MA2213 Numerical Analysis I

Single-precision floating point format: 1/8/23 (sign/exponent/mantissa)

Sign bit is 1=negative, 0=positive. Actual exponent is obtained by subtracting 127.

Denormalized numbers: exponent=0 is denormalized. Then the actual exponent is -126, and implicit digit is 0 instead of 1. There are 2^24-1 different denormalized values, because +0 and -0 are the same meaning.

Largest number: exponent = 0b11111110 (actual exponent is 127), mantissa = 2^23-1; because exponent=0b11111111 is special meaning (inf and nan, both have +ve and -ve versions). X/0b11111111/0 == inf, X/0b11111111/!0 == nan

Significant digits: p^* approximates p to t significant digits in base b if t is the largest non-negative integer such that $\frac{|p-p^*|}{|p|} \leq 0.5 \times b^{1-t}$

Number of decimal significant digits in decimal representation of float: $0.5 \times 2^{1-24} \approx 5.96 \times 10^{-8} < 0.5 \times 10^{1-7}$ hence there are 7 significant digits.

In practice, just round number after t digits.

In general, a number expressed as single-precision float will be approximated to 24 sig digits in binary (unless it is too big->inf or too small->denormalized).

Three ways to measure error: actual error = $p - p^*$; absolute error = $|p - p^*|$; relative error = $\frac{|p-p^*|}{|p|}$ (where $p \neq 0$). p is the actual value and p^* is the approximation.

Finite-digit arithmetic: Using three-digit arithmetic -> three significant digits. After every operation, round to three significant digits.

In general, it is better to sum up numbers with closer magnitude first. Kahan's algorithm calculates the sum of a list of numbers more accurately.

Algorithms and numerical error: Numerical error = round-off error + truncation error (when truncating a series)

When doing summation of (infinite) series, we can split the terms into separate series so that we don't repeat too many computations.

If $g(x) \coloneqq a_0 + \cfrac{b_1}{a_1+\cfrac{b_2}{a_2+\cfrac{b_3}{a_3+\ddots}}}$ then $P_0 = 1$ and $Q_0 = 0$ and $P_1 = a_0$ and $Q_1 = 1$ and $P_k = a_{k-1}P_{k-1} + b_{k-1}P_{k-2}$ and $Q_k = a_{k-1}Q_{k-1} + b_{k-1}Q_{k-2}$ and kth approx. is $\frac{P_k}{Q_k}$.

Some summations:

$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \frac{n^2}{2} + \frac{n}{2}$ $\quad$ $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} = \frac{n^3}{3} + \frac{n^2}{2} + \frac{n}{6}$ $\quad$ $\sum_{i=1}^{n} i^3 = \left(\frac{n(n+1)}{2}\right)^2 = \frac{n^2(n+1)^2}{4} = \frac{n^4}{4} + \frac{n^3}{2} + \frac{n^2}{4}$

Determinant as area of unit object after transformation.

Cramer's rule: If $A \in M_{n\times n}$ is invertible then the unique solution of Ax=b is $x_i = \frac{det\, A_i}{det\, A}$ for i=1,2,…,n; where A_i is the matrix obtained by replacing the ith column of A with the column vector b.

$$det\begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc; det\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} = aei + bfg + cdh - afh - bdi - ceg$$

Gaussian elimination. O(n^3). Have to pick a pivot row whose element is nonzero. Swapping can be O(1) if we use an additional layer of indirection (i.e. maintain an array of indices).

Cramer's rule vs Gaussian elimination: Difference is just that n-digit arithmetic may produce different results. For 2x2 matrices, the Gaussian elimination is usually more accurate (lesser residual) than Cramer's rule. Both are O(n^3) but Gaussian elim has lower constant factor.

(n-1)*n*(n+1)/3 subs/mults in Gaussian elim; (n-1)*n/2 divs in Gaussian elim.

Pivoting strategies for gaussian elimination (because it is dangerous to check whether a float equals to zero):

- Partial pivoting: Find the entry with largest magnitude in the pivot column.
- Complete pivoting: Find the entry with largest magnitude in the whole matrix, and eliminate that column first.
- Scaled partial pivoting: Find the entry with largest (<magnitude of entry>/<largest magnitude in whole row>). Takes extra O(n) divisions per pivot, and O(n^2) comparisons per pivot, so the total is O(n^2) divisions and O(n^3) comparisons.
- Scaled partial pivoting, but only scale once: Before running gaussian elimination, determine the scaling factor of each row (i.e. largest element in the row) once, and use those factors for the whole gaussian elimination algorithm. O(n^2) additional operations, but might have greater error.

Gaussian elimination to find the determinant – after getting an upper triangular matrix, the determinant is the product of the diagonal entries. In any upper or lower triangular matrix, the determinant is the product of the diagonal entries.

LU Matrix factorisation: Two row operations:

- $E_i \longleftarrow E_i - m_{ij}E_j$ (multiply the jth row by m_{ij} (ratio of values of pivot column in ith row to jth row) and subtract the result from the ith row, and $i > j$)
  - Equivalent to left-multiplying $\left(I - m_{ij}\mathbf{e}_i\mathbf{e}_j^T\right)$ to the augmented matrix
    - Lower-triangular matrix with all diagonal entries equal 1, and all non-diagonal entries equal 0 except its (i,j)-element is $-m_{ij}$.
- $E_i \longleftrightarrow E_j$ (exchange ith row with jth row)
  - Equivalent to left-multiplying $\left(I - \left(\mathbf{e}_i - \mathbf{e}_j\right)\left(\mathbf{e}_i - \mathbf{e}_j\right)^T\right)$ to the augmented matrix
    - Symmetric matrix with diagonal entries equal 1, except (i,i)-element and (j,j)-element equal 0; all non-diagonal entries equal 0, except (i,j)-element and (j,i)-element equal 1
- …where $\mathbf{e}_i$ is the vector $(0,\dots,0,1,0,\dots,0)^T$ whose ith entry is nonzero

Theorem 1: From the same row E_j, can eliminate pivot element in all subsequent rows E_i with a single matrix:
Suppose $j \in [1, n-1]$ and $i_1, i_2, \dots, i_k \in [j+1, n]$, then $\prod_{r=1}^{k}\left(I - m_{i_r j}\mathbf{e}_{i_r}\mathbf{e}_j^T\right) = I - \left(\sum_{r=1}^{k} m_{i_r j}\mathbf{e}_{i_r}\right)\mathbf{e}_j^T$.
Let $\mathbf{m}_j \coloneqq \left(0,\dots 0, m_{j+1,j}, \dots, m_{nj}\right)^T = \sum_{r=1}^{k} m_{i_r j}\mathbf{e}_{i_r}$. Hence $\left(\prod_{j=n-1}^{1}\left(I - \mathbf{m}_j\mathbf{e}_j^T\right)\right)A = U$. (A is the original matrix, U is the row-echelon form.)

Theorem 2: Can reverse the multi-row elimination operation:
Suppose $j < n$, and $\mathbf{m}_j \coloneqq \left(0,\dots 0, m_{j+1,j}, \dots, m_{nj}\right)^T = \sum_{r=1}^{k} m_{i_r}\mathbf{e}_{i_r}$ then $\left(I - \mathbf{m}_j\mathbf{e}_j^T\right)^{-1} = \left(I + \mathbf{m}_j\mathbf{e}_j^T\right)$.
Hence $A = \left(\prod_{j=1}^{n-1}\left(I + \mathbf{m}_j\mathbf{e}_j^T\right)\right)U$.

Theorem 3: Suppose $i_1 \le i_2 \le \cdots \le i_k < n$ and for any $j = 1, \dots, k$, $\mathbf{m}_j$ is an n-dimensional column vector whose first i_j components are zero. Then $\prod_{j=1}^{k}\left(I + \mathbf{m}_j\mathbf{e}_{i_j}^T\right) = I + \sum_{j=1}^{k}\mathbf{m}_j\mathbf{e}_{i_j}^T$. Hence $A = \left(I + \sum_{j=1}^{k}\mathbf{m}_j\mathbf{e}_{i_j}^T\right)U$.

Theorem 4: If a linear system with coefficient matrix A can be solved by gaussian elimination without pivoting, then there exists a unique unit lower-triangular matrix (i.e. main diagonal has all 1) L such that $U = L^{-1}A$ is an upper-triangular matrix.
Actually $L = \left(I + \sum_{j=1}^{k}\mathbf{m}_j\mathbf{e}_{i_j}^T\right)$.

After computing U and L in O(n^3), we can solve Ax=b in O(n^2) by solving Ly=b and Ux=y (it is O(n^2) to do backward substitution on a triangular matrix).

Lagrange interpolation: Finding the unique polynomial of degree≤n that passes through n+1 points:

$$\left(x_0, f(x_0)\right), \left(x_1, f(x_1)\right), \dots, \left(x_n, f(x_n)\right)$$

To find the polynomial, we solve for the coefficients $a_0, a_1, \dots, a_n$ in $\begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{pmatrix}\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}$.

The matrix in the expression above is a Vandermonde matrix. Vandermonde matrix with distinct $x_0, x_1, \dots, x_n$ are linearly independent (and have determinant = $\prod_{0 \le i < j \le n}\left(x_j - x_i\right)$), hence the solution exists and is unique.

Lagrange basis polynomial: $L_k(x) = \frac{(x-x_0)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}$. At $x = x_k$, $L_k(x_k) = 1$; at all other x_i, $L_k(x_i) = 0$.
Then the unique polynomial is $P_n(x) \coloneqq y_0L_0(x) + y_1L_1(x) + \cdots + y_nL_n(x)$.

Error analysis of Lagrange interpolating polynomial:

Thm: $\exists \xi \in (\min\{x, x_0, x_1, \dots, x_n\}, \max\{x, x_0, x_1, \dots, x_n\})$ s.t. $f(x) = P_n(x) + \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)(x-x_1)\dots(x-x_n)$.
To find the maximum error (as a function of x), find the maximum value of $\left|f^{(n+1)}(\xi)\right|$ in the required interval.

Divided differences: 0th divided difference: $f[x] \coloneqq f(x)$. nth divided difference: $f[x_0, \dots, x_n] \coloneqq \frac{f[x_1,\dots,f_n]-f[x_0,\dots,x_{n-1}]}{x_n-x_0}$.
(Note: $f[x_0, \dots, x_n]$ is independent of the order of $x_0, \dots, x_n$)
Then $P_n(x) = f[x_0] + f[x_0, x_1](x-x_0) + \cdots + f[x_0, x_1, \dots, x_n](x-x_0)(x-x_1)\dots(x-x_{n-1})$.

(Modified) Horner's method:

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

(i.e. can evaluate the polynomial at any x in O(n))

Runge's phenomenon: When using more points for Lagrange interpolation yields worse results.
Runge's function: $f(x) = \frac{1}{1+25x^2}$ (in general, when $\left|f^{(n)}(x)\right|$ increases rapidly as n increases, Lagrange interpolation might not work)

Cubic spline interpolation: Given $x_0, x_1, \dots, x_n$ and $f(x_0), f(x_1), \dots, f(x_n)$, want to find $S(x)$

- $S(x_k) = f(x_k)$ for all k = 0,1,…,n
- $S(x)$ is piecewise cubic in each range $[x_{k-1}, x_k]$
- $S(x)$ is second-order differentiable
- Either $S''(x_0) = S''(x_n) = 0$ (natural) or $S'(x_0) = f'(x_0), S'(x_n) = f'(x_n)$ (clamped)

Then $S_k(x) = M_{k-1}\frac{(x-x_k)^3}{6(x_{k-1}-x_k)} + M_k\frac{(x-x_{k-1})^3}{6(x_k-x_{k-1})} + A_kx + B_k$ (since $S_k''(x)$ is linear, and $M_k \coloneqq S''(x_k)$)
where $A_k = \frac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}} - \frac{1}{6}(M_k - M_{k-1})(x_k - x_{k-1})$ and $B_k = \frac{x_kf(x_{k-1})-x_{k-1}f(x_k)}{x_k-x_{k-1}} + \frac{1}{6}(M_kx_{k-1} - M_{k-1}x_k)(x_k - x_{k-1})$
For natural boundary conditions, $M_1, M_2, \dots, M_{n-1}$ is the solution for $\begin{pmatrix} 2 & \lambda_1 & & \\ \mu_2 & 2 & \lambda_2 & \\ & \mu_3 & \ddots & \lambda_{n-2} \\ & & \mu_{n-1} & 2 \end{pmatrix}\begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-1} \end{pmatrix} =$
$\begin{pmatrix} 6f[x_0, x_1, x_2] \\ 6f[x_1, x_2, x_3] \\ \vdots \\ 6f[x_{n-2}, x_{n-1}, x_n] \end{pmatrix}$, and $M_0 = M_n = 0$.
For clamped boundary conditions, $M_0, M_1, \dots, M_n$ is the solution for $\begin{pmatrix} 2 & \lambda_0 & & \\ \mu_1 & 2 & \lambda_1 & \\ & \mu_2 & \ddots & \lambda_{n-1} \\ & & \mu_n & 2 \end{pmatrix}\begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ M_n \end{pmatrix} = \begin{pmatrix} 6f[x_0, x_0, x_1] \\ 6f[x_0, x_1, x_2] \\ \vdots \\ 6f[x_{n-1}, x_n, x_n] \end{pmatrix}$,
where $f[x_0, x_0] = f'(x_0)$, $\lambda_0 = \mu_n = 1$, and $\mu_k = \frac{x_k-x_{k-1}}{x_{k+1}-x_{k-1}}$ and $\lambda_k = \frac{x_{k+1}-x_k}{x_{k+1}-x_{k-1}}$.

Least squares approximation: Given m points, find a polynomial $p_n(x)$ minimising $\sum_{i=0}^{m}(p_n(x_i) - y_i)^2$.
(equivalently, want to find $\mathbf{a} = (a_0, \dots, a_n)^T$ minimising $(X\mathbf{a} - \mathbf{y})^T(X\mathbf{a} - \mathbf{y})$ where $X = \begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & \cdots & x_m^n \end{pmatrix}$.)

Thm: $\mathbf{a}$ minimises $(X\mathbf{a} - \mathbf{y})^T(X\mathbf{a} - \mathbf{y}) \iff (X^TX)\mathbf{a} = (X^T\mathbf{y})$ (called the *normal equation*).

Weighted least squares approximation: Given m points, find a polynomial $p_n(x)$ minimising $\sum_{i=0}^{m} w_i(p_n(x_i) - y_i)^2$.
(equivalently, want to minimise $(X\mathbf{a} - \mathbf{y})^TW(X\mathbf{a} - \mathbf{y})$ where $W \coloneqq \text{diag}\{w_0, \dots, w_n\}$).
To do so, minimise $\left(\sqrt{W}X\mathbf{a} - \sqrt{W}\mathbf{y}\right)^T\left(\sqrt{W}X\mathbf{a} - \sqrt{W}\mathbf{y}\right)$ where $\sqrt{W} \coloneqq \text{diag}\{\sqrt{w_0}, \dots, \sqrt{w_n}\}$;
i.e. solve $\left(\tilde{X}^T\tilde{X}\right)\mathbf{a} = \left(\tilde{X}^T\tilde{\mathbf{y}}\right)$ where $\tilde{X} = \sqrt{W}X$ and $\tilde{\mathbf{y}} = \sqrt{W}\mathbf{y}$.

Trapezoidal rule: $\int_a^b f(x)dx = \frac{b-a}{2}\big(f(a) + f(b)\big) - \frac{1}{12}(b-a)^3f''(\xi)$ for some $\xi \in [a, b]$.

Simpson's rule: $\int_a^b f(x)dx = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right) - \frac{1}{90}\left(\frac{b-a}{2}\right)^5 f^{(4)}(\xi)$ for some $\xi \in [a,b]$.

Newton-Cotes formulas:

For odd n, $\int_a^b f(x)dx = \sum_{k=0}^n w_k f(x_k) + \frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!}\int_0^n s(s-1)\dots(s-n)ds$, where $h \coloneqq \frac{b-a}{n} = x_k - x_{k-1}$;

For even n, $\int_a^b f(x)dx = \sum_{k=0}^n w_k f(x_k) + \frac{h^{n+3} f^{(n+2)}(\xi)}{(n+2)!}\int_0^n s^2(s-1)\dots(s-n)ds$, where $h \coloneqq \frac{b-a}{n} = x_k - x_{k-1}$;

where $w_k \coloneqq \int_a^b L_k(x)dx = \frac{b-a}{n}\int_0^n \prod_{j\in\{0,\dots,k-1,k+1,\dots,n\}} \frac{x-j}{k-j}dx$.

Trapezoidal rule & Simpson's rule are special cases of Newton-Cotes formulas (where n=1 & n=2 respectively).

Degree of accuracy: The integer n s.t. all polynomials with degree≤n will have exact numerical integration.
Trapezoidal rule = 1, Simpson's rule = 3, Newton-Cotes (odd n) = n, Newton-Cotes (even n) = n+1

Composite trapezoidal rule: $\int_a^b f(x)dx = h\left(\frac{1}{2}f(x_0) + \sum_{k=1}^{n-1} f(x_k) + \frac{1}{2}f(x_n)\right) - \frac{h^3}{12}\sum_{k=1}^n f''(\xi_k)$

If $f \in C^2$ (on $[a,b]$) then $\int_a^b f(x)dx = h\left(\frac{1}{2}f(x_0) + \sum_{k=1}^{n-1} f(x_k) + \frac{1}{2}f(x_n)\right) - \frac{(b-a)h^2}{12}f''(\xi)$ for some $\xi \in [a,b]$.
It has *second-order convergence* because the error term is proportional to h^2.

Composite Simpson's rule: If n is even and $f \in C^4$ then
$\int_a^b f(x)dx = \frac{h}{3}\left(f(a) + f(b) + 2\sum_{k=1}^{\frac{n}{2}-1} f(x_{2k}) + 4\sum_{k=1}^{\frac{n}{2}} f(x_{2k-1})\right) - \frac{\mathrm{b-a}}{180}\mathrm{h}^4\mathrm{f}^{(4)}(\xi)$ for some $\xi \in [a,b]$.

Contraction mapping: Function $g\colon [a,b] \to [a,b]$ where $\exists k < 1$ such that $\forall x,y \in [a,b], |g(x) - g(y)| \le k|x-y|$.

Every contraction mapping has a unique fixed point.
$\forall x \in [a,b]$, applying $x \leftarrow g(x)$ repeatedly will form a sequence whose limit is the fixed point.

If $\exists k < 1$ such that $\forall x \in (a,b), |g'(x)| < k$, then $g(x)$ is a contraction mapping.

Fixed-point iteration: Given $f(x) = 0$, we can define $g(x) = x + \beta(x)f(x)$ for some nonzero function β, and solve $f(x) = 0$ by finding the fixed point of $g(x)$.

Newton's method: Also finding the solution for $f(x) = 0$, but do $x_{n+1} \coloneqq x_n - \frac{f(x_n)}{f'(x_n)}$.

Thm: If $f \in C^2$ (on $[a,b]$) and $\exists x^* \in (a,b)$ such that $f(x^*) = 0$ and $f'(x^*) \neq 0$, then $\exists \delta > 0$ such that $\forall x_0 \in [x^* - \delta, x^* + \delta]$, the sequence $\{x_n\}$ generated using Newton's method tends to x^*.

Matrix definitions:

- Symmetric matrix: $A = A^T$
- Orthogonal matrix: $Q^TQ = I$ (or $QQ^T = I$) (or $Q^T = Q^{-1}$)
- Diagonal matrix: All entries that are not on the main diagonal are zero
- Upper triangular matrix: A square matrix such that $a_{ij} = 0 \quad \forall\, i > j$.
- Lower triangular matrix: A square matrix such that $a_{ij} = 0 \quad \forall\, i < j$.
- Positive definite matrix: For every nonzero column vector z, $z^TMz > 0$.
- Positive semi-definite matrix: For every nonzero column vector z, $z^TMz \ge 0$.
- Negative definite matrix: For every nonzero column vector z, $z^TMz < 0$.
- Negative semi-definite matrix: For every nonzero column vector z, $z^TMz \le 0$.
- Indefinite matrix: A matrix that is not positive semi-definite and not negative semi-definite.

Tools:

- Eigenvalues of a triangular matrix are its diagonal entries
- The product of the eigenvalues of a matrix is the determinant
- The sum of the eigenvalues of a matrix is the trace
- Symmetric matrices have all eigenvalues, and corresponding eigenvectors are mutually orthogonal