Numerical Analysis Cheat sheet (for paper exam)

Daniel Lin

Based on lectures by Dr Sheehan Olver

This note is for final exams only. It includes key results of theorems and key methodologies for solving exam questions. The following contents are omitted as they are non-examinable

- Convergence of Poisson DE approximation
- FFT
- All codes

Contents

1 Basics of Linear algebra **3**
1.1 Rank-nullity theorem . . . 4
1.2 Non-singularity . . . 5
1.3 Spectral Theorem . . . 5

2 Numbers **5**
2.1 Integers . . . 5
2.2 Floating Point Numbers . . . 6
2.3 Floating Point Errors . . . 7

3 Differentiation **8**

4 Asymptotic costs **9**

5 Matrix **10**
5.1 Matrix multiplication . . . 10
5.2 Permutation . . . 10
5.3 Reflection and Rotation . . . 11

6 Decomposition of matrices and Applications **12**
6.1 QR decomposition . . . 12
6.2 LU, PLU decomposition . . . 13
6.3 Cholesky decomposition . . . 14
6.4 Vector and matrix norm . . . 14

6.5 SVD decomposition 15

7 Conditional Numbers **16**

8 Differential Equations **17**
8.1 Integration 17
8.2 Time-evolution and Euler method 17
8.3 Non-linear simple case 18
8.4 Laplace and Poisson equation 18
8.5 Convergence 19

9 Fourier series **20**
9.1 Discrete Fourier Transform (DFT) 21
9.2 Tricks for finding Fourier coefficients 22

10 Orthogonal polynomials **22**
10.1 Classical polynomials 24
10.2 List of classical polynomials 27

11 Interpolation polynomials **27**
11.1 Interpolation to Quadrature 28

1 Basics of Linear algebra

This section mainly summarises conclusions of non-square matrices, which is almost completely omitted in the course, but turns out to be very important in many places.

Matrix is closely related to linear transformations. Assume U, V are vector spaces over the same field $\boldsymbol{F}$ with dimension n, m respectively. Assume B_U, B_V are basis of U, V respectively, and $T : U \to V$ is a linear transformation. Matrix $A \in \mathbb{R}^{m\times n}$ represent the linear transformation T if $y = T(x) \Leftrightarrow [y]_{B_V} = A[x]_{B_U}$.

Conjugate transpose of A is defined as $A^* := \overline{A}^T$. Note as long as A, B has appropriate dimensions

$$(AB)^* = B^*A^* \quad (AB)^T = B^TA^T \quad \text{but } \overline{AB} = \overline{A}\,\overline{B}$$

many definitions of families of matrices use transpose and conjugate transpose

- *Symmetric* if $A^T = A$
- *Skew-symmetric* if $A^T = -A$
- *Orthogonal* if A is real-valued and $A^TA = I$
- *Hermitian* if $A^* = A$
- *Skew-Hermitian* if $A^* = -A$
- *Unitary* if $A^*A = I$
- *Normal* if $A^*A = AA^*$

Note if A is real valued, orthogonal $\Rightarrow$ unitary $\Rightarrow$ normal.

$\operatorname{tr} A := \sum_{i=1}^{q} a_{ii}$ where $q := \min\{m, n\}$. If $A \in \mathbb{C}$, then $\operatorname{tr} A^*A = \operatorname{tr} AA^* = \sum_{i,j} |a_{ij}|^2$. So

$$\operatorname{tr} AA^* = 0 \Leftrightarrow A = 0$$

for real matrix, replace A^* by A^T.

Ways to understand matrix multiplication

- $Ax = \sum_i x_i \boldsymbol{c}_i$ where $\boldsymbol{c}_i$ are columns of A.
- $y^TA = \sum_i y_i \boldsymbol{r}_i$ where row vectors $\boldsymbol{r}_i$ are rows of A.
- For square matrices $A, B \in M_n(F)$:

$$A\begin{pmatrix} | & & | \\ b_1 & \cdots & b_n \\ | & & | \end{pmatrix} = \begin{pmatrix} | & & | \\ Ab_1 & \cdots & Ab_n \\ | & & | \end{pmatrix}, \begin{pmatrix} - & r_1^T & - \\ & \vdots & \\ - & r_n^T & - \end{pmatrix} B = \begin{pmatrix} - & r_1^TB & - \\ & \vdots & \\ - & r_n^TB & - \end{pmatrix}$$

i.e. Left multiplication by A is multiplying columns by A, right multiplication by B is multiplying rows by B.

- If $A \in F^{m\times p},\ B \in F^{m\times q}$ where F is arbitrary field, denote a_k, b_k as the k'th column of A, B, then

$$A^T B = [a_i^T b_j]$$

i.e. i, j entry is scalar $a_i^T b_j$.
If $A \in F^{m\times p},\ B \in F^{n\times p}$, then

$$AB^T = \sum_{k=1}^{p} a_k b_k^T$$

. i.e. the summation of outer product(a matrix) of a_k, b_k.

1.1 Rank-nullity theorem

$\operatorname{rank} A^T = \operatorname{rank} A$.
Rank-nullity theorem:

- $\dim \text{ColSpace}\, A + \ker A = \operatorname{rank} A + \ker A = n$
- $\dim \text{RowSpace}\, A + \ker A^T = \operatorname{rank} A + \ker A^T = m$

Rank of combined matrix

- If $A \in F^{m\times p},\ B \in F^{m\times q}$, then

$$\text{ColSpace}\, A + \text{ColSpace}\, B = \text{ColSpace}\, \begin{bmatrix} A & B \end{bmatrix}$$

- If $A \in F^{m\times p},\ B \in F^{n\times p}$, then

$$\ker A \cap \ker B = \ker \begin{bmatrix} A \\ B \end{bmatrix}$$

Given $A \in F^{m\times p}, b \in F^n$, the linear system $Ax = b$ have at least one solution iff $\operatorname{rank} \begin{bmatrix} A & b \end{bmatrix} = \operatorname{rank} A$.

Note elementary operations(on rows or on columns) do not change rank of a matrix, so r.r.e.f. of A has the same rank as A.

Properties of rank

- If $A \in F^{m\times n}$, $1 \le \operatorname{rank} A \le \min\{m, n\}$.
- $\operatorname{rank} \tilde{A} \le \operatorname{rank} A$ where $\tilde{A}$ is A with some rows and/or columns deleted.
- (Sylvester inequality)If $A \in F^{m\times p},\ B \in F^{p\times n}$, then

$$(\operatorname{rank}(A) + \operatorname{rank}(B)) - p \le \operatorname{rank} AB \le \min\{\operatorname{rank} A, \operatorname{rank} B\}$$

- (rank-sum inequality) If $A, B \in F^{m\times n}$, then

$$|\operatorname{rank} A - \operatorname{rank} B| \leq \operatorname{rank}(A+B) \leq \operatorname{rank} A + \operatorname{rank} B$$

- If $A \in \mathbb{C}^{m\times n}$, then $\operatorname{rank}(A^*A) = \operatorname{rank} A^* = \operatorname{rank} A^T = \operatorname{rank} \overline{A} = \operatorname{rank} A$.
- If $A \in F^{m\times m}, C \in F^{n\times n}$ are non-singular, $B \in F^{m,n}$, then

$$\operatorname{rank} AB = \operatorname{rank} B = \operatorname{rank} BC = \operatorname{rank} ABC$$

- $\operatorname{rank} A = \operatorname{rank} B$ iff exists non-singular X, Y s.t. $B = XAY$.

1.2 Non-singularity

Non-square matrices can have left/right inverse. Left inverse of A is B s.t. $BA = I$, right inverse of A is C s.t. $AC = I$. If $A \in F^{m\times n}$ and $\operatorname{rank} A = n$, then left inverse exists. If $\operatorname{rank} A = m$, then right inverse exists.
But there is no guarantee left inverse = right inverse for non-square matrices. So $U^TU = I$ CANNOT imply $UU^T = I$.
$(A^{-1})^T = (A^T)^{-1}$ so we can denote it as A^{-T}. If $A \in GL(n, \mathbb{C})$ (group of invertible matrices over $\mathbb{C}$), then $(A^{-1})^* = (A^*)^{-1}$ so we can denote it as A^{-*}.

1.3 Spectral Theorem

Eigenvectors of symmetric matrix are all orthogonal. So P (matrix of unit Eigenvectors of symmetric matrix A) must be orthogonal.
Hermitian case: If A is Hermitian ($A = A^*$), then A is diagonalisable.
Normal case: A is normal ($AA^* = A^*A$) $\Leftrightarrow$ exists unitary U s.t. $A = UDU^*$ where D is diagonal.
Note every unitary matrix and every real-valued orthogonal matrix is automatically normal.

2 Numbers

2.1 Integers

Binary: $(\cdots)_2$, Decimals: $(\cdots)_{10}$.
Decimals to Binary Use division with remainder by 2 repeated. Each time divide the quotient further. For example,

$$11/2 = 5...\mathbf{1}, \quad 5/2 = 2...\mathbf{1}, \quad 2/2 = \mathbf{1}...\mathbf{0}$$

Put 1 first, then fill the remainders in REVERSE order , so binary expression is $(1011)_2$.

Decimals to signed binary Leave the first bit for sign: 0 - positive, 1 - negative.

Other bits: same as unsigned for positive integer. For negative numbers, revert every bit then plus 1.
Signed binary integer to decimals Treat as usual for positive number. For negative numbers, subtract 2^p. (p is total number of bits, including sign bit)

2.2 Floating Point Numbers

1. $F^{\text{normal}}_{\sigma,Q,S}$ - set of normal FPN
$\sigma \in \mathbb{N}$ - bias, $Q \in \mathbb{N}$ - exponent bits, $S \in \mathbb{N}$ - mantissa bits. For FPN $(s\boldsymbol{e}\boldsymbol{m})_2$ (where s - sign bit, $\boldsymbol{e}$ - exponent bits and $\boldsymbol{m}$ - mantissa bits), it represents

$$(-1)^s 2^{((\boldsymbol{e})_{10}-\sigma)}(1.\boldsymbol{m})_{10}$$

Common values of σ, Q, S: 64-bits $(1023, 11, 52)$, 32-bits $(127, 8, 23)$, 16-bits $(15, 5, 10)$

2. $F^{\text{sub}}_{\sigma,Q,S}$ - set of subnormal/denormalised FPN
Any number has bit string $(s\boldsymbol{0}\boldsymbol{m})_2$ and it represents

$$(-1)^s 2^{-\sigma+1}(0.\boldsymbol{m})_{10}$$

3. F^{special} - the special FPNs (i.e. ± 0, $\pm\infty$, NaN)

1. $\boldsymbol{e} = \boldsymbol{0}, \boldsymbol{m} = \boldsymbol{0}$ – ± 0
2. $\boldsymbol{e} = \boldsymbol{0}, \boldsymbol{m} \neq \boldsymbol{0}$ – subnormal numbers
3. $\boldsymbol{e} = \boldsymbol{1}, \boldsymbol{m} = \boldsymbol{0}$ – $\pm\infty$
4. $\boldsymbol{e} = \boldsymbol{1}, \boldsymbol{m} \neq \boldsymbol{0}$ – NaN

$\boldsymbol{1}$ means all bits are 1.

Decimal system to binary for real numbers This only applies to some numbers in $[0, 1)$. Multiply by 2, take the integer part, and repeat the process with the decimal part. e.g.

$$0.8125 \times 2 = 1.625 \to 1$$

$$0.625 \times 2 = 1.25 \to 1$$

$$0.25 \times 2 = 0.5 \to 0$$

$$0.5 \times 2 = 1.0 \to 1$$

so binary expression is $(0.1101)_2$
If the above process fails to converge, simply subtract the largest powers of 2 not resulting in negative numbers repeatedly.

Binary to decimals (real numbers)
For recurring ones, try to convert to the form $(2^{-s_1}+...+2^{-s_n})(1.001001001)_2$. (Number of 0s between 1 s can change) Then use geometric series.

The rounding functions

- $f|^{\text{up}}(x)$ - smallest binary decimal y s.t. $x \leq y$
- $f|^{\text{down}}(x)$ - largest binary decimal y s.t. $x \geq y$
- $f|^{\text{near}}(x)$ - pick the nearer one of above. (There may be a tie, pick the one with last bit 0)

If not specified, $f|^{\text{near}}(x)$ is used. Because of rounding, FP additions, multiplications etc. are not associative.

Floating point arithmetic
Addition is divided into these steps: shift bit of one number to align bits, add the significant parts, perform necessary rounding, and normalise. (i.e. keep integer part 1)

2.3 Floating Point Errors

If $\tilde{x}$ is the approximated value of x, and $\tilde{x} = x + \delta_a$. $|\delta_a|$ is defined as absolute error. Rewrite as $\tilde{x} = x(1+\delta_r)$, where $\delta_r = \delta_a/x$. $|\delta_r|$ is defined as relative error.

Machine epsilon: $\epsilon_{m,S} = 2^{-S}$ where S is the number of significant bits in FPN.
Normalised range: Range of numbers that can be written as normal FPN.

$$N = [\min F^{\text{norm}}, \max F^{\text{norm}}]$$

For $f|^{\text{near}}(x)$ $(x \in N)$ relative error $\leq \frac{\epsilon_{m,S}}{2}$. And for $f|^{\text{up}}(x)$, $f|^{\text{down}}(x)$ relative error $< \epsilon_{m,S}$.

Be careful, some numbers can be expressed in FPN exactly, e.g. $1.25/2 = 0.75 = 3 \times 2^{-2}$, so $1.25 \oslash 2$ has no error.

Bounding multiplication Under any precision and RoundNearest mode, $1.1 \otimes 1.2 \leq (1.1+\delta_1) \times (1.2+\delta_2) = 1.1 \times 1.2 + 1.1\delta_2 + 1.2\delta_1 + \delta_1\delta_2$ absolute error $|1.1\delta_2 + 1.2\delta_1 + \delta_1\delta_2| \leq \frac{\epsilon_m}{2}(1.1+1.2+1) = 3.3\epsilon_m$
Rules for bounding: keep calculation simple! Round 0.6 to 1, round 3.4 to 4. You are not asked to find sharpest bound.

Bounding division For $1 \oslash 1.1$, Let δ_1 be error for rounding 1.1 and δ_2 for division. So $1 \oslash 1.1 = \frac{1}{1.1(1+\delta_1)}(1+\delta_2)$. But

$$\left|\frac{1}{1+\delta_1} - 1\right| = \left|\frac{\delta_1}{1+\delta_1}\right| \leq \frac{\epsilon_m}{2}\frac{1}{1-1/2}(\text{Since } |\delta_1| \leq 1/2) = \epsilon_m$$

So we can express $\frac{1}{1+\delta_1}$ as $1+\delta_3$ where $|\delta_3| \leq \epsilon_m$. Now $\frac{1}{1.1(1+\delta_1)}(1+\delta_2) = \frac{1}{1.1}(1+\delta_3)(1+\delta_2)$.

Useful inequalities:

- $e^x < 3$ if $x \leq 1$
- if $a > 0$, $\theta \in [0, \pi/2a]$, then $2a\theta/\pi \leq \sin(a\theta) \leq a\theta$

3 Differentiation

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} =: r_f$$

Bound the error of this estimation using Taylor series: $f(x+h) = f(x) + f'(x)h + \frac{f''(t)}{2}h^2$ where t is between x and $x+h$. So $r_f \approx f'(x) + \frac{f''(t)}{2}h$, then the error $\frac{f''(t)}{2}h$ is bounded by

$$\delta_{x,h} \leq \frac{M}{2}h, \quad \text{where } M := \sup_{x \leq t \leq x+h}(f''(t))$$

Floating Point estimation
Assume estimation of $f(x)$ is $f^{\mathrm{FP}}(x)$ with absolute error δ_x^f s.t. $|\delta_x^f| \leq c\epsilon_m$. And assume $h = 2^{-n}$, $n \in \mathbb{N}$ to avoid error when dividing h. We have

$$\frac{f^{\mathrm{FP}}(x+h) \ominus f^{\mathrm{FP}}(x)}{h} = \frac{f(x+h) + \delta_{x+h}^f - f(x) - \delta_x^f}{h}(1+\delta_1) \quad \text{where } (1+\delta_1) \text{ is error due to } \ominus$$

$$= \frac{f(x+h) - f(x)}{h}(1+\delta_1) + \frac{\delta_{x+h}^f - \delta_x^f}{h}(1+\delta_1)$$

Then using Taylor expansion in the same way, (sometimes we need more terms for Taylor expansion)

$$= f'(x) + f'(x)\delta_1 + \frac{f''(t)}{2}h(1+\delta_1) + \frac{\delta_{x+h}^f - \delta_x^f}{h}(1+\delta_1)$$

So the error δ^D is the last three terms,

$$|\delta^D| \leq |f'(x)|\frac{\epsilon_m}{2} + Mh + \frac{4c}{h}\epsilon_m$$

where we used $1+\delta_1 \leq 2$. As $h \to 0$, this error will decrease first and then increase. To get the best result, the last two terms should be balanced, i.e. $Mh \approx \frac{4c}{h}\epsilon_m$, so $h \approx \sqrt{\frac{4c}{M}\epsilon_m} \propto \sqrt{\epsilon_m}$.

Dual number estimation Commutative ring $\mathbb{D} := \{a + b\epsilon : a, b \in \mathbb{R}\}$ where $\epsilon^2 := 0$. $(a+b\epsilon)^n = a^n + bna^{n-1}\epsilon$ so for polynomial $p(x)$:

$$p(a+b\epsilon) = p(a) + bp'(a)\epsilon$$

For general function $f(x)$:

$$f(a+b\epsilon) = f(a) + bf'(a)\epsilon$$

e.g. $sin(1+\epsilon) = sin(1) + cos(1)\epsilon$. We may use these basic functions to estimate derivative of more complex ones. e.g. $f(x) = \exp(x^2 + e^x)$, find $f'(1)$:

$$f(1+\epsilon) = \exp((1 + 2\epsilon + e + e\epsilon)) = \exp(1+e) + \exp(1+e)(2+e)\epsilon$$

so $f'(1) = \exp(1+e)(2+e)$.

Second derivative:

```
f(Dual(Dual(a, 1), Dual(1, 0))).b.b
```

Outer Dual is a new dual of dual numbers. i.e. Dual{Dual}. We denote its ϵ by $\tilde{\epsilon}$.

$$\begin{aligned} f((a+b\epsilon) + \tilde{\epsilon}(c+d\epsilon)) &= f(a+b\epsilon) + f'(a+b\epsilon)\tilde{\epsilon}(c+d\epsilon) \\ &= f(a) + b\epsilon f'(a) + (f'(a) + b\epsilon f''(a))(c+d\epsilon)\tilde{\epsilon} \end{aligned}$$

4 Asymptotic costs

$f(n) = O(\phi(n)), n \to \infty$ iff $\left|\frac{f(n)}{\phi(n)}\right|$ is bounded for large n.
$f(n) = o(\phi(n)), n \to \infty$ iff $\lim_{n\to\infty} \frac{f(n)}{\phi(n)} = 0$
$f(n) \sim o(\phi(n)), n \to \infty$ iff $\lim_{n\to\infty} \frac{f(n)}{\phi(n)} = 1$
Same definitions apply at x_0 by replacing $n \to \infty$ by $x \to x_0$. e.g. $f(n) = O(\phi(n))$ as $x \to x_0$ iff $\left|\frac{f(n)}{\phi(n)}\right|$ is bounded for neighbourhood of x_0.

Rules:

$$cO(\phi(n)) = O(\phi(n)), O(c\phi(n)) = O(\phi(n)) \quad \text{where } c \text{ is constant}$$

$$O(\phi(n)) + o(\phi(n)) = O(\phi(n))$$

$$O(\phi(n))O(\psi(n)) = O(\phi(n)\psi(n))$$

$$O(\phi(n)) + O(\psi(n)) = O(|\phi(n)| + |\psi(n)|)$$

If $g(n) = O(f(n)), f(n) = o(\phi(n))$, then $g(n) = o(\phi(n))$

Note $\log(n) = o(n)$ might be useful.

5 Matrix

5.1 Matrix multiplication

Matrix multiplication in column fashion:

$$A\boldsymbol{x} = \boldsymbol{a}_1 x_1 + ...\boldsymbol{a}_n x_n$$

cost: $O(mn)$ if $A \in F^{m\times n}$.

Solving matrix equation $A\boldsymbol{x} = \boldsymbol{b}$, Upper triangular case:

$$\begin{pmatrix} u_{1,1} & u_{1,2} & \dots & u_{1,n} \\ 0 & \ddots & & \vdots \\ 0 & 0 & \ddots & u_{n-1,n} \\ 0 & 0 & 0 & u_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ ... \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ ... \\ b_{n-1} \\ b_n \end{pmatrix}$$

$$x_k = \frac{b_k - \sum_{j=1}^{n-k} u_{k,k+j} x_{k+j}}{u_{k,k}}$$

Lower triangular case:

$$\begin{pmatrix} l_{1,1} & 0 & 0 & 0 \\ l_{2,1} & \ddots & 0 & 0 \\ \vdots & & \ddots & 0 \\ l_{n,1} & l_{n,2} & \dots & l_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ ... \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ ... \\ b_{n-1} \\ b_n \end{pmatrix}$$

$$x_k = \frac{b_k - \sum_{j=1}^{k-1} l_{k,j} x_j}{l_{k,k}}$$

Banded matrix
Banded matrix B only has main diagonal, u sub-diagonals above, l sub-diagonals below. The k'th entry of $B\boldsymbol{x}$, is

$$\sum_{j=\max(1,k-l)}^{\min(n,k+1)} b_{k,j} x_j$$

5.2 Permutation

Cauchy notation:

$$P_\sigma = \begin{pmatrix} 1 & 2 & \dots & n \\ \sigma_1 & \sigma_2 & \dots & \sigma_n \end{pmatrix}$$

We can permute a vector

$$P\boldsymbol{v} = \begin{pmatrix} v_{\sigma_1} \\ \vdots \\ v_{\sigma_n} \end{pmatrix}$$

Warning: It is sending entry σ_k to position k, not k to σ_k
Matrix representation:

$$P = \begin{pmatrix} e_{\sigma_1^{-1}} & e_{\sigma_2^{-1}} & \dots & e_{\sigma_n^{-1}} \end{pmatrix}$$

$P^{-1} = P^T$. So

$$P = \begin{pmatrix} e_{\sigma_1}^T \\ e_{\sigma_2}^T \\ \vdots \\ e_{\sigma_n}^T \end{pmatrix}$$

Side note: P is orthogonal iff $e_i^T P^T P e_j = \delta_{i,j}$ for all i, j.

5.3 Reflection and Rotation

Rotation matrices Q_θ are all orthogonal.

$$Q_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

Here are some useful reflection matrices:
Reflection in direction of $\mathbf{v}$, where $\|\mathbf{v}\| = 1$ is

$$Q_{\mathbf{v}} = I - 2\mathbf{v}\mathbf{v}^T$$

Properties of $Q_{\mathbf{v}}$:

- symmetric, orthogonal.
- $\mathbf{v}$ is Eigenvector with Eigenvalue -1, all other Eigenvalues are 1 ($g(1) = n - 1$)
- $\det(Q_{\mathbf{v}}) = -1$

Note rank $(\mathbf{v}\mathbf{v}^T) = 1$ for any vector $\mathbf{v}$.

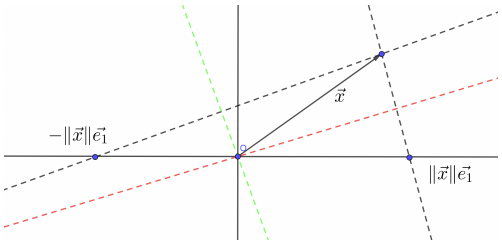

Figure 1: Householder reflection

Householder reflection: $\vec{x} \mapsto \|\vec{x}\|\vec{e_1}$. Let $\vec{y} = \vec{x} - \|\vec{x}\|\vec{e_1}$

$$Q_{\vec{w}} = I - 2\vec{w}\vec{w}^T \quad \text{where } \vec{w} = \frac{\vec{y}}{\|\vec{y}\|}$$

is the required reflection matrix. Reflection to $-\|\vec{x}\|\vec{e_1}$ is done by replacing $\vec{y}$ with $\vec{x} + \|\vec{x}\|\vec{e_1}$.
By default, we choose to reflect to $-sign(x_1)\|\boldsymbol{x}\|\vec{e}_1$. Same works when finding maps s.t. $\vec{x} \mapsto \|\vec{x}\|\vec{e_i}$.

6 Decomposition of matrices and Applications

6.1 QR decomposition

Given matrix $A_{m\times n}$ $(m \geq n)$, $A = QR$ where $Q_{m\times m}$ is orthogonal $(Q^T = Q^{-1})$ and $R_{m\times n}$ is right triangular. If $m > n$, $R = \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix}$ where $\hat{R} \in F^{n\times n}$ is upper triangular.

reduced QR decomposition
$A = \hat{Q}\hat{R}$. $\hat{Q}_{m\times n}$ takes first n columns of Q and $\hat{R}$(upper-triangular) takes first n rows of R. Note $\hat{Q}^T\hat{Q} = I$ again but we cannot invert $\hat{Q}$.
Method 1
If $A = \begin{pmatrix} \boldsymbol{a}_1| & \boldsymbol{a}_2| & ... & |\boldsymbol{a}_n \end{pmatrix}$ and $\hat{Q} = \begin{pmatrix} \boldsymbol{q}_1| & \boldsymbol{q}_2| & ... & |\boldsymbol{q}_n \end{pmatrix}$. $\forall 1 \leq j \leq n$,

$$span(\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_j) = span(\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_j)$$

Beginning with $j = 1$, for each j,

$$\boldsymbol{q}_j = \frac{\boldsymbol{v}_j}{\|\boldsymbol{v}_j\|}, \text{ where } \boldsymbol{v}_j = \boldsymbol{a}_j - \sum_{k=1}^{j-1}(\boldsymbol{q}_k^T\boldsymbol{a}_j)\boldsymbol{q}_k, \quad \boldsymbol{r}_j = \begin{pmatrix} \boldsymbol{q}_1^T\boldsymbol{a}_j \\ ... \\ \boldsymbol{q}_{j-1}^T\boldsymbol{a}_j \\ \|v_j\| \end{pmatrix}$$

where $\boldsymbol{r}_j$ are columns of R. Computation cost: $O(mn^2)$

Method 2
Take Q_i to be the householder reflection on $a_j[j : end]$, then $Q_n...Q_1A$ is right-triangular, call it R. Then $A = QR$ where $Q = Q_1^T...Q_n^T$. Computational cost: $O(m^2n^2)$.

Application: Least Square Problem Find $\boldsymbol{x}$ s.t. $\|A\boldsymbol{x} - \boldsymbol{b}\|^2$ is minimised:

$$\min_{x\in\mathbb{R}^n} \|A\boldsymbol{x} - \boldsymbol{b}\| = \min_{x\in\mathbb{R}^n} \|\hat{R}\boldsymbol{x} - \boldsymbol{c}\| + \|(Q^T\boldsymbol{b})_{n+1:m}\| \quad \text{where } c := (Q^T\boldsymbol{b})_{1:n} = \hat{Q}^T\boldsymbol{b}$$

so equivalent to finding $\boldsymbol{x}$ that minimises $\|\hat{R}\boldsymbol{x} - \boldsymbol{c}\|$

6.2 LU, PLU decomposition

LU decomposition is $A = LU$, PLU is $A = P^T LU$. (P - permutation)

One-column lower triangular matrix

$$\mathcal{L}_j = \left\{ I + \begin{pmatrix} \mathbf{0}_j \\ \boldsymbol{l} \end{pmatrix} \boldsymbol{e}_j^T \ : \ \boldsymbol{l} \in \mathbb{R}^{n-j} \right\}$$

Inverse of $L_j \in \mathcal{L}_j$:

$$L_j^{-1} = I - \begin{pmatrix} \mathbf{0}_j \\ \boldsymbol{l} \end{pmatrix} \boldsymbol{e}_j^T$$

Multiplication of $L_j \in \mathcal{L}_j, L_k \in \mathcal{L}_k$:

$$L_j L_k = I + \begin{pmatrix} \mathbf{0}_j \\ \boldsymbol{l}_1 \end{pmatrix} \boldsymbol{e}_j^T + \begin{pmatrix} \mathbf{0}_k \\ \boldsymbol{l}_2 \end{pmatrix} \boldsymbol{e}_k^T$$

Pivoting property: If $P_\sigma = \begin{pmatrix} I_j & 0 \\ 0 & \tilde{P} \end{pmatrix}$, then

$$P_\sigma L_j = \widetilde{L_j} P_\sigma$$

where $\widetilde{L_j} \in \mathcal{L}_j$ is the result of applying $\tilde{P}$ to $\boldsymbol{l}$

LU decomposition
If

$$L_1 = \begin{pmatrix} 1 & & & \\ -\frac{a_{21}}{a_{11}} & 1 & & \\ \vdots & & \ddots & \\ -\frac{a_{n1}}{a_{11}} & & & 1 \end{pmatrix}$$

then $L_1 A$ has 0 entries on first column except $1,1$ entry. Repeat this process on $L_1 A[2 : end, 2 : end]$, continue until getting

$$L_{n-1} ... L_2 L_1 A = U$$

then $A = LU$ where $L = L_1^{-1} L_2^{-1} ... L_{n-1}^{-1}$.
PLU decomposition
Sometimes we need to swaps rows to avoid problem of dividing very small numbers. Say P_1 swaps row 1 and row k where k maximises $|a_{k1}|$. Then we pick L_1 as in LU decomposition for $P_1 A$. Repeat this process, we can get

$$L_{n-1} P_{n-1} ... L_2 P_2 L_1 P_1 A = U$$

use pivoting property to obtain $L' P A = U$, then $A = P^T (L')^{-1} U$.

6.3 Cholesky decomposition

Properties of positive definite matrix:

- diagonal elements $a_{kk} > 0$.
- If V is non-singular, A is positive definite, then $V^T AV$ is positive definite. V can be permutation P_σ.

Theorem. A is symmetric positive definite $\Leftrightarrow$ it has Cholesky decomposition $A = LL^T$ where diagonal elements of L are all positive.
So if for symmetric matrix A, Cholesky decomposition fails (at some stage, first entry is negative), A cannot be positive definite.

Algorithm: Define

$$A_1 := A,\ \boldsymbol{v}_k := A_k[2 : n-k+1, 1],\ \alpha_k := A_k[1,1]$$

$$A_{k+1} := A_k[2 : n-k+1, 2 : n-k+1] - \frac{\boldsymbol{v}_k \boldsymbol{v}_k^T}{\alpha_k}$$

Then

$$L = \begin{pmatrix} \sqrt{a_1} & & & \\ \frac{v_1[1]}{\sqrt{a_1}} & \sqrt{a_2} & & \\ \vdots & \ddots & \ddots & \\ \frac{v_1[n-1]}{\sqrt{a_1}} & \cdots & \frac{v_{n-1}[1]}{\sqrt{a_{n-1}}} & \sqrt{a_n} \end{pmatrix}$$

The induction step is given by:

$$\begin{pmatrix} \alpha & \mathbf{v}^T \\ \mathbf{v} & K \end{pmatrix} = \begin{pmatrix} \sqrt{a} & \\ \mathbf{v}/\sqrt{a} & I \end{pmatrix} \begin{pmatrix} 1 & \\ & K - \frac{\mathbf{v}\mathbf{v}^T}{\alpha} \end{pmatrix} \begin{pmatrix} \sqrt{a} & \mathbf{v}^T/\sqrt{a} \\ & I \end{pmatrix}$$

Positive symmetric matrix can also be decomposed into UU^T where U is upper triangular.

6.4 Vector and matrix norm

For vector x, $\|x\|_2 \le \|x\|_1 \le \sqrt{n}\|x\|_2$ and $\|x\|_\infty \le \|x\|_2 \le \sqrt{n}\|x\|_\infty$.
Fröbenius norm $\|A\|_F$: treat matrix A as vector and find its 2-norm. Note $\|A\|_F = \sqrt{tr(A^T A)}$, And if Q is orthogonal, $\|QA\|_F = \|A\|_F$.

Induced Matrix Norm Given $A_{m\times n}$ and $\|\cdot\|_X$ on $\mathbb{R}^m$, $\|\cdot\|_Y$ on $\mathbb{R}^n$.

$$\|A\|_{X\to Y} := \sup_{\boldsymbol{v}:\|v\|_X=1} \|Av\|_Y$$

For square matrix, $\|A\|_X := \|A\|_{X\to X}$.
Properties of Induced Matrix Norm (abbreviated as $\|\cdot\|$ below)

- Triangular inequality: $\|A+B\| \leq \|A\| + \|B\|$
- Homogenecity: $\|cA\| = |c|\|A\|$
- Positive-definite: $\|A\| = 0 \Rightarrow A = 0$
- $\|Ax\|_Y \leq \|A\|_{X\to Y}\|x\|_X$
- Multiplicative inequality $\|AB\|_{X\to Z} \leq \|A\|_{Y\to Z}\|B\|_{X\to Y}$.

Note for induced p-norm

$$\left\| \begin{pmatrix} A \\ B \end{pmatrix} \right\|_p^p = \|A\|_p^p + \|B\|_p^p$$

Special cases:
$\|A\|_1 = \max_j \|\boldsymbol{c}_j\|_1$ (max 1-norm of columns)
$\|A\|_\infty$ equals max 1-norm of rows.
$\|A\|_{1\to\infty}$ equals $\max_{k,j} |a_{k,j}|$
$\|A\|_2$ = largest singular value. If A is diagonal, $\|A\|_2 = \max_k |d_k|$ (max diagonal entry). If Q is orthogonal, $\|Q\|_2 = 1$.
$\|A\|_2 \leq \|A\|_F \leq \sqrt{r}\|A\|_2$ where $r = rank(A)$. So if rank $A = 1$, 2-norm equals Fröbenius norm.
$\||A|\|_1 = \|A\|_1, \||A|\|_\infty = \|A\|_\infty, \||A|\|_F = \|A\|_F$, but $\||A|\|_2 \neq \|A\|_2$ in general. But one can derive

$$\|A\|_2 \leq \sqrt{s}\||A|\|_2$$

where $s = \text{rank}\,|A|$.

6.5 SVD decomposition

Reduced version
$A_{m\times n}$ with $rank(A) = r$ can be decomposed into $U\Sigma V^T$. $U_{m\times r}$, $V_{n\times r}$ have orthonormal columns. Σ is diagonal matrix with diagonal entries $d_1 \geq d_2 ... \geq d_r > 0$.
Full version
$A = U\Sigma V^T$. $U_{m\times m}, V_{n\times n}$ are orthogonal and Σ has only diagonal entries $d_{k,k}$.

Gram matrix A^TA. $Ker(A^TA) = Ker(A)$ and Eigenvalues of A^TA are non-negative. The matrix is symmetric so spectrum theorem applies. Note singular values of A are exactly square root of Eigenvalues of A^TA.

Singular values and matrix norms

- $\|A\|_F^2 = \sigma_1^2 + ... + \sigma_m^2$
- $\|A\|_2$ is the largest singular value
- $\|A^{-1}\|_2$ is reciprocal of the smallest non-zero singular value

$rank(A)$ = number of non-zero singular values = number of non-zero Eigenvalues of A^TA

Application: low-rank approximation
If $k < r = rank(A)$, and A has SVD $A = U\Sigma V^T$. Let U_k, Σ_k, V_k be first k columns of U, Σ, V respectively. Then $A_k := U_k\Sigma_k V_k^T$ is the best 2-norm approximation to matrix A:

$$\forall B \text{ with rank } k, \|A - A_k\|_2 \leq \|A - B\|_2$$

7 Conditional Numbers

Proposition. *If $|\epsilon_i| \leq \epsilon$ and $n\epsilon < 1$, then*

$$\prod_{i=1}^{n}(1+\epsilon_i) = 1 + \theta_n$$

where $|\theta_n| \leq \frac{n\epsilon}{1-n\epsilon}$

Proposition (dot product backward error)**.** *The floating point dot product satisfies*

$$dot(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} + \delta\boldsymbol{x})^T\boldsymbol{y}$$

where $|\delta\boldsymbol{x}| \leq \frac{n\epsilon_m}{2-n\epsilon_m}|\boldsymbol{x}|$

Proposition (matrix multiplication backward error)**.** *Consider floating point matrix multiplication on $A \in \mathbb{R}^{m\times n}$*

$$mul(A, \boldsymbol{x}) = \begin{pmatrix} dot(A[1,:], \boldsymbol{x}) \\ dot(A[2,:], \boldsymbol{x}) \\ \vdots \\ dot(A[m,:], \boldsymbol{x}) \end{pmatrix}$$

then $mul(A, \boldsymbol{x}) = (A + \delta A)\boldsymbol{x}$ where entries of δA: δa satisfies $|\delta a| \leq \frac{n\epsilon_m}{2-n\epsilon_m}|a|$. So

$$\|\delta A\|_1 \leq \frac{n\epsilon_m}{2-n\epsilon_m}\|A\|_1$$

$$\|\delta A\|_2 \leq \sqrt{\min\{m,n\}}\frac{n\epsilon_m}{2-n\epsilon_m}\|A\|_2$$

$$\|\delta A\|_\infty \leq \frac{n\epsilon_m}{2-n\epsilon_m}\|A\|_\infty$$

Conditional number: $K(A) := \|A\|\|A^{-1}\|$, $K_p(A)$ denotes $\|A\|_p\|A^{-1}\|_p$. Note $K_2(A) = \sigma_1/\sigma_n$ (σ_i are singular values)

Theorem (Relative error)**.** *If* $\|\delta A\| \leq \|A\|\epsilon$*, then*

$$\frac{\|\delta A\boldsymbol{x}\|}{\|A\boldsymbol{x}\|} \leq K(A)\epsilon$$

the general conclusion is if $p = 1, \infty$

$$\frac{\|mul(A, \boldsymbol{x}) - A\boldsymbol{x}\|_p}{\|A\boldsymbol{x}\|_p} \leq K_p(A)\frac{n\epsilon_m}{2 - n\epsilon_m}$$

for $p = 2$*, multiply* $\sqrt{\min\{m, n\}}$ *in front.*

8 Differential Equations

8.1 Integration

$$u(a) = c,\ u'(x) = f(x) \quad \forall x \in [a, b]$$

define $a =: x_1, x_2, ..., x_n := b$ so each interval $[x_k, x_{k+1}]$ has length $(b-a)/(n-1)$. Using forward difference one can find a linear system

$$\begin{pmatrix} e_1^T \\ D_h \end{pmatrix} \boldsymbol{u} = \begin{pmatrix} c \\ \boldsymbol{f} \end{pmatrix}$$

Finding $\boldsymbol{u}$ approximates $u(x)$ at x_i. Matrix$\begin{pmatrix} e_1^T \\ D_h \end{pmatrix}$ is lower bi-diagonal so the system can be solved by forward substitution:

$$u_1 = c,\ u_{k+1} = u_k + hf(x_k)$$

If using central difference instead, $\boldsymbol{f}$ becomes $(f(m_1) \ \ldots \ f(m_{n-1}))^T$ where $m_k := (x_{k+1} + x_k)/2$ are midpoints.

8.2 Time-evolution and Euler method

Scalar version:

$$u(0) = c,\ u'(t) - a(t)u(t) = f(t) \quad \forall t \in [0, T]$$

Vector version:

$$\boldsymbol{u}(0) = \boldsymbol{c},\ \boldsymbol{u}'(t) - A(t)\boldsymbol{u}(t) = \boldsymbol{f}(t) \quad \forall t \in [0, T]$$

Forward Euler

$$\begin{pmatrix} 1 & & & \\ -a(t_1) - 1/h & 1/h & & \\ & \ddots & \ddots & \\ & & -a(t_{n-1}) - 1/h & 1/h \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} c \\ f(t_1) \\ \vdots \\ f(t_{n-1}) \end{pmatrix}$$

Iteration version:

$$u_1 = c,\ u_{k+1} = (1 + ha(t_k))u_k + hf(t_k)$$

Generalised version to vector DE:

$$\boldsymbol{u}_1 = \boldsymbol{c},\ \boldsymbol{u}_{k+1} = (I + hA(t_k))\boldsymbol{u}_k + h\boldsymbol{f}(t_k)$$

Backward Euler

$$\begin{pmatrix} 1 & & & \\ -1/h & 1/h - a(t_2) & & \\ & \ddots & \ddots & \\ & & -1/h & 1/h - a(t_n) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} c \\ f(t_2) \\ \vdots \\ f(t_n) \end{pmatrix}$$

this gives iteration process

$$u_1 = c,\ u_{k+1} = (1 - ha(t_{k+1}))^{-1} (u_k + hf(t_{k+1}))$$

generalise to vector differential equation:

$$\boldsymbol{u}_1 = \boldsymbol{c},\ \boldsymbol{u}_{k+1} = (I - hA(t_{k+1}))^{-1} (\boldsymbol{u}_k + h\boldsymbol{f}(t_{k+1}))$$

8.3 Non-linear simple case

Consider

$$\boldsymbol{u}(0) = \boldsymbol{c},\ \boldsymbol{u}'(t) = f(t, \boldsymbol{u}(t)) \quad \forall\, t \in [0, T]$$

forward Euler method gives iteration process:

$$\boldsymbol{u}_1 = \boldsymbol{c},\ \boldsymbol{u}_{k+1} = \boldsymbol{u}_k + h\boldsymbol{f}(t_k, \boldsymbol{u}_k)$$

8.4 Laplace and Poisson equation

Poisson equation

$$\nabla^2 u = f(\boldsymbol{x})$$

with Dirichlet boundary condition

$$u(\boldsymbol{x}) = g(\boldsymbol{x}) \text{ for } \boldsymbol{x} \in D$$

where D is the boundary of region.
Laplace equation Poisson equation with $f = 0$.

Consider simple case $f : [a, b] \to \mathbb{R}$.

$$u(a) = c_0,\ u''(x) = f(x),\ u(b) = c_1$$

Using

$$u''(x_k) \approx \frac{u_{k-1} - 2u_k + u_{k+1}}{h^2}$$

$$\begin{pmatrix} 1 & 0 & & & \\ \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{1}{h^2} & -\frac{2}{h^2} & \frac{1}{h^2} \\ & & & 0 & 1 \end{pmatrix} \boldsymbol{u} = \begin{pmatrix} c_0 \\ f(x_2) \\ \vdots \\ f(x_{n-1}) \\ c_1 \end{pmatrix}$$

The above linear system is equivalent to

$$T\boldsymbol{u} := \begin{pmatrix} 1 & & \\ & \frac{1}{h^2}\Delta & \\ & & 1 \end{pmatrix} \boldsymbol{u} = \begin{pmatrix} c_0 \\ f(x_2) - \frac{c_0}{h^2} \\ f(x_3) \\ \vdots \\ f(x_{n-2}) \\ f(x_{n-1}) - \frac{c_1}{h^2} \\ c_1 \end{pmatrix}$$

The Laplacian of this graph

$$\Delta := \begin{pmatrix} -1 & 1 & & & \\ 1 & -2 & \ddots & & \\ & 1 & \ddots & 1 & \\ & & \ddots & -2 & 1 \\ & & & 1 & -1 \end{pmatrix}$$

is negative definite, i.e. $-\Delta = LL^T$ for some L.

8.5 Convergence

Toeplitz matrix. matrices that are constant on all diagonals, i.e.

$$A = \begin{pmatrix} a_0 & a_{-1} & a_{-2} & \dots & \dots & a_{-(n-1)} \\ a_1 & a_0 & a_{-1} & \ddots & & \vdots \\ a_2 & a_1 & \ddots & \dots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_{-1} & a_{-2} \\ \vdots & & \ddots & a_1 & a_0 & a_{-1} \\ a_{n-1} & \dots & \dots & a_2 & a_1 & a_0 \end{pmatrix}$$

One special case where we can find inverse

$$\begin{pmatrix} 1 & & & & \\ -l & 1 & & & \\ & -l & 1 & & \\ & & \ddots & \ddots & \\ & & & -l & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & & & & \\ l & 1 & & & \\ l^2 & l & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \\ l^{n-1} & \dots & l^2 & l & 1 \end{pmatrix}$$

Convergence of forward/backward Euler

$$u(0) = c,\ u'(t) = au(t) + f(t) \text{ for } t \in [0, 1]$$

If u is twice differentiable and u'' is uniformly bounded, then

$$\|\boldsymbol{u} - \boldsymbol{u}_{\text{ex}}\|_\infty = O(1/n)$$

$\boldsymbol{u}$ is approximation and $\boldsymbol{u}_{\text{ex}}$ is exact solution.

9 Fourier series

If f is 2π-periodic,

$$f(\theta) \approx \sum_{k=-m}^{m} \hat{f}_k e^{ik\theta} =: f_m(\theta)$$

where $\hat{f}_k := \frac{1}{2\pi} \int_0^{2\pi} f(\theta) e^{-ik\theta}\, d\theta$.

Theorem. *If $\|\hat{f}\|_1 := \sum_{k=-\infty}^{\infty} |\hat{f}_k| < \infty$, then $f_m(\theta) \to f(\theta)$.*
For condition to hold: f is periodic, f'' exists and is uniformly bounded. Higher derivative existence, faster convergence.

.

Fourier and Taylor If $0 = \hat{f}_{-1} = \hat{f}_{-2} = \dots$, and $\hat{f}_k$ converges absolutely. Let $z := e^{i\theta}$,

$$f(z) = \sum_{k=0}^{\infty} \hat{f}_k e^{ik\theta} = \sum_{k=0}^{\infty} \hat{f}_k z^k$$

Estimation of coefficients: Cut $[0, 2\pi]$ into n pieces. If $n = 2m + 1$, approximate $\hat{f}_{-m}, ..., \hat{f}_m$ using trapezium rule. If $n = 2m$, approximate coefficients $\hat{f}_{-m}, ..., \hat{f}_{m-1}$ instead.

Trapezium rule and average of function For 2π-periodic function f,

$$\int_0^{2\pi} f(\theta)\, d\theta \approx \frac{2\pi}{n} \sum_{j=1}^{n-1} f(\theta_j) =: 2\pi \Sigma_n[f(\theta)]$$

where $n = 2m+1$ and $\theta_j := \frac{2\pi j}{n}$ partitions $[0, 2\pi]$.

For Fourier coefficients:

$$\hat{f}_k \approx \frac{1}{n}\sum_{j=0}^{n-1} f(\theta_j)e^{-ik\theta j} = \Sigma_n[f(\theta)e^{-ik\theta}] =: \hat{f}_k^n$$

Proposition.

$$\Sigma_n[e^{ik\theta}] = \begin{cases} 1, & \textit{if } k = ..., -n, 0, n, 2n, ... \\ 0, & \textit{otherwise} \end{cases}$$

So

$$\Sigma_n[e^{ik\theta}e^{-ij\theta}] = \begin{cases} 1, & \text{if } j = k + pn \text{ for some } p \in \mathbb{Z} \\ 0, & \text{otherwise} \end{cases}$$

therefore,

$$\hat{f}_k^n = \sum_{p=-\infty}^{\infty} \hat{f}_{k+pn} \qquad (1)$$

If $0 = \hat{f}_{-1} = \hat{f}_{-2} = \ldots$, $\sum_{k=0}^{\infty} |\hat{f}_k| < \infty$

$$f_n(\theta) := \sum_{k=0}^{n-1} \hat{f}_k^n e^{ik\theta} \to f(\text{uniformly})$$

so trapezium rule approximation converges to true function.

$\hat{f}_{k+pn}^n = \hat{f}_k^n$ for any $p \in \mathbb{Z}$. So given Taylor coefficients $(\hat{f}_k^n)_{k=0,\ldots,n-1}$, $n = 2m+1$, to build Fourier coefficients $(\hat{f}_k^n)_{k=-m,\ldots,0,1,\ldots,m}$, use permutation

$$\begin{pmatrix} 1 & \cdots & m & | & m+1 & \cdots & n \\ m+2 & \cdots & n & | & 1 & \cdots & m+1 \end{pmatrix}$$

9.1 Discrete Fourier Transform (DFT)

Dividing $[0, 2\pi]$ into n pieces $0 =: \theta_0, \theta_1, ..., \theta_n =: 2\pi$,

$$\begin{pmatrix} \hat{f}_0^n \\ \vdots \\ \hat{f}_{n-1}^n \end{pmatrix} = \frac{1}{\sqrt{n}} Q_n \begin{pmatrix} f(\theta_0) \\ \vdots \\ f(\theta_{n-1}) \end{pmatrix}$$

where matrix Q_n is

$$\frac{1}{\sqrt{n}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ 1 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \cdots & \omega^{(n-1)^2} \end{pmatrix} \quad \text{where } \omega = e^{i\frac{2\pi}{n}}$$

this matrix is unitary. i.e. $Q_n^{-1} = Q_n^*$.
Corollary: $f_n(\theta_j) := \sum_{k=0}^{n-1} \hat{f}_k^n e^{ik\theta_j} = f(\theta_j)$, same applies to Fourier series (a sum from $-m$ to m instead). This means correct interpolation at θ_j.

9.2 Tricks for finding Fourier coefficients

Find expression for $\hat{f}_k$ first, then consider $0 \le k \le n-1$ (or use $-n \le k \le -1$, choose the convenient range) $\forall m \in \mathbb{Z}$,

$$\hat{f}_{k+mn}^n = \sum_{p=-\infty}^{\infty} \hat{f}_{k+pn}$$

By a famous integral in complex analysis, if $k \in \mathbb{Z}, C = \{e^{i\theta} : \theta \in [0, 2\pi]\}$

$$\oint_C \frac{1}{z^k}\, dz = \begin{cases} 2\pi i & \text{if } z = -1 \\ 0 & \text{otherwise} \end{cases}$$

substitution $z = e^{-i\theta}$ or $z = e^{i\theta}$ helps you with lots of integrals. e.g.

$$\cos\theta = \frac{e^{i\theta} + e^{-i\theta}}{2}$$

use $z = e^{i\theta}$ for $\frac{b}{b - ce^{i\theta}}$ where $b > c$, and $z = e^{-i\theta}$ for $b < c$. Taylor series can be helpful here.

10 Orthogonal polynomials

Graded polynomials A sequence of polynomials $(p_n)_{n\ge 0}$ is s.t. $\deg(p_n) = n$, denote $p_n := c_n x^n + O(x^{n-1})$ where $c_n \neq 0$.
Given integrable weight function $w(x)$ s.t. $w(x) > 0$ when $x \in (a, b)$,

$$\langle f, g\rangle := \int_a^b f(x)g(x)w(x)\, dx$$

Orthogonal Polynomials(OPs) Graded polynomials $(p_n)_{n\ge 0}$ s.t. $\langle p_n, p_m\rangle = 0$ when $m \neq n$.
Note $\|p_n\|^2 := \langle p_n, p_n\rangle = \int_a^b p_n^2(x)w(x)\, dx > 0$. And orthogonal $(q_n)_{n\ge 0}$ is orthonormal if $\|q_n\|^2 = 1$. Perform Gram-Schmidt process on $\{1, x, ..., x^n\}$ to obtain a unique sequence of monic OPs, or use three-term recurrence.

Proposition (Base expansion)**.** *If* $\deg r(x) = n$*, and* $(p_n)_{n\ge 0}$ *are OPs, then*

$$r(x) = \sum_{k=0}^{n} \frac{\langle p_k, r\rangle}{\|p_k\|^2} p_k(x)$$

i.e. $\{p_0, ..., p_n\}$ *is an orthogonal basis of set of all polynomials with degree* $\le n$*.*

Proposition (Orthogonal to lower degree)**.** *Fix $n \in \mathbb{N}$ and weight $w(x)$. Given OPs $(p_k)_{0\le k\le n}$. For any polynomial $p(x)$,*

$$\begin{cases} \deg p = n \\ \langle p, r\rangle = 0 \text{ whenever } \deg r < n \end{cases} \iff p(x) = cp_n(x)$$

.

Generating OPs using three-term recurrence
First, determine $p_0(x)$ and weight function, then

$$a_n = \frac{\langle xp_n, p_n\rangle}{\|p_n\|^2}, \quad b_n = \frac{\langle xp_n, p_{n+1}\rangle}{\|p_{n+1}\|^2}, \quad c_{n-1} = \frac{\|p_n\|^2}{\|p_{n-1}\|^2} b_{n-1}$$

(set $b_n = 1$ to get monic OPs.) and

$$b_0 p_1(x) = xp_0(x) - a_0 p_0(x), \quad b_n p_{n+1}(x) = xp_n(x) - c_{n-1}p_{n-1}(x) - a_n p_n(x)$$

Special cases:

- If (p_n) is monic, $b_n = 1$.
- If (p_n) is orthonormal, $c_n = b_n$.
- If $w(x)$ defined on $[-U, U]$ is even function, $a_n = 0$. (e.g. two kinds of Chebyshev, Legendre)

Matrix form three-term recurrence let $P(x) := (p_0(x) \mid p_1(x) \mid p_2(x) \mid p_3(x) \cdots)$,

$$xP(x) = P(x)X$$

where matrix X is

$$\begin{pmatrix} a_0 & c_0 & & \\ b_0 & a_1 & c_1 & \\ & b_1 & a_2 & \ddots \\ & & \ddots & \ddots \end{pmatrix}$$

X^T is called Jacobi matrix. If OPs are orthonormal, X will be symmetric. If we are dealing with general degree n polynomial $f(x) = \sum_{k=0}^{n} d_k p_k(x)$, then $f(x) = P(x)\boldsymbol{d}$ where $\boldsymbol{d} = \begin{pmatrix} d_0 & d_1 & \cdots & d_n \end{pmatrix}^T$, and

$$xf(x) = P(x)X\boldsymbol{d}$$

and given polynomial $a(x)$, we have $a(x)f(x) = P(x)a(X)\boldsymbol{d}$ where $a(X)$ is the polynomial $a(x)$ applied to matrix X.

Given basic low degree polynomial e.g. $f(x) = x^2$, expand f into $\sum d_k p_k(x)$ by $d_k = \langle f, p_k\rangle / \|p_k\|^2$. Then for any polynomial $a(x)$, $a(x)f(x) = P(x)a(X)\boldsymbol{d}$.

Orthonormal case If given OPs (p_k) with Jacobi X, Jacobi of (q_k) ($q_k := \frac{p_k}{\|p_k\|}$) is $D^{-1}XD$ where $D = diag(1/\|p_1\|, \cdots 1/\|p_n\|)$. So $\hat{a}_n = a_n$, $\hat{b}_n = \sqrt{b_n c_n}$.

10.1 Classical polynomials

Chebyshev Polynomial(first kind)
$w(x) = \frac{1}{\sqrt{1-x^2}}$ on $[-1,1]$, $T_0(x) = 1$, $T_n(x) = 2^{n-1}x^n + O(x^{n-1})$.

$T_n(\cos(\theta)) = \cos n\theta$, so $T_n(x) = \cos(n \arccos x)$. (trick $x = \cos\theta$ is very useful)
Roots of T_n: $\cos\left(\frac{(j-1/2)\pi}{n}\right)$ for $j = 1, ..., n$.
Three-term recurrence: $xT_0(x) = T_1(x), \quad T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$.

Note if $f(x) = \sum_{k=0}^{\infty} a_k T_k(x)$, then

$$f(\cos\theta) = \sum_{k=0}^{\infty} a_k \cos k\theta = \sum_{k=0}^{\infty} a_k \left(\frac{e^{ik\theta} + e^{-ik\theta}}{2}\right)$$

$$= \sum_{k=-\infty}^{-1} \frac{a_k}{2} e^{ik\theta} + a_0 + \sum_{k=1}^{\infty} \frac{a_k}{2} e^{ik\theta}$$

so $a_0 = \hat{f}_0, a_k = 2\hat{f}_k$. where $\hat{f}_k$ are the Fourier coefficients.

Chebyshev Polynomial(second kind)
$w(x) = \sqrt{1-x^2}$ on $[-1,1]$, $U_0(x) = 1$, $U_n(x) = 2^n x^n + O(x^{n-1})$.
Exact formulae:

$$U_n(x) = \frac{\sin(n+1)\theta}{\sin\theta}$$

Roots of U_n: $\cos\left(\frac{k\pi}{n+1}\right)$ for $k = 1, ..., n$.
Three term recurrence:

$$xU_0(x) = \frac{1}{2}U_1(x), \quad xU_n(x) = \frac{1}{2}U_{n-1}(x) + \frac{1}{2}U_{n+1}(x)$$

Fourier results may be helpful:

$$\int_{-\pi}^{\pi} \sin(n\theta)\sin(m\theta)\, d\theta = \int_{-\pi}^{\pi} \cos(n\theta)\cos(m\theta)\, d\theta = \pi\delta_{m,n}$$

Legendre polynomials
$w(x) = 1$ on $[-1,1]$, $P_n(x) = \frac{1}{2^n}\binom{2n}{n} x^n + O(x^{n-1})$.
Rodriguez formulae:

$$P_n(x) = \frac{1}{(-2)^n n!}\left(\frac{d}{dx}\right)^n (1-x^2)^n$$

Steps of proof:

- Verify such P_n are graded polynomials.

- orthogonal to all lower degrees on $[-1, 1]$. (Using integration by part and the fact that for any polynomial p, first $n-1$ derivatives of p^n has factor p.)
- have correct normalising constants $k_n = \frac{1}{2^n}\binom{2n}{n}$. (Differentiate n times, but only taking care of highest power of x)

First two terms:

$$P_0(x) = 1, P_1(x) = x, \quad P_n(x) = \frac{(2n)!}{2^n(n!)^2}x^n - \frac{(2n-2)!}{2^n(n-2)!(n-1)!}x^{n-2} + O(x^{n-4})$$

Three-term recurrence formulae:

$$xP_0(x) = P_1(x), \quad (2n+1)xP_n(x) = nP_{n-1}(x) + (n+1)P_{n+1}(x)$$

proof: Let $r(x) := (2n+1)xP_n(x) - nP_{n-1}(x) - (n+1)P_{n+1}(x)$ (orthogonal to all polynomials $O(x^{n-2})$), prove coefficients for x^{n+1}, x^n, x^{n-1} of r are 0.

Name	$w(x)$	Interval	Leading coefficient	Three-term Recurrence	Exact Express/Rodriguez
Chebyshev(first) T_n	$\frac{1}{\sqrt{1-x^2}}$	$[-1,1]$	2^{n-1} if $n \neq 0$ 1 if $n = 0$	$b_0 = 1, a_0 = 0$ for $n \geq 1$, $b_n = c_{n-1} = 1/2$, $a_n = 0$	$\cos(n \arccos(x))$
Chebyshev(second) U_n	$\sqrt{1-x^2}$	$[-1,1]$	2^n	$b_0 = 1/2, a_0 = 0$ for $n \geq 1$, $b_n = c_{n-1} = 1/2$, $a_n = 0$	$\frac{\sin(n+1)\theta}{\sin\theta}$
Legendre P_n	1	$[-1,1]$	$\frac{(2n)!}{2^n(n!)^2}$	$b_0 = 1, a_0 = 0$ for $n \geq 1$, $b_n = \frac{n+1}{2n+1}, c_{n-1} = \frac{n}{2n+1}, a_n = 0$	$\frac{1}{(-2)^n n!}\left(\frac{d}{dx}\right)^n (1-x^2)^n$
Hermite H_n	e^{-x^2}	$\mathbb{R}$	2^n	$b_n = 1/2, c_{n-1} = n, a_n = 0$	$(-1)^n \exp(x^2)\frac{d^n}{dx^n}\exp(-x^2)$
Laguerre $L_n^{(\alpha)}$ $\alpha > -1$	$e^{-x}x^\alpha$	$(0,\infty)$	$\frac{(-1)^n}{n!}$	$b_n = -(n+1), c_{n-1} = -(n+\alpha)$ $a_n = 2n+\alpha+1$	$\frac{1}{n!e^{-x}x^\alpha}\frac{d^n}{dx^n}\left(e^{-x}x^{n+\alpha}\right)$
Jacobi $P_n^{(a,b)}$ $a, b > -1$	$(1-x)^a(1+x)^b$	$[-1,1]$	$\frac{(n+a+b+1)_n}{2^n n!}$	very complex	$\frac{1}{(-2)^n n! w(x)} \times ...$ $\frac{d^n}{dx^n}\left(w(x)(1-x^2)^n\right)$
Ultraspherical $C_n^{(\lambda)}$ $\lambda \neq 0, \lambda > -1/2$	$(1-x^2)^{\lambda-1/2}$	$[-1,1]$	$\frac{2^n(\lambda)_n}{n!}$	$b_n = \frac{n+1}{2(n+\lambda)}, c_{n-1} = \frac{n+2\lambda-1}{2(n+\lambda)}$ $a_n = 0$	$\frac{(2\lambda)_n}{(-2)^n(\lambda+1/2)_n n! w(x)} \times ...$ $\frac{d^n}{dx^n}\left(w(x)(1-x^2)^n\right)$

10.2 List of classical polynomials

This section provides first few polynomials of each family and constants to turn them into orthonormal polynomials.
Chebyshev's(first kind)

$$T_n : 1,\ x,\ 2x^2-1,\ 4x^3-3x,\ 8x^4-8x^2+1,\ 16x^5-20x^3+5x, ...$$

$$q_0(x) = \frac{1}{\sqrt{\pi}}, \quad \text{for } n>0, q_n(x) = T_n(x)\frac{\sqrt{2}}{\sqrt{\pi}}$$

Chebyshev's(second kind):

$$U_n : 1,\ 2x,\ 4x^2-1,\ 8x^3-4x,\ 16x^4-12x^2+1,\ 32x^5-32x^3+6x, ...$$

$$q_n(x) = U_n(x)\frac{\sqrt{2}}{\sqrt{\pi}}$$

Legendre:

$$P_n : 1,\ x,\ \frac{3}{2}x^2-\frac{1}{2},\ \frac{5}{2}x^3-\frac{3}{2}x,\ \frac{35}{8}x^4-\frac{15}{4}x^2+\frac{3}{8}, ...$$

$$q_n(x) = P_n(x)\frac{\sqrt{2n+1}}{\sqrt{2}}$$

Hermite:

$$H_n : 1,\ 2x,\ 4x^2-2,\ 8x^3-12x,\ 16x^4-48x^2+12,\ 32x^5-160x^3+120x$$

$$q_n(x) = H_n(x)\frac{1}{\sqrt{\pi^{1/2}2^n n!}}$$

Laguerre:

$$q_n(x) = L_n^{(\alpha)}(x)\frac{\sqrt{n!}}{\sqrt{\Gamma(n+\alpha+1)}}$$

11 Interpolation polynomials

Given discting $x_1, ..., x_n \in \mathbb{R}$ and samples $f_1, ..., f_n \in \mathbb{R}$, degree $n-1$ interpolatory polynomial $p(x)$ is a polynomial s.t. $p(x_i) = f_i, \deg p = n-1$.
Assume $p(x) = \sum_{k=0}^{n-1} c_k x^k$,

$$\begin{pmatrix} p(x_1) \\ \vdots \\ p(x_n) \end{pmatrix} = \begin{pmatrix} 1 & x_1 & \cdots & x_1^{n-1} \\ 1 & x_2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^{n-1} \end{pmatrix} \begin{pmatrix} c_0 \\ \vdots \\ c_{n-1} \end{pmatrix}$$

the Vandermonde matrix V is invertible. Degree $n-1$ interpolation polynomial exists and is unique.

Theorem (Lagrange's interpolation)**.** *Let*

$$l_k(x) := \prod_{j=1 \ j\neq k}^{n} \frac{x - x_j}{x_k - x_j}$$

by uniqueness of interpolatory polynomial, $p(x) = f_1 l_1(x) + \cdots + f_n l_n(x)$.

Best choice of $x_1, \cdots x_n$: roots of $q_n(x)$ (where (q_n) are orthonormal OPs). For all OP $p_n(x)$, there are exactly n distinct roots.

Truncated Jacobi Matrix

$$X_n := \begin{pmatrix} a_0 & b_0 & & \\ b_0 & \ddots & \ddots & \\ & \ddots & a_{n-2} & b_{n-2} \\ & & b_{n-2} & a_{n-1} \end{pmatrix}$$

zeros of q_n are the Eigenvalues of X_n i.e. $Q_n^T X_n Q_n = diag(x_1, \cdots, x_n)$, where

$$Q_n = \begin{pmatrix} q_0(x_1) & \cdots & q_0(x_n) \\ \vdots & \ddots & \vdots \\ q_{n-1}(x_1) & \cdots & q_{n-1}(x_n) \end{pmatrix} \begin{pmatrix} \alpha_1^{-1} & & \\ & \ddots & \\ & & \alpha_n^{-1} \end{pmatrix}$$

$$\alpha_j := \sqrt{q_0(x_j)^2 + \cdots + q_{n-1}(x_j)^2}$$

Q_n is orthogonal.

Normalise OPs before use (as truncated Jacobi is symmetric).

11.1 Interpolation to Quadrature

Quadrature from Lagrange interpolation
Target: approximate $\int_a^b f(x)w(x)\ dx$ by $\sum_{j=1}^n w_j f(x_j)$ where $x_j, w_j \in \mathbb{R}$. Lagrange interpolation $\sum_{j=1}^n f(x_j) l_j(x)$ yields

$$\sum_{j=1}^{n} \underbrace{\int_a^b w(x) l_j(x)\ dx}_{=:w_j} f(x_j) =: \sum_{n}^{w,\boldsymbol{x}} [f]$$

find $\int_a^b x^k w(x)\ dx$ for $1 \leq k \leq n-1$ to calculate w_j. Note if p is polynomial with $\deg p < n$,

$$\sum_{n}^{w,\boldsymbol{x}} [p] = \int_a^b p(x)w(x)\ dx$$

Gaussian Quadrature
Given family of orthonormal polynomials $(q_n)_{n\geq 0}$ and its roots $x_1, ..., x_n$, Gaussian quadrature is:

$$\sum_n^w [f] := \sum_{j=1}^n w_j f(x_j)$$

where $w_j := \frac{1}{\alpha_j^2} = \frac{1}{q_0(x_j)^2+\cdots+q_{n-1}(x_j)^2}$.
Another expression of w_j:

$$w_j = \int_a^b w(x)\, dx\, Q_n[1,j]^2, \quad Q_n[1,j] = \frac{q_0(x_j)}{\alpha_j}$$

Proposition (Discrete Orthogonality)**.** *For $0 \leq l, m \leq n-1$, we have*

$$\sum_n^w [q_l q_m] = \delta_{lm} = \langle q_l, q_m \rangle$$

.
Interpolation by Gaussian Quadrature

$$f_n(x) := \sum_{k=0}^{n-1} c_k^n q_k(x)$$

where $c_k^n := \sum_n^w [f q_k]$. Note f_n depends on f.
Equivalent form (for calculations):

$$\begin{pmatrix} w_1 q_0(x_1) & \cdots & w_n q_0(x_n) \\ \vdots & & \vdots \\ w_1 q_{n-1}(x_1) & \cdots & w_n q_{n-1}(x_n) \end{pmatrix} \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix} = \begin{pmatrix} c_0^n \\ \vdots \\ c_{n-1}^n \end{pmatrix}$$

where c_k^n are the coefficients in $f_n(x)$, $w_j = \frac{1}{q_0(x_j)^2+\cdots+q_{n-1}(x_j)^2}$

Theorem. *$f_n(x_j) = f(x_j)$ for $j = 1, ..., n$*

Corollary. $\sum_n^w [f] = \sum_n^{w,\boldsymbol{x}} [f]$.

Exactness of Gaussian quadrature for polynomials extends up to degree $2n-1$. (instead of $n-1$)