1. Floating-point arithmetic

- **Fixed point representation**:

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

- **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 range and precision.
- **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. $\varepsilon_M = \beta^{-k+1}$.
- $\mathrm{fl}(x)$ 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

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

- **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

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

  where $\rho = \left(0. \, d_{k+1} ...\right)$.
- **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:

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

- $x \oplus y = \mathrm{fl}(\mathrm{fl}(x) + \mathrm{fl}(y))$ 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 $\leq n$.
- $\mathrm{conv}\{x_0, \ldots, x_n\}$ is smallest closed interval containing $\{x_0, \ldots, x_n\}$.
- **Taylor's theorem**: for function f, if for $t \in \mathcal{P}_n$, $t^{(j)}(x_0) = f^{(j)}(x_0)$ for $j \in \{0, \ldots, n\}$ 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 $j \in \{0, \ldots, n\}$.
- **Cauchy's theorem**: let $p \in P_n$ interpolate f at $\left\{x_j\right\}^{(j=0)^n}$, then

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

- **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**:
$$\left[x_j\right] f := f\left(x_j\right)$$
$$\left[x_j, x_k\right] f := \frac{\left[x_j\right] f - [x_k] f}{x_j - x_k}, \quad [x_k, x_k] f := \lim_{y \to x_k} [x_k, y] = f'(x_k)$$
$$\left[x_j, ..., x_k, y, z\right] f := \frac{\left[x_j, ..., x_k, y\right] f - \left[x_j, ..., x_k, z\right] f}{y - z}$$
These can be computed incrementally as new nodes are added.
- **Newton construction**: Interpolating polynomial p is
$$\begin{aligned} p(x) = [x_0] f + (x - x_0)[x_0, x_1] f + (x - x_0)(x - x_1)[x_0, x_1, x_2] f \\ + \cdots + (x - x_0) \cdots (x - x_{n-1})[x_0, ..., x_n] f \end{aligned}$$

- **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,

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

  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,

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