Math 151A: Applied Numerical Methods

Prof. M. Zhou | Winter 2025

Contents

1 Review

Theorems (Calculus):

1. **IVT**: Let $f \in C[a,b]$ and let $k \in \mathbb{R}$ between $f(a), f(b)$. Then $\exists\, c \in [a,b]$ s.t. $f(c) = k$.
2. **MVT**: Let $f \in C[a,b]$ be differentiable on (a,b). Then $\exists\, c \in (a,b)$ s.t. $f'(c) = \frac{f(b)-f(a)}{b-a}$.

Theorem (Taylor's Theorem). Let $f \in C^n[a,b]$, and let $x_0 \in [a,b]$. Assume f^{n+1} exists on $[a,b]$. Then $\forall\, x \in [a,b]$, $\exists\, \xi_x \in [x_0, x]$ s.t. $f(x) = P_n(x) + R_n(x)$, where:

$$\boxed{\begin{aligned} P_n(x) &= f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)}{2}(x-x_0)^2 + \ldots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n \\ R_n(x) &= \frac{f^{(n+1)}(\xi_x)}{(n+1)!}(x-x_0)^{n+1} \end{aligned}}$$

Integration by Parts: $\int u(x)v'(x)dx = u(x)v(x) - \int u'(x)v(x)dx$

Error: Let p be an approximation of p^*:

1. ***Absolute error*** $\left(|p - p^*|\right)$ vs. ***relative error*** $\left(\left|\frac{p-p^*}{p^*}\right|\right)$ $[p \neq 0]$
2. ***Underflow error***: When a small number rounds to 0 (after subtracting near-equal #s, e.g.)
3. ***Overflow error***: When a large number rounds to $\pm\infty$ (from dividing by a small #, e.g.)

Big-O Notation: Say that a function $f(x)$ is $O(g(x))$ as $x \to \infty$ if $\exists$ constants $M > 0, x_0 \in \mathbb{R}$ s.t.:

$$\boxed{|f(x)| \leq M\,|g(x)| \;\; \forall\, x > x_0}$$

Say that $f(x)$ is $O(g(x))$ as $x \to a$ if $\exists\, M, \delta > 0$ s.t.:

$$\boxed{|f(x)| \leq M\,|g(x)| \;\; \forall\, x \text{ s.t. } |x-a| \leq \delta}$$

Orders of Convergence

A convergent sequence $\{x_n\}_{n\geq 1}$ with limit x is said to converge with order $\alpha \geq 1$ to x if $\exists$ constant $L \in (0, \infty)$ s.t.:

$$\boxed{\lim_{n\to\infty} \frac{|x_{n+1} - x|}{|x_n - x|^\alpha} = L}$$

2 Floating Point

Floating point: computational form for representing a decimal number; can denote by $Fl(x) = x+\epsilon$

1. **Single-precision/short/float32**: ~6-9 [base 10] decimal digits
2. **Double-precision/long/float64**: ~15-17 decimal digits
3. In general: machine accuracy given by $\epsilon \approx 10^{-16}$

The IEEE *binary64* Floating Point

IEEE binary64 floating point format (64 bits total) is divided into 3 parts: (i) the ***sign bit*** [1 bit], (ii) the ***exponent*** [11 bits], and (iii) the ***significand*** [52 bits]

$$\boxed{\underline{\text{FP}} : (-1)^{\text{sign}} \cdot [1.\text{significand}]_2 \cdot 2^{\text{exponent}-2}}$$

Floating Point:

1. Sign bit indicates sign of a number, including ± *zero*
2. Exponent bits are unsigned, range from 0 to 2047 $\xrightarrow[-1024]{}$ -1022 to +1023
   - -1023 (all 0s), +1024 (all 1s) reserved for special numbers

Finite-Digit Arithmetic: For numbers that cannot be represented exactly, have to either ***chop*** [remove extra digits] or ***round*** [round to nearest representable number]

- When doing arithmetic: chop/round after every operation
- Accumulate error with every operation → can use ***nested arithmetic***: factor common terms to reduce the total # operations [FLOPs: floating point operations] performed

3 Root-Finding

Bisection Search

1. Start with search region $[a_0, b_0]$
2. At every iteration, evaluate $f(\frac{b_n+a_n}{2})$
3. $f > 0 \implies$ choose $a_{n+1} = a_n, b_{n+1} = \frac{b_n+a_n}{2}$; else, choose $a_{n+1} = \frac{b_n+a_n}{2}, b_{n+1} = b_n$

Convergence of Bisection Method: Let $f \in C[a,b]$ s.t. $\text{sign}[f(a)] \neq \text{sign}[f(b)]$. Then the sequence $\{p_n\}_{n\geq 1}$ generated by the bisection method converges globally to a root p of f with error:

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

Fixed-Point Iteration

Def: A ***fixed point*** of a function g is a point p s.t. $g(p) = p$. (Note: f.p. of $g \Leftrightarrow$ root of $x - g(x)$)

Theorems (Fixed Points):

1. **Existence**: Let $g \in C[a,b]$ with $a \leq g(x) \leq b \; \forall \, x \in [a,b]$; then $\exists \, p \in [a,b]$ s.t. $g(p) = p$.
2. **Uniqueness**: If $g'(x)$ exists on (a,b) and $\exists$ constant $0 < k < 1$ s.t. $|g'(x)| \leq k \; \forall \, x \in (a,b)$, then the aforementioned fixed point p is unique.

Fixed-Point Iteration

Given $g \in C[a,b]$ s.t. $g(x) \in [a,b] \; \forall \, x \in [a,b]$ and initial guess $p_0 \in [a,b]$, can find a sequence p_n converging to fixed point p.

Let $\{p_n\}_{n\geq 1}^{\infty}$ def. by $\underline{p_n = g(p_{n-1})}$; if $p_{n n \geq 1}$ converges to $p \in [a,b]$, then p is a fixed point of g.

Theorems (Fixed Point Iteration):

1. **Fixed-Point Theorem**: Let g as above. Suppose g' exists on (a,b) and $\exists$ constant $k \in (0,1)$ s.t. $|g'(x)| \leq k \; \forall \, x \in (a,b)$. Then for any $p_0 \in [a,b]$, the sequence $\{p_n\}_{n\geq 1}$ converges to the unique fixed point $p \in [a,b]$ of g [*Corollary*: with error $\underline{|p_n - p| \leq k^n \cdot \max(p_0 - a, b - p_0)}$].
2. **Convergence**: $\{p_n\}_{n\geq 1}$ converges linearly to p with $\lim_{n\to\infty} \frac{|p_n - p|}{|p_{n-1} - p|} = |g'(p)| \leq k < 1$.

Newton's Method

Given an initial guess p_0 s.t. $|p_0 - p|$ small, have update rule:

$$\boxed{x^{k+1} := x^k - \frac{f(x^k)}{f'(x^k)}}$$

Theorems (Newton's Method):

1. **Convergence**: Let $f \in C^2[a,b]$. If $p \in (a,b)$ satisfies $f(p) = 0, f'(p) \neq 0$, then $\exists\ \delta > 0$ s.t. Newton's method converges to p for any $p_0 \in [p-\delta, p+\delta]$.

2. **Convergence Order**: Let $g = x - \frac{f(x)}{f'(x)} \in C^\alpha[a,b]$ $(\alpha > 2)$, and let $p \in [a,b]$. Assume $g(p) = p$ and $g'(p) = \ldots = g^{\alpha-1}(p)$, but $g^\alpha(p) \neq 0$. Then $p_n \to p$ with order α.

Secant Method

Have update rule:

$$\boxed{x^{k+1} := x^k - \frac{x^k - x^{k-1}}{f(x^k) - f(x^{k-1})} f(x^k)}$$

Modified Newton's Method

Def: A root p of f is a zero of multiplicity m for f if, for $x \neq p$, can write:

$$\boxed{\underline{f(x) = (x-p)^m q(x)} \text{ for } q(x) \text{ s.t. } q(p) \neq 0}$$

Theorems (Modified Newton):

1. Let $f \in C^m(a,b)$ and $p \in (a,b)$.

   Then p is a zero with multiplicity m iff $\underline{0 = f(p) = f'(p) = \ldots = f^{m-1}(p)}$, but $\underline{f^m(p) \neq 0}$.

2. *Corollary*: Let $m \geq 1$, p a zero with multiplicity m. Then the function $\mu(x) := \frac{f(p)}{f'(p)}$ has a zero of multiplicity 1 at p.

   $\to$ **Modified Newton's Method**: Newton's method on $\mu(x)$:

$$\boxed{p_{n+1} = p_n - \frac{\mu(p_n)}{\mu'(p_n)} = p_n - \frac{f(p_n)f'(p_n)}{[f'(p_n)]^2 - f(p_n)f''(p_n)}}$$

Orders of Convergence (Global for bisection; local otherwise)

Method	**Order**
Bisection	1
Fixed-Point	1 if $\lvert g'(p)\rvert \in (0,1)$, ≥ 2 if $g'(p) = 0$
Newton's	≥ 2 if $g'(p) \neq 0$; 1 otherwise
Secant	$(1+\sqrt{5})/2 \approx 1.618$

$(*)$ **Fixed-Point Iteration**: Order = smallest n satisfying $g^{(n)}(p) \neq 0$ (?)

Theorem (Aitken's Δ^2 Method for Accelerating Convergence)

Assume $\{p_n\}_{n\geq 1}$ converges linearly to p, and that for n large, $(p_{n+2}-p)(p_n-p) > 0$. Then the sequence $\{\hat{p}\}_{n\geq 1}$ satisfies $\lim_{n\to\infty} \frac{|\hat{p}_n - p|}{|p_n| - p} = 0$, where:

$$\boxed{\hat{p}_n = p_n - \frac{(p_{n+1}-p_n)^2}{p_{n+2} - 2p_{n+1} + p_n} \text{ for } n \geq 0}$$

4 Data Fitting

Lagrange Polynomials

Define the basis elements for $k = 0, \ldots, n$:

$$L_{k,n} := \frac{(x - x_0) \ldots (x - x_{k-1})(x - x_{k+1}) \ldots (x - x_n)}{(x_k - x_0) \ldots (x_k - x_{k-1})(x_k - x_{k+1}) \ldots (x_k - -x_n)} = \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i}$$

This yields the (unique) Lagrange polynomial for the points $x_0, \ldots, x_n$ [degree $n-1$]:

$$\boxed{P(x) = \sum_{k=0}^{n} f(x_k) L_{k,n}(x) = \sum_{k=0}^{n} f(x_k) \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{x - x_i}{x_k - x_i}}$$

Theorem (**Lagrange Interpolation Error**). Let $x_0, \ldots, x_n \in [a, b]$ be distinct, and let $f \in C^{n+1}[a, b]$. Then for each $x \in [a, b]$, $\exists\ \epsilon(x) \in [a, b]$ within the interval spanned by $x_0, \ldots, x_n$ s.t.:

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

The Divided Differences Method

Let $x_0, \ldots, x_n$ be distinct, and let $P(x)$ be the Lagrange polynomial of $\{(x_i, f(x_i)\}$. ***Newton's divided differences*** is an expression for $P(x)$ in the following form:

$$\boxed{P_n(x) := a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots + a_n(x - x_0) \ldots (x - x_{n-1})}$$

Define the divided differences by:

1. 0^{th} divided difference: $f[x_i] = f(x_i)$
2. 1^{st} divided difference: $f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}$
3. k^{th} divided difference $f[x_i, x_{i+1}, \ldots, x_{i+k}] = \frac{f[x_{i+1}, \ldots, x_{i+k}] - f[x_i, \ldots, x_{i+k-1}]}{x_{i+k} - x_i}$

Newton's Divided Difference Interpolating Polynomial

The divided difference polynomial coefficients a_i are given by $\underline{a_k = f[x_0, x_1, \ldots, x_k]}$:

$$\boxed{P(x) = f(x_0) + \sum_{k=1}^{n} f[x_0, \ldots, x_k](x - x_0)(x - x_1) \ldots (x - x_{k-1})}$$

Divided Differences for Equally-Spaced Points: Assume nodes are arranged consecutively with equal spacing $h = x_{i+1} - x_i$ $[i = 0, \ldots, n-1]$ between nodes. Let $x = x_0 + sh$; then:

$$\boxed{P_n(x) = P_n(x_0 + sh) = f[x_0] + \sum_{k=1}^{n} (\binom{s}{k}) k! h^k f[x_0, x_1, \ldots, x_k]}$$

Lagrange Polynomial Methods:

- Can use the divided differences method to iteratively construct higher- and higher-order polynomials (up to degree n)
  - Start from degree 0, use to build degree 1, etc.

<u>**Non-Equispaced Polynomial Interpolation**</u>

Runge's phenomenon: Even for "well-behaved" $f \in C^{n+1}[a, b]$, the interpolation error may still be large if the nodes $\{x_i\}_{i=0,\ldots,n}$ are equispaced.

Theorem (**Runge's phenomenon**). If the Lagrange polynomial nodes $\{x_i\}_{i=0,\ldots,n}$ are equispaced (i.e. $x_i = x_0 + ih$ for $i = 0, \ldots, n$) and $x_0 = a, x_n = b$, then:

$$\boxed{\max_{x \in [a,b]} \prod_{j=0}^{n} |x - x_j| \leq \frac{1}{4} h^{n+1} n! \quad \Longrightarrow \quad |f(x) - P(x)| \leq \frac{M}{n+1} \cdot \frac{h^{n+1}}{4}}$$

To reduce error: instead of equispaced nodes, use **Chebyshev nodes** (<u>$a = -1, b = 1$</u>):

$$\boxed{\tilde{x}_i = \cos\left(\frac{2i+1}{2n+2}\pi\right), \quad i = 0, \ldots, n}$$

The Chebyshev nodes minimize the error:

$$\max_{x \in [a,b]} \prod_{j=0}^{n} |x - x_j|$$

Theorem (**Chebyshev Polynomials**). Let $T_n(x) = \prod_{k=0}^{n} (x - \tilde{x}_k)$ be the n^{th}-order Chebyshev polynomial, where $\tilde{x}_k$ are the n^{th}-order Chebyshev nodes. Then $T_n(x)$ satisfies:

1. <u>$T_n(x) = 2^{-n} \cos\Big((n+1)\arccos(x)\Big)$</u> $\forall\, x \in (-1, 1)$
2. The T_n's are recursive: <u>$T_{n+1}(x) = xT_n(x) - \frac{1}{4}T_{n-1}(x)$</u>

Cubic Splines

A ***cubic spline*** for a function f on $[a, b]$ (with nodes $a = x_0 < x_1 < \ldots < x_n = b$) is a function $S(x)$ satisfying:

1. **Piecewise Cubic Polynomial**: on each subinterval $I_j = [x_j, x_{j+1}]$ for $j = 0, 1, \ldots, n-1$, $S(x)$ is a cubic polynomial of the form (for $x \in I_j$):

$$\boxed{S(x) = S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3}$$

2. **Interpolation**: $S(x)$ satisfies $S(x_j) = f(x_j)$ for $j = 0, 1, \ldots, n$
3. **Continuity & Differentiability**: $S(x)$ is continuous and has continuous 1^{st} & 2^{nd} derivatives

Boundary conditions: Need 2 addl. constraints to obtain $(4n) \times (4n)$ linear system:

1. **Natural boundary condition**: $S''(x_0) = S''(x_n) = 0$
2. **Clamped boundary condition**: For $f'(x_0), f'(x_n)$ known: $S'(x_0) = f'(x_0)$, $S'(x_n) = f'(x_n)$

Constraints:

1. Interpolation: $S_j(x_j) = f(x_j)$, $S_{j+1}(x_{j+))} = f(x_{j+1})$ for $j = 0, 1, \ldots, n-1$ [$(n+1)$-many]
2. $S(x) \in \mathcal{C}^2$: $S_j^{(k)}(x_{j+1}) = S_{j+1}^{(k)}(x_{j+1})$ for $j = 0, \ldots, n-2$ and $k = 0, 1, 2$ [$3 \times (n-1)$-many]
3. Boundary conditions: 2

→ **Theorem**: The spline interpolant is unique.

5 Numerical Differentiation

First-order methods: Evaluate limit defn. of derivative for finite h:

$$\textbf{Forward diff}:\ f'(x_0) \approx \frac{f(x_0+h)-f(x_0)}{h}, \quad \textbf{backward diff}:\ f'(x_0) \approx \frac{f(x_0)-f(x_0-h)}{h}$$

- *Error*: $\frac{f(x_0+h)-f(x_0)}{h} = f'(x_0) + \frac{h}{2}f''(\xi)$ [$\xi \in [x_0, x_0+h]$, by Taylor] $\to e \leq \frac{h}{2}M$ for $f' \leq M$

Second-order methods: Difference of finite difference equations

$$\to \textbf{ Central difference}:\ \frac{f(x_0+h)-f(x_0-h)}{2h} = f'(x_0) + \frac{f^{(3)}(\xi_1)f^{(3)}(\xi_2)}{12}h^2 \quad [\mathrm{O}(h^2)]$$

Richardson extrapolation: Combine low-order formulas (with different h) to generate higher-order results

- Based on Taylor: want to cancel lower-order terms in formulas' error expressions
- R.E. for forward difference $[2D^+_{\frac{h}{2}}f(x_0) - D^+_h f(x_0)]$:

$$\boxed{\frac{-f(x_0+h)+4f(x_0+\frac{h}{2})-3f(x_0)}{h} = f'(x_0) + \frac{1}{6}h^2\left(\frac{1}{2}f^{(3)}(\xi_2) - f^{(3)}(\xi_1)\right)}$$

Differentiation round-off error: $\tilde{f}(x) = f(x) + e(x)$

$$\implies \left| f'(x_0) - \frac{\tilde{f}(x_0+h)-\tilde{f}(x_0-h)}{2h} \right| \leq \frac{\epsilon}{h} + \frac{h^2}{6}M \quad [\text{assuming } e(x) \leq \epsilon,\ f^{(3)} \leq M]$$

6 Numerical Integration

Numerical Quadrature: Approximate $\int_a^b f(x)dx$ by a discrete weighted sum of $f(x_i)$s:

$$\int_a^b f(x)dx \approx \sum_{i=0}^{N} w_i f(x_i)$$

Weighted MVT: Let $h \in C(a,b)$ and g integrable on (a,b). If $g(x)$ does not change sign on $[a,b]$, then $\exists\ c \in (a,b)$ s.t. $\int_a^b h(x)g(x)dx = h(c)\int_a^b g(x)dx$

Newton-Cotes: Approximate $f(x)$ by its Lagrange polynomial $P(x)$

1. **Trapezoidal Rule** [2 points]:

$$\int_a^b f(x)dx = \left[\frac{h}{2}f(a) + \frac{h}{2}f(b)\right] - \frac{h^3}{12}f''(c)$$

2. **Simpson's Rule** [3 equispaced points]: $\int_a^b f(x)dx = \frac{h}{3}\left[f(x_0) + 4f(x_1) + f(x_2)\right] - h^5\frac{f^{(4)}(\eta)}{90}$

3. **General Newton-Cotes**: Given $(n+1)$ equispaced points $a = x_0 < \ldots < x_n = b$:

$$\boxed{\int_a^b f(x)dx \approx \sum_{i=0}^{n} w_i f(x_i) \text{ with } w_i = \int_a^b L_i(x)dx = \int_a^b \prod_{\substack{j=0 \\ j\neq i}}^{a} \frac{x - x_j}{x_i - x_j}dx}$$

Degree of precision: Largest integer n for which a formula is exact for $x^k\ \forall\ k = 0, 1, \ldots, n$ [i.e. $E[x^k] = 0$ for $k = 0, 1, \ldots, n$, but $E[x^{n+1}] \neq 0$].

- Trapezoidal Rule: 2nd order, degree of precision 1
- Simpson's Rule: 4th order, degree of precision 3
- Newton-Cotes: $(n+1)$ nodes $\rightarrow$ degree of exactness n

Composite Numerical Integration: Higher-order Newton-Cotes susceptible to Runge's phenomenon $\rightarrow$ use lower-order Newton-Cotes on subintervals

$$\int_a^b f = \int_{x_0}^{x_1} f + \int_{x_1}^{x_2} f + \ldots + \int_{x_{n-1}}^{x_n} f$$

- Composite Trapezoidal Rule [$x_i = x_0 + ih$]:

$$\int_a^b f(x)dx = \sum_{i=0}^{n-1}\int_{x_i}^{x_{i+1}} f(x)dx = \frac{h}{2}\left(f(a) + 2\sum_{i=0}^{n-1} f(x_i) + f(b)\right) - \frac{h^2}{12}(b-a)f''(\xi)$$

- Composite Simpson's Rule:

$$\int_a^b f(x)dx = \frac{h}{3}\left(f(a) + 2\sum_{i=1}^{\frac{n}{2}-1} f(x_{2i}) + 4\sum_{i=1}^{\frac{n}{2}} f(x_{2i-1}) + f(b)\right) - \frac{h^4}{180}(b-a)f^{(4)}(\xi)$$

- Round-off error: $e(h) \leq (b-a)\epsilon = hn\epsilon$ [stable as $h \to 0$]

Motivation (Gaussian Quadrature): Want to find n nodes x_i and weights c_i such that the formula $\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i)$ is exact for polynomials of degree $\leq 2n-1$

Orthogonal Polynomials: Define the inner product of functions on $[-1,1]$ by

$$\boxed{\langle f, g \rangle = \int_{-1}^{1} f(x)g(x)dx}$$

$\to$ Two functions f, g are ***orthogonal*** if $\langle f, g \rangle = 0$.

Recall (Gram-Schmidt): The vector project of a vector v onto a vector u is defined by:

$$\boxed{\mathrm{proj}_u(v) = \frac{\langle v, u \rangle}{\langle u, u \rangle} u}$$

$\to$ Given vectors $v_1, v_2, \ldots$, the ***Gram-Schmidt*** finds $u_1, u_2, \ldots$ orthogonal by:

$$\boxed{u_i = v_i - \mathrm{proj}_{u_1}(v_1) - \ldots - \mathrm{proj}_{u_{i-1}}(v_{i-1})}$$

Legendre Polynomials: The ***Legendre polynomials*** are the set of orthogonal polynomials obtained from performing the Gram-Schmidt process on $\{1, x, x^2, \ldots\}$; ex:

$$p_0(x) = 1; \quad p_1(x) = x; \quad p_2(x) = \frac{3x^2-1}{2}; \quad p_3(x) = \frac{5x^3-3x}{2}$$

<u>**Gaussian Quadrature**</u>: Let $\{x_i\}_{i=1}^{n}$ be the roots of the n-degree Legendre polynomial (assumed real, distinct). Then the Gaussian quadrature rule is:

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i) \text{ with weights } w_i = \int_{-1}^{1} \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \text{ for } i = 1, 2, \ldots, n$$

- Weights based on Lagrange polynomial; Gaussian quadrature exact for $f \in \mathbb{P}_{2n-1}$
- Error term: for some $\xi \in [a, b]$, $\frac{(b-a)^{2n+1}(n!)^4}{(2n+1)[2n!]^3} f^{(2n)}(\xi)$
- Convert unbounded domains into bounded: via change of variables
  Ex: $\int_0^\infty e^{-x^2} dx \to$ change of vars with $z = \frac{x}{1+x}$ $[z(0) = 0, z(\infty) = 1]$

7 Direct Methods for Solving Linear Systems

Gaussian Elimination: Solve $Ax = b$ [A invertible] via 2-phase process:

1. Transform $Ax = b$ into a new system $Ux = y$, where U is upper-triangular
   - Notation: Eliminate x_i by $(E_j - \frac{a_{ji}}{a_{ii}} E_i) \to (E_j)$ for $j = i+1, i+2, \ldots, n$
2. Solve $Ux = y$ via backward substitution

Cost of G.E.: $\underline{\mathbf{O}(n^3)}$

1. **Variable elimination**: $\mathrm{O}(n^3)$
   - One variable: $(n-i)$ divisions, $(n-i)(n-i+1)$ multiplications & subtractions
   - Over n variables: $\frac{2n^3+3n^2-5n}{6}$ mult/div, $\frac{n^3-n}{3}$ add/subtract
2. **Backward substitution**: $\mathrm{O}(n^2)$
   - One variable: $(n-i)$ multiplications, $(n-i+1)$ adds, 1 div & 1 subtract
   - Total: $\frac{1}{2}(n^2+n)$ mult/div, $\frac{1}{2}(n^2-n)$ add/subtract

Partial Pivoting: To avoid round-off error: to choose which variable to eliminate, select the row E_p [$p \geq i$] with largest $|a_{pi}|$, then swap E_p with E_i and eliminate $x_i' = x_p$

- Round-off error: From dividing by small or subtracting nearly-equal

LU Decomposition: Factor $A = LU$ [L, U lower- and upper-Δ], then solve $LUx = b$ via (i) $Ly = b$ into (ii) $Ux = y$, using forward/backward substitution

- Gaussian Elimination: Compute

$$A^{(n)}x = M^{(n-1)}M^{(n-2)}\ldots M^{(2)}M^{(1)}x = b^{(n)} = M^{(n-1)}M^{(n-2)}\ldots M^{(2)}M^{(1)}b$$

- **LU Factorization**:

$$\boxed{L = (M^{(1)})^{-1}(M^{(2)})^{-1}\ldots(M^{(n)})^{-1}, \quad U = A^{(n)}}$$

- Inverting Gaussian transformation matrices:

$$M^{(i)} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -m_{i+1,i} & \ddots & \\ & & \vdots & & \ddots & \\ & & -m_{n,i} & & & 1 \end{pmatrix} \implies (M^{(i)})^{-1} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & +m_{i+1,i} & \ddots & \\ & & \vdots & & \ddots & \\ & & +m_{n,i} & & & 1 \end{pmatrix}$$

Overall L:

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ m_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \dots & 1 \end{pmatrix}$$

- With row swaps: $A = P^{-1}LU$
- Cost of LU decomp.: $O(n^3)$ factor $\rightarrow$ O(n^2) solve

Special Matrices

Diagonally Dominant: A matrix A is diagonally dominant if:

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \text{ for } i = 1, 2, \dots, n$$

- Strictly DD $\implies$ A nonsingular, G.E. is numerically stable w.r.t. round-off and no swaps
- "Numerically stable": Large divisors, unequal subtractands in all expressions

SPD: A matrix A is SPD if if its symmetric ($A^T = A$) and

$$x^T A x > 0 \; \forall \; x \neq 0 \quad \text{[Equiv.: All } \lambda\text{s} > 0]$$

A SPD $\implies$ A invertible, no row swaps needed for G.E.

Cholesky factorization: A is SPD iff $\exists$ L lower-Δ with positive diagonal entries s.t.:

$$A = LL^T \quad [\leftarrow \text{this is } A\text{'s LU decomp, } L^T = U]$$

Cholesky Algorithm

1. Set $l_{11} = \sqrt{a_{11}}$
2. For $j - 2, \dots, n$, set $l_{j1} = a_{j1}/l_{11}$
3. For $i = 2, \dots, n-1$:
   (a) Set $l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}$
   (b) For $j = i+1, \dots, n$, set:
   $$l_{ji} = \frac{a_{ji} - \sum_{k=1}^{i-1} l_{jk} l_{ik}}{l_{ii}}$$
4. Set $l_{nn} = \sqrt{a_{nn} - \sum_{k=1}^{n-1} l_{nk}^2}$

Tridiagonal matrices: All entries zero except main diagonal and its two adjacent diagonals

- A tridiag $\implies$ can do GE, LU decomp in $O(n)$
- LU factorization has L zero except main/lower diags, U zero except main/upper diags

8 Indirect Methods for Solving Linear Systems

Jacobi Method: For D, L, U diagonal, strictly lower-Δ, strictly upper-Δ portions of A:

$$\boxed{x^{(k+1)} = D^{-1}(b - (L+U)x^k), \qquad x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)}\right)}$$

- Requires nonzero diagonal entries (can ensure via row/column swaps)
- Guaranteed convergence under certain conditions (e.g. A strictly diagonally dominant)

Gauss-Seidel: Use earlier components of $x^{(k+1)}$ to inform later ones:

$$\boxed{x^{(k+1)} = (D+L)^{-1}(b - Ux^{(k)}), \qquad x_i^{(k+1)} = \frac{1}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij} x_j^{(k+1)} + \sum_{j>i} a_{ij} x_j^{(k)}\right)}$$

- Has some weaker convergence conditions (e.g. A SPD)
- **Successive Over-Relaxation/SOR**: Scales step size by ω [$x_{GS}^{(k+1)}$: GS iterate]

$$x^{(k+1)} = (1-\omega)x^{(k)} + \omega x_{GS}^{(k+1)}$$

  - $\omega = 1$ is GS; $\omega < 1$, $1 < \omega < 2$ under-/over-relaxation; $\omega > 2$ is divergence

Convergence for Iterative Methods: For matrix A with eigenvalues $\lambda_1, \ldots, \lambda_n$, define ***spectral radius*** as $\rho(A) = \max\{|\lambda_1|, \ldots, |\lambda_n|\}$

$\rightarrow$ *Theorem*: $\rho(A) < 1$ iff $\lim_{k\to\infty} A^k = 0$

- Iterative methods: $x^{(k+1)} = Bx^{(k)} + g \implies e^{(k+1)} = Be^{(k)} \implies e^{(k)} = B^k e^{(0)}$ [$e^{(k)} = x^{(k)-x^*}$]
  Converges for arbitrary $x^{(0)}$ iff $\lim_{k\to\infty} B^k = 0 \iff \underline{\rho(B) < 1}$