1. Floating-point arithmetic

- **Fixed point representation**:

$$x = \pm(d_1 d_2 ... d_{k-1}.\, d_k ... d_n)_\beta$$

- **Floating-point representation**:

$$x = (0.\, d_1 ... d_{k-1}) \beta^{d_k ... d_n - B}$$

  where B is an **exponent bias**.
- If $d_1 \neq 0$ then the floating point system is **normalised** and each float has a unique representation.
- **binary64**: stored as

$$s e_{10} ... e_0 d_1 ... d_{52}$$

  where s is the **sign** (0 for positive, 1 for negative), $e_{10} ... e_0$ is the **exponent**, and $d_1 ... d_{52}$ is the **mantissa**. The bias is 1023. The number represented is

$$\begin{cases} (-1)^s (1.\, d_1 ... d_{52})_2 2^e \text{ if } e \neq 0 \text{ or } 2047 \\ (-1)^s (0.\, d_1 ... d_{52})_2 2^{-1022} \text{ if } e = 0 \end{cases}$$

  where $e = (e_{10} ... e_0)_2$ $e = 2047$ is used to store NaN, $\pm\infty$. The first case $e \neq 0$ is a **normal** representation, the $e = 0$ case is a **subnormal representation**.
- Floating-point numbers have **finite precision**: exists $\varepsilon_M > 0$ such that $\text{fl}(x) = \text{fl}((1+\varepsilon)x)$ for all $\varepsilon < \varepsilon_M$.
- Floating-point numbers have finite range: exists $m_{\max}$ and $m_{\min}$ such that fl defined only when $m_{\min} \leq |x| \leq m_{\max}$.
- **Underflow**: where floating point calculation result is smaller than smallest representable float. Result is set to zero.
- **Overflow**: where floating point calculation result is larger than largest representable float. **Floating-point exception** is raised.
- **Machine epsilon ε_M**: difference between smallest representable number greater than 1 and 1. $\varepsilon_M = \beta^{-k+1}$.
- $\text{fl}(x)$ maps real numbers to floats.
- **Chopping**: rounds towards zero. Given $x = \left(0.\, d_1 ... d_k d_{k+1} ...\right)_\beta \cdot \beta^e$, if the float has k mantissa digits, then

$$\text{fl}_{\text{chop}}(x) = (0.\, d_1 ... d_k) \cdot \beta^e$$

- **Rounding**: rounds to nearest. Given $x = \left(0.\, d_1 ... d_k d_{k+1} ...\right)_\beta \cdot \beta^e$, if the float has k mantissa digits, then

$$\tilde{\text{fl}}_{\text{round}}(x) = \begin{cases} (0.\, d_1 ... d_k)_\beta \cdot \beta^e & \text{if } \rho < \frac{1}{2} \\ \left((0.\, d_1 ... d_k)_\beta + \beta^{-k}\right) \cdot \beta^e & \text{if } \rho \geq \frac{1}{2} \end{cases}$$

  where $\rho = (0.\, d_{k+1} ...)$.

- **Relative rounding error**:

$$\varepsilon_x = \frac{\mathrm{fl}(x) - x}{x} \Longleftrightarrow \mathrm{fl}(x) = x(1 + \varepsilon_x)$$

- $$\left|\frac{\mathrm{fl}_{\mathrm{chop}} - x}{x}\right| \leq \beta^{-k+1}, \quad \left|\frac{\tilde{\mathrm{fl}}_{\mathrm{round}}(x) - x}{x}\right| \leq \frac{1}{2}\beta^{-k+1}$$

- **Round-to-nearest half-to-even**: fairer rounding than regular rounding for discrete values. In the case of a tie, round to nearest even integer:

$$\mathrm{fl}_{\mathrm{round}}(x) = \begin{cases} (0.\,d_1...d_k)_\beta \cdot \beta^e & \text{if } \rho < \frac{1}{2} \text{ or } \left(\rho = \frac{1}{2} \text{ and } d_k \text{ is even}\right) \\ \left((0.\,d_1...d_k)_\beta + \beta^{-k}\right) \cdot \beta^e & \text{if } \rho > \frac{1}{2} \text{ or } \left(\rho = \frac{1}{2} \text{ and } d_k \text{ is odd}\right) \end{cases}$$

- $x \oplus y = \mathrm{fl}(\mathrm{fl}(x) + \mathrm{fl}(y))$ and similarly for $\otimes$, $\ominus$, $\oplus$.
- Relative error in $x \pm y$ can be large:

$$\mathrm{fl}(x) \pm \mathrm{fl}(y) - (x \pm y) = x(1 + \varepsilon_x) \pm y\left(1 + \varepsilon_y\right) - (x \pm y) = x\varepsilon_x \pm y\varepsilon_y$$

  so relative error is

$$\frac{x\varepsilon_x \pm y\varepsilon_y}{x \pm y}$$

- In general, $x \oplus (y \oplus z) \neq (x \oplus y) \oplus z$
- For some computations, can avoid round-off errors (usually caused by subtraction of numbers close in value) e.g. instead of

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

  compute

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \cdot \frac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}} = \frac{-2c}{b + \sqrt{b^2 - 4ac}}$$

2. Polynomial Interpolation

- $\mathcal{P}_n$ is set of polynomials of degree $\leq n$.
- $\mathrm{conv}\{x_0, ..., x_n\}$ is smallest closed interval containing $\{x_0, ..., x_n\}$.
- **Taylor's theorem**: for function f, if for $t \in \mathcal{P}_n$, $t^{(j)}(x_0) = f^{(j)}(x_0)$ for $j \in \{0, ..., n\}$ then

$$f(x) - t(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)^{n+1}$$

  for some $\xi \in \mathrm{conv}\{x_0, x\}$ (**Lagrange form of remainder**).
- **Polynomial interpolation**: given nodes $\left\{x_j\right\}_{j=0}^{n}$ and function f, there exists unique $p \in \mathcal{P}_n$ such that p interpolates f: $p\left(x_j\right) = f\left(x_j\right)$ for $j \in \{0, ..., n\}$.
- **Cauchy's theorem**: let $p \in P_n$ interpolate f at $\left\{x_j\right\}^{(j=0)^n}$, then

$$\forall x \in \operatorname{conv}\{x_j\}, f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0) \cdots (x - x_n) \quad \text{for some } \xi \in \operatorname{conv}\{x_j\}$$

- **Chebyshev polynomials**:

$$T_n(x) = \cos\left(n \cos^{-1}(x)\right), \quad x \in [-1, 1]$$

- $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$.
- Roots of $T_n(x)$ are $x_j = \cos\left(\pi\left(j + \frac{1}{2}\right) / n\right)$ for $j \in \{0, ..., n-1\}$. Local extrema at $y_j = \cos(j\pi / n)$ for $j \in \{0, ..., n-1\}$.
- Let $\omega_n(x) = (x - x_0) \cdots (x - x_n)$, $\{x_j\}_{j=0}^n \subset [-1, 1]$ (if $\{x_j\} \not\subset [-1, 1]$ so interval is $[a, b]$, then we can map $x_j \to a + \frac{1}{2}\left(x_j + 1\right)(b - a)$). Then $\sup_{x \in [-1,1]} |\omega_n(x)|$ attains its min value iff $\{x_j\}$ are zeros of $T_{n+1}(x)$. Also,

$$2^{-n} \leq \sup_{x \in [-1,1]} |\omega_n(x)| < 2^{n+1}$$

- **Convergence theorem**: let $f \in C^2([-1, 1])$, $\{x_j\}_{j=0}^n$ be zeros of Chebyshev polynomial $T_{n+1}(x)$ and $p_n \in \mathcal{P}_n$ interpolate f at $\{x_j\}$. Then

$$\sup_{x \in (-1,1)} \left|f(x) - p_n(x)\right| \to 0 \quad \text{as } n \to \infty$$

- **Weierstrass' theorem**: let $f \in C^0([a, b])$. $\forall \varepsilon > 0$, exists polynomial p such that

$$\sup_{x \in (a,b)} |f(x) - p(x)| < \varepsilon$$

- **Lagrange construction**: basis polynomials given by

$$L_k(x) = \prod_{j \neq k} \frac{x - x_j}{x_k - x_j}$$

  satisfy $L_k\left(x_j\right) = \delta_{jk}$. Then

$$p(x) = \sum_{k=0}^{n} L_k(x) f(x_k)$$

  interpolates f at $\{x_j\}$.
- **Note**: Lagrange construction not often used due to computational cost and as we have to recompute from scratch if $\{x_j\}$ is extended.
- **Divided difference operator**:

$$[x_j] f := f(x_j)$$

$$[x_j, x_k] f := \frac{[x_j] f - [x_k] f}{x_j - x_k}, \quad [x_k, x_k] f := \lim_{y \to x_k} [x_k, y] = f'(x_k)$$

$$[x_j, ..., x_k, y, z] f := \frac{[x_j, ..., x_k, y] f - [x_j, ..., x_k, z] f}{y - z}$$

  These can be computed incrementally as new nodes are added.
- **Newton construction**: Interpolating polynomial p is

$$p(x) = [x_0]f + (x - x_0)[x_0, x_1]f + (x - x_0)(x - x_1)[x_0, x_1, x_2]f$$
$$+ \cdots + (x - x_0) \cdots (x - x_{n-1})[x_0, ..., x_n]f$$

- **Hermite construction**: for nodes $\{x_j\}_{j=0}^n$, exists unique $p_{2n+1} \in \mathcal{P}_{2n+1}$ that interpolates f and f' at $\{x_j\}$. Can be found using Newton construction, using nodes $(x_0, x_0, x_1, x_1, ..., x_n, x_n)$. Generally, if $p'(x_k) = f'(x_k)$ is needed, include x_k twice. If $p^{(n)}(x_k) = f^{(n)}(x_k)$ is needed, include x_k $n + 1$ times.
- If $y_0, ..., y_k$ is permutation of $x_0, ..., x_k$ then $[y_0, ..., y_k]f = [x_0, ..., x_k]f$.
- Interpolating error is

  $$f(x) - p(x) = (x - x_0) \cdots (x - x_n)[x_0, ..., x_n, x]f$$

  which gives

  $$[x_0, ..., x_{n-1}, x]f = \frac{f^{(n+1)}(\xi)}{(n+1)!}$$
- **Range reduction**: when computing a function e.g. $f(x) = \arctan(x)$, $f(-x) = -f(x)$ and $f(1 / x) = \frac{\pi}{2} - f(x)$ so only need to compute for $x \in [0, 1]$.

3. Root finding

- **Intermediate value theorem**: if f continuous on $[a, b]$ and $f(a) < c < f(b)$ then exists $x \in (a, b)$ such that $f(x) = c$.
- **Bisection**: let $f \in C^0([a_n, b_n])$, $f(a_n)f(b_n) < 0$. Then set $m_n = (a_n + b_n) / 2$ and

  $$(a_{n+1}, b_{n+1}) = \begin{cases} (m_n, b_n) \text{ if } f(a_n)f(m_n) > 0 \\ (a_n, m_n) \text{ if } f(b_n)f(m_n) > 0 \end{cases}$$

  Then:
  - $b_{n+1} - a_{n+1} = \frac{1}{2}(b_n - a_n)$.
  - By intermediate value theorem, exists $p_n \in (a_n, b_n)$ with $f(p_n) = 0$.
  - $|p_n - m_n| \leq 2^{-(n+1)}(b_0 - a_0)$.
- **False position**: same as bisection except set m_n as x intercept of line from $(a_n, f(a_n))$ to $(b_n, f(b_n))$:

  $$m_n = b_n - \frac{f(b_n)}{f(b_n) - f(a_n)}(b_n - a_n)$$
- Bisection and false position are **bracketing methods**. Always work but slow.
- **Fixed-point iteration**: rearrange $f(x_*) = 0$ to $x_* = g(x_*)$ then iterate $x_{n+1} = g(x_n)$.
- f is **Lipschitz continuous** if for some L,

  $$|f(x) - f(y)| \leq L|x - y|$$
- Space of Lipschitz functions on X is $C^{0,1}(X)$.
- Smallest such L is **Lipschitz constant**.
- Every Lipschitz function is continuous.
- Lipschitz constant is bounded by derivative:

$$\sup_{x \neq y} \frac{|f(x) - f(y)|}{|x - y|} \leq \sup_x |f'(x)|$$

- f is **contraction** if Lipschitz constant $L < 1$.
- **Contraction mapping or Banach fixed point theorem**: if g is a contraction and $g(X) \subset X$ (g maps X to itself) then:
  - Exists unique solution $x_* \in X$ to $g(x) = x$ and
  - The fixed point iteration method converges $x_n \to x_*$.
- **Local convergence theorem**: Let $g \in C^1([a, b])$ have fixed point $x_* \in (a, b)$ with $|g'(x_*)| < 1$. Then with x_0 sufficiently close to x_*, fixed point iteration method converges to x_*.
  - If $g'(x_*) > 0$, $x_n \to x_*$ monotonically.
  - If $g'(x_*) < 0$, $x_n - x_*$ alternates in sign.
  - If $|g'(x_*)| > 1$, iteration method almost always diverges.
- $x_n \to x_*$ with **order at least $\alpha > 1$** if

$$\lim_{n \to \infty} \frac{|x_{n+1} - x_*|}{|x_n - x_*|^\alpha} = \lambda < \infty$$

  If $\alpha = 1$, then $\lambda < 1$ is required.
- **Exact order of convergence** of $x_n \to x_*$:

$$\alpha := \sup \left\{ \beta : \lim_{n \to \infty} \frac{|x_{n+1} - x_*|}{|x_n - x_*|^\beta} < \infty \right\}$$

  Limit must be < 1 for $\alpha = 1$.
- Convergence is **superlinear** if $\alpha > 1$, **linear** if $\alpha = 1$ and $\lambda < 1$, **sublinear** otherwise.
- If $g \in C^2$, then with fixed point iteration,

$$\frac{|x_{n+1} - x_*|}{|x_n - x_*|} \to |g'(x_*)| \text{ as } n \to \infty$$

  so $x_n \to x_*$ superlinearly if $g'(x_*) = 0$ and linearly otherwise.
- If $g \in C^N$, fixed point iteration converges with order $N > 1$ iff

$$g'(x_*) = \cdots = g^{(N-1)}(x_*) = 0, \quad g^{(N)}(x_*) \neq 0$$

- **Newton-Raphson**: fixed point iteration with $g(x) = x - f(x) / f'(x)$

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

- For Newton-Raphson, $g'(x_*) = 0$ so quadratic convergence.
- Can use Newton-Raphson to solve $1 / x - b = 0$:

$$x_{n+1} = x_n - \frac{1 / x_n - b}{-1 / x_n^2} = x_n(2 - b x_n)$$

- **Newton-Raphson in d dimensions**:

$$\underline{x}_{n+1} = \underline{x}_n - (Df)^{-1}\left(\underline{x}_n\right)\underline{f}\left(\underline{x}_n\right)$$

where Df is **Jacobian**.

- **Secant method**: approximate $f'(x_n) \approx \frac{f(x_n)-f(x_{n-1})}{x_n - x_{n-1}}$ with Newton-Raphson:

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

4. Numerical differentiation

- **Taylor expansion**:

$$f(x \pm h) = f(x) \pm h f'(x) + \frac{h^2}{2!} f''(x) \pm \frac{h^3}{3!} f'''(x) + \cdots$$

- **Forward difference approximation**:

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2} f''(\xi), \quad \xi \in \operatorname{conv}\{x, x+h\}$$

with $h > 0$.

- **Backward difference approximation**: forward difference but with $h < 0$.
- **Centred difference approximation**:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{12}\left(f'''(\xi\_) + f'''\left(\xi_+\right)\right), \quad \xi_\pm \in [x-h, x+h]$$

- **Richardson extrapolation**: for approximation of $R(x;0)$ of the form

$$R(x;h) = R^{(1)}(x;h) = R(x;0) + a_1(x)h + a_2(x)h^2 + a_3(x)h^3 + \cdots$$

we have

$$R^{(1)}(x;h/2) = R(x;0) + a_1(x)\frac{h}{2} + a_2(x)\frac{h^2}{4} + a_3(x)\frac{h^3}{8} + \cdots$$

This gives **second order approximation**:

$$R^{(2)}(x;h) = 2R^{(1)}(x;h/2) - R^{(1)}(x;h) = R(x;0) - a_2(x)\frac{h^2}{2} + \cdots$$

Similarly,

$$R^{(3)}(x;h) = \frac{4R^{(2)}(x;h/2) - R^{(2)}(x;h)}{3} = R(x;0) + \tilde{a}_3(x)h^3 + \cdots$$

is **third order approximation**. Generally,

$$R^{(n+1)}(x;h) = \frac{2^n R^{(n)}(x;h/2) - R^{(n)}(x;h)}{2^n - 1} = R(x;0) + O\left(h^{n+1}\right)$$

5. Linear systems

- A **symmetric** if $A^T = A$.

- **Hermitian conjugate**: $(A^*)_{ij} = \overline{A_{ji}}$. A **Hermitian** if $A^* = A$.
- A **non-singular** iff $\forall b \in K^n$, exists solution $x \in K^n$ to $Ax = b$ ($K = \mathbb{R}$ or $\mathbb{C}$).
- If A non-singular, exists exactly one solution x to $Ax = b$ and unique A^{-1} such that $\forall b \in K^n, x = A^{-1}b$.
- A non-singular iff $\det(A) \neq 0$.
- A **positive-definite** iff $x \cdot Ax > 0 \ \forall x \neq 0$.
- A **positive-semidefinite** iff $x \cdot Ax \geq 0 \ \forall x \in K^n$.
- L **lower-triangular** iff $L_{ij} = 0$ for $i < j$.
- U **upper-triangular** iff $U_{ij} = 0$ for $i > j$.
- Can solve $Lx = b$ by **forward substitution**: for $j = 1, ..., n$:

$$x_j = \frac{b_j - \sum_{k=1}^{j-1} L_{jk} x_k}{L_{jj}}$$

- Can solve $Ux = b$ by **backward substitution**: for $j = n, ..., 1$:

$$x_j = \frac{b_j - \sum_{k=j+1}^{n} U_{jk} x_k}{U_{jj}}$$

- If A not upper/lower triangular, use **Gaussian elimination** to reduce A to upper triangular U using addition of multiple of row to another row. If leading element in current row is zero, swap with row below.
- **Gaussian elimination with row pivoting**: at sth stage of Gaussian elimination, if largest element in sth column is in row j, swap row j and row s, then proceeed as usual. This gives more accurate results.
- For operation count, assume each arithmetic operation takes one **flop**.
- When asked about **order** of operation count, include **constant multiple** as well as highest power of n.
- ***LU* decomposition**: write $A = LU$, then solve $Ly = b$, then $Ux = y$ with backward/forward substitution. Better when solving with multiple b.
- **Frobenius matrix of index s**: diagonal elements are 1, other elements zero except for s th colum below main diagonal.
- Any Frobenius matrix can be written

$$F_{ij}^{(s)} = \delta_{ij} - f_i^{(s)} e_j^{(s)}$$

  where $e^{(s)}$ is sth unit vector, $f^{(s)} = \left(0, ..., 0, f_{s+1}^{(s)}, ..., f_n^{(s)}\right)$ or

$$F^{(s)} = I - f^{(s)} \otimes e^{(s)}$$

  where $(v \otimes w)_{ij} = v_i w_j$ is tensor product.
- Inverse of Frobenius matrix is Frobenius matrix of same index:

$$G^{(s)} = I + f^{(s)} \otimes e^{(s)}$$

- $G^{(1)} \cdots G^{(s)} = I + \sum_{r=1}^{s} f^{(r)} \otimes e^{(r)}$
- If A can be transform to upper triangular U by Gaussian eliminiation without pivoting, then exists lower triangular L such that $A = LU$. L given by

$$L_{ii} = 1, \quad L_{is} = A_{is}^{(s-1)} \ / \ A_{ss}^{(s-1)}$$

where $A^{(s-1)}$ is matrix at $(s-1)$th stage of Gaussian elimination ($A^0 = A$ is initial matrix).

- Any non-singular A can be written as $PA = LU$ where L is **permutation (pivot) matrix** (each row and column has exactly one 1 and all other elements are 0).
- **Norm** of vector space V: map $\|\cdot\| : V \to \mathbb{R}$ with:
  - **Triangle inequality**: $\|x + y\| \leq \|x\| + \|y\|$.
  - **Linearity**: $\|\alpha x\| = |a|\|x\|$.
  - **Positivity**: $\|x\| \geq 0$ and $\|x\| = 0 \Longrightarrow x = 0$.
- **Seminorm** $|[x]|$: norm except non-zero vectors with $|[x]| = 0$.
- $\boldsymbol{l_p}$ **norm**: for $p \geq 1$,

$$\|x\|_p := \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$

- $\boldsymbol{l_\infty}$ **norm**:

$$\|x\|_\infty := \max_i |x_i|$$

- Matrix **row-sum norm**:

$$\|A\|_{\text{row}} := \max_{i=1,\ldots,n} \sum_{j=1}^{n} |A_{ij}|$$

- Matrix **column-sum norm**:

$$\|A\|_{\text{col}} := \max_{j=1,\ldots,n} \sum_{i=1}^{n} |A_{ij}|$$

- **Frobenius norm**:

$$\|A\|_{\text{Fro}} := \left(\sum_{i,j=1}^{n} |A_{ij}|^2\right)^{1/2}$$

- For n dimensional vector space V, $\text{Hom}(V)$ is vector space of $n \times n$ matrices.
- Given norm $\|\cdot\|$ on V, **induced norm** on $\text{Hom}(V)$ is

$$\|A\| := \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|$$

- Properties of induced norm:
  - $\|Ax\| \leq \|A\|\|x\|$, $x \in V$, $A \in \text{Hom}(V)$.
  - $\|AB\| \leq \|A\|\|B\|$, $A, B \in \text{Hom}(V)$.
- **Spectral radius** of matrix:

$$\rho(A) := \max\{|\lambda| : \lambda \text{ eigenvalue of } A\}$$

- We have these equalities:
  - $\|A\|_1 = \|A\|_{\text{col}}$.
  - $\|A\|_2 = \max\left\{\sqrt{|\lambda|} : \lambda \text{ eigenvalue of } A^T A\right\} = \rho\left(A^T A\right)^{1/2} = \rho\left(AA^T\right)^{1/2}$

- $\|A\|_\infty = \|A\|_{\text{row}}$.
- **Condition number** of A with respect to norm $\|\cdot\|_*$:

$$\kappa_*(A) := \left\|A^{-1}\right\|_* \|A\|_*$$

- If $\|B\| < 1$ for any submultiplicative matrix norm $\|\cdot\|$,

$$B^k \to 0 \quad \text{as } k \to \infty$$

Also,

$$B^k \to 0 \quad \text{as } k \to \infty \Longleftrightarrow \rho(B) < 1$$

- **Richardson's method for lineary systems**: $Ax = b$ so $x = x + w(b - Ax)$ for some w. So iterate

$$x^{(k+1)} = x^{(k)} + w\left(b - Ax^{(k)}\right)$$

Residual: $r^{(k)} := x^{(k)} - x$ satisfies

$$r^{(k+1)} = (I - wA)r^{(k)} \Longrightarrow r^{(k)} = (I - wA)^k r^{(0)}$$

So iteration converges iff $(I - wA)^k \to 0 \Longleftrightarrow \rho(I - wA) < 1$
- **Jacobi's method**: split A into $A = D - E - F$, D diagonal, E strictly lower triangular, F strictly upper triangular. Rewrite $Ax = b$ as $Dx = (E + F)x + b$, and iterate

$$x^{(k+1)} = D^{-1}\left((E + F)x^k + b\right)$$

Residual satisfies $r^{(k+1)} = D^{-1}(E + F)r^{(k)}$ so iteration converges iff $\left(D^{-1}(E + F)\right)^k \to 0$. Converges if A **strictly diagonally dominant** ($|a_{ii}| > \sum_{j \neq i} |a_{ij}|$ for all i).
- **Gauss-Seidel method**: iterate

$$(D - E)x^{(k+1)} = Fx^{(k)} + b$$

Residual satisfies $r^{(k+1)} = (D - E)^{-1} F r^{(k)}$. Converges if A strictly diagonally dominant.

6. L^2 approximations and orthogonal polynomials

- **Inner product over vector space V**: map $(\cdot, \cdot) : V \times V \to \mathbb{C}$ satisfying:
  - $(\alpha u + \beta u', v) = \alpha(u, v) + \beta(u', v)$.
  - $(u, v) = \overline{(v, u)}$.
  - $(u, u) \geq 0$ and $(u, u) = 0 \Longleftrightarrow u = 0$.
- For $V = C^0([a, b])$, define inner product

$$(u, v)_{L^2_w(a,b)} := \int_a^b u(x)v(x)w(x)\,\mathrm{d}x$$

where **weight function** $w(x) > 0$ except at finite set of points. $w(x) = 1$ if not specified.
- Inner product induces norm $\|u\| = \sqrt{(u, u)}$.
- Let V inner product space, X linear subspace of V. Then the $\tilde{p} \in X$ that minimises

$$E(p) = \|f - p\|^2$$

satisfies

$$\forall p \in X, (f - \tilde{p}, p) = 0 \Longleftrightarrow (f, p) = (p, \tilde{p}) \Longleftrightarrow \left(f, \varphi_k\right) = \left(\tilde{p}, \varphi_k\right) \quad \forall k$$

where X spanned by $\left\{\varphi_k\right\}$. So if $\tilde{p} = \tilde{p}_0\varphi_0 + \cdots + \tilde{p}_K\varphi_K$ then

$$\left(f, \varphi_k\right) = \sum_j \left(\varphi_j, \varphi_k\right)\tilde{p}_j$$

- **Gram-Schmidt**: to construct orthogonal basis $\left\{\hat{\varphi}_k\right\}$ from non-orthogonal basis $\left\{\varphi_k\right\}$:
  - $\hat{\varphi}_0 = \varphi_0$.
  - $\hat{\varphi}_k = \varphi_k - \sum_{j=0}^{k-1} \frac{\left(\varphi_k, \hat{\varphi}_j\right)}{\left\|\hat{\varphi}_j\right\|^2}\hat{\varphi}_j$ where norm is respect to given inner product.
- Properties of orthogonal basis:
  - Unique up to normalisation: if $\left\{\varphi_j^*\right\}$ is another orthogonal basis, then $\varphi_j^* = c_j\hat{\varphi}_j$ for some constant c_j.
  - Has exactly k simple roots in (a, b).
- **Recurrence formula** to recursively calculate orthogonal basis:

$$\hat{\varphi}_{k+1} = \frac{1}{\left\|\hat{\varphi}_k\right\|}x\hat{\varphi}_k(x) - \frac{\left(x\hat{\varphi}_k, \hat{\varphi}_k\right)}{\left\|\hat{\varphi}_k\right\|^3}\hat{\varphi}_k(x) - \frac{\left\|\hat{\varphi}_k\right\|}{\left\|\hat{\varphi}_{k-1}\right\|}\hat{\varphi}_{k-1}(x)$$

7. Numerical integration

- Want to approximate

$$I(f) := \int_a^b f(x)w(x)\,\mathrm{d}x$$

  with **quadrature formula**:

$$Q_n(x) = \sum_{k=0}^{n} \hat{\sigma}_k f(x_k)$$

  for **nodes** $\{x_k\}$ and **coefficients** $\{\hat{\sigma}_k\}$.
- Q_n has **degree of exactness** r if $Q_n(x^j) = I(x^j)$ for all $j \leq r$, and $Q_n(x^{r+1}) \neq I(x^{r+1})$.
- By linearity, if Q_n has degree of exactness r, then $Q_n(p) = I(p)$ for all $p \in P_r$.
- **Interpolatory quadrature**: given nodes $\{x_k\}$, find p that interpolates f at nodes, $f(x_k) = p(x_k)$ and find integral of p. E.g. with Lagrange interpolation,

$$I_n(f) := \int_a^b p(x)\,\mathrm{d}x = \sum_{k=0}^{n} f(x_k)\int_a^b L_k(x)$$

  Let $t = (x - a) \,/\, (b - a)$ then

$$\int_a^b L_k(x)\,\mathrm{d}x = (b-a)\int_0^1 \prod_{l\neq k} \frac{t-t_l}{t_k-t_l}\,\mathrm{d}t =: (b-a)\sigma_k$$

so

$$I_n(f) = (b-a)\sum_{k=0}^{n} \sigma_k f(x_k)$$

- Degree of exactness of I_n is n.
- **Newton-Cotes** formula: interpolatory quadrature with equidistant nodes.
- **Closed Newton-Cotes** formula: Newton-Cotes with $x_0 = a$ and $x_n = b$, so $t_k = \frac{k}{n}$.
- If nodes symmetric, $t_{n-k} = 1 - t_k$ then $\sigma_{n-k} = \sigma_k$.
- **Rectangle method**:

$$I_0(f) = (b-a)f\left(\frac{a+b}{2}\right)$$

- If p interpolates f at $\{x_k\} \subset [a,b]$ then for all $x \in [a,b]$,

$$f(x) - p(x) = \frac{\omega_{n+1}(x)}{(n+1)!} f^{(n+1)}(\xi)$$

  where $\omega_{n+1}(x) = (x-x_0)\cdots(x-x_n)$ and $\xi \in (a,b)$.
- $$|I(f) - I_n(f)| \leq \frac{1}{(n+1)!} \max_{\xi\in[a,b]} \left|f^{(n+1)}(\xi)\right| \int_a^b |w_{n+1}(x)|\,\mathrm{d}x$$
- **Composite quadrature**: divide $[a,b]$ into m subintervals $\{[x_{i-1}, x_i]\}_{i=1}^m$ of each length $h = \frac{b-a}{m}$ and apply interpolatory quadrature to each subinterval, then add each of these together.
- **Trapezium rule**: use composite with closed Newton-Cotes formula with $n = 1$: $I_1(f) = (b-a)\frac{f(a)+f(b)}{2}$ to give

$$C_{1,m}(f) = \frac{b-a}{m}\left(\frac{1}{2}f(x_0) + f(x_1) + \cdots + f(x_{m-1}) + \frac{1}{2}f(x_m)\right)$$

- **Simpson's $\frac{1}{3}$ rule**: use composite with closed Newton-Cotes formula with $n = 2$: $I_2(f) = (b-a)\left(\frac{1}{6}f(a) + \frac{2}{3}f\left(\frac{a+b}{2}\right) + \frac{1}{6}f(b)\right)$ to give

$$C_{2,m}(f) = \frac{b-a}{m}\left(\frac{1}{6}f(x_0) + \frac{2}{3}f\left(x_{\frac{1}{2}}\right) + \frac{1}{3}f(x_1) + \cdots + \frac{1}{3}f(x_{m-1}) + \frac{2}{3}f\left(x_{m-\frac{1}{2}}\right) + \frac{1}{6}f(x_m)\right)$$

- To compute error bounds for composite, add individual error bounds for each of the individual quadratures.
- Interpolatory formula

$$G_n = \sum_{k=0}^{n} \rho_k f(x_k)$$

  obtains highest degree of exactness $2n+1$ iff nodes $\{x_k\}$ chosen so that $\hat{p}(x) = (x-x_0)\cdots(x-x_n)$ satisfies

$$\forall p \in P_n, \quad (\hat{p}, p) = 0$$

$\{x_k\}$ must be roots of $\varphi_{n+1} \in P_{n+1}$ where $\left\{\varphi_j\right\}$ are orthogonal polynomials with respect to inner product $(\cdot, \cdot)_{a,b,w}$ Then coefficients given by

$$\rho_k = \int_a^b \prod_{l \neq k} \frac{x - x_l}{x_k - x_l} w(x) \, \mathrm{d}x$$

where w is weight function.