \frac{b-a}{n}.$$

Composite Trapezoidal Rule.

$$\int_a^b f(x)\,dx = \frac{h}{2}\big(f_0+2f_1+\ldots+2f_{n-1}+f_n\big) - \frac{f''(\xi)}{12}(b-a)h^2, \ \xi \in (a,b).$$

Composite Midpoint Rule.

$$\int_a^b f(x)\,dx = h\big(f_{1/2} + f_{3/2} + \cdots + f_{n-1/2}\big) + \frac{f''(\xi)}{24}(b-a)h^2.$$

Composite Simpson's Rule.

$$\int_a^b f(x)\,dx = \sum_{i=0}^{n-1} \frac{h}{6}\big(f(x_i)+4f(x_{i+1/2})+f(x_{i+1})\big) - \frac{f^{(4)}(\xi)}{180}(b-a)h^4.$$

Definition 2.7 (Spectral Accuracy). Both the composite trapezoidal and midpoint rule exhibit spectral accuracy for periodic functions

$$f \in C^m_{\text{periodic}}[a,b] \quad \Longrightarrow \quad E_n(f) = O(h^m).$$

Definition 2.8 (Gaussian Quadrature Rules). Consider numerical approximation of integrals of the form

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

where $w(x) \geq 0$ is a given weight function and the interval $[a,b]$ may be finite or infinite. Seek a quadrature rule of the form

$$\int_a^b f(x)\,w(x)\,dx = \sum_{i=0}^{n} a_i f(x_i) + E_n(f),$$

where $x_0, \ldots, x_n$ are the nodes, $a_0, \ldots, a_n$ are the weights, and $E_n(f)$ is the error term.

Definition 2.9 (Degree of Exactness). The *degree of exactness* (or precision) of the quadrature rule is the largest integer d such that

$$E_n(p) = 0 \quad \text{for all } p \in \mathbb{P}_d.$$

If the quadrature rule satisfies

$$E_n(p) = 0 \quad \text{for all } p \in \mathbb{P}_n,$$

then we call the rule *interpolatory*.

Definition 2.10 (Interpolatory Quadrature). Let $\ell_i(x)$ be the Lagrange basis polynomials associated with the nodes $x_0, \ldots, x_n$. Then,

$$\int_a^b f(x)w(x)\,dx = \sum_{i=0}^{n} f(x_i) \int_a^b \ell_i(x)w(x)\,dx + E_n(f).$$

Thus,

$$a_i = \int_a^b \ell_i(x)w(x)\,dx.$$

If $f \in \mathbb{P}_n$, then $f^{(n+1)} \equiv 0$, hence $E_n(f) = 0$. Therefore, interpolatory quadrature has degree of exactness at least n. The maximum possible degree of exactness is

$$d + 1 \text{ (\# of eqs)} \leq (n+1) + (n+1) \text{ (\# of unknown)} \ \Rightarrow \ d \leq 2n+1.$$

Theorem 2.11. *Given an integer k with $0 \leq k \leq n+1$, the quadrature rule has degree of exactness $d = n + k$ if and only iff*

(1) *The rule is interpolatory (exact for $\mathbb{P}_n$).*

(2) *For all $p \in \mathbb{P}_{k-1}$,*

$$\int_a^b \left(\prod_{i=0}^{n}(x - x_i)\right) p(x)w(x)\,dx = 0.$$

Proof. ($\Rightarrow$) Assume $d = n + k$. Then $d \geq n$, so the rule is interpolatory. Let $p \in \mathbb{P}_{k-1}$. Then

$$\prod_{i=0}^{n}(x - x_i)p(x) \in \mathbb{P}_{n+k}.$$

By exactness,

$$\int_a^b \prod_{i=0}^{n}(x - x_i)p(x)w(x)\,dx = \sum_{j=0}^{n} a_j \prod_{i=0}^{n}(x_j - x_i)p(x_j) = 0.$$

($\Leftarrow$) Let $p \in \mathbb{P}_{n+k}$. Then we can write

$$p(x) = q(x)\prod_{i=0}^{n}(x - x_i) + r(x),$$

where $q \in \mathbb{P}_{k-1}$ and $r \in \mathbb{P}_n$. Thus,

$$\int_a^b p(x)w(x)\,dx = \int_a^b q(x)\prod_{i=0}^{n}(x-x_i)w(x)\,dx + \int_a^b r(x)w(x)\,dx$$

$$= \sum_{i=0}^{n} a_i r(x_i) = \sum_{i=0}^{n} a_i r(x_i) + \sum_{i=0}^{n} q(x_i)\prod_{j=0}^{n}(x_i - x_j) = \sum_{i=0}^{n} a_i p(x_i).$$

□

Definition 2.12 (Gauss Quadrature). If $k = n+1$, then the degree of exactness is $d = 2n+1$, which is the best possible. Such a rule is called *Gaussian quadrature*. For a general interval $[a,b]$, define

$$t = \frac{b-a}{2}x + \frac{a+b}{2}, \quad x \in [-1,1].$$

$$\int_a^b f(t)\,dt = \int_{-1}^{1} f\left(\frac{b-a}{2}x + \frac{a+b}{2}\right)\frac{b-a}{2}\,dx \approx \sum_{j=0}^{n} b_j f(t_j),$$

where

$$t_j = \frac{b-a}{2}x_j + \frac{a+b}{2}, \quad b_j = \frac{b-a}{2}a_j.$$

- $\displaystyle\int_{-1}^{1} f(x)\,dx \ \Rightarrow$ roots of Legendre polynomials
- $\displaystyle\int_0^{\infty} f(x)e^{-x}\,dx \ \Rightarrow$ roots of Laguerre polynomials
- $\displaystyle\int_{-\infty}^{\infty} f(x)e^{-x^2}\,dx \ \Rightarrow$ roots of Hermite polynomials
- $\displaystyle\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}\,dx \ \Rightarrow$ roots of Chebyshev polynomials

Theorem 2.13 (Positivity of Weights). *All quadrature weights*

$$a_i > 0.$$

Since $\ell_i(x)^2 \in \mathbb{P}_{2n}$,

$$0 < \int_a^b \ell_i(x)^2 w(x)\,dx = \sum_{j=0}^{n} a_j \ell_i(x_j)^2 = a_i.$$

Theorem 2.14 (Quadrature Error Formula). *Define the quadrature error*

$$E_n(f) = \int_a^b f(x)w(x)\,dx - \sum_{i=0}^{n} a_i f(x_i).$$

Let $H_{2n+1}(x)$ be the Hermite interpolating polynomial satisfying

$$H_{2n+1}(x_i) = f(x_i), \quad H'_{2n+1}(x_i) = f'(x_i), \quad i = 0, \dots, n.$$

Then

$$E_n(f) = \int_a^b \big(f(x) - H_{2n+1}(x)\big)w(x)\,dx.$$

Using the Hermite interpolation error formula,

$$f(x) - H_{2n+1}(x) = \frac{f^{(2n+2)}(\xi(x))}{(2n+2)!}\prod_{i=0}^{n}(x-x_i)^2,$$

we obtain

$$E_n(f) = \int_a^b \frac{f^{(2n+2)}(\xi(x))}{(2n+2)!}\prod_{i=0}^{n}(x-x_i)^2 w(x)\,dx.$$

By the mean value theorem, there exists $\xi \in (a,b)$ such that

$$E_n(f) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!}\int_a^b \prod_{i=0}^{n}(x-x_i)^2 w(x)\,dx.$$

Definition 2.15 (Gauss–Lobatto and Gauss–Radau Quadrature). To include both endpoints a and b, choose interior nodes $x_1, \dots, x_{n-1}$ such that

$$\int_a^b \prod_{i=1}^{n-1}(x-x_i)\,p(x)\,(x-a)(x-b)\,w(x)\,dx = 0, \quad \forall p \in \mathbb{P}_{n-2}.$$

This leads to choosing $x_1, \dots, x_{n-1}$ as the roots of the orthogonal polynomial $\pi_{n-1}(x)$ with respect to the modified weight

$$(x-a)(b-x)w(x) > 0 \quad \text{on } [a,b].$$

Including both endpoints gives **Gauss–Lobatto quadrature**. Including exactly one endpoint gives **Gauss–Radau quadrature**.

Definition 2.16 (Richardson Extrapolation Methods). Suppose an approximation $A(h)$ satisfies

$$A^* = A(h) + Ch^k + o(h^k),$$

where A^* is the exact value. Evaluating the approximation at h/t,

$$A^* = A\left(\frac{h}{t}\right) + C\left(\frac{h}{t}\right)^k + o(h^k), \quad t^k A^* = t^k A\left(\frac{h}{t}\right) + Ch^k + o(h^k).$$

$$A^* = \frac{t^k A\left(\frac{h}{t}\right) - A(h)}{t^k - 1} + o(h^k).$$

Definition 2.17 (Romberg Integration). Romberg integration applies Richardson extrapolation recursively to the composite trapezoidal rule, producing a sequence of increasingly accurate approximations by systematically eliminating lower-order error terms.

Definition 2.18 (Adaptive Algorithm). Evaluate $R(h)$, $R(h/2)$, estimate $E(h/2)$. If $|E| <$ tol, accept $Q = R(h/2)$. Otherwise, split the interval

$$Q_1 = \text{fuc}\left(a, \tfrac{a+b}{2}, \tfrac{\text{tol}}{2}\right), \quad Q_2 = \text{fuc}\left(\tfrac{a+b}{2}, b, \tfrac{\text{tol}}{2}\right).$$

Set $Q = Q_1 + Q_2$.

3 Nonlinear Equations

3.1 Iteration, Convergence, and Efficiency

Definition 3.1 (Linear Convergence). A sequence $\{x_n\}$ converges to α (at least) linearly if

$$|x_n - \alpha| \le \varepsilon_n,$$

where $\{\varepsilon_n\}$ is a positive sequence satisfying

$$\lim_{n\to\infty} \frac{\varepsilon_{n+1}}{\varepsilon_n} = C, \quad 0 < C < 1.$$

If $C = 1$, the convergence is called *sublinear*.

Definition 3.2 (Convergence of Order p). We say $\{x_n\}$ converges to α with (at least) order $p > 1$ if

$$|x_n - \alpha| \le \varepsilon_n, \quad \lim_{n\to\infty} \frac{\varepsilon_{n+1}}{\varepsilon_n^p} = C, \quad C > 0.$$

Definition 3.3 (Stopping Rules). Typical stopping criteria include $|x_n - x_{n-1}| < \varepsilon_a$, $|x_n - x_{n-1}| < \varepsilon_r|x_n|$, $|f(x_n)| < \varepsilon_f$.

3.2 The Methods of Bisection and Sturm Sequences

Theorem 3.4 (Intermediate Value Theorem). *Let $f \in C[a,b]$. If a value u lies between $f(a)$ and $f(b)$, then there exists $c \in (a,b)$ s.t.*

$$f(c) = u.$$

Definition 3.5 (The Methods of Bisection). Assume $f \in C[a,b]$ and $f(a)f(b) < 0$.

(1) Set $x_1 = \frac{a+b}{2}$.
(2) Evaluate $f(x_1)$.
(3) Check the sign of $f(a)f(x_1)$.
(4) Replace either a or b by x_1 to form a smaller interval.

After n iterations,

$$|x_n - \alpha| \le \frac{b-a}{2^n}.$$

Moreover,

$$\frac{\varepsilon_{n+1}}{\varepsilon_n} = \frac{1}{2} \quad \text{for all } n,$$

so the bisection method has *linear convergence*.

3.3 Newton's Method

Definition 3.6 (Newton's Method).

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \dots$$

Let α be a simple root of f. Then

$$x_{n+1} - \alpha = (x_n - \alpha)\left(1 - \frac{f(x_n) - f(\alpha)}{(x_n - \alpha)f'(x_n)}\right) = (x_n - \alpha)^2 \frac{[x_n, x_n, \alpha]f}{[x_n, x_n]f}.$$

Therefore,

$$\frac{x_{n+1} - \alpha}{(x_n - \alpha)^2} = \frac{[x_n, x_n, \alpha]f}{[x_n, x_n]f}.$$

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

Thus, Newton's method converges *quadratically*. Newton's method is *locally convergent*, i.e., one must start sufficiently close to the desired root in order to achieve quadratic convergence.

3.4 Secant Method

Definition 3.7 (Secant Method). The secant method is given by

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n), \quad n = 1, 2, \dots$$

It requires two starting values x_0, x_1 with $x_0 \neq x_1$. Assume $x_n \to \alpha$, where α is a root of f, i.e. $f(\alpha) = 0$.

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

$$= (x_n - \alpha)\left(1 - \frac{[x_n, \alpha]f}{[x_{n-1}, x_n]f}\right) = (x_n - \alpha)(x_{n-1} - \alpha)\frac{[x_{n-1}, x_n, \alpha]f}{[x_{n-1}, x_n]f}.$$

Taking limits as $n \to \infty$,

$$\lim_{n\to\infty} \left|\frac{x_{n+1} - \alpha}{x_n - \alpha}\right| = 0,$$

so the convergence is faster than linear. Define

$$\varepsilon_n = |x_n - \alpha|.$$

Then for n large,

$$\varepsilon_{n+1} = C\,\varepsilon_n \varepsilon_{n-1}, \quad C = \left|\frac{[x_{n-1}, x_n, \alpha]f}{[x_{n-1}, x_n]f}\right| \to \left|\frac{f''(\alpha)/2}{f'(\alpha)}\right| > 0.$$

Let $E_n = C\varepsilon_n$, then

$$E_{n+1} = E_n E_{n-1} \implies \log E_{n+1} = \log E_n + \log E_{n-1}$$

$$t^2 - t - 1 = 0 \implies t_{1,2} = \frac{1 \pm \sqrt{5}}{2} \implies \log E_n \sim -(\frac{1+\sqrt{5}}{2})^n$$

$$\varepsilon_n \sim C^{-1} e^{-(\frac{1+\sqrt{5}}{2})^n} \implies \lim_{n\to\infty} \frac{\varepsilon_{n+1}}{(\varepsilon_n)^{\frac{1+\sqrt{5}}{2}}} = C^{\frac{1+\sqrt{5}}{2}-1}$$

Hence the order of convergence is

$$p = \frac{1+\sqrt{5}}{2} \approx 1.618.$$

Theorem 3.8 (Local Convergence of Newton and Secant Methods). *Let α be a simple root of f*

$$f(\alpha) = 0, \quad f'(\alpha) \neq 0.$$

Assume $f \in C^2[a, b]$. Then there exists $\delta > 0$ such that Newton's method and the secant method converge locally for initial guesses sufficiently close to α.

3.5 Fixed Point Iteration

Definition 3.9 (Fixed Point Iteration). If $f(\alpha) = 0$, then $\alpha = \varphi(\alpha)$, so α is a fixed point of φ. Iterate

$$x_{n+1} = \varphi(x_n), \quad n = 0, 1, 2, \dots$$

If $x_n \to \alpha$, then α solves the original equation.

Theorem 3.10 (Fixed Point Theorem (Contraction Mapping Theorem)). *Let $\varphi : [a, b] \to [a, b]$ and assume*

$$\varphi \in C^1[a, b],$$

and there exists $0 < \gamma < 1$ such that

$$|\varphi'(x)| \leq \gamma, \quad \forall x \in [a, b].$$

Then there exists a unique fixed point $\alpha \in [a, b]$, and the fixed point iteration converges for any initial value $x_0 \in [a, b]$.

Proof. Let

$$F(x) = \varphi(x) - x.$$

Then,

$$F(a) = \varphi(a) - a \geq 0, \quad F(b) = \varphi(b) - b \leq 0.$$

Hence there exists $\alpha \in [a, b]$ such that

$$\alpha = \varphi(\alpha).$$

If there exists another fixed point $\alpha_1 \neq \alpha$, then

$$|\alpha - \alpha_1| = |\varphi(\alpha) - \varphi(\alpha_1)| = |\varphi'(\xi)|\,|\alpha - \alpha_1| \leq \gamma|\alpha - \alpha_1| < |\alpha - \alpha_1|,$$

a contradiction. □

Theorem 3.11 (Local convergence of fixed point iteration). *Let α be a fixed point of φ. Assume*

$$\varphi'(\alpha) = \varphi''(\alpha) = \cdots = \varphi^{(p-1)}(\alpha) = 0, \quad \varphi^{(p)}(\alpha) \neq 0, \quad p \geq 1,$$

and $\varphi \in C^p(I_\varepsilon)$ where $I_\varepsilon = [\alpha - \varepsilon, \alpha + \varepsilon]$. Define

$$M(\varepsilon) = \max_{t \in I_\varepsilon} |\varphi'(t)|.$$

Choose ε such that $M(\varepsilon) \leq \gamma < 1$. Then for $x_0 \in I_\varepsilon$, the iteration $x_{n+1} = \varphi(x_n)$ converges to α. Moreover, its order of convergence is p.

Proof. For $x \in I_\varepsilon$, by MVT, for some ξ between x and α,

$$|\varphi(x) - \alpha| = |\varphi(x) - \varphi(\alpha)| = |\varphi'(\xi)|\,|x - \alpha| \leq \gamma|x - \alpha| < |x - \alpha| \leq \varepsilon.$$

So $\varphi(x) \in I_\varepsilon$, and we may apply the contraction fixed point theorem.

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

Hence,

$$\lim_{n\to\infty} \left|\frac{x_{n+1} - \alpha}{(x_n - \alpha)^p}\right| = \left|\frac{1}{p!}\varphi^{(p)}(\alpha)\right| \neq 0.$$

Therefore, the convergence is of order p. □

3.6 Systems of Nonlinear Equations

Consider solving $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^d$.

$$\mathbf{J}(\mathbf{x}_n)\,\mathbf{p}_n = -\mathbf{f}(\mathbf{x}_n), \quad \mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{p}_n.$$

Let $\mathbf{A}_n$ approximate $\mathbf{J}(\mathbf{x}_n)$.

$$\mathbf{A}_n(\mathbf{x}_n - \mathbf{x}_{n-1}) = \mathbf{f}(\mathbf{x}_n) - \mathbf{f}(\mathbf{x}_{n-1}).$$

Define $\mathbf{p}_{n-1} = \mathbf{x}_n - \mathbf{x}_{n-1}$, $\mathbf{g}_{n-1} = \mathbf{f}(\mathbf{x}_n) - \mathbf{f}(\mathbf{x}_{n-1})$.

Assume a rank-one update,

$$\mathbf{A}_n = \mathbf{A}_{n-1} + \mathbf{u}_{n-1}\mathbf{p}_{n-1}^T.$$

Plugging into the secant condition,

$$\mathbf{A}_{n-1}\mathbf{p}_{n-1} + \mathbf{u}_{n-1}(\mathbf{p}_{n-1}^T\mathbf{p}_{n-1}) = \mathbf{g}_{n-1}.$$

Thus,

$$\mathbf{u}_{n-1} = \frac{\mathbf{g}_{n-1} - \mathbf{A}_{n-1}\mathbf{p}_{n-1}}{\mathbf{p}_{n-1}^T\mathbf{p}_{n-1}}.$$

For a rank-one update,

$$(\mathbf{A} + \mathbf{x}\mathbf{y}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{x}\mathbf{y}^T\mathbf{A}^{-1}}{1 + \mathbf{y}^T\mathbf{A}^{-1}\mathbf{x}}.$$

Sherman–Morrison Formula

$$\mathbf{A}_n^{-1} = \mathbf{A}_{n-1}^{-1} - \frac{\mathbf{A}_{n-1}^{-1}(\mathbf{g}_{n-1} - \mathbf{A}_{n-1}\mathbf{p}_{n-1})\mathbf{p}_{n-1}^T\mathbf{A}_{n-1}^{-1}}{\mathbf{p}_{n-1}^T\mathbf{p}_{n-1} + \mathbf{p}_{n-1}^T\mathbf{A}_{n-1}^{-1}(\mathbf{g}_{n-1} - \mathbf{A}_{n-1}\mathbf{p}_{n-1})}.$$

4 Initial Value Problems for ODEs

4.1 Taylor Expansion Methods

Using the Taylor series,

$$y(t_{n+1}) = y(t_n) + h_n y'(t_n) + \frac{h_n^2}{2} y''(t_n) + \cdots + \frac{h_n^p}{p!} y^{(p)}(t_n) + O(h_n^{p+1}).$$

$$y''(t) = \frac{d}{dt} f(t, y(t)) = \frac{\partial f}{\partial t}(t, y(t)) + \frac{\partial f}{\partial y}(t, y(t))\, f(t, y(t)).$$

4.2 Integral Formulation and Quadrature

Integrate $y'(t) = f(t, y(t))$ over $[t_n, t_{n+1}]$,

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t))\, dt.$$

Trapezoidal Method (Implicit, Second-Order)

$$y_{n+1} = y_n + \frac{h_n}{2}\Big(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\Big).$$

Heun's / Improved Euler Method (Explicit, Second-Order)

$$Y = y_n + h_n f(t_n, y_n),$$

$$y_{n+1} = y_n + \frac{h_n}{2}\Big(f(t_n, y_n) + f(t_{n+1}, Y)\Big).$$

Midpoint Method (Implicit)

$$y_{n+1} = y_n + h\, f\Big(t_{n+\frac{1}{2}},\ \frac{y_n + y_{n+1}}{2}\Big).$$

RK2 Method (Explicit)

$$Y = y_n + \frac{h}{2} f(t_n, y_n),$$

$$y_{n+1} = y_n + h\, f(t_{n+\frac{1}{2}}, Y).$$

General s-stage Runge–Kutta methods (Butcher tableau)

$$Y_i = y_n + h \sum_{j=1}^{s} a_{ij}\, f(t_n + c_j h,\ Y_j), \quad i = 1, \ldots, s,$$

$$y_{n+1} = y_n + h \sum_{j=1}^{s} b_j\, f(t_n + c_j h,\ Y_j).$$

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

First order condition

$$\sum_{j=1}^{s} a_{ij} = c_i, \quad i = 1, \ldots, s, \qquad \sum_{j=1}^{s} b_j = 1.$$

Second order condition

$$\sum_{j=1}^{s} b_j c_j = \frac{1}{2}.$$

Third order condition

$$\sum_{j=1}^{s} b_j c_j^2 = \frac{1}{3}, \qquad \sum_{i=1}^{s} \sum_{j=1}^{s} b_i a_{ij} c_j = \frac{1}{6}.$$

Forward Euler

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

Backward Euler

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

Explicit midpoint

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array}$$

Implicit midpoint

$$\begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}$$

Classical RK4

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

4.3 Error and Stability

Theorem 4.1. *For a general one-step p-th order explicit method*

$$y_{n+1} = y_n + h\, \varphi(t_n, y_n, h),$$

let

$$\overline{y}_{n+1} = y(t_n) + h\, \varphi(t_n, y(t_n), h).$$

Then

$$\begin{aligned} |E_{n+1}| &\le |y_{n+1} - \overline{y}_{n+1}| + |\overline{y}_{n+1} - y(t_{n+1})| \\ &\le |y_n + h\varphi(t_n, y_n, h) - y(t_n) - h\varphi(t_n, y(t_n), h)| + h|T_n| \\ &\le |y_n - y(t_n)| + h\,|\varphi(t_n, y_n, h) - \varphi(t_n, y(t_n), h)| + h|T_n| \\ &\le (1 + hL_\varphi)|E_n| + h\, c\, h^p. \end{aligned}$$

Assume

$$|\varphi(t, y_1, h) - \varphi(t, y_2, h)| \le L_\varphi\, |y_1 - y_2|$$

$$|E_n| \le Ch^p\, \frac{1}{L_\varphi} \left(e^{(b-a)L_\varphi} - 1\right) \sim O(h^p).$$

Definition 4.2 (Absolute Stability). Generally, for a one-step method, when applied to the test problem

$$y' = \lambda y,$$

we have

$$y_{n+1} = R(z) y_n, \quad z := \lambda h.$$

The *absolute stability region* is

$$\{z \in \mathbb{C} \mid |R(z)| \le 1\}.$$

Definition 4.3 (A-stability). A method whose absolute stability region contains the entire left half-plane is called *A-stable*.

Theorem 4.4 (Dahlquist). *Any A-stable multistep method is at most second-order accurate.*

Definition 4.5 (L-stability). A method is *L-stable* if it is A-stable and

$$|R(z)| \to 0 \quad \text{as } |z| \to \infty.$$

Definition 4.6 ($A(\alpha)$-stability). Let $\arg(z) = \pi$ represent the negative real axis. If the wedge

$$\pi - \alpha < \arg(z) < \pi + \alpha$$

is contained in the absolute stability region, then the method is called

$$A(\alpha)\text{-stable.}$$

In particular,

$$A\text{-stable} \iff A\Big(\frac{\pi}{2}\Big)\text{-stable.}$$

4.4 Multistep Methods

Definition 4.7 (Multistep Methods). Generally, an s-step method can be written as

$$y_{n+1} = -\sum_{j=1}^{s} \alpha_j y_{n+1-j} + h \sum_{j=0}^{s} \beta_j f(t_{n+1-j}, y_{n+1-j}).$$

If $\beta_0 \neq 0$, the method is implicit

$$y_{n+1} - h\beta_0 f(t_{n+1}, y_{n+1}) = -\sum_{j=1}^{s} \alpha_j y_{n+1-j} + h \sum_{j=1}^{s} \beta_j f(t_{n+1-j}, y_{n+1-j}).$$

Definition 4.8 (Local Truncation Error). The local truncation error (LTE) is defined by

$$T_n := \frac{1}{h} \sum_{j=0}^{s} \alpha_j \, y(t_{n+1-j}) - \sum_{j=0}^{s} \beta_j \, f(t_{n+1-j}, y(t_{n+1-j})).$$

Definition 4.9 (The Adams Family).

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t))\, dt.$$

Approximate f by an interpolating polynomial

$$f(t) \approx p(t)$$

through previous values $f(t_\ell, y_\ell)$. Thus

$$\alpha_0 = 1, \quad \alpha_1 = -1, \quad \alpha_j = 0 \quad (j > 1).$$

Accuracy

$$O(h^s),$$

with $s = 1$ giving forward Euler. If using $t_n, t_{n-1}, \ldots, t_{n+1-s}$ to do interpolation (s points), we obtain the s-step explicit Adams method, called the Adams-Bashforth method.
Accuracy

$$O(h^{s+1}),$$

with $s = 0$ giving backward Euler. If using $t_{n+1}, t_n, \ldots, t_{n+1-s}$ to do interpolation ($s + 1$ points), we obtain the s-step implicit Adams method, called the Adams-Moulton method.

Definition 4.10 (Predictor-Corrector Methods).
Predictor (second-order Adams-Bashforth):

$$Y = y_n + \frac{h}{2}\Big(3f(t_n, y_n) - f(t_{n-1}, y_{n-1})\Big).$$

Corrector (third-order Adams-Moulton):

$$y_{n+1} = y_n + \frac{h}{12}\Big(5f(t_{n+1}, Y) + 8f(t_n, y_n) - f(t_{n-1}, y_{n-1})\Big).$$

Overall, the scheme is explicit and

$$O(h^3).$$

Definition 4.11 (Backward Differentiation Formula (BDF)). Interpolate y using

$$t_{n+1}, t_n, \ldots, t_{n+1-s}$$

($s + 1$ points), and set

$$p'(t_{n+1}) = f(t_{n+1}, y_{n+1}).$$

This gives the s-step BDF, with accuracy

$$O(h^s).$$

Thus

$$\beta_0 \neq 0, \quad \beta_j = 0 \quad (j > 0).$$

The two-step BDF formula is

$$\frac{3y_{n+1} - 4y_n + y_{n-1}}{2h} = f(t_{n+1}, y_{n+1}).$$

(Dahlquist) The BDF-s methods satisfy the root condition for

$$s \leq 6.$$

4.5 Convergence

Definition 4.12 (Zero-Stability of One-Step Method).

$$y_{n+1} = y_n + h\varphi(t_n, y_n, h), \quad (R_h\vec{u})_n := \frac{1}{h}(u_{n+1} - u_n) - \varphi(t_n, u_n, h).$$

The one-step method is called *zero stable* if there exists $K > 0$, not depending on h, such that for arbitrary two grid functions $\vec{u}, \vec{v}$, hold

$$\|\vec{u} - \vec{v}\|_\infty \leq K\Big(|u_0 - v_0| + \|R_h\vec{u} - R_h\vec{v}\|_\infty\Big)$$

for all $h \leq h_0$.

Theorem 4.13. *If $\varphi(t, y, h)$ satisfies the Lipschitz condition*

$$|\varphi(t, y_1, h) - \varphi(t, y_2, h)| \leq L_\varphi |y_1 - y_2|$$

for all

$$t \in [a, b], \quad y_1, y_2 \in \mathbb{R}, \quad h \leq h_0,$$

then the one-step method is zero stable.

Proof. For two arbitrary grid functions $\vec{u}, \vec{v}$,

$$u_{n+1} = u_n + h\varphi(t_n, u_n, h) + h(R_h\vec{u})_n,$$

$$v_{n+1} = v_n + h\varphi(t_n, v_n, h) + h(R_h\vec{v})_n.$$

$$u_{n+1} - v_{n+1} = u_n - v_n + h\big[\varphi(t_n, u_n, h) - \varphi(t_n, v_n, h)\big] + h\big[(R_h\vec{u})_n - (R_h\vec{v})_n\big].$$

$$|u_{n+1} - v_{n+1}| \leq |u_n - v_n| + hL_\varphi|u_n - v_n| + h\|R_h\vec{u} - R_h\vec{v}\|_\infty$$

$$= (1 + hL_\varphi)|u_n - v_n| + h\|R_h\vec{u} - R_h\vec{v}\|_\infty.$$

$$\|\vec{u} - \vec{v}\|_\infty \leq K\Big(|u_0 - v_0| + \|R_h\vec{u} - R_h\vec{v}\|_\infty\Big).$$

□

Theorem 4.14. *If the one-step method has LTE $O(h^p)$ and zero stable, then it converges and*

$$\|\vec{y} - \vec{y}^*\|_\infty = O(h^p),$$

where $\vec{y}$ is the numerical solution and $\vec{y}^*$ is the exact solution.

Proof.

$$\|\vec{y} - \vec{y}^*\|_\infty \le K\Big(|y_0 - y^*(t_0)| + \|R_h\vec{y} - R_h\vec{y}^*\|_\infty\Big).$$

Since

$$|y_0 - y^*(t_0)| = 0, \quad \|R_h\vec{y} - R_h\vec{y}^*\|_\infty = O(h^p),$$

then

$$\|\vec{y} - \vec{y}^*\|_\infty = O(h^p).$$

Thus

consistency (LTE) + stability (zero-stability) $\Longrightarrow$ convergence.

□

Definition 4.15 (Zero-Stability of Linear Multistep Method). Consider the linear multistep method

$$\frac{dy}{dt} = f(t,y), \quad \sum_{j=0}^{s} \alpha_j y_{n+1-j} = h\sum_{j=0}^{s} \beta_j f(t_{n+1-j}, y_{n+1-j}).$$

It is called *zero stable* if there exists $K > 0$, not depending on h, such that for arbitrary two grid functions $\vec{u}, \vec{v}$, there holds

$$\|\vec{u} - \vec{v}\|_\infty \le K\Big(\max_{0\le i\le s-1} |u_i - v_i| + \|R_h\vec{u} - R_h\vec{v}\|_\infty\Big)$$

for all $h \le h_0$.

Theorem 4.16. *Assume f satisfies the Lipschitz condition with L_f, and the roots of the characteristic polynomial*

$$\rho(\xi) = \sum_{j=0}^{s} \alpha_j \xi^{s-j} \quad (\text{assume } \alpha_0 = 1)$$

satisfy the root condition

(1) *all roots satisfy $|\xi_j| \le 1$ for $j = 1, \dots, s$,*
(2) *any root with $|\xi_j| = 1$ must be simple (multiplicity 1).*

Then the linear multi-step method is zero stable.

Proof. For two arbitrary grid functions $\vec{u}, \vec{v}$,

$$\sum_{j=0}^{s} \alpha_j u_{n+1-j} = h\sum_{j=0}^{s} \beta_j f(t_{n+1-j}, u_{n+1-j}) + h(R_h\vec{u})_n,$$

$$\sum_{j=0}^{s} \alpha_j v_{n+1-j} = h\sum_{j=0}^{s} \beta_j f(t_{n+1-j}, v_{n+1-j}) + h(R_h\vec{v})_n.$$

Let

$$e_n = u_n - v_n.$$

Then

$$\sum_{j=0}^{s} \alpha_j e_{n+1-j} = \varphi_{n+1},$$

where φ_{n+1} collects the right-hand-side differences. Since $\alpha_0 = 1$,

$$e_{n+1} = -\sum_{j=1}^{s} \alpha_j e_{n+1-j} + \varphi_{n+1}.$$

Define

$$E_{n+1} = \begin{pmatrix} e_{n+1} \\ e_n \\ \vdots \\ \vdots \\ e_{n+2-s} \end{pmatrix}, \quad E_n = \begin{pmatrix} e_n \\ e_{n-1} \\ \vdots \\ \vdots \\ e_{n+1-s} \end{pmatrix} \in \mathbb{R}^s.$$

Then

$$\begin{pmatrix} e_{n+1} \\ e_n \\ \vdots \\ \vdots \\ e_{n+2-s} \end{pmatrix} = \begin{pmatrix} -\alpha_1 & -\alpha_2 & \cdots & -\alpha_s \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1\ 0 \end{pmatrix} \begin{pmatrix} e_n \\ e_{n-1} \\ \vdots \\ \vdots \\ e_{n+1-s} \end{pmatrix} + \begin{pmatrix} \varphi_{n+1} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}.$$

That is,

$$E_{n+1} = AE_n + b_{n+1}.$$

The characteristic polynomial of A,

$$\det(\xi I - A),$$

is $\rho(\xi)$. The root condition implies

$$\|A^n\|_\infty \le M \quad \forall n$$

for some M. Iterating,

$$\begin{aligned} E_{n+1} &= AE_n + b_{n+1} = A(AE_{n-1} + b_n) + b_{n+1} \\ &= A^2E_{n-1} + Ab_n + b_{n+1} \\ &\ \ \vdots \\ &= A^{n-s+2}E_{s-1} + \sum_{m=s}^{n+1} A^{n+1-m}b_m. \end{aligned}$$

Hence

$$E_n = A^{n-(s-1)}E_{s-1} + \sum_{m=s}^{n} A^{n-m}b_m.$$

Therefore

$$\|E_n\|_\infty \le M\|E_{s-1}\|_\infty + M\sum_{m=s}^{n} \|b_m\|_\infty.$$

Since

$$|e_n| \le \|E_n\|_\infty, \quad \|E_{s-1}\|_\infty = \max_{0\le i\le s-1} |e_i|,$$

we get

$$|e_n| \le M\left\{\max_{0\le i\le s-1} |e_i| + \sum_{m=s}^{n} |\varphi_m|\right\}.$$

$$|\varphi_m| \le h\beta L_f \sum_{j=0}^{s} |u_{m-j} - v_{m-j}| + h\|R_h\vec{u} - R_h\vec{v}\|_\infty,$$

where

$$\beta = \max_{0\le j\le s} |\beta_j|.$$

Then

$$\begin{aligned} |e_n| &\le M\left\{\max_{0\le i\le s-1} |e_i| + \sum_{m=s}^{n}\left(h\beta L_f \sum_{j=0}^{s} |u_{m-j} - v_{m-j}| + h\|R_h\vec{u} - R_h\vec{v}\|_\infty\right)\right\} \\ &\le M\left\{\max_{0\le i\le s-1} |e_i| + (s+1)h\beta L_f \sum_{m=0}^{n} |e_m| + Nh\,\|R_h\vec{u} - R_h\vec{v}\|_\infty\right\}. \end{aligned}$$

Hence

$$\begin{aligned} \big(1 - h\beta ML_f(s+1)\big)|e_n| \le M\{&\max_{0\le i\le s-1} |e_i| + (s+1)h\beta L_f \sum_{m=0}^{n-1} |e_m| \\ &+(b-a)\|R_h\vec{u} - R_h\vec{v}\|_\infty\}. \end{aligned}$$

Choose $h \le h_0$ such that

$$1 - h\beta M L_f(s+1) \ge \frac{1}{2}.$$

Then

$$|e_n| \le 2M \left\{ \max_{0\le i\le s-1} |e_i| + (s+1)h\beta L_f \sum_{m=0}^{n-1} |e_m| + (b-a)\|R_h\vec{u} - R_h\vec{v}\|_\infty \right\}.$$

Define

$$A = 2M(s+1)\beta L_f, \quad B = 2M \left\{ \max_{0\le i\le s-1} |e_i| + (b-a)\|R_h\vec{u} - R_h\vec{v}\|_\infty \right\}.$$

Then

$$|e_n| \le Ah \sum_{m=0}^{n-1} |e_m| + B.$$

$$|e_n| \le B(1+hA)^n.$$

Therefore

$$|e_n| \le Be^{nhA} \le Be^{(b-a)A}.$$

So

$$|u_n - v_n| \le 2Me^{(b-a)2M\beta L_f(s+1)} \left\{ \max_{0\le i\le s-1} |u_i - v_i| + (b-a)\|R_h\vec{u} - R_h\vec{v}\|_\infty \right\}.$$

Thus one can choose

$$K = \max\left\{ 2Me^{(b-a)2M\beta L_f(s+1)},\ 2M(b-a)e^{(b-a)2M\beta L_f(s+1)} \right\},$$

so that

$$|u_n - v_n| \le K \left\{ \max_{0\le i\le s-1} |u_i - v_i| + \|R_h\vec{u} - R_h\vec{v}\|_\infty \right\}.$$

□

Theorem 4.17 (Convergence of multi-step method). *If the multi-step method has LTE of $O(h^p)$ and*

$$y_i - y^*(t_i) = O(h^p), \quad i = 0, 1, \dots, s-1,$$

and is zero stable, then it converges and

$$\|\vec{y} - \vec{y}^*\|_\infty = O(h^p).$$

Proof.

$$\|\vec{y} - \vec{y}^*\|_\infty \le K \left(\max_{0\le i\le s-1} |y_i - y^*(t_i)| + \|R_h\vec{y} - R_h\vec{y}^*\|_\infty \right).$$

Since

$$\max_{0\le i\le s-1} |y_i - y^*(t_i)| = O(h^p)$$

and, by the LTE assumption,

$$\|R_h\vec{y} - R_h\vec{y}^*\|_\infty = O(h^p),$$

we obtain

$$\|\vec{y} - \vec{y}^*\|_\infty = O(h^p).$$

□

Definition 4.18 (Absolute Stability for Multistep Methods). Consider the test equation

$$y' = \lambda y, \quad \lambda \in \mathbb{C}, \quad \Re(\lambda) < 0.$$

$$\sum_{j=0}^{s} \alpha_j y_{n+1-j} = h\lambda \sum_{j=0}^{s} \beta_j y_{n+1-j},$$

hence

$$\sum_{j=0}^{s} (\alpha_j - z\beta_j) y_{n+1-j} = 0, \quad z = h\lambda.$$

Define the *stability polynomial*

$$\Pi(\xi; z) = \sum_{j=0}^{s} (\alpha_j - z\beta_j)\, \xi^{s-j}.$$

The region of absolute stability for a multistep method is

$$\{ z \in \mathbb{C} : \ \Pi(\xi; z) \text{ satisfies the root condition} \}.$$

4.6 Fourier Transform

For the complex exponential basis

$$\pi_k(x) = e^{ikx}, \quad k \in \mathbb{Z},$$

$$(\pi_k, \pi_j) := \int_0^{2\pi} e^{ikx} e^{-ijx}\, dx = \begin{cases} 0, & k \ne j, \\ 2\pi, & k = j. \end{cases}$$

Define the inner product

$$(u, v) = \int_0^{2\pi} u\overline{v}\, dx.$$

The best least-squares approximation to f using $\{\pi_k\}$ is

$$v(x) = \sum_{k=-\ell}^{\ell-1} c_k e^{ikx},$$

where

$$c_k = \frac{(f, \pi_k)}{(\pi_k, \pi_k)} = \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-ikx}\, dx.$$

As $\ell \to \infty$, this becomes the full Fourier series

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

When f is only given at discrete points. Let

$$\Delta x = \frac{2\pi}{m}, \quad x_j = \frac{2\pi}{m} j, \quad j = 0, 1, \dots, m-1,$$

and assume $m > 2\ell$. Define the discrete inner product

$$(u, v)_\Delta = \frac{2\pi}{m} \sum_{i=0}^{m-1} u_i \overline{v_i}.$$

Then

$$(\pi_k, \pi_j)_\Delta = \frac{2\pi}{m} \sum_{i=0}^{m-1} e^{ikx_i} e^{-ijx_i} = \begin{cases} 0, & k \ne j, \\ 2\pi, & k = j. \end{cases}$$

Hence, the discrete Fourier coefficients are

$$\widetilde{c}_k = \frac{(f, \pi_k)_\Delta}{(\pi_k, \pi_k)_\Delta} = \frac{1}{2\pi} \cdot \frac{2\pi}{m} \sum_{j=0}^{m-1} f(x_j) e^{-ikx_j} = \frac{1}{m} \sum_{j=0}^{m-1} f(x_j) e^{-i\frac{2\pi}{m}kj}.$$

If $m = 2\ell$, then

$$\widetilde{v}(x_j) = \sum_{k=-\ell}^{\ell-1} \widetilde{c}_k e^{ikx_j} = \sum_{k=-\ell}^{\ell-1} \widetilde{c}_k e^{i\frac{2\pi}{m}kj}.$$

Define

$$\omega = e^{-2\pi i/m}, \quad \omega^m = 1.$$

Then

$$\widetilde{v}(x_i) = \sum_{k=-\ell}^{\ell-1} \widetilde{c}_k (\overline{\omega})^{ki},$$

and

$$\widetilde{c}_k = \frac{1}{m} \sum_{j=0}^{m-1} f(x_j) e^{-i\frac{2\pi}{m}kj} = \frac{1}{m} \sum_{j=0}^{m-1} f(x_j) \omega^{kj}.$$

Therefore,

$$\widetilde{v}(x_i) = \sum_{k=-\ell}^{\ell-1} \widetilde{c}_k (\overline{\omega})^{ki} = \frac{1}{m} \sum_{j=0}^{m-1} f(x_j) \left(\sum_{k=-\ell}^{\ell-1} \omega^{k(j-i)} \right).$$

So for $m = 2\ell$, we have the discrete Fourier transform (DFT)

$$\widetilde{c}_k = \frac{1}{m} \sum_{j=0}^{m-1} f(x_j) e^{-i\frac{2\pi}{m}kj}, \quad k = -\ell, \ldots, \ell - 1,$$

and the inverse relation

$$f(x_j) = \sum_{k=-\ell}^{\ell-1} \widetilde{c}_k e^{i\frac{2\pi}{m}kj}, \quad j = 0, \ldots, m-1.$$

Define

$$\widehat{f}_k = \sum_{j=0}^{m-1} f(x_j) e^{-i\frac{2\pi}{m}kj}, \quad k = 0, \ldots, m-1,$$

with inverse

$$f(x_j) = \frac{1}{m} \sum_{k=0}^{m-1} \widehat{f}_k e^{i\frac{2\pi}{m}kj}, \quad j = 0, \ldots, m-1.$$

FFT. Let

$$\omega = e^{-2\pi i/m} =: \omega_m,$$

and assume m is a power of 2. Then

$$\widehat{f}_k = \sum_{j=0}^{m-1} f(x_j) \omega_m^{kj} = \sum_{j=0}^{m/2-1} \left(f_{2j} \omega_m^{(2j)k} + f_{2j+1} \omega_m^{(2j+1)k} \right).$$

Since

$$\omega_m^{2k} = \omega_{m/2}^k,$$

this becomes

$$\widehat{f}_k = \sum_{j=0}^{m/2-1} f_{2j} \omega_{m/2}^{jk} + \omega_m^k \sum_{j=0}^{m/2-1} f_{2j+1} \omega_{m/2}^{jk}, \quad k = 0, \ldots, \frac{m}{2} - 1.$$

Define

$$\widehat{f}_k^{\text{even}} = \sum_{j=0}^{m/2-1} f_{2j} \omega_{m/2}^{jk}, \quad \widehat{f}_k^{\text{odd}} = \sum_{j=0}^{m/2-1} f_{2j+1} \omega_{m/2}^{jk}.$$

Then

$$\widehat{f}_k = \widehat{f}_k^{\text{even}} + \omega_m^k \widehat{f}_k^{\text{odd}}, \quad k = 0, \ldots, \frac{m}{2} - 1.$$

Also,

$$\widehat{f}_{k+m/2} = \sum_{j=0}^{m/2-1} f_{2j} \omega_m^{j(k+m/2)} + \omega_m^{k+m/2} \sum_{j=0}^{m/2-1} f_{2j+1} \omega_m^{jk}.$$

Using

$$\omega_m^{m/2} = e^{-\pi i} = -1,$$

we get

$$\widehat{f}_{k+m/2} = \sum_{j=0}^{m/2-1} f_{2j} \omega_{m/2}^{jk} - \omega_m^k \sum_{j=0}^{m/2-1} f_{2j+1} \omega_{m/2}^{jk},$$

that is,

$$\widehat{f}_{k+m/2} = \widehat{f}_k^{\text{even}} - \omega_m^k \widehat{f}_k^{\text{odd}}, \quad k = 0, \ldots, \frac{m}{2} - 1.$$

If $P(m)$ denotes the cost, then

$$P(m) = 2P(m/2) + O(m).$$

$$P(m) = 4P(m/4) + O(2m) = 8P(m/8) + O(3m) = \cdots$$

If $m = 2^n$, then $n = \log_2 m$, so

$$P(m) = 2^n P(1) + O(nm) = O(m) + O(m \log m) = O(m \log m).$$

Acknowledgments

To my parents and teachers, whose guidance and support have shaped who I am today. And to my beloved Sunny Sun, your companionship and encouragement enable me to go further on my journey.

References

[1] Walter Gautschi. 2011. *Numerical analysis.* Springer Science & Business Media.