2016-04-26

notessum - Summary

computer arithmetic

- Simplified form for presenting a floating-point number x in **base** b: $x = (f)_b \times b^{(e)_b}$
- $f = \pm(d_1 d_2 \cdots d_t)_b$ **mantissa** (or **significant**); $0 \leq f < 1$.
- $e = \pm(c_{s-1}\, c_{s-2} \cdots c_0)_b$ integer **exponent** (or **characteristic**); $E_{min} \leq e \leq E_{max}$
- A computer in which numbers are represented as above is said to have t base-b digits **precision**. The exponent goveners the **range** of representable numbers.
- *Term*: **normalized** mantissa, **significant** digits, OFL (overflow level, N_{max}), UFL (underflow level, N_{min}), **overflow**, **underflow**
- The real numbers that are exactly representable as floating-point numbers are discrete, belong to $[-N_{max}, -N_{min}] \cup \{0\} \cup [N_{min}, N_{max}]$, and are denser towards 0.
- For the rest of the real numbers:
  - those in $[-N_{max}, -N_{min}] \cup [N_{min}, N_{max}]$ are rounded/chopped (x represented as $fl(x)$);
  - those in $(-\infty, -N_{max}) \cup (N_{max}, \infty)$ overflow;
  - those in $(-N_{min}, 0) \cup (0, N_{min})$ underflow.
- **Relative round-off error (representation error)**: $\delta = \frac{fl(x)-x}{x}$
- Bound for δ: $|\delta| \leq \frac{1}{2} b^{1-t}$, i.e. $fl(x)$ is correct in t significant b-digits.
- *Terms*: **absolute** error, **relative** error
- All computer operations are designed so that **the error of a computation is the round-off error of the correct result.**
- **Machine epsilon** ϵ_{mach} : The smallest (non-normalised) floating-point number with the property $1 + \epsilon_{mach} > 1$.
- Bounds for ϵ_{mach} and δ:
  - $\frac{1}{2} b^{1-t} \leq \epsilon_{mach} \leq b^{1-t}$
  - $-\epsilon_{mach} \leq \delta \leq \epsilon_{mach}$
  - $0 < UFL < \epsilon_{mach} < OFL$.
- **Saturation**: The phenomenon in which a non-zero number is added to another and the latter is left unchanged.
- **Catastrophic cancellation**: The phenomenon in which adding nearly opposite (or subtracting nearly equal) numbers results in having no (or very few) correct digits (i.e. the past errors in the nearly opposite or nearly equal numbers, propagate from the least significant digits to the most significant ones).
- **Condition number** of a function: $\kappa_f(x) = |\frac{xf'(x)}{f(x)}|$; measure of how the error in x propagates in $f(x)$ (error propagation in computation)
  - | relative forward error | ≈ condition number × |relative backward error|
  - $$\frac{y - \hat{y}}{y} \approx \kappa_f \frac{x - \hat{x}}{x}, y = f(x)$$

    - It does not change by re-writing a certain computation in an equivalent way.
- **Stability** refers to the error propagation in a numerical algorithm, i.e. the particular way a certain computation is carried out, and may change if the computation is performed by a different algorithm.

matrices, solving square linear systems

- *Matrix terminology:* triangular (upper/lower), unit triangular (upper/lower), tridiagonal, (*l*,*u*)-banded, permutation, elementary Gauss transformation, orthogonal, positive definite, diagonally dominant, symmetric
- *Important algorithms:* Gauss elimination, Gauss elimination with partial pivoting (applicable to all matrices ,gives L unit lower triangular, U upper triangular and P permutaion, such that $PA = LU$), back substituion, forward substitution
- *Important flops counts:*

matrix	**LU**	**LU piv(part)**	**LU piv compl.**	**f/s**	**b/s**	**sol of m sys.**	**inverse**
general $n \times n$	$\frac{n^3}{3}$	$\frac{n^3}{3}$	$\frac{2n^3}{3}$	$\frac{n^2}{2}$	$\frac{n^2}{2}$	$\frac{n^3}{3} + mn^2$	n^3
(*l*,*u*)-banded	nlu	$2nlu$	-	ln	un	$2nlu + m(l+u)n$	$n^2(l+u)$
symmetric	$\frac{n^3}{6}$	-	-	-	-	-	-

- General $n \times m$ matrix times $m \times k$ matrix (or vector for $k = 1$): nmk

norms, condition numbers

- Definition of a norm: three properties
    - $\|x\| \geq 0, \|x\| = 0$ iff $x = 0 \in S$
    - $\|\alpha x\| = \mid \alpha \mid \|x\|$ for all scalars α
    - $\|x + y\| \leq \|x\| + \|y\|$ (triangle or Minowski's inequality)
- Common *vector norms*:
    - max (infinity) $\|x\|_\infty \equiv \max_{i=1}^n \{\mid x_i \mid\}$
    - Euclidean (2-norm) $\|x\|_2 \equiv \sqrt{(x,x)} \equiv (\sum_{i=1}^n x_i^2)^{1/2}$
    - one-norm $\|x\|_1 \equiv \sum_{i=1}^n \mid x_i \mid$
- For any vector $x \in \mathbb{R}^n$, we have $\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1$.
- Comman *matrix norms*:
    - p-norms $\|A\|_p \equiv \max_{x \neq 0} \{\frac{\|Ax\|_p}{\|x\|_p}\}, p = 1, 2, \infty$
    - row norms $\|A\|_\infty = \max_{i=1}^n \{\sum_{j=1}^n \mid a_{ij} \mid\}$
    - col norms $\|A\|_1 = \max_{j=1}^n \{\sum_{i=1}^n \mid a_{ij} \mid\}$
- For the p-norms, another three properties hold
    - $\|Ax\| \leq \|A\|\|x\|$
    - $\|I\| = 1$
    - $\|AB\| \leq \|A\|\|B\|$
- **Condition number** of a non-singular matrix A: $\kappa_a(A) = \|A\|_a \|A^{-1}\|_a$. It is a measure of the relative sensitivity of the solution x of $Ax = b$ to relative changes in A and b:

- $$\frac{\|x-\hat{x}\|_a}{\|x\|_a} \leq \kappa_a(A)\left(\frac{\|b-\hat{b}\|_a}{\|b\|_a}+\frac{\|A-\hat{A}\|_a}{\|A\|_a}+\frac{\|r\|_a}{\|b\|_a}\right)$$

nonlinear equations -- terms and concepts

- General form: $f(x)=0$
- **Multiplicity** of root: If $f \in \mathbb{C}^m$ and $f(x^*)=0, f'(x^*)=0, \cdots, f^{(m-1)}(x^*)=0$, but $f^{(m)}(x^*) \neq 0$, for some $m \geq 1$, then x^* is a root of multiplicity m.
- General form of a $n \times n$ system of nonlinear equations: $f: \mathbb{R}^n \to \mathbb{R}^n, \bar{f}(\bar{x})=\bar{0}$
- **Jacobian** matrix: $(J(\bar{x}))_{ij} = \frac{df_i}{dx_j}(\bar{x})$
- **Fixed point** x of function $g(x)$: $x=g(x)$.
- **Contraction** in set $S \subset \mathbb{R}^n$: a function $g: \mathbb{R}^n \to \mathbb{R}^n$ for which there is constant λ, with $0 \leq \lambda < 1$, such that
  - $$\|g(x)-g(z)\| \leq \lambda \|x-z\|, \forall x, z \in S. \ (1)$$
- 4 theorems about existence and uniqueness or roots or fixed pts.
  - Theorem 1 (Bolzano) Existence of a root of $f(x)$: If $f(x)$ is continuous in $[a,b], f(a) \cdot f(b) < 0$, then there exists at least one root of $f(x)$ in (a,b).
  - Theorem 2 Existence of a fixed point of $g(x)$: If $g(x)$ is continous $I=[a,b]$, $g(x) \in I, \forall x \in I$ (i.e. $g(x)$ maps I to itself), then there exists at least one fixed point of $g(x)$ in I.
  - Theorem 3 Uniqueness of a root of $f(x)$: If $f(x)$ is differentiable in $I=(a,b), f'(x) \neq 0, \forall x \in I$, there exists a root of $f(x)$ in I, then the root is unique in I.
  - Theorem 4 Existence and uniqueness of a fixed point of $g(x)$: If g is contraction in $I=[a,b]$, $g(x) \in I, \forall x \in I$ (i.e. $g(x)$ maps I to itself), then there exists a unique fixed point x^* of $g(x)$ in I.

numerical methods for solving nonlinear equations

- General nonlinear solver
  - guess $x^{(0)}$ (and possibly $x^{(-1)}$ or more)
  - for $k=1, \cdots,$ maxit
    - compute $x^{(k)}$ using previous approximations and information from f
    - if stopping criterion satisfied exit, endif
  - endfor

rate of convergence of sequences / iterative methods

- Sequence $x^{(0)}, x^{(1)}, \cdots$ converges to x^*, with rate of convergence β and asymptotic error constant C:
  - $$\lim_{k \to \infty} \frac{\|x^{(k+1)}-x^*\|}{\|x^{(k)}-x^*\|^\beta} = C$$
- See Theorem 6 about rate of convergence of fixed-pt iteration methods.

numerical methods for solving nonlinear equations

- *Bisection* method

    - Bisection is applicable when f is continuous and there are two points L and R ($L < R$), such that $f(L)f(R) < 0$. Bisection chooses $M = L + (R - L)/2$ as next approximation and continuous to $[L, M]$ if $f(L)f(M) < 0$, or to $[M, R]$ if $f(M)f(R) < 0$. Bisection always converges (when applicable) and is of first order. Requires one function evaluation per iteration.
- *Newton's* method
    - Newton's is applicable if f is differentiable and $f'(x^{(k)}) \neq 0$:
        - $$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}$$
    - Converges if f is twice differentiable and if $x^{(0)}$ *close enough* to root. When it converges, it is usually of second order. Requires one function and one derivative evaluation per iteration.
- General form of *fixed-point iteration*
    - Assume we have found (or been given) function $g(x)$ such that
        - $$f(x) = 0 \iff x = g(x)$$
    - Then, the associated fixed-point iteration method computes
        - $$x^{(k+1)} = g(x^{(k)})$$
    - Newton's is a fixed-point iteration method with $g(x) = x - f(x)/f'(x)$.
    - 3 theorems convergence (and rate) of fixed-point iteration methods
        - Theorem 4b (extension of Thm 4) Convergence of fixed-point iteration $x^{(k+1)} = g(x^{(k)})$: If g is continuous in $I = [a, b]$ with constant λ, $g(x) \in I, \forall x \in I$, then there exists a unique fixed point x^* of g in I, $\forall x^{(0)} \in I$, the iteration scheme $x^{(k+1)} = g(x^{(k)})$ converges to $x^*, \mid x^{(k)} - x^* \mid \leq \lambda^k \mid x^{(0)} - x^* \mid \leq \lambda^k \max\{x^{(0)} - a, b - x^{(0)}\}$
        - Theorem 5 Convergence of fixed-point iteration $x^{(k+1)} = g(x^{(k)})$: If $g(x)$ has a fixed point x^*, $g(x)$ has continuous derivative in an open interval containing x^*, $|g'(x^*)| < 1$, then there exists an open interval I that contains x^*, i.e. $I = (x^* - r^*, x^* + r^*), r^* > 0$, such that $\forall x^{(0)} \in I$, the iteration scheme $x^{(k+1)} = g(x^{(k)})$ converges to x^*
        - Theorem 6: If g has a fixed point α, $x^{(k+1)} = g(x^{(k)})$ converges to α, $g \in \mathbb{C}^\beta$ near α, $g'(\alpha) = g''(\alpha) = \cdots = g^{(\beta-1)}(\alpha) = 0$ and $g^{(\beta)}(\alpha) \neq 0$, then the rate of convergence of $x^{(k+1)} = g(x^{(k)}$ is β and the asymptotic error constant is $\frac{1}{\beta!} \mid g^{(\beta)}(\alpha)|$
- *Secant* method
    - Secant is applicable if f is continuous and $f(x^{(k)}) \neq f(x^{(k-1)})$:
        - $$x^{(k+1)} = x^{(k)} - f(x^{(k)}) \frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})}$$
    - Secant does not always converge; when it converges it is of order approximately 1.6. Requires one function evaluation per iteration.
- *Newton's* for systems
    - Newton's is applicable if all components of f are differentiable (w.r.t. each variable) and $J(\bar{x}^{(k)})$ is nonsingular:
        - $$\bar{x}^{(k+1)} = \bar{x}^{(k)} - J^{-1}(\bar{x}^{(k)})\bar{f}(\bar{x}^{(k)})$$

polynomial interpolation

- *Generic interpolation problem*: Given data $(x_i, y_i), i = 0, \cdots, n$, find a function (possibly a polynomial) $p(x)$, that interpolates the data, i.e. $p(x_i) = y_i, i = 0, \cdots, n$.
- *Existence and uniqueness theorem*: Given data $(x_i, y_i), i = 0, \cdots, n$, with x_i distinct, there exists a unique polynomial $p_n(x)$ of degree n or less that interpolates the data.
- *Polynomial interpolation error formula*: Let $p_n(x)$ be the polynomial of degree n or less that interpolates f at $x_i, i = 0, \cdots, n$. If f has $n + 1$ continuous derivatives, then, for any x,
  - $$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \Pi_{i=0}^{n}(x - x_i),$$
  - for some $\xi \in \text{ospr}\{x, x_0, x_1, \cdots, x_n\}$, that depends on x.
  - Notes: If $a = \min_i\{x_i\}$ and $b = \max_i\{x_i\}$, then $\text{ospr}\{x_0, x_1, \cdots, x_n\} = (a, b)$.
  - If $a \leq x \leq b$, $\text{ospr}\{x, x_0, x_1, \cdots, x_n\} = (a, b)$.
  - If $x < a$, $\text{ospr}\{x, x_0, x_1, \cdots, x_n\} = (x, b)$.
  - If $b < x$, $\text{ospr}\{x, x_0, x_1, \cdots, x_n\} = (a, x)$.

construction of polynomial interpolants

- *Properly posed polynomial interpolation problem*: Given data $(x_i, y_i), i = 0, \cdots, n$, with x_i distinct, find the (unique) polynomial $p_n(x)$ of degree n or less that interpolates the data.
- Three ways of constructing:
  - (1) Monomials, Vandermonde matrix: easy for small n, very inefficient and inaccurate (unstable) for large n, construction requires the solution of an $(n + 1) \times (n + 1)$ dense and ill-conditioned linear system, easy to evaluate (Horner's), differentiate and intergrate.
  - (2) Lagrange basis functions, explicit formula: often convenient for mathematical manipulation, easy to construct, not very stable, reasonably easy to evaluate, hard to differentiate and integrate.
  - (3) Newton's basis functions, recursive algorithm, NDD table: reasonably efficient to construct, stable, easy to evaluate (Horner's), relatively hard to differentiate and integrate, adding data is easy.

problems with polynomial interpolation

- Polynomial interpolants of high degree
  - oscillate a lot close to the endpoints;
  - lead to ill-conditioned computations;
  - are sensitive to small perturbations of the coefficients;
  - are costly to construct (NDD table takes $O(n^2)$ time for n data points) and evaluate (the evaluation on one point takes $O(n)$ time, while the number of evaluation points is often much larger than the number of data points).
- Polynomial interpolants do not guarantee that the error decreases as the number of data increases.
- Chebyshev data points minimize the $\Pi_{i=0}^{n}(x - x_i)$ part of the error among all choices of $n + 1$ data points in a given interval, and partly only address the oscillatory behaviour problem.
- Piecewise polynomial interpolation is a good alternative, that resolves all the above problems.

piecewise polynomial interpolation

- Piecewise polynomial of degree k defined w.r.t. knots $x_i, i = 0, \cdots, n$:

- $$p(x) = \begin{cases} a_{01} + a_{11}x + a_{21}x^2 + \cdots + a_{k1}x^k \text{ for } x_0 \leq x < x_1 \\ a_{02} + a_{12}x + a_{22}x^2 + \cdots + a_{k2}x^k \text{ for } x_1 \leq x < x_2 \\ \cdots \cdots \\ a_{0n} + a_{1n}x + a_{2n}x^2 + \cdots + a_{kn}x^k \text{ for } x_{n-1} \leq x < x_n \end{cases}$$

- There are $n(k+1)$ coefficients -- free parameters, degrees of freedom -- in the representation of p. Piecewise polynomials may not be continuous or may not have continuous derivatives of some order on the knots.
- By imposing continuity conditions on the interior knots, some coefficients are no longer free parameters.
- If we impose continuity conditions up to derivatives of order $k-1$, we get splines.
- Splines have $n(k+1) - (n-1)k = n + k$coefficients -- free parameters, degrees of freedom.
- Example: **Linear splines** are piecewise polynomial of degree 1 which are $\mathbb{C}^0$ continuous on the knots $x_i, i = 1, \cdots, n-1$, and have $n+1$ degrees of freedom.
- Example: **Cubic splines** are piecewise polynomial of degree 3 which are $\mathbb{C}^2$ continous on the knots $x_i, i = 1, \cdots, n-1$, and have $n+3$ degrees of freedom.

linear splines

- A **linear splines** $L(x)$ takes the form:
  - $$L(x) = \begin{cases} L_1(x) & \equiv a_{01} + a_{11}x \text{ for } x_0 \leq x < x_1 \\ L_2(x) & \equiv a_{02} + a_{12}x \text{ for } x_1 \leq x < x_2 \\ & \cdots \cdots \\ L_n(x) & \equiv a_{0n} + a_{1n}x \text{ for } x_{n-1} \leq x < x_n \end{cases}$$
  - and satisfies the $n-1$ *continuity conditions*
  - $$L_i(x_i) = L_{i+1}(x_i), i = 1, \cdots, n-1. (A)$$
- The **linear spline interpolant** $L(x)$ of data $(x_i, y_i), i = 0, \cdots, n$, is the linear spline (i.e. a pp of degree 1 satisfying (A)), satisfying the *interpolation conditions*
  - $$L(x_i) = y_i, i = 0, \cdots, n. (B)$$
- The linear spline interpolant $L(x)$ of data $(x_i, y_i), i = 0, \cdots, n$, is given by
  - $$L(x) = \left\{ y_{i-1} \frac{x - x_i}{x_{i-1} - x_i} + y_i \frac{x - x_{i-1}}{x_i - x_{i-1}} \right. \text{ for } x_{i-1} \leq x \leq x_i, i = 1, \cdots, n. (C)$$
- Thus, $L(x)$ is given by a ready-to-evaluate formula. (No construction cost.) The evaluation of a linear spline requires $O(1)$ time for one evaluation point.

cubic splines

- A **cubic spline** $C(x)$ takes the form:
  - $$C(x) = \begin{cases} C_1(x) & \equiv a_{01} + a_{11}x + a_{21}x^2 + a_{31}x^3 \text{ for } x_0 \leq x < x_1 \\ C_2(x) & \equiv a_{02} + a_{12}x + a_{22}x^2 + a_{32}x^3 \text{ for } x_1 \leq x < x_2 \\ & \cdots \cdots \\ C_n(x) & \equiv a_{0n} + a_{1n}x + a_{2n}x^2 + a_{3n}x^3 \text{ for } x_{n-1} \leq x < x_n \end{cases}$$

    - and satisfies the $3(n-1)$ *continuity conditions*
    - $$C_i(x_i) = C_{i+1}(x_i), i = 1, \cdots, n-1, (0a)$$
      $$C_i'(x_i) = C_{i+1}'(x_i), i = 1, \cdots, n-1, (0b)$$
      $$C_i''(x_i) = C_{i+1}''(x_i), i = 1, \cdots, n-1, (0c)$$
- A **cubic spline interpolant** $C(x)$ of data $(x_i, y_i, i = 0, \cdots, n,$ is a cubic spline (i.e. a pp of degree 3 satisfying (0a)-(0b)-(0c)), satisfying the *interpolation conditions*
    - $$C(x_i) = y_i, i = 0, \cdots, n,$$
    - and a set of two other conditions, referred to as *end-conditions.*
- Typical sets of end-conditions
    - *Clamped (derivative) end-contions:*
        - $$C'(x_i) = y_i', i = 0, n, (2a)$$
        - if $y_i', i = 0, n,$ are given.
    - *Approximate clamped (derivative) end-conditions:*
        - $$C'(x_i) = \text{ approximation to } y_i', i = 0, n, (2b)$$
        - if $y_i', i = 0, n,$ are not given (the approximations to y_i' can be formed by cubic polynomial interpolation based on 4 endpoint values).
    - *Natural end-conditions:*
        - $$C''(x_i) = 0, i = 0, n. (2c)$$
    - *Not-a-knot end-conditions:*
        - $$C''' \text{ continuous on } x_i, i = 1, n-1. (2d)$$
    - These are equivalently written as
        - $$C_i'''(x_i) = C_{i+1}'''(x_i), i = 1, n-1.$$
    - Note: only **one** of these sets (pairs) of end-conditions is satisfied.
- Note: There is no ready-to-evaluate (no cost) formula for cubic spline interpolants, such as (C) for linear spline interpolants.
- It can be shown that the computation to construct a cubic spline interpolant (i.e. to compute the coefficients $a_{ij}, j = 0, 1, 2, 3, i = 1, \cdots, n,$ of $C(x)$) given data $(x_i, y_i), i = 0, \cdots, n,$ is equivalent to solving an $(n+1) \times (n+1)$ tridiagonal symmetric linear system, i.e. $O(n)$. This linear system is usually well-conditioned.
- The evaluation of a cubic spline requires $O(1)$ time for one evluation point.

spine interpolation error bounds

- Linear spline interpolant error bound: Let $L(x)$ be the linear spline w.r.t $x_i, i = 0, \cdots n,$ interpolating f at $x_i, i = 0, \cdots, n.$ If $f(x) \in \mathbb{C}^2$, then for $x \in [x_0, x_n]$,

- $$|f(x) - L(x)| \leq \frac{1}{8} \max_{x_0 \leq x \leq x_n} |f^{(2)}(x)| \max_{i=1}^{n} (x_i - x_{i-1})^2$$

.

- The linear spline error decreases by a factor of approximately 4, when n doubles. Linear spline interpolation is of order 2.
- Cubic spline interpolant error bound: Let $C(x)$ be the (clamped) cubic spline w.r.t. $x_i, i = 0, \cdots, n$, interpolating f at $x_i, i = 0, \cdots, n$. If $f(x) \in \mathbb{C}^4$, then for $x \in [x_0, x_n]$,
  - $$|f(x) - C(x)| \leq \frac{5}{384} \max_{x_0 \leq x \leq x_n} |f^{(4)}| \max_{i=1}^{n} (x_i - x_{i-1})^4.$$
- The cubic spline error decreases by a factor of approximately 16, when n doubles. Cubic spline interpolation is of order 4.
- Piecewise polynomial interpolants guarantee that the error decreases as the number of data increases.
- Piecewise polynomial interpolants do not (normally) suffer from oscillatory behaviour.
- Notice the different use of term "order" (and "rate") for convergence of spline interpolants and for convergence of iterative methods for nonlinear equations.