Math 128A Notes

Henry Yan

September 2023

Introduction

Suppose $\{\beta_n\}$ converges to 0, and $\{\alpha_i\}$ converges to α. If there is a positive constant K such that

$$|\alpha_n - \alpha| \leq k|\beta_n| \text{ for large } n,$$

then we say that $\{\alpha_n\}$ converges to α with a **rate of convergence** $O(|\beta_n|)$:

$$\alpha_n = \alpha + O(|\beta_n|).$$

Suppose that $\lim_{h\to 0} G(h) = 0$ and $\lim_{h\to 0} F(h) = L$. If there exists a positive number K such that

$$|F(h) - L| \leq K|G(h)|$$

for all sufficiently small h, then $F(h) = L + O(G(h))$.

Theorem 2.1: Suppose that $f \in C[a,b]$ and $f(a)f(b) < 0$. The Bisection method generates a sequence $\{p_n\}_{n=1}^{\infty}$ approximating a zero p of f with

$$|p_n - p| \leq \frac{b-a}{2^n}, n \geq 1.$$

Definition 2.2: The number p is a *fixed point* for a given function g if $g(p) = p$.

Given a root-finding problem $f(p) = 0$, we can define functions $g(x)$ with a fixed point at p in multiple ways: $g(x) = x - f(x), g(x) = x + 3f(x)$, etc.

Conversely, given function g with fixed point at p, then the function $f(x) = x - g(x)$ has a root at p.

Definition 2.2 - Fixed point iteration: Given initial approximation p_0, define *Fixed Point Iteration*

$$p_n = g(p_{n-1}), n = 1, 2, \ldots,$$

if iteration converges to p, then

$$p = \lim_{n\to\infty} p_n = \lim_{n\to\infty} g(p_{n-1}) = g(p).$$

Theorem 2.3:

1. If $g \in C[a,b]$ and $g(x) \in [a,b]$ for all $x \in [a,b]$, then g has at least one fixed point in $[a,b]$.
2. If, in addition, $g'(x)$ exists on (a,b) and a positive constant $k < 1$ exists with $|g'(x)| \leq k$, for all $x \in (a,b)$, then there is exactly one fixed point in $[a,b]$.

Theorem 2.4: Let $g \in C[a,b]$ be such that $g(x) \in [a,b]$, for all $x \in [a,b]$. Suppose, in addition, that g' exists on (a,b) and that a constant $0 < k < 1$ exists with

$$|g'(x)| \leq k, \forall x \in [a,b].$$

Then, for any number p_0 in $[a,b]$, the sequence defined by

$$p_n = g(p_{n-1}), n \geq 1,$$

converges to the unique fixed point p in $[a,b]$.

Newton's Method: $p_{k+1} = p_k - \frac{f(p_k)}{f'(p_k)}, k = 0, 1, \ldots$. Start with p_0 being a reasonable approximation for p, where p is a root of f.

When doing Newton's method, if $|p - p_0|$ is small, we can expect a fast convergence. Otherwise, Newton's method might not even converge to p.

Then *secant method* is just linear approximation (poor mans Newton's method).

Theorem 2.6: Let $f \in C^2[a,b]$. If $p \in (a,b)$ such that $f(p) = 0$ and $f'(p) \neq 0$, then there exists a $\delta > 0$ such that Newton's method generates a sequence $\{p_n\}_{n=1}^{\infty}$ converging to p for any initial approximation $p_0 \in [p - \delta, p + \delta]$.

Definition 2.7 - Asymptotic Error Constant: Suppose $\{p_n\}_{n=0}^{\infty}$ is a sequence that converges to p with $p_n \neq p$ for all n. If positive constants λ and α exist with

$$\lim_{n \to \infty} \frac{|p_{n+1} - p|}{|p_n - p|^{\alpha}} = \lambda,$$

then $\{p_n\}_{n=0}^{\infty}$ converges to p with *order* α, with *asymptotic error constant* λ.

Definition - Types of Convergence:

1. If $\alpha = 2$, then $\{p_n\}$ converges *quadratically.*
2. If $\alpha = 1$ and $\lambda < 1$, then $\{p_n\}$ converges *linearly.*
3. If $\alpha = 1$ and $\lambda = 0$, then $\{p_n\}$ converges *super-linearly.*
4. If $\alpha = 1$ and $\lambda \geq 1$, then $\{p_n\}$ converges *sub-linearly.*

Theorem 2.8 - FPI Converges Linearly: Let $g \in C[a,b]$ such that $g(x) \in [a,b]$ and there exists a positive constant $k < 1$ exists with

$$|g'(x)| \leq k, \forall x \in (a,b).$$

If $g'(p) \neq 0$, then for any number $p_0 \neq p$ in $[a,b]$, the sequence

$$p_n = g(p_{n-1}), n \geq 1,$$

converges only linearly to the unique fixed point p in $[a,b]$.

For a fixed point method to converge quadratically, we need to have both $g(p) = p$, and $g'(p) = 0$.

The bisection method is also considered converge linearly.

Definition - Simple Root: p is a *simple root* of f if $f(p) = 0$ and $f'(p) \neq 0$.

Theorem: The function $f \in C[a,b]$ has a simple root at p in (a,b) iff $f(x) = (x-p)q(x)$, where $\lim_{x\to p} q(x) \neq 0$.

Definition - (Root of Multiplicity): A solution p of $f(x) = 0$ is a *root of multiplicity* m (for integer m) of f if for $x \neq p$, we can write

$$f(x) = (x-p)^m q(x), \text{ where } \lim_{x\to p} q(x) \neq 0.$$

Theorem: The function $f \in C^m[a,b]$ has a root of multiplicity m at p in (a,b) iff

$$0 = f(p) = f'(p) = f''(p) = \cdots = f^{(m-1)}(p), \text{ but } f^{(m)}(p) \neq 0.$$

Definition - Modified Newton's Method:

$$p_{n+1} = p_n - \frac{f(p_n)f'(p_n)}{f'(p_n)^2 - f(p_n)f''(p_n)}.$$

Definition - Aitken's Acceleration for a given $\{p_n\}$:

$$\widehat{p}_n = \{\Delta^2\}(p_k) = p_k - \frac{(p_{k+1}-p_k)^2}{p_{k+2} - 2p_{k+1} + p_k}.$$

Theorem 2.14: Suppose that $\{p_n\}$ is sequence that converges linearly to limit p and that

$$\lim_{n\to\infty} \frac{p_{n+1}-p}{p_n - p} < 1.$$

Then Aitken's acceleration sequence $\{\widehat{p}_n\}$ converges to p faster than $\{p_n\}$ in the sense that

$$\lim_{n\to\infty} \frac{\widehat{p}_n - p}{p_n - p} = 0.$$

Definition - Steffenson's Method: Given initial approximation p_0, convergent fixed point iteration map $g(x)$ with $g'(p) \neq 1$, *Steffenson's Method* is

$$p_n = G(p_{n-1}), n = 1, 2, \ldots, G(x) = x - \frac{(g(x)-x)^2}{g(g(x)) - 2g(x) + x}.$$

Theorem: Steffenson's Method converges at least quadratically.

Theorem: Horner's Method further computes

$$P(x) = (x - x_0)Q(x) + b_0, \text{ with } Q(x) = b_n x^{n-1} + b_{n-1}x^{n-2} + \cdots + b_2 x + b_1.$$

Muller's Method: Choose parabola $P(x) = a(x-p_2)^2 + b(x-p_2) + c$, where a, b, c satisfy

$$\begin{aligned} f(p_0) &= a(p_0 - p_2) + b(p_0 - p_2) + c, \\ f(p_1) &= a(p_1 - p_2) + b(p_1 - p_2) + c, \\ f(p_2) &= c. \end{aligned}$$

Theorem: Assume that Muller's Method converges to a root p and that $f'''(x)$ is continuous with $f'(p) \neq 0$.

Then Muller's Method converges with order $\mu \approx 1.84$, where $\mu^3 = \mu^2 + \mu + 1$.

Interpolation is basically connecting the dots of the data we have. **Extrapolation** is extending the data we have beyond the data points we have.

Polynomial Interpolation: Given $n+1$ distinct points $(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))$, we can find a $\leq n$ degree polynomial

$$P(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0,$$

where $P(x_i) = f(x_i), i = 0, 1, \ldots, n$. Then

$$f(x_j) = P(x_j) = f(x_0)L_0(x_j) + f(x_1)L_1(x_j) + \cdots + f(x_n)L_n(x_j),$$

where $L_i(x) = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$. Alternatively, $L_i(x_j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases}$

Lemma: Let $f \in C^n[a,b]$ be such that $f(x_1) = 0, f(x_2) = 0, \ldots, f(x_n) = 0$, where $a \leq x_1 < x_2 < \cdots < x_n \leq b$ are mutually distinct. Then there exists a $\xi \in [a,b]$ such that

$$f^{(n-1)}(\xi) = 0.$$

Theorem: Suppose $x_0, x_1, \ldots, x_n$ are distinct numbers in the interval $[a,b]$ and $f \in C^{n+1}[a,b]$. Then, for each $x \in [a,b]$, a number $\xi(x)$ between $x_0, x_1, \ldots, x_n$ (hence $\in (a,b)$) exists with

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)(x - x_1)\ldots(x - x_n),$$

where $P(x)$ is the interpolating polynomial.

Corollary: Suppose $x_0, x_1, \ldots, x_n$ are distinct numbers in the interval $[a,b]$ and f is polynomial of degree at most n, then $P(x) = f(x)$.

Theorem: Assume that the nodal points $x_0, x_1, \ldots, x_n$ are mutually distinct, then the interpolating polynomial $P(x)$ of degree $\leq n$ is unique.

Neville's Method: Let $Q(x)$ interpolate $f(x)$ at $x_0, x_1, \ldots, x_k$. Let $\widehat{Q}(x)$ interpolate $f(x)$ at $x_1, \ldots, x_k, x_{k+1}$. Then

$$P(x) = \frac{(x - x_{k+1})Q(x) - (x - x_0)\widehat{Q}(x)}{x_0 - x_{k+1}}$$

interpolates $f(x)$ at $x_0, x_1, \ldots, x_k, x_{k+1}$.

Divided Differences: Given $n+1$ distinct points $(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))$,

$$P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)\ldots(x - x_{n-1}),$$

with $P(x_i) = f(x_i), i = 0, 1, \ldots, n$. It follows that $a_0 = P(x_0) = f(x_0)$ and

$$\frac{P(x) - f(x_0)}{x - x_0} = a_1 + a_2(x - x_1) + \cdots + a_n(x - x_1)\ldots(x - x_{n-1}).$$

Theorem - Newton Divided Difference: Suppose that $f \in C^n[a,b]$ and $x_0, x_1, \ldots, x_n$ are distinct nodes in $[a,b]$. Then a number $\xi \in (a,b)$ exists such that

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

Recursive Newton Divided Differences, for all i, j,

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

Then in

$$P(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \dots + a_n(x - x_0)(x - x_1)\dots(x - x_{n-1}),$$

we have

$$\begin{aligned} a_0 &= f[x_0] \\ a_1 &= f[x_0, x_1] \\ &\vdots \\ a_n &= f[x_0, x_1, \dots, x_n] \end{aligned}$$

Then $f = a_i + (x - x_i) \cdot f, i = 0, 1, \dots, n-1$, and output is $f = P(x)$.

Definition - Forward Differences (FD): $x_1 = x_0 + h, x_2 = x_0 + 2h$. Then

$$f[x_0, x_1] = f[x_0, x_0 + h] = \frac{f(x_0 + h) - f(x_0)}{h} \qquad \text{(first order FD)},$$

$$f[x_0, x_1, x_2] = f[x_0, x_0 + h, x_0 + 2h] = \frac{f[x_1, x_2] - f[x_0, x_1]}{2h} = \frac{f(x_2) + f(x_0) - 2f(x_1)}{2h^2} \qquad \text{(second order FD)}.$$

Definition - Backward Differences (BD): $x_{n-1} = x_n - h, x_{n-2} = x_n - 2h$. Then

$$f[x_{n-1}, x_n] = \frac{f(x_n) - f(x_n - h)}{h} \qquad \text{(first order BD)},$$

$$f[x_{n-2}, x_{n-1}, x_n] = \frac{f(x_n) + f(x_{n-2}) - 2f(x_{n-1})}{2h^2} \qquad \text{(second order BD)}.$$

Double Nodes - Linear Approximation: Given two distinct points $(x_0, f(x_0)), (x_1, f(x_1))$, the interpolating polynomial of degree ≤ 1 is given by $P(x) = a_0 + a_1(x - x_0)$, where

$$a_0 = f(x_0), a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

Definition - Lagrange Interpolation: The interpolating polynomial

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

Theorem 3.3 - Lagrange Interpolation Error: Given $n + 1$ distinct points $(x_0, f(x_0)), \dots, (x_n, f(x_n))$, there exists $\xi \in [x_0, x_n]$ such that

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)(x - x_1)\dots(x - x_n).$$

Definition - Hermite Interpolation: Given $n+1$ distinct points, $(x_0, f(x_0), f'(x_0)), \ldots, (x_n, f(x_n), f'(x_n))$. Interpolating polynomial of degree $\leq 2n+1$ with

$$H(x_0) = f(x_0), H'(x_0) = f'(x_0)$$
$$\vdots$$
$$H(x_n) = f(x_n), H'(x_n) = f'(x_n)$$

Since there are $2(n+1) = 2n+2$ conditions above, we get a $2n+1$ degree interpolating polynomial $H(x)$.

An alternative form for Hermite interpolation is

$$H(x) = \sum_{j=0}^{n} f(x_j) H_j(x) + \sum_{j=0}^{n} f'(x_j) \widehat{H}_j(x),$$

where $H_j(x) = (1 - 2(x - x_j)L_j'(x_j))L_j^2(x)$ and $\widehat{H}_j(x) = (x - x_j)L_j^2(x)$, with

$$L_j(x) = \prod_{i \neq j} \frac{x - x_i}{x_j - x_i}.$$

Theorem - Hermite Interpolation Error: Suppose $x_0, x_1, \ldots, x_n$ are distinct numbers in the interval $[a, b]$ and $f \in C^{2n+2}[a, b]$. Then, for each $x \in [a, b]$, a number $\xi(x)$ between $x_0, x_1, \ldots, x_n$ (and hence $\in [a, b]$) exists with

$$f(x) = H(x) + \frac{f^{(2n+2)}(\xi(x))}{(2n+2)!}(x - x_0)^2(x - x_1)^2 \ldots (x - x_n)^2,$$

where $H(x)$ is the Hermite interpolating polynomial.

Definition - Splines Equations: For *cubic Spline interpolate* $S(x) \in C^3[x_0, x_n]$, we have $S_j(x_j) = f(x_j) = a_j, 0 \leq j \leq n$. Define

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3, \text{ for } x \in [x_j, x_{j+1}], 0 \leq j \leq n-1.$$

Then, for $j = 1, \ldots, n-1$, define $h_j = x_{j+1} - x_j$,

$$(3.21) h_{j-1}c_{j-1} + 2(h_{j-1} + h_j)c_j + h_j c_{j+1} = 3\left(\frac{a_{j+1} - a_j}{h_j} - \frac{a_j - a_{j-1}}{h_{j-1}}\right),$$

which gives $n-1$ equations and has $n+1$ unknowns. Thus we need at least 2 more conditions/equations.

Solving Spline's Equations:

3.15. $a_{j+1} = a_j + b_j h_j + c_j h_j^2 + d_j h_j^3$;

3.16. $b_{j+1} = b_j + 2c_j h_j + 3d_j h_j^2$;

3.17. $c_{j+1} = c_j + 3d_j h_j$;

3.18. $a_{j+1} = a_j + b_j h_j + \dfrac{h_j^2}{3}(2c_j + c_{j+1})$;

3.19. $b_{j+1} = b_j + h_j(c_j + c_{j+1})$;

3.20. $b_j = \frac{1}{h_j}(a_{j+1} - a_j) - \frac{h_j}{3}(2c_j + c_{j+1})$.

Splines equations essentially just create piece-wise cubic functions to estimate graphs that aren't as nicely estimated by polynomials.

Definition - Natural Splines: In the case that $S_0''(x_0) = S_{n-1}''(x_n) = 0$,

$$c_0 = S_0''(x_0)/2 = 0, c_n = S_{n-1}''(x_n) = 0,$$

so we have $n-1$ unknowns.

Definitions - Clamped Splines: $S_0'(x_0) = f'(x_0) = b_0, S_{n-1}'(x_n) = f'(x_n) = b_n$.

Cubic Bezier Polynomial: Given nodes (endpoints) $(x_0, y_0), (x_1, y_1)$, the guidepoint for (x_0, y_0) is $(x_0 + \alpha_0, y_0 + \beta_0)$ and the guidepoint for (x_1, y_1) is $(x_1 - \alpha_1, y_1 - \beta_1)$. The unique cubic Bezier polynomial $x(t)$ on $[0, 1]$ satisfying $x(0) = x_0, x(1) = x_1, x'(0) = \alpha_0, x'(1) = \alpha_1$ is

$$x(t) = [2(x_0 - x_1) + 3(\alpha_0 + \alpha_1)]t^3 + [3(x_1 - x_0) - 3(\alpha_1 + 2\alpha_0)]t^2 + 3\alpha_0 t + x_0.$$

Numerical Differentiation (Week 7 Lecture)

Direct Approximation: For small h,

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} - \frac{1}{2}hf''(\xi).$$

Alternatively, we could have used a three-point midpoint formula, which would give an approximation of the form

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

Then assuming rounding errors $e(f(x_0 + h)), e(f(x_0 - h)) < \epsilon$ and $|f^{(3)}(\xi)| \leq M$, we have an upper bound on the error of our approximation $e(h) = \dfrac{\epsilon}{h} + \dfrac{Mh^2}{6}$.

This $e(h) = \dfrac{\epsilon}{h} + \dfrac{Mh^2}{6}$ is minimized at $h_{min} = \left(\frac{3\epsilon}{M}\right)^{1/3} = O(\epsilon^{1/3})$, with $e(h_{min}) = \frac{1}{2}(9M\epsilon^2)^{1/3} = O(\epsilon^{2/3})$.

Numerical Differentiation using Polynomial Interpolation:

$$f'(x_k) = \sum_{j=0}^{n} f(x_j)L_j'(x_k) + \left(\frac{f^{(n+1)}(\xi(x_k))}{(n+1)!}\prod_{i\neq k}(x_k - x_i)\right),$$

where $L_j'(x_k) = \begin{cases} \dfrac{1}{x_j - x_k}\displaystyle\prod_{i\neq j,k}\dfrac{x_k - x_i}{x_j - x_i}, & \text{for } k \neq j, \\ \displaystyle\sum_{i\neq j}\dfrac{1}{x_j - x_i}, & \text{for } k = j. \end{cases}$

Similarly,

$$f''(x_0) = \frac{f(x_0 + h) + f(x_0 - h) - 2f(x_0)}{h^2} - \frac{h^2}{12}f^{(4)}(\xi).$$

Linear Interpolation: $f(x) = P_1(x) + \frac{1}{2}f''(\xi(x))(x - x_0)(x - x_1), P_1(x)$.

Numerical Integration: Choose $n + 1$ points $a \leq x_0 < x_1 < \cdots < x_n \leq b$.

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}\prod_{j=0}^{n}(x - x_j), P(x) = \sum_{j=0}^{n} f(x_j)L_j(x).$$

Then the approximate integration

$$\int_a^b f(x)dx = \sum_{j=0}^{n} a_j f(x_j), \text{ with } a_j = \int_a^b L_j(x)dx.$$

Definition - Degree of Precision (DoP): Integer n such that quadrature formula is exact for $f(x) = x^k$, for each $k = 0, 1, \ldots, n$ but inexact for $f(x) = x^{n+1}$.

Theorem: Quadrature formula is exact for all polynomials of degree at most n.

Simpson's Rule: For $n = 3, x_0 = a, x_1 = \frac{a+b}{2}, x_2 = b, h = \frac{b-a}{2}$,

$$\int_a^b f(x)dx \approx \frac{h}{3}(f(x_0) + 4f(x_1) + f(x_2)) - \frac{f^{(4)}(\xi)}{90}h^5.$$

Composite Simpson's Rule: $n = 2m, x_j = a + jh, h = \frac{b-a}{n}, 0 \le j \le n$,

$$\begin{aligned}
\int_a^b f(x)dx &\approx \sum_{j=1}^{m} \left(\frac{h}{3}(f(x_{2(j-1)}) + 4f(x_{2j-1}) + f(x_{2j})) - \frac{f^{(4)}(\xi)}{90}h^5 \right) \\
&= \frac{h}{3}(f(a) + 2\sum_{j=1}^{m-1} f(x_{2j}) + 4\sum_{j=1}^{m} f(x_{2j-1}) + f(b)) - \frac{h^5}{90}\sum_{j=1}^{m} f^{(4)}(\xi_j) \\
&= \frac{h}{3}(f(a) + 2\sum_{j=1}^{m-1} f(x_{2j}) + 4\sum_{j=1}^{m} f(x_{2j-1}) + f(b)) - \frac{(b-a)h^4}{180}f^{(4)}(\xi).
\end{aligned}$$

Composite Trapezoidal Rule: $x_j = a + jh, h = \frac{b-a}{n}, 0 \le j \le n$.

$$\int_a^b f(x)dx = \frac{h}{2}\left(f(a) + 2\sum_{j=1}^{n-1} f(x_j) + f(b) \right) - \frac{(b-a)h^2}{12}f''(\xi).$$

Recursive Composite Trapezoidal: $n = 2^k, h_j = \frac{b-a}{2^j}$.

$$\begin{aligned}
\int_a^b f(x)dx &\approx \frac{h_k}{2}\left(f(a) + 2\sum_{j=1}^{n-1} f(x_j) + f(b) \right) - \frac{(b-a)h_k^2}{12}f''(\mu) \\
&= \frac{h_k}{2}\left(f(a) + 2\sum_{j=1}^{n-1} f(x_j) + f(b) \right) + \sum_{j=1}^{\infty} K_j h_k^{2j} \\
&= R_{k,1} + \sum_{j=1}^{\infty} K_j h_k^{2j},
\end{aligned}$$

where $R_{k,1} = \frac{h_k}{2}\left(f(a) + 2\sum_{j=1}^{n-1} f(x_j) + f(b)\right)$. Additionally,

$$
\begin{aligned}
R_{1,1} &= \frac{h_0}{2}(f(a)+f(b)) = \frac{b-a}{2}(f(a)+f(b)) = \mathcal{N}_1(h_0)\\
R_{2,1} &= \frac{1}{2}(R_{1,1} + h_0 f(a+h_1)) = \mathcal{N}_1\left(\frac{h_0}{2}\right)\\
&\vdots\\
R_{k,1} &= \frac{1}{2}\left(R_{k-1,1} + h_{k-2}\sum_{j=1}^{2^{k-2}} f(a+(2j-1)h_{k-1})\right) = \mathcal{N}_1\left(\frac{h_0}{2^{k-1}}\right)
\end{aligned}
$$

The $R_{k,1}$ are the **Romberg Extrapolations**.

Thus composite Simpson's Rule can be rewritten as

$$\int_a^b f(x)dx \approx R_{k,1} + \sum_{j=2}^{\infty} K_j h^{2j}, \text{ for } n = 2^k.$$

Crying baby Principle (in adaptive algorithms): Add more points in regions of inadequate accuracy.

Presumed innocence: Accuracy adequate unless detected otherwise.

$S(a,b)$ is Simpson's Rule approximation on $[a,b]$.

Adaptive Quadrature:

$$\int_a^b f(x)dx = S(a, \frac{a+b}{2}) + S(\frac{a+b}{2}, b) - \frac{1}{16}\left(\frac{h^5}{90}\right) f^{(4)}(\hat{\xi}),$$

where $\hat{\xi} \in (a,b)$. For a given tolerance τ, if approximate error $\frac{1}{16}\left(\frac{h^5}{90}\right) f^{(4)}(\hat{\xi}) \approx \frac{1}{15}|S(a,b) - (S(a,\frac{a+b}{2}) + S(\frac{a+b}{2}, b))| < \tau$, then $\int_a^b f(x)dx$ is sufficiently accurate.

Gaussian Quadrature: Given $n > 0$, choose both distinct nodes $x_1, \dots, x_n \in [-1,1]$ and weights $c_1, \dots, c_n$, so quadrature

$$\int_{-1}^1 f(x)dx \approx \sum_{j=1}^n c_j f(x_j)$$

has highest possible degree of precision (DoP). Since we have $2n$ parameters, we can choose quadrature that is exact for $2n$ monomials $f(x) = 1, x, x^2, \dots, x^{2n-1}$. Choosing

$$c_i = \int_{-1}^1 L_i(x)dx$$

gives the highest DoP $= 2n-1$.

Note that we can change the bounds of any integral from

$$\int_a^b f(x)dx = \frac{b-a}{2}\int_{-1}^1 f\left(\frac{(b-a)t + (a+b)}{2}\right) dt.$$

Use the table

Table 4.12

n	Roots $r_{n,i}$	Coefficients $c_{n,i}$
2	0.5773502692	1.0000000000
	−0.5773502692	1.0000000000
3	0.7745966692	0.5555555556
	0.0000000000	0.8888888889
	−0.7745966692	0.5555555556
4	0.8611363116	0.3478548451
	0.3399810436	0.6521451549
	−0.3399810436	0.6521451549
	−0.8611363116	0.3478548451
5	0.9061798459	0.2369268850
	0.5384693101	0.4786286705
	0.0000000000	0.5688888889
	−0.5384693101	0.4786286705
	−0.9061798459	0.2369268850

Figure 1: Gaussian Quadrature Table of Values

Gaussian Quadrature by Legendre Polynomials: $P_0(x) = 1, P_1(x) = x,$

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x), \text{ for } n \geq 1.$$

Then $P_n(x)$ always has n distinct roots $x_1, \dots, x_n \in (-1, 1)$ and has DoP $2n-1$. This can be seen by long dividing a polynomial $P(x)$ of degree $2n-1$ by $P_n(x)$ to get $P(x) = P_n(x)Q(x) + R(x)$.

The Gaussian quadrature by Lagrange polynomials are mutually orthogonal, meaning

$$\int_{-1}^{1} P_n(x)P_j(x)dx = 0, \text{ for } j < n.$$

Theorem:

$$\int_{-1}^{1} P_n(x)Q(x)dx = 0,$$

for any polynomial $Q(x)$ with degree $< n$.

Gaussian Quadrature by Hermite Interpolation: Given $n+1$ points $(x_i, f(x_i), f'(x_i))$, the Gaussian quadrature derived from the Hermite polynomial $H(x)$ is

$$\begin{aligned}\int_{-1}^{1} f(x)dx &= \int_{-1}^{1} H(x)dx + \int_{-1}^{1} R(x)dx \\ &= c_1 f(x_1) + c_2 f(x_2) + \cdots + c_n f(x_n) + \frac{f^{(2n)}(\xi)}{(2n)!}\int_{-1}^{1}(x-x_0)^2(x-x_1)^2\dots(x-x_n)^2,\end{aligned}$$

where we denote the last term (error) above by R and have

$$\begin{aligned}R &= \frac{f^{(2n)}(\xi)}{(2n)!}\int_{-1}^{1}(x-x_0)^2(x-x_1)^2\dots(x-x_n)^2 \\ &= \frac{2^{2n}(n!)^3(n-1)!}{(2n+1)!(2n)!(2n-1)!}f^{(2n)}(\xi) = O\left(\frac{4^{-n}|f^{(2n)}(\xi)|}{(2n)!}\right)\end{aligned}$$

Double Integrals: When computing double integrals, treat it as the integral of an integral function, i.e.

$$\int_a^b \int_c^d f(x,y)dydx = \int_a^b g(x)dx, \text{ where } g(x) = \int_c^d f(x,y)dy.$$

We approximate

$$\int_a^b g(x)dx = c_1 g(x_1) + \cdots + c_n g(x_n) + R(g).$$

We further approximate

$$g(x_i) = \int_c^d f(x_i,y)dy = \widehat{c_1} f(x_i,y_1) + \cdots + \widehat{c_m} f(x_i,y_m) + \widehat{R}(f(x_i,\cdot)).$$

Then

$$\begin{aligned}
\int\int_R f(x,y)dA &= \left(\sum_{i=1}^n c_i g(x_i)\right) + R(g) \\
&= \left(\sum_{i=1}^n c_i \left(\left(\sum_{j=1}^m \widehat{c_j} f(x_i,y_j)\right) + \widehat{R}(f(x_i,\cdot))\right)\right) + R(g) \\
&= \sum_{i=1}^n \sum_{j=1}^m c_i \widehat{c_j} f(x_i,y_j) + \left(\sum_{i=1}^n c_i \widehat{R}(f(x_i,\cdot)) + R(g)\right),
\end{aligned}$$

where the right bracketed term is the error term.

Composite Simpson's Rule:

$$\int\int_R f(x,y)dA = \sum_{i=1}^n \sum_{j=1}^m c_i \widehat{c_j} f(x_i,y_j) - \frac{(b-a)(d-c)}{180}\left(k^4 \frac{\partial^4 f}{\partial^4 y}(\widehat{\xi},\widehat{\eta}) + h^4 \frac{\partial^4 f}{\partial^4 x}(\xi,\eta)\right),$$

where k is the step size for y and h is the step size for x.

A **2-Dimensional Gaussian quadrature** gives

$$\begin{aligned}
\int_a^b \int_c^d f(x,y)dydx &= \frac{(b-a)(d-c)}{4} \int_{-1}^1 \widehat{g}(u)du, \text{ where } \widehat{g}(u) = \int_{-1}^1 f\left(\frac{a+b}{2} + \frac{b-a}{2}u, \frac{c+d}{2} + \frac{d-c}{2}v\right)dv \\
&= \frac{(b-a)(d-c)}{4} \sum_{i=1}^n \sum_{j=1}^m c_i \widehat{c_j} f(x_i,y_j).
\end{aligned}$$

Definition - Initial value ODE: Initial value conditions $\theta(t_0) = \theta_0, \theta'(t_0) = \theta_0'$.

Definition - Lipschitz Condition: A function $f(t,y)$ satisfies a **Lipschitz condition** in the variable y on a set $D \subset \mathbb{R}^2$ if a constant $L > 0$ exists with

$$|f(t,y_1) - f(t,y_2)| \le L|y_1 - y_2|,$$

whenever $(t,y_1),(t,y_2)$ are in D. L is the **Lipschitz constant**.

Definition - Convex Set: A set $D \subset \mathbb{R}^2$ is *convex* if whenever $(t_1,y_1),(t_2,y_2)$ are in D, the line segment joining the two points is entirely contained in D.

Theorem: Suppose $f(t,y)$ is defined on a convex set $D \subset \mathbb{R}^2$. If a constant $L > 0$ exists with

$$\left|\frac{\partial f}{\partial y}(t,y)\right| \le L, \text{ for all } (t,y) \in D,$$

then f satisfies a Lipschitz condition with Lipschitz constant L.

Definition - Well-posed: An ODE is *well-posed* if a unique ODE solution exists, and small changes (perturbations) to the ODE cause small changes to the solution.

The initial value problem

$$\frac{dy}{dt} = f(t, y), a \le t \le b, y(a) = \alpha$$

is said to be *well-posed* if

1. A unique solution, $y(t)$, to the problem exists.
2. There exist constants $\epsilon_0 > 0$ and $k > 0$ such that for any ϵ, with $\epsilon_0 > \epsilon > 0$, whenever $\delta(t)$ is continuous with $|\delta(t)| < \epsilon$ for all t in $[a, b]$, and when $|\delta_0| < \epsilon$, the initial-value problem

$$\frac{dz}{dt} = f(t, z) + \delta(t), a \le t \le b, z(a) = \alpha + \delta_0,$$

has unique solution $z(t)$ that satisfies

$$|z(t) - y(t)| < k\epsilon, \text{ for all } t \in [a, b].$$

Theorem: Suppose

$$D = \{(t, y) | a \le t \le b \text{ and } -\infty < y < \infty\}.$$

If f is continuous and satisfies a Lipschitz condition in the variable y on the set D, then the initial-value problem

$$\frac{dy}{dt} = f(t, y), a \le t \le b, y(a) = \alpha$$

is well-posed.

Definition - Euler's Method: Given initial value problem $\frac{dy}{dt} = f(t, y), a \le t \le b, y(a) = \alpha$. Select *Mesh points* $t_j = a + jh$, for $j = 0, 1, 2, \ldots, N$, where $h = (b - a)/N$.

For each j, compute the 2-term Taylor expansion

$$y(t_{j+1}) = y(t_j) + hy'(t_j) + \frac{h^2}{2} y''(\xi_j) = y(t_j) + hf(t_j, y(t_j)) + \frac{h^2}{2} y''(\xi_j).$$

Setting $w_0 = \alpha$, the above becomes

$$w_{j+1} = w_j + hf(t_j, w_j), j = 0, 1 \ldots, N - 1.$$

Theorem: Suppose that in an initial value ODE where $f(t, y)$ is continuous and satisfies Lipschitz condition. So $|f(t, y_1) - f(t, y_2)| \le L|y_1 - y_2|$. Let $w_0, w_1, \ldots, w_N$ be the approximations generated by Euler's method for some N, then for each $j = 0, 1, \ldots, N$,

$$|y(t_j) - w_j| \le \frac{hM}{2L} \left(e^{L(t_j - a)} - 1 \right),$$

where $M = \max_{t \in [a,b]} |y''(t)|$.

2nd Order Taylor Polynomials: $w_{j+1} = w_j + hT^{(2)}(t_j, w_j)$, for $j = 0, 1, \ldots, N - 1$, where $T^{(2)}(t, w) = f(t, w) + f'(t, w) = f(t, w) + \frac{\partial f}{\partial t}(t, w) + \frac{\partial f}{\partial y}(t, w) f(t, w)$.

n-th Order Taylor Polynomials: $w_{j+1} = w_j + hT^{(n)}(t_j, w_j), j = 0, 1, \ldots,$ where $T^{(n)}(t_j, w_j) = f(t_j, w_j) +$

$\frac{h}{2}f'(t_j, w_j) + \cdots + \frac{h^{n-1}}{n!}f^{(n-1)}(t_j, w_j)$, where each $f^{(k)}$ is the k-th total derivative.

When doing n-th order Taylor polynomials, LTE is n-th order, meaning

$$\tau_{j+1}(h) = \frac{y(t_{j+1}) - y(t_j)}{h} - T^{(n)}(t_j, y(t_j)) = \frac{h^n}{(n+1)!}y^{(n+1)}(\xi_j).$$

Theorem - First order Taylor expansion in two variables: Let $(t_0, y_0), (t, y) \in D, \Delta_t = t - t_0, \Delta_y = y - y_0$, then $f(t, y) = P_1(t, y) + R_1(t, y)$, where for some $(\xi, \mu) \in D$,

$$P_1(t, y) = f(t_0, y_0) + \Delta_t \frac{\partial f}{\partial t}(t_0, y_0) + \Delta_y \frac{\partial f}{\partial y}(t_0, y_0)$$
$$R_1(t, y) = \frac{\Delta_t^2}{2}\frac{\partial^2 f}{\partial t^2}(\xi, \mu) + \Delta_t \Delta_y \frac{\partial^2 f}{\partial t \partial y}(\xi, \mu) + \frac{\Delta_y^2}{2}\frac{\partial^2 f}{\partial y^2}(\xi, \mu).$$

Reverse Theorem of Above: Let $P_1(t, y) = f(t_0, y_0) + \Delta_t \frac{\partial f}{\partial t}(t_0, y_0) + \Delta_y \frac{\partial f}{\partial y}(t_0, y_0)$, defined same as above, for very small Δ_t, Δ_y and for $t = t_0 + \Delta_t, y = y_0 + \Delta_y$, then

$$P_1(t, y) = f(t, y) - R_1(t, y).$$

2nd Order Runge-Kutta Method (Midpoint Method): $w_0 = \alpha, w_{j+1} = w_j + hf(t_j + h/2, w_j + h/2 f(t_j, w_j))$.

Modified Euler's Method: $w_{j+1} = w_j + \frac{h}{2}(f(t_j, w_j) + f(t_{j+1}, w_j + hf(t_j, w_j)))$.

Local Truncation Error (LTE): For the above two methods,

$$\tau_{j+1}(h) = \frac{y(t_{j+1}) - y(t_j)}{h} - (a_1 f(t_j, y(t_j)) + a_2 f(t_j + \alpha_2, y(t_j) + \delta_2 f(t_j, y(t_j)))) = O(h^2),$$

for $1 = a_1 + a_2, h/2 = a_2\alpha_2, \frac{h}{2}f(t_j, y(t_j)) = a_2\delta_2$.

Runge-Kutta Order Four

$$\begin{aligned}
w_0 &= \alpha,\\
k_1 &= hf(t_i, w_i),\\
k_2 &= hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_1\right),\\
k_3 &= hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_2\right),\\
k_4 &= hf(t_{i+1}, w_i + k_3),\\
w_{i+1} &= w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),
\end{aligned}$$

Figure 2: Runge-Kutta Method of Order 4

Adaptive Error Control: $w_{j+1} = w_j + h\phi(t_j, w_j, h), h = t_{j+1} - t_j$

$$\tau_{j+1}(h) = \frac{y(t_{j+1}) - y(t_j)}{h} - \phi(t_j, y(t_j), h) = \frac{y(t_{j+1}) - w_{j+1}}{h} = O(h^n).$$

Given a tolerance $\epsilon > 0$, we would like to estimate **largest step-size** h for which $|\tau_{j+1}(h)| \lesssim \epsilon$. Define

$$\tilde{w}_{j+1} = \tilde{w}_j + h\tilde{\phi}(t_j, \tilde{w}_j, h), \text{ for } j \geq 0.$$

Assume $\tilde{w}_j \approx y(t_j)$, then $\phi(t, w, h)$ is an order-n method.

Let $\tilde{\phi}(t_j, w_j, h)$ is an order-$(n+1)$ method

$$\tilde{\tau}_{j+1}(h) = \frac{y(t_{j+1}) - y(t_j)}{h} - \tilde{\phi}(t_j, y(t_j), h) = \frac{y(t_{j+1}) - \tilde{w}_{j+1}}{h} = O(h^{n+1}).$$

$$\textbf{LTE Estimate}: \ \tau_{j+1}(h) \approx \frac{\tilde{w}_{j+1} - w_{j+1}}{h}$$

New step size qh should satisfy

$$qh \lesssim \left|\frac{\epsilon h}{\tilde{w}_{j+1} - w_{j+1}}\right|^{1/n} h = \left|\frac{\epsilon}{\tilde{\phi}(t_j, w_j, h) - \phi(t_j, w_j, h)}\right|^{1/n} h$$

Runge-Kutta-Fehlberg: Step-size selection procedure: first compute

$$q = \left|\frac{\epsilon h}{2(\tilde{w}_{j+1} - w_{j+1})}\right|^{1/n},$$

then let $h = \begin{cases} 0.1h, & \text{if } q \leq 0.1, \\ qh, & \text{if } 0.1 < q < 4 \\ 4h, & \text{if } q \geq 4, \end{cases}$, and set $h = \min(h, h_{max})$ to prevent step size from being too large. If step size is too small, i.e. $h < h_{min}$, then declare failure.

Runge-Kutta-Fehlberg: 4^{th} order method, 5^{th} order estimate

$$\begin{aligned}
w_{j+1} &= w_j + \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5, \\
\tilde{w}_{j+1} &= w_j + \frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6, \quad \text{where} \\
k_1 &= h\,f\,(t_j, w_j), \\
k_2 &= h\,f\left(t_j + \frac{h}{4}, w_j + \frac{1}{4}\,k_1\right), \\
k_3 &= h\,f\left(t_j + \frac{3h}{8}, w_j + \frac{3}{32}\,k_1 + \frac{9}{32}\,k_2\right), \\
k_4 &= h\,f\left(t_j + \frac{12h}{13}, w_j + \frac{1932}{2197}\,k_1 - \frac{7200}{2197}\,k_2 + \frac{7296}{2197}\,k_3\right), \\
k_5 &= h\,f\left(t_j + h, w_j + \frac{439}{216}\,k_1 - 8\,k_2 + \frac{3680}{513}\,k_3 - \frac{845}{4104}\,k_4\right), \\
k_6 &= h\,f\left(t_j + \frac{h}{2}, w_j - \frac{8}{27}\,k_1 + 2\,k_2 - \frac{3544}{2565}\,k_3 + \frac{1859}{4104}\,k_4 - \frac{11}{40}\,k_5\right).
\end{aligned}$$

Figure 3: Runge-Kutta-Fehlberg, 4-th order method

Runge-Kutta Method LTE: Errors:

$$\text{actual} = \left|\frac{y(t_j) - w_j}{h_j}\right|, \qquad \text{estimate} = \left|\frac{\tilde{w}_j - w_j}{h_j}\right|.$$

Multi-Step Methods: For each $0 \le j \le N-1$, integrate ODE:

$$y(t_{j+1}) - y(t_j) = \int_{t_j}^{t_{j+1}} f(t, y(t))dt \approx \int_{t_j}^{t_{j+1}} P(t)dt.$$

4-th order Adams-Bashforth Method (explicit, 4-step):

$$\begin{aligned} w_{j+1} &= w_j + \frac{h}{24}(55f(t_j, w_j) - 59f(t_{j-1}, w_{j-1}) + 37f(t_{j-2}, w_{j-2}) - 9f(t_{j-3}, w_{j-3})) \\ LTE &= \frac{251}{720} f^{(4)}(\xi, y(\xi))h^4 \end{aligned}$$

4-th order Adams-Moulton Method (implicit, 3-step):

$$\begin{aligned} w_{j+1} &= w_j + \frac{h}{24}(9f(t_{j+1}, w_{j+1}) + 19f(t_j, w_j) - 5f(t_{j-1}, w_{j-1}) + f(t_{j-2}, w_{j-2})) \\ LTE &= -\frac{19}{720} f^{(4)}(\xi, y(\xi))h^4 \end{aligned}$$

Predictor-Corrector (PC):

1. Adams-Bashforth **Predictor**: $w^p_{j+1} = w_j + \frac{h}{24}(55f(t_j, w_j) - 59f(t_{j-1}, w_{j-1}) + 37f(t_{j-2}, w_{j-2}) - 9f(t_{j-3}, w_{j-3}))$
2. Adams-Moulton **Corrector**: $w_{j+1} = w_j + \frac{h}{24}(9f(t_{j+1}, w^p_{j+1}) + 19f(t_j, w_j) - 5f(t_{j-1}, w_{j-1}) + f(t_{j-2}, w_{j-2}))$

Step-size Selection:

$$\text{LTE estimate} : \tau_{j+1}(h) = \frac{y_{j+1} - w_{j+1}}{h} \approx -\frac{19}{270} \frac{w_{j+1} - w^p_{j+1}}{h}.$$

Then we can choose a conservative value for q, letting

$$q = 1.5 \left(\frac{\epsilon h}{|w_{j+1} - w^p_{j+1}|} \right)^{1/4}.$$

Then set

$$h = \begin{cases} h \max(q, 0.1), & \text{if } q < 1, \\ h \min(q, 4), & \text{if } q > 2, \\ h, & \text{if } 1 \le q \le 2. \end{cases}$$

Finally, let $h = \min(h, h_{max})$. If $h < h_{min}$, then declare failure.

Predator and Prey Model: Let x = prey population, and y = predator population. Then

$$x' = \alpha x - \beta xy, y' = -\gamma y + \delta xy.$$

A system of m first-order ODEs:

$$\begin{aligned} \frac{du_1}{dt} &= f_1(t, u_1, \dots, u_m) \\ \frac{du_2}{dt} &= f_2(t, u_1, \dots, u_m) \\ &\vdots \\ \frac{du_m}{dt} &= f_m(t, u_1, \dots, u_m), \end{aligned}$$

where $a \le t \le b$, with m initial conditions $u_1(a) = \alpha_1, u_2(a) = \alpha_2, \ldots, u_m(a) = \alpha_m$. In vector form, this looks like $u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix}$, $\alpha = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix}$, and $f(t,u) = \begin{pmatrix} f_1(t, u_1, \ldots, u_m) \\ f_2(t, u_1, \ldots, u_m) \\ \vdots \\ f_m(t, u_1, \ldots, u_m) \end{pmatrix}$. We can then rewrite $f(t,u) = \alpha$.

The same can be done for higher order ODEs, and thus, every ODE is essentially a first order ODE of vectors.

Definition - Vector Lipschitz condition: The function $f(t,u)$ for $u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix} \in \mathbb{R}^m$ defined on the set $D = \{(t,u) | a \le t \le b, -\infty < u_j < \infty, 1 \le j \le m\}$ satisfies a Lipschitz condition on D if

$$|f(t,u) - f(t,z)| \le L \sum_{j=1}^{m} |u_j - z_j|, \text{ where } z = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{pmatrix} \in \mathbb{R}^m,$$

for a constant L and all $(t,u), (t,z) \in D$.

Theorem: $f(t,u)$ satisfies a Lipschitz condition with Lipschitz constant L on D if

$$\left| \frac{\partial f}{\partial u_j}(t,u) \right| \le L, \quad j = 1, 2, \ldots, m.$$

Theorem: Suppose that $f_j(t,u)$ satisfies a Lipschitz condition with Lipschitz constant L on D for all $1 \le j \le m$. Then the system of initial value ODEs has a unique solution $u = u(t)$ for all $t \in [a,b]$.

The vector ODE method is the exact same as the scalar ODE method, just with vectors instead.

Definition - Consistency:

$$\lim_{h \to 0} \max_{0 \le j \le N} |\tau_j(h)| = 0, x_j = a + jh.$$

Definition - Convergent:

$$\lim_{h \to 0} \max_{0 \le j \le N} |y(t_j) - w_j| = 0.$$

Theorem: Suppose a one-step method with $w_0 = \alpha$, and $w_{j+1} = w_j + h\phi(t_j, w_j, h)$, for $j = 0, 1, \ldots$. Suppose that $\phi(t, w, h)$ is continuous and satisfies Lipschitz condition with Lipschitz constant L, for $0 < h < h_0$. Define $D = \{(t, w, h) | a \le t \le b, -\infty < w < \infty, 0 < h < h_0\}$. Then the method is

1. stable,
2. convergent $\iff$ consistent $\iff$ $\phi(t, y, 0) = f(t, y)$, for $a \le t \le b$,
3. $|y(t_j) - w_j| \le \frac{\tau(h)}{L} e^{L(t_j - a)}$, where $\tau(h) = \max_{0 \le j \le N} |\tau_j(h)|$.

Finite Recurrence Relation: Given *recurrence relation* $w_{j+1} = a_{m-1}w_j + a_{m-2}w_{j-1} + \cdots + a_0 w_{j+1-m}$, we have *characteristic polynomial* $P(\mu) = \mu^m - (a_{m-1}\mu^{m-1} + \cdots + a_1\mu + a_0)$. If $P(\mu)$ has m distinct roots $\mu_1, \ldots, \mu_m$, then

$$w_j = c_1\mu_1^j + c_2\mu_2^j + \cdots + c_m\mu_m^j, j = 0, 1, \ldots, m-1, m, \ldots$$

for constants $c_1, c_2, \ldots, c_m$ determined by the equations for $0 \le j \le m-1$.

Root condition: Every root μ_i of $P(\mu)$ must satisfy $|\mu_i| \le 1$.

Multistep Methods

The multistep method with $w_0 = \alpha, w_1 = \alpha_1, \ldots, w_{m-1} = \alpha_{m-1}$ for $j = m-1, m, \ldots$ is given by

$$w_{j+1} = a_{m-1}w_j + a_{m-2}w_{j-1} + \cdots + a_0 w_{j+1-m} + hF(t_j, w_{j+1}, w_j, \ldots, w_{j+1-m}),$$

but we often assume $F = 0$.

Stability Adjectives: Assume multistep method satisfies root condition, then

1. **Strongly stable**: $\mu = 1$ is the only root of $P(\mu)$ with magnitude 1.
2. **Weakly stable**: $P(\mu)$ has more than one distinct root with magnitude 1.
3. Otherwise, method is **unstable**.

Theorem: Assume multistep method with $w_0 = \alpha, w_1 = \alpha_1, \ldots, w_{m-1} = \alpha_{m-1}$. Let $w_{j+1} = a_{m-1}w_j + a_{m-2}w_{j-1} + \cdots + a_0 w_{j+1-m} + hF(t_j, w_{j+1}, w_j, \ldots, w_{j+1-m})$. If the method is consistent, then

$$\text{stable} \iff \text{convergent} \iff \text{satisfies root condition.}$$

The LTE

$$\tau_{j+1}(h) = \frac{y(t_{j+1}) - (a_{m-1}y(t_j) + a_{m-2}y(t_{j-1}) + \cdots + a_0 y(t_{j+1-m}))}{h} - F(t_j, y(t_{j+1}), y(t_j), \ldots, y(t_{j+1-m})).$$

For a multistep method, **consistency** means

$$\lim_{h\to 0} \max_{m \le j \le N} |\tau_j(h)| = 0, \qquad \lim_{h \to 0} \max_{0 \le j \le m-1} |y(t_j) - \alpha_j| = 0.$$

Convergence has the same definition as in the single-step case.

Stiff ODEs

Numerical Stability: Small error in α implies small error in w_j.

Test equation: $\frac{dy}{dt} = \lambda y, y(0) = \alpha$, for $\lambda < 0$. Multistep method gives

$$w_{j+1} = a_{m-1}w_j + a_{m-2}w_{j-1} + \cdots + a_0 w_{j+1-m} + \lambda h(b_m w_{j+1} + b_{m-1}w_j + \cdots + b_0 w_{j+1-m}),$$

or

$$(1 - \lambda h b_m)w_{j+1} - (a_{m-1} + \lambda h b_{m-1})w_j - \cdots - (a_0 + \lambda h b_0)w_{j+1-m} = 0.$$

The **characteristic polynomial** for the test equation is

$$Q(z, \lambda h) = (1 - \lambda h b_m)z^m - (a_{m-1} + \lambda h b_{m-1})z^{m-1} - \cdots - (a_0 + \lambda h b_0).$$

Assume the characteristic polynomial has distinct roots $\beta_1, \beta_2, \ldots, \beta_m$, then $\exists$ constants $c_1, c_2, \ldots, c_m$ such that

$$w_j = c_1(\beta_1)^j + c_2(\beta_2)^j + \cdots + c_m(\beta_m)^j, \text{ for } j = 0, 1, 2, \ldots.$$

For convergence and stability, we require that $|\beta_k| < 1, k = 1, 2, \ldots, m$.

We define the **region of absolute stability** R as

$$R = \{\lambda h \in \mathbb{C}|\ \ |\beta_k| < 1,\ \text{for all zeros } \beta_k \text{ of } Q(z, \lambda h)\}.$$

Euler's Method: $w_{j+1} = w_j + hf(t_j, w_j) = (1 + \lambda h)w_j$, and the characteristic polynomial is

$$Q(z, \lambda h) = z - (1 + \lambda h) = 0,$$

with root $\beta_1 = 1 + \lambda h$. So $R = \{\lambda h \in \mathbb{C}|\ \ |1 + \lambda h| < 1\}$.

Implicit Trapezoid: $R = \{\lambda h \in \mathbb{C}|\ \ \Re(\lambda h) < 0\}$.

Definition - A-Stable: Implicit Trapezoid is A-*stable*.

Chapter 6 - Matrices

Elementary matrix operations: Given a system of equations as rows of a matrix E_i, we have the following elementary matrix operations which do not alter the determinant

1. Multiplying a row by a non-zero constant λ, this operation is defined by $((\lambda E_i) \to (E_i))$,
2. Adding the multiple (λ) of a row E_j to some row E_i, i.e. $((E_i + \lambda E_j) \to (E_i))$,
3. Transposing rows, i.e. $((E_j) \leftrightarrow (E_i))$.

Partial pivoting, or **maximal column pivoting**, pivots by which row has the largest value in the current pivot column.
Scaled partial pivoting defines a scale factor s_i, the greatest magnitude for each row i, and interchanges rows by the greatest ratio of the current pivot position to scale factor for that row. So if we're just starting the pivot process, we consider A_{i1}/s_i to determine which row to interchange with the topmost row at each step.
Complete pivoting exchanges the entry, at position row i and column j, on the k-th step, searching through $a_{ij}, k \leq i, j \leq n$. Additional time for complete pivoting is $n(n-1)(2n+5)/6$.

Inverse: To compute A^{-1}, we simply need to solve

$$AX = I,\ \text{solving } Ax_j = e_j.$$

$(AB)^T = B^T A^T, (A^{-1})^T = (A^T)^{-1}$.

For a matrix $A \in \mathbb{R}^{n \times n}$, **minor** $M_{ij} = \det((n-1) \times (n-1)$ submatrix of A without row i and column j.) Then

$$\det(A) = \sum_{i=1}^{n} (-1)^{i+j} a_{ij} M_{ij}, \forall 1 \leq j \leq n.$$

Properties of Determinant: For $A \in \mathbb{R}^{n \times n}$

1. $\det(A^T) = \det(A)$.
2. If A^{-1} exists, then $\det(A^{-1}) = (\det(A))^{-1}$.
3. For $B \in \mathbb{R}^{n \times n}$, $\det(AB) = \det(A)\det(B)$.
4. $\det(A) \neq 0 \iff Ax = b$ has a unique solution.

More stuff about determinants:

1. If $\widetilde{A}$ is obtained from A through elementary operation $(E_i) \leftrightarrow (E_j)$, for $i \neq j$, then $\det(\widetilde{A}) = -\det(A)$; that is to say row swaps change signs of determinant.

2. Other elementary matrix operations do not affect the determinant.

3. If A is upper or lower-triangular, then the determinant of A is equal to the trace of A.

To compute the determinant of A, we use pivoting to turn A into an upper-triangular matrix $\widetilde{A}$, then

$$\det(A) = (-1)^{\#\text{ of row swaps}}\,\mathrm{Tr}(\widetilde{A}).$$

A **leading principle sub-matrix of** $A \in \mathbb{R}^{n\times n}$ is

$$A_k = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \dots & a_{kk} \end{pmatrix},$$

for each $k = 1, 2, \dots, n$.

Theorem 6.25: A symmetric matrix A is positive definite if and only if each of its leading principal submatrices has a positive determinant.

Matrix Factorization Basic Facts: Define $l_{js} = \dfrac{a_{js}}{a_{ss}}, 1+s \leq j \leq n$ and $\overline{a}_{jk} = a_{jk} - l_{js}a_{sk}, 1+s \leq j, k \leq n$.

► *Blessing* #1: $A \longrightarrow \boxed{\text{matrix-matrix product}} \longrightarrow$ **new** A

$$A = \left(\begin{array}{c|ccc} 1 & & & \\ \hline l_{21} & 1 & & \\ \vdots & & \ddots & \\ l_{n1} & & & 1 \end{array}\right) \cdot \left(\begin{array}{c|ccc} a_{11} & a_{12} & \cdots & a_{1n} \\ \hline 0 & \overline{a}_{22} & \cdots & \overline{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \overline{a}_{n2} & \cdots & \overline{a}_{nn} \end{array}\right)$$

Figure 4: Blessing #1 from GE God

1.

2. Let upper-triangular matrix $U = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ & u_{22} & \dots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{pmatrix}$ be A post Gaussian elimination.

3. Define lower-triangular matrix $L = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \dots & 1 \end{pmatrix}$.

4. Then $A = LU$.

A **permutation matrix** $P = (p_{ij})$ is a matrix obtained by rearranging the rows of the identity matrix I_n.

For example, $P_{2,3} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$.

$P_{k,s} \cdot A$ is A with rows k and s interchanged.

Then using $PA = LU$, can solve $Ax = b$ by $P \cdot Ax = L \cdot Ux = Pb$, or $x = U^{-1}(L^{-1}(P \cdot b))$.

Strictly Diagonal Dominant (SDD) Matrices: An $n \times n$ matrix $A = (a_{ij})$ is SDD if

$$|a_{ii}| > \sum_{i \neq j=1}^{n} |a_{ij}| \text{ holds for } i = 1, 2, \ldots, n.$$

Gaussian Elimination succeeds without pivoting on SDD matricies. This is because SDD is invariant under each step of $L \cdot U$ factorization.

Symmetric Positive Definite (SPD) Matrices: An $n \times n$ matrix $A = (a_{ij})$ is SPD if

$$A = A^T, x^T Ax > 0, \text{ for any non-zero} x \in \mathbb{R}^n,$$

where

$$x^T Ax = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

Gaussian Elimination with partial pivoting (GEPP) succeeds without pivoting on SPD matricies. This is because SPD is invariant under each step of $L \cdot U$ factorization.

Cholesky Factorization is a special LU factorization for SPD matrices: $A = LDL^T$, where A is symmetric, L is a lower-triangular matrix and D is a diagonal matrix. In terms of a LU factorization, $U = DL^T$.

An $n \times n$ matrix A is tri-diagonal if all the entries not on the 3 main diagonals of A are 0.

Natural Splines equations in matrix form: For spline coefficients $c_{j}{}_{j=1}^{n-1}$,

$$A \cdot \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{pmatrix} = RHS,$$

where

$$A = \begin{pmatrix} 2(h_0 + h_1) & h_1 & & & \\ h_1 & 2(h_1 + h_2) & h_2 & & \\ & \ddots & \ddots & \ddots & \\ & & h_{n-3} & 2(h_{n-3} h_{n-2}) & h_{n-2} \\ & & & h_{n-2} & 2(h_{n-2} + h_{n-1}) \end{pmatrix}$$

The remaining matrix is tri-diagonal after each step of $L \cdot U$ factorization. This whole process takes $3n$ operations.