Basic identities of matrix/vector ops

$(A+B)^T=A^T+B^T$, $(AB)^T=B^TA^T$, $(A^{-1})^T=(A^T)^{-1}$, $(AB)^{-1}=B^{-1}A^{-1}$

For $A\in\mathbb{R}^{m\times n}$, A_{ij} is the i-th **ROW** then j-th **COLUMN**

$(A^T)_{ij}=A_{ji}$, $(AB)_{ij}=A_{i*}\cdot B_{*j}=\sum_k A_{ik}B_{kj}$

$(Ax)_i=A_{i*}\cdot x=\sum_j A_{ij}x_j$, $x^Ty=y^Tx=x\cdot y=\sum_i x_iy_i$

$x^TAx=\sum_i\sum_j A_{ij}x_ix_j$

Scalar-multiplication + addition distributes over:
- **column-blocks** => $\lambda A+B=\lambda[A_1|\dots|A_c]+[B_1|\dots|B_c]=[\lambda A_1+B_1|\dots|\lambda A_c+B_c]$
- **row-blocks** => $\lambda A+B=\lambda[A_1;\dots;A_r]+[B_1;\dots;B_r]=[\lambda A_1+B_1;\dots;\lambda A_r+B_r]$

Matrix-multiplication distributes over:
- **column-blocks** => $AB=A[B_1|\dots|B_p]=[AB_1|\dots|AB_p]$
- **row-blocks** => $AB=[A_1;\dots;A_p]B=[A_1B;\dots;A_pB]$
- **outer-product sum** => $AB=[A_1|\dots|A_p][B_1;\dots;B_p]=\sum_{i=1}^p A_iB_i$
- e.g. for $A=[a_1|\dots|a_n]$, $B=[b_1;\dots;b_n]$ => $AB=\sum_i a_ib_i$

What is a projection
- A **projection** $\pi:V\to V$ is an **endomorphism** such that $\pi\circ\pi=\pi$ i.e. it leaves its image unchanged (its idempotent)
- A **square matrix** P such that $P^2=P$ is called a **projection matrix**
- It is called an **orthogonal projection matrix** if $P^2=P=P^*$ (conjugate-transpose)
- Eigenvalues of a **projection matrix** must be 0 or 1
- Because $\pi:V\to V$ is a **linear map**, its **image space** $U=\mathrm{im}(\pi)$ and **null space** $W=\ker(\pi)$ are **subspaces** of V
- π is the **identity operator** on U
- The **linear map** $\pi^*=I_V-\pi$ is **also** a projection with $W=\mathrm{im}(\pi^*)=\ker(\pi)$ and $U=\ker(\pi^*)=\mathrm{im}(\pi)$, i.e. they swapped
- ∗ π is a projection **along** W **onto** U
- ∗ π^* is a projection **along** U **onto** W
- ∗ π^* is the **identity operator** on W
- V can be decomposed as $V=U\oplus W$ meaning every vector $x\in V$ can be uniquely written as $x=u+w$
- ∗ $u\in U$ and $u=\pi(x)$
- ∗ $w\in W$ and $w=x-\pi(x)=(I_V-\pi)(x)=\pi^*(x)$
- An **orthogonal projection** further satisfies $U\perp W$ i.e. the **image** and **kernel** of π are **orthogonal subspaces**
- infact they are eachother's **orthogonal compliments**, i.e. $U^\perp=W$, $W^\perp=U$ (*because finite-dimensional vectorspaces*)
- so we have $\pi(x)\cdot\pi(y)=\pi(x)\cdot y=x\cdot\pi(y)$
- or equivalently, $\pi(x)\cdot(y-\pi(y))=(x-\pi(x))\cdot\pi(y)=0$

Projection properties
- By Cauchy–Schwarz inequality we have $\|\pi(x)\|\le\|x\|$
- The **orthogonal projection onto the line** containing vector u is $\mathrm{proj}_u=\hat u\hat u^T$, which can also be written as $\mathrm{proj}_u(v)=\frac{u}{u\cdot u}u$
- $\hat u=\frac{u}{\|u\|}$ so $\hat u$ a **unit vector on the line** containing $\hat u$
- So we get $\mathrm{proj}_u(v)=\hat u\hat u^Tv=\frac{1}{\|u\|\|u\|}uu^Tv=\frac{1}{\|u\|^2}u(u\cdot v)=\frac{u\cdot v}{u\cdot u}u$
- A special case of $\pi(x)\cdot(y-\pi(y))=0$ is $u\cdot(v-\mathrm{proj}_u(v))=0$ since $\mathrm{proj}_u(u)=u$
- If $U\subseteq\mathbb{R}^n$ is a k-dimensional subspace with **orthonormal basis (ONB)** $\langle u_1,\dots,u_k\rangle\in\mathbb{R}^n$
- Let $U=[u_1|\dots|u_k]\in\mathbb{R}^{n\times k}$ be the matrix of columns $u_1,\dots,u_k$
- Then **orthogonal projection onto the subspace** U is $\pi_U=UU^T$
- Can be rewritten as $\pi_U(v)=\sum_i(u_i\cdot v)u_i$
- If $\langle u_1,\dots,u_k\rangle$ is **not orthonormal**, then "normalizing factor" $(U^TU)^{-1}$ is added => $\pi_U=U(U^TU)^{-1}U^T$
- ∗ For **line subspaces** $U=\mathrm{span}\{u\}$, we have $(U^TU)^{-1}=(u^Tu)^{-1}=1/(u\cdot u)=1/\|u\|^2$

Gram-Schmidt method to generate orthonormal basis from any linearly independent vectors
- Gram-Schmidt is **iterative** [[#What is a projection|projection]] => we use **current** j-**dim subspace**, to get **next** $(j+1)$-**dim subspace**
- Assume **orthonormal basis (ONB)** $\langle q_1,\dots,q_j\rangle\in\mathbb{R}^m$ for j-**dim subspace** $U_j\subset\mathbb{R}^m$
- ∗ Let $Q_j=[q_1|\dots|q_j]\in\mathbb{R}^{m\times j}$ be the matrix of columns $q_1,\dots,q_j$
- ∗ $P_j=Q_jQ_j^T$ is [[#Projection properties|orthogonal projection]] **onto** U_j
- ∗ $P_\perp=I_m-Q_jQ_j^T$ is [[#Projection properties|orthogonal projection]] **onto** $(U_j)^\perp$ (*orthogonal compliment*)
- Assume $a_{j+1}\notin U_j$ => unique decomposition $a_{j+1}=v_{j+1}+u_{j+1}$
- ∗ $v_{j+1}=P_j(a_{j+1})\in U_j$ => discard it!!
- ∗ $u_{j+1}=P_\perp(a_{j+1})\in(U_j)^\perp$ => we're after this!!
- ∗ $q_{j+1}=\hat u_{j+1}$ => we have ***next*** **ONB** $\langle q_1,\dots,q_{j+1}\rangle$ for U_{j+1} => start next iteration
- ∗ $u_{j+1}=(I_m-Q_jQ_j^T)a_{j+1}=a_{j+1}-Q_jc_j$ where $c_j=[q_1\cdot a_{j+1},\dots,q_j\cdot a_{j+1}]^T$
- ∗ **Notice**: $Q_jc_j=\sum_{i=1}^j(q_i\cdot a_{j+1})q_i=\sum_{i=1}^j\mathrm{proj}_{q_i}(a_{j+1})$, so rewrite as $u_{j+1}=a_{j+1}-\sum_{i=1}^j(q_i\cdot a_{j+1})q_i=a_{j+1}-\sum_{i=1}^j\mathrm{proj}_{q_i}(a_{j+1})$
- Let $a_1,\dots,a_n\in\mathbb{R}^m$ $(m\ge n)$ be linearly independent, i.e. basis of n-**dim subspace** $U_n=\mathrm{span}\{a_1,\dots,a_n\}$
- We apply Gram-Schmidt to build **ONB** $\langle q_1,\dots,q_n\rangle\in\mathbb{R}^m$ for $U_n\subset\mathbb{R}^m$
- $j=1$ => $u_1=a_1$ and $q_1=\hat u_1$, i.e. ***start of iteration***
- $j=2$ => $u_2=a_2-(q_1\cdot a_2)q_1$ and $q_2=\hat u_2$ ***and so on...***
- Linear independence ***guarantees*** that $a_{j+1}\notin U_j$
- **For exams**: more efficient to compute as
  1) Gather $Q_j=[q_1|\dots|q_j]\in\mathbb{R}^{m\times j}$ **all-at-once**
  2) Compute $c_j=[q_1\cdot a_{j+1},\dots,q_j\cdot a_{j+1}]^T\in\mathbb{R}^j$ **all-at-once**
  3) Compute $Q_jc_j\in\mathbb{R}^m$, and subtract from a_{j+1} **all-at-once**

Properties of dot product & (induced) norm
- $x^Ty=y^Tx=x\cdot y=\sum_i x_iy_i$
- $x\cdot y=\|a\|\|b\|\cos\hat{xy}$
- $x\cdot y=y\cdot x$
- $x\cdot(y+z)=x\cdot y+x\cdot z$
- $\alpha x\cdot y=\alpha(x\cdot y)$
- $x\cdot x=\|x\|^2=0\iff x=0$
- for $x\ne0$, we have $x\cdot y=x\cdot z\iff x\cdot(y-z)=0$
- $|x\cdot y|\le\|x\|\|y\|$ (**Cauchy-Schwartz inequality**)
- $\|u+v\|^2+\|u-v\|^2=2\|u\|^2+2\|v\|^2$ (**parallelogram law**)
- $\|u+v\|\le\|u\|+\|v\|$ (**triangle inequality**)
- $u\perp v\iff\|u+v\|^2=\|u\|^2+\|v\|^2$ (**pythagorean theorem**)
- $\|c\|^2=\|a\|^2+\|b\|^2-2\|a\|\|b\|\cos\hat{ba}$ (**law of cosines**)

Properties of linear independence
- Let $v_1,\dots,v_k\in\mathbb{R}^m$ be linearly independent
- $v_i\ne0$ (proof by contradiction)

Transformation matrix of linear map w.r.t. bases
- For linear map $f:\mathbb{R}^n\to\mathbb{R}^m$, ordered bases $\langle b_1,\dots,b_n\rangle\in\mathbb{R}^n$ and $\langle c_1,\dots,c_m\rangle\in\mathbb{R}^m$
- $A=F_{CB}\in\mathbb{R}^{m\times n}$ is the **transformation-matrix** of f w.r.t. to bases B and C
- $f(b_j)=\sum_{i=1}^m A_{ij}c_i$ => each b_j basis gets mapped to a linear combination of $\sum_i a_ic_i$ bases
- If f^{-1} exists (*i.e. its bijective and* $m=n$) then $(F_{CB})^{-1}=F^{-1}{}_{BC}$ (*where* $F^{-1}{}_{BC}$ *is the* ***transformation-matrix*** *of* f^{-1})
- The transformation matrix of the identity map is called change-in-basis matrix
- The identity matrix I_m represents $\mathrm{id}_{\mathbb{R}^m}$ w.r.t. the standard basis $E_m=\langle e_1,\dots,e_m\rangle$ i.e. $I_m=I_{EE}$
- If $B=\langle b_1,\dots,b_m\rangle$ is a basis of $\mathbb{R}^m$, then $I_{EB}=[b_1|\dots|b_m]$ is the transformation matrix from B to E
- $I_{BE}=(I_{EB})^{-1}$, so => $F_{CB}=I_{CE}F_{EE}I_{EB}$
- Dot-product uniquely determines a vector w.r.t. to basis
- If $a_j=x\cdot b_j$ then $x=\sum_i a_ib_i$, we call a the coordinate-vector of x w.r.t. to B
- **Rank-nullity theorem**: $\dim(\mathrm{im}(f))+\dim(\ker(f))=\mathrm{rk}(A)+\dim(\ker(A))=n$, i.e. properties of transformation-matrices/liner maps correspond
- f is injective/monomorphism **iff** $\ker(f)=\{0\}$ **iff** A is full-rank

Orthogonality concepts
- $u\perp v\iff u\cdot v=0$ i.e. u and v are orthogonal
- u and v are orthonormal **iff** $u\perp v$, $\|u\|=1=\|v\|$
- $A\in\mathbb{R}^{n\times n}$ is orthogonal **iff** $A^{-1}=A^T$
- Columns of $A=[a_1|\dots|a_n]$ are orthonormal basis (ONB) $C=\langle a_1,\dots,a_n\rangle\in\mathbb{R}^n$, so $A=I_{EC}$ is change-in-basis matrix
- Orthogonal transformations preserve lengths/angles/distances => $\|Ax\|=\|x\|_2$, $\widehat{AxAy}=\hat{xy}$
- ∗ Therefore can be seen as a succession of reflections and planar rotations
- $\det(A)=1$ or $\det(A)=-1$, and all **eigenvalues** of A are s.t. $|\lambda|=1$
- $A\in\mathbb{R}^{m\times n}$ is semi-orthogonal **iff** $A^TA=I$ or $AA^T=I$
- If $n>m$ then **all** m **rows** are orthonormal vectors
- If $m>n$ then **all** n **columns** are orthonormal vectors
- $U\perp V\subset\mathbb{R}^n\iff u\cdot v=0$ for all $u\in U$, $v\in V$, i.e. they are **orthogonal subspaces**
- **Orthogonal compliment** of $U\subset\mathbb{R}^n$ is the subspace $U^\perp=\{x\in\mathbb{R}^n\mid\forall y\in\mathbb{R}^n:x\perp y\}=\{x\in\mathbb{R}^n\mid\forall y\in\mathbb{R}^n:\|x\|\le\|$
- $\mathbb{R}^n=U\oplus U^\perp$ and $(U^\perp)^\perp=U$ (*because finite dimensional*)
- $U\perp V\iff U^\perp\perp V$ and vice-versa (*because finite dimensional*)
- $Y\subseteq X\Rightarrow X^\perp\subseteq Y^\perp$ and $X\cap X^\perp=\{0\}$
- Any $x\in\mathbb{R}^n$ can be uniquely decomposed into $x=x_i+x_k$, $x_i\in U$ and $x_k\in U^\perp$
- For matrix $A\in\mathbb{R}^{m\times n}$ and for row-space $R(A)$, column-space $C(A)$ and null space $\ker(A)$
- $R(A)^\perp=\ker(A)$ and $C(A)^\perp=\ker(A^T)$
- Any $b\in\mathbb{R}^m$ can be uniquely decomposed into two
- ∗ $b=b_i+b_k$ where $b_i\in C(A)$ and $b_k\in\ker(A^T)$
- ∗ $b=b_i+b_k$ where $b_i\in R(A)$ and $b_k\in\ker(A)$

Back-to-basics: revise a-levels trigonometry
- $a^2+b^2=c^2$ (**Pythagorean theorem**)
- $c=\sqrt{a^2+b^2-2ab\cdot\cos\gamma}$ (**law of cosines**)
- $\frac{\sin A}{a}=\frac{\sin B}{b}=\frac{\sin C}{c}$ (**law of sines**)
- TODO: angles, triangles, identities, etc.

Vector norms (beyond euclidean)
- **vector norms** are such that: $\|x\|=0\iff x=0$, $|\lambda x|=|\lambda|\|x\|$, $\|x+y\|\le\|x\|+\|y\|$
- ℓ_p norms: $\|x\|_p=\left(\sum_{i=1}^n|x_i|^p\right)^{1/p}$
- $p=1$: $\|x\|_1=\sum_{i=1}^n|x_i|$
- $p=2$: $\|x\|_2=\sqrt{\sum_{i=1}^n x_i^2}=\sqrt{x\cdot x}$
- $p=\infty$: $\|x\|_\infty=\lim_{p\to\infty}\|x\|_p=\max_{1\le i\le n}|x_i|$
- Any two norms in $\mathbb{R}^n$ are equivalent, meaning there exist $r>0,s>0$ such that: $\forall x\in\mathbb{R}^n, r\|x\|_a\le\|x\|_b\le s\|x\|_a$

  $\|x\|_\infty\le\|x\|_2\le\|x\|_1$
- Equivalence of ℓ_1,ℓ_2 and ℓ_∞ => $\|x\|_2\le\sqrt n\|x\|_\infty$

  $\|x\|_1\le\sqrt n\|x\|_2$
- Induce **metric** $d(x,y)=\|y-x\|$, has additional properties:
- Translation invariance: $d(x+w,y+w)=d(x,y)$
- Scaling: $d(\lambda x,\lambda y)=|\lambda|d(x,y)$

Matrix norms
- **Matrix norms** are such that: $\|A\|=0\iff A=0$, $|\lambda A|=|\lambda|\|A\|$, $\|A+B\|\le\|A\|+\|B\|$
- Matrices $\mathbb{R}^{m\times n}$ are a vector space so **matrix norms** ***are vector norms***, all results apply
- **Sub-multiplicative matrix norm** (assumed by default) is also such that $\|AB\|\le\|A\|\|B\|$
- Common matrix norms, for some $A\in\mathbb{R}^{m\times n}$:
- $\|A\|_1=\max_j\|A_{*j}\|_1$
- $\|A\|_2=\sigma_1(A)$ i.e. largest singular value of A (*square-root of [[tutorial 3#Singular Value Decomposition (SVD) & Singular Values|largest eigenvalue]] of* A^TA *or* AA^T)
- $\|A\|_\infty=\max_i\|A_{i*}\|_1$, note that $\|A\|_1=\|A^T\|_\infty$
- **Frobenius norm**: $\|A\|_F=\sqrt{\sum_{i=1}^m\sum_{j=1}^n|A_{ij}|^2}$
- A matrix norm $\|\cdot\|$ on $\mathbb{R}^{m\times n}$ is **consistent** with the vector norms $\|\cdot\|_a$ on $\mathbb{R}^n$ and $\|\cdot\|_b$ on $\mathbb{R}^m$ if
- for all $A\in\mathbb{R}^{m\times n}$ and $x\in\mathbb{R}^n$ => $\|Ax\|_b\le\|A\|\|x\|_a$
- If $a=b$, $\|\cdot\|$ is **compatible** with $\|\cdot\|_a$
- Frobenius norm is **consistent** with ℓ_2 norm i.e. $\|Av\|_2\le\|A\|_F\|v\|_2$
- For a vector norm $\|\cdot\|$ on $\mathbb{R}^n$, the **subordinate matrix norm** $\|\cdot\|$ on $\mathbb{R}^{n\times n}$ is $\|A\|=\max\{\|Ax\|:x\in\mathbb{R}^n,\|x\|=1\}$
- Alternative expressions:
  $\|A\|=\max\{\|Ax\|:x\in\mathbb{R}^n,\|x\|=1\}$
  $=\max\left\{\frac{\|Ax\|}{\|x\|}:x\in\mathbb{R}^n,x\ne0\right\}$
  $=\max\{\|Ax\|:x\in\mathbb{R}^n,\|x\|\le1\}$
- Vector norms are **compatible** with their **subordinate matrix norms**
- For $p=1,2,\infty$ matrix norm $\|\cdot\|_p$ is **subordinate** to the vector norm $\|\cdot\|_p$ (*and thus* ***compatible*** *with*)

Trick for proofs: "picking a vector"
Often times you might want to pick a vector to **prove a bound**: say the index M is special (e.g. maybe $\|A_{M*}\|_1=\max_i\|A_{i*}\|_1$) - Then you could pick a vector x_M based **on a function of** M - e.g. $(x_M)_j=\mathrm{sgn}(A_{Mj})$ can help prove $x_M\cdot A_{M*}=\|A_{M*}\|_1$ -

e.g. $(x_M)_j=\begin{cases}1 & j=M,\\ 0 & j\ne M\end{cases}$ can help prove other properties

Properties of matrices
- Consider $A\in\mathbb{R}^{m\times n}$
- If $Ax=x$ for all x then $A=I$
- A is symmetric **iff** $A=A^T$
- A is Hermitian, iff $A=A^\dagger$, i.e. its equal to its conjugate-transpose
- AA^T and A^TA are **symmetric** (*and* ***positive semi-definite***)
- For real matrices, **Hermitian/symmetric** are equivalent conditions
- Every eigenvalue λ_i of **Hermitian** matrices is real
- ∗ and geometric multiplicity of λ_i = geometric multiplicity of λ_i
- ∗ and eigenvectors x_1,x_2 associated to distinct eigenvalues λ_1,λ_2 are **orthogonal**, i.e. $x_1\perp x_2$
- A is triangular **iff** all entries above (***lower-triangular***) or below (***upper-triangular***) the main diagonal are **zero**
- **Triangular matrices** => $|A|=\prod_i a_{ii}$, i.e. the product of diagonal elements
- A is diagonal **iff** $A_{ij}=0, i\ne j$, i.e. if all off-diagonal entries are zero
- *Sometimes refers to rectangular matrices*, but most often **square matrices**
- Written as $\mathrm{diag}_{m\times n}(\mathbf{a})=\mathrm{diag}_{m\times n}(a_1,\dots,a_p), p=\min(m,n)$, where $\mathbf{a}=[a_1,\dots,a_p]^T\in\mathbb{R}^p$ diagonal entries of A
- For $x\in\mathbb{R}^n$, $Ax=\mathrm{diag}_{m\times n}(a_1,\dots,a_p)[x_1\dots x_n]^T=[a_1x_1\dots a_px_p\ 0\dots0]$ (*if* $p=m$ *those tail-zeros don't exist*)
- Consider $\mathrm{diag}_{m\times n}(\mathbf{b})$, then $\mathrm{diag}_{m\times n}(\mathbf{a})+\mathrm{diag}_{m\times n}(\mathbf{b})=\mathrm{diag}_{m\times n}(\mathbf{a}+\mathbf{b})$
- Consider $\mathrm{diag}_{n\times k}(c_1,\dots,c_q), q=\min(n,k)$, then $\mathrm{diag}_{m\times n}(a_1,\dots,a_p)\mathrm{diag}_{n\times k}(c_1,\dots,c_q)=\mathrm{diag}_{m\times k}(a_1c_1,\dots$
- ∗ Where $r=\min(p,q)=\min(m,n,k)$, and $\mathbf{s}\in\mathbb{R}^s, s=\min(m,k)$
- **Inverse of square-diagonals** => $\mathrm{diag}(a_1,\dots,a_n)^{-1}=\mathrm{diag}(a_1^{-1},\dots,a_n^{-1})$, i.e. diagonals **cannot be zero** (*division by zero undefined*)
- **Determinant of square-diagonals** => $|\mathrm{diag}(a_1,\dots,a_n)|=\prod_i a_i$ (*since they are technically triangular matrices*)
- For square A, the **trace of** A is the **sum if its diagonals**, i.e. $\mathrm{tr}(A)$
- The **(column) rank** of A is number of linearly independent columns, i.e. $\mathrm{rk}(A)$
- I.e. its the **number of pivots** in **row-echelon-form**
- ∗ I.e. its the **dimension** of the **column-space** $\mathrm{rk}(A)=\dim(C(A))$
- ∗ I.e. its the **dimension** of the **image-space** $\mathrm{rk}(A)=\dim(\mathrm{im}(f_A))$ of linear map $f_A(x)=Ax$
- The **(row) rank** of A is number of linearly independent rows
- The **row/column ranks** are **always the same**, hence $\mathrm{rk}(A)=\dim(C(A))=\dim(R(A))=\dim(C(A^T))=\mathrm{rk}(A^T)$
- A is full-rank **iff** $\mathrm{rk}(A)=\min(m,n)$, i.e. its as linearly independent as possible
- Two matrices $A,\tilde A\in\mathbb{R}^{m\times n}$ are **equivalent** if there exist two invertible matrices $P\in\mathbb{R}^{m\times m}$ and $Q\in\mathbb{R}^{n\times n}$ such that $A=P\tilde AQ^{-1}$
- Two matrices $A,\tilde A\in\mathbb{R}^{n\times n}$ are **similar** if there exists an invertible matrix $P\in\mathbb{R}^{n\times n}$ such that $A=P\tilde AP^{-1}$
- **Similar** matrices are equivalent, with $Q=P$
- A is diagonalisable **iff** A is similar to some diagonal matrix D

Properties of determinants
- Consider $A\in\mathbb{R}^{n\times n}$, then $A_{ij}{}'\in\mathbb{R}^{(n-1)\times(n-1)}$ the (i,j)-**minor matrix** of A, obtained by deleting i-**th row** and j-**th column** from A
- Then we define **determinant** of A, i.e. $\det(A)=|A|$, as
- $\det(A)=\sum_{k=1}^n(-1)^{i+k}A_{ik}\det(A_{ik}{}')$, i.e. expansion along i-**th row** *(for any i)
- $\det(A)=\sum_{k=1}^n(-1)^{k+j}A_{kj}\det(A_{kj}{}')$, i.e. expansion along j-**th column** (*for any* j)
- When $\det(A)=0$ we call A a **singular matrix**
- Common determinants
- For $n=1$, $\det(A)=A_{11}$
- For $n=2$, $\det(A)=A_{11}A_{22}-A_{12}A_{21}$
- $\det(\mathbf{I}_n)=1$
- **Multi-linearity in columns/rows**: if $A=[a_1|\dots|a_j|\dots|a_n]=[a_1|\dots|\lambda x_i+\mu y_i|\dots|a_n]$ then $\det(A)=\lambda\det([a_1|\dots|x_i|\dots|a_n])+\mu\det([a_1|\dots|y_i|\dots|a_n])$
- And the exact same linearity property for **rows** $A=[a_1;\dots;a_n]$
- *Immediately* leads to: $|A|=|A^T|$, $|\lambda A|=\lambda^n|A|$, and $|AB|=|BA|=|A||B|$ (*for any* $B\in\mathbb{R}^{n\times n}$)
- **Alternating**: if any **two columns** of A are **equal** (*or any* ***two rows*** *of* A *are* ***equal***), then $|A|=0$ (its singular)
- *Immediately* from this (and **multi-linearity**) => if **columns** (*or rows*) are linearly-dependent (*some are* linear combinations of *others*) then $|A|=0$
- Stated in other terms => $\mathrm{rk}(A)<n\iff|A|=0$ <=> $\mathrm{RREF}(A)\ne I_n\iff|A|=0$ (*reduced row-echelon-form*) <=> $C(A)\ne\mathbb{R}^n\iff|A|=0$ (*column-space*)
- For more equivalence to the above, see invertible matrix theorem
- Interaction with **EROs**/**ECOs**:
- **Swapping** rows/columns **flips the sign**, e.g. $\det([a_1|\dots|a_i|\dots|a_j|\dots|a_n])=-\det([a_1|\dots|a_j|\dots|a_i|\dots|a_n])$
- **Scaling** a row/column by $\lambda\ne0$ will **scale the determinant** by λ (*by multi-linearity*)
- ∗ Remember to **scale by** λ^{-1} to **maintain equality**, i.e. $\det(A)=\lambda^{-1}\det([a_1|\dots|\lambda a_i|\dots|a_n])$
- **Addition** of rows/columns does **not change determinant**
- **Link to invertable matrices** => $|A^{-1}|=|A|^{-1}$ which means A is invertible $\iff|A|\ne0$ (*because division by zero undefined*), i.e. **singular matrices** are **not invertible**
- For block-matrices:
- $\det\begin{pmatrix}A & B\\ 0 & D\end{pmatrix}=\det(A)\det(B)=\det\begin{pmatrix}A & 0\\ C & D\end{pmatrix}$
- $\det\begin{pmatrix}A & B\\ C & D\end{pmatrix}=\det(A)\det(D-CA^{-1}B)=\det(D)\det(A-BD^{-1}C)$ if A or D are invertible, *respectively*
- **Sylvester's determinant theorem:** $\det(I_m+AB)=\det(I_n+BA)$
- **Matrix determinant lemma**:
- $\det(\mathbf{A}+\mathbf{u}\mathbf{v}^T)=(1+\mathbf{v}^T\mathbf{A}^{-1}\mathbf{u})\det(\mathbf{A})$
- $\det(\mathbf{A}+\mathbf{U}\mathbf{V}^T)=\det(\mathbf{I}_m+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U})\det(\mathbf{A})$
- $\det(\mathbf{A}+\mathbf{U}\mathbf{W}\mathbf{V}^T)=\det(\mathbf{W}^{-1}+\mathbf{V}^T\mathbf{A}^{-1}\mathbf{U})\det(\mathbf{W})\det(\mathbf{A})$

Tricks for computing determinant
- If block-triangular matrix then apply $\det\begin{pmatrix}A & B\\ 0 & D\end{pmatrix}=\det(A)\det(B)$
- If *close* to triangular matrix apply **EROs**/**ECOs** to get it there, then its just product of diagonals
- If **Cholesky/LU/QR** is possible and cheap then do it, then apply $|AB|=|A||B|$
- If all else fails, try to find row/column with **MOST** zeros
- Perform minimal **EROs/ECOs** to get that row/column to be **all-but-one** zeros
- ∗ Don't forget to keep track of sign-flipping & scaling-factors
- Do Laplace expansion along *that* row/column => notice **all-but-one** minor matrix determinants **go to zero**

Representing EROs/ECOs as transformation matrices
- For $A\in\mathbb{R}^{m\times n}$, suppose a sequence of:
- **EROs** transform $A\rightsquigarrow_{\mathrm{EROs}}A'$ => there is matrix R s.t. $RA=A'$
- **ECOs** transform $A\rightsquigarrow_{\mathrm{ECOs}}A'$ => there is matrix C s.t. $AC=A'$
- Both transform $A\rightsquigarrow_{\mathrm{EROs+ECOs}}A'$ => there are matrices R,C s.t. $RAC=A'$
- **FORWARD:** to compute these **transformation matrices**:
- Start with $[I_m\mid A\mid I_n]$, i.e. A and identity matrices
- For every **ERO** on A, do the same to **LHS** (i.e. I_m)
- For every **ECO** on A, do the same to **RHS** (i.e. I_n)
- Once done, you should get $[I_m\mid A\mid I_n]\rightsquigarrow[R\mid A'\mid C]$ with $RAC=A'$
- If the sequences of **EROs** and **ECOs** were $R_1,\dots,R_\lambda$ and $C_1,\dots,C_\mu$ *respectively*
- $R=R_\lambda\cdots R_1$ and $C=C_1\cdots C_\mu$, so $(R_\lambda\cdots R_1)A(C_1\cdots C_\mu)=A'$
- $R^{-1}=R_1^{-1}\cdots R_\lambda^{-1}$ and $C^{-1}=C_\mu^{-1}\cdots C_1^{-1}$, where R_i^{-1},C_j^{-1} are **inverse EROs/ECOs** *respectively*
- **BACKWARD:** once $R_1,\dots,R_\lambda$ and $C_1,\dots,C_\mu$ for which $RAC=A'$ are **known**, starting with $[I_m\mid A\mid I_n]$
- For $i=1\to\lambda$ perform R_i on A, perform $R_{\lambda-i+1}^{-1}$ on **LHS** (i.e. I_m)
- For $j=1\to\mu$ perform C_j on A, perform $C_{\mu-j+1}^{-1}$ on **RHS** (i.e. I_n)
- You should get $[I_m\mid A\mid I_n]\rightsquigarrow[R^{-1}\mid A'\mid C^{-1}]$ with $A=R^{-1}A'C^{-1}$
- You can mix-and-match the **forward/backward** modes
- i.e. inverse operations in inverse order for one, and operations in normal order for the other
- e.g. you can do $[I_m\mid A\mid I_n]\rightsquigarrow[R^{-1}\mid A'\mid C]$ to get $AC=R^{-1}A'$ => useful for LU factorization

Eigen-values/vectors
- Consider $A\in\mathbb{R}^{n\times n}$, non-zero $\mathbf{x}\in\mathbb{C}^n$ is an **eigenvector** with **eigenvalue** $\lambda\in\mathbb{C}$ for A if $A\mathbf{x}=\lambda\mathbf{x}$
- If $A\mathbf{x}=\lambda\mathbf{x}$ then $A(k\mathbf{x})=\lambda(k\mathbf{x})$ for $k\ne0$, i.e. $k\mathbf{x}$ is also an **eigenvector**
- A has at most n distinct **eigenvalues**
- The set of all eigenvectors associated with eigenvalue λ is called **eigenspace** E_λ of A
- $E_\lambda=\ker(A-\lambda I)$
- The **geometric multiplicity** of λ is $\dim(E_\lambda)=\dim(\ker(A-\lambda I))$
- The **spectrum** $Sp(A)=\{\lambda_1,\dots,\lambda_n\}$ of A is the set of all eigenvalues of A
- The **characteristic polynomial** of A is $P(\lambda)=|A-\lambda I|=\sum_{i=0}^n a_i\lambda^i$
- $a_0=|A|$, $a_{n-1}=(-1)^{n-1}\mathrm{tr}(A)$, $a_n=(-1)^n$
- $\lambda\in\mathbb{C}$ is eigenvalue of A **iff** λ is a root of $P(\lambda)$
- The **algebraic multiplicity** of λ is the **number of times** it is **repeated as root** of $P(\lambda)$
- $1\le$ geometric multiplicity of $\lambda\le$ algebraic multiplicity of λ
- Let $\lambda_1,\dots,\lambda_n\in\mathbb{C}$ be (*potentially non-distinct*) **eigenvalues** of A, with $\mathbf{x}_1,\dots,\mathbf{x}_n\in\mathbb{C}^n$ their **eigenvectors**
- $\mathrm{tr}(A)=\sum_i\lambda_i$ and $\det(A)=\prod_i\lambda_{ij}$
- A is diagonalisable **iff** there exist a basis of $\mathbb{R}^n$ consisting of $\mathbf{x}_1,\dots,\mathbf{x}_n$
- A is diagonalisable **iff** $r_i=g_i$, where r_i = geometric multiplicity of λ_i and g_i = geometric multiplicity of λ_i
- **Eigenvalues** of A^k are $\lambda_1,\dots,\lambda_n$
- Let $P=[\mathbf{x}_1|\dots|\mathbf{x}_n]$, then $AP=[\lambda_1\mathbf{x}_1|\dots|\lambda_n\mathbf{x}_n]=[\mathbf{x}_1|\dots|\mathbf{x}_n]\mathrm{diag}(\lambda_1,\dots,\lambda_n)=PD$
- => if P^{-1} exists then
- $A=PDP^{-1}$, i.e. A is diagonalisable
- $P=\mathbf{I}_{EB}$ is **change-in-basis** matrix for basis $B=\langle\mathbf{x}_1,\dots,\mathbf{x}_n\rangle$ of eigenvectors
- If $A=\mathbf{F}_{EE}$ is transformation-matrix of linear map f, then $\mathbf{F}_{EE}=\mathbf{I}_{EB}\mathbf{F}_{BB}\mathbf{I}_{BE}$
- **Spectral theorem**: if A is Hermitian then P^{-1} exists, so:
- If $\mathbf{x}_i,\mathbf{x}_j$ associated to different eigenvalues then $\mathbf{x}_i\perp\mathbf{x}_j$
- If associated to same eigenvalue λ then **eigenspace** E_λ has spanning-set $\{\mathbf{x}_{\lambda_i},\dots\}$
- ∗ $\mathbf{x}_1,\dots,\mathbf{x}_n$ are linearly independent => apply Gram-Schmidt $\mathbf{q}_{\lambda_i},\dots\leftarrow\mathbf{x}_{\lambda_i},\dots$
- ∗ Then $\{\mathbf{q}_{\lambda_i},\dots\}$ is orthonormal basis (ONB) of E_λ
- $Q=\langle\mathbf{q}_1,\dots,\mathbf{q}_n\rangle$ is an ONB of $\mathbb{R}^n$ => $\mathbf{Q}=[\mathbf{q}_1|\dots|\mathbf{q}_n]$ is orthogonal matrix i.e. $\mathbf{Q}^{-1}=\mathbf{Q}^T$
- $\mathbf{q}_1,\dots,\mathbf{q}_n$ are still eigenvectors of A => $A=\mathbf{QDQ}^T$ (***spectral decomposition***)
- $A=\mathbf{QDQ}^T$ can be interpreted as scaling in direction of its eigenvectors:
  1) Perform a succession of reflections/planar rotations to change coordinate-system
  2) Apply scaling by λ_i to each dimension $\mathbf{q}_i$
  3) Undo those reflections/planar rotations

Extension to $\mathbb{C}^n$
- Standard inner product: $\langle x,y\rangle=x^\dagger y=\sum_i\overline{x_i}y_i$
- **Conjugate-symmetric**: $\langle x,y\rangle=\overline{\langle y,x\rangle}$
- Standard (*induced*) norm: $\|x\|=\sqrt{\langle x,y\rangle}=\sqrt{x^\dagger y}$
- We can [[tutorial 1#Eigen-values/vectors|diagonalise]] real matrices in $\mathbb{C}$ which lets us diagonalise more matrices than before

Least Square Method
- If we are solving $A\mathbf{x}=\mathbf{b}$ and $\mathbf{b}\notin C(A)$, *i.e. no solution*, then **Least Square Method** is:
- Finding $\mathbf{x}$ which **minimizes** $\|A\mathbf{x}-\mathbf{b}\|_2$
- Recall for $A\in\mathbb{R}^{m\times n}$ [[tutorial 1#Orthogonality concepts|we have unique decomposition for any $\mathbf{b}\in\mathbb{R}^m$]]: $\mathbf{b}=\mathbf{b}_i+\mathbf{b}_k$
- ∗ where $\mathbf{b}_i\in C(A)$ and $\mathbf{b}_k\in\ker(A^T)$
- $\|A\mathbf{x}-\mathbf{b}\|_2$ is minimized $\iff\|A\mathbf{x}-\mathbf{b}_i\|_2=0\iff A\mathbf{x}=\mathbf{b}_i$
- $A^TA\mathbf{x}=A^T\mathbf{b}$ is the **normal equation** which gives solution to least square problem: $\|A\mathbf{x}-\mathbf{b}\|_2$ is minimized $\iff A\mathbf{x}=\mathbf{b}_i\iff A^TA\mathbf{x}=A^T\mathbf{b}$

Linear Regression
- Let $y=f(t)=\sum_{j=1}^n s_jf_j(t)$ be a **mathematical model**, where f_j are **basis functions** and s_j are **parameters**
- Let (t_i,y_i), $1\le i\le m, m\gg n$ be a set of **observations**, and $\mathbf{t},\mathbf{y}\in\mathbb{R}^m$ are vectors representing those **observations**
- $f_j(\mathbf{t})=[f_j(t_1),\dots,f_j(t_m)]^T$ is a vector **transformed under** f_j
- $A=[f_1(\mathbf{t})|\dots|f_n(\mathbf{t})]\in\mathbb{R}^{m\times n}$ is a matrix of columns
- $\mathbf{z}=[s_1,\dots,s_n]^T$ is vector of parameters
- Then we get equation $A\mathbf{z}=\mathbf{y}$ => **minimizing** $\|A\mathbf{z}-\mathbf{y}\|_2$ is the solution to Linear Regression
- So applying **LSM** to $A\mathbf{z}=\mathbf{y}$ is *precisely* what **Linear Regression** is
- We can use normal equations for this => $\|A\mathbf{z}-\mathbf{y}\|_2$ is minimized $\iff A^TA\mathbf{z}=A^T\mathbf{y}$
- Solution to **normal equations** unique **iff** A is full-rank, i.e. it has linearly-independent columns

Back to basics: multinomial expansion + manipulations on $\sum$ / $\prod$
- $$(x_1+x_2+\dots+x_m)^n=\sum_{\substack{k_1+k_2+\dots+k_m=n\\ k_1,k_2,\dots,k_m\ge0}}\binom{n}{k_1,k_2,\dots,k_m}x_1^{k_1}\cdot x_2^{k_2}\cdots$$
- where $\binom{n}{k_1,k_2,\dots,k_m}=\frac{n!}{k_1!k_2!\cdots k_m!}$
- TODO: figure out wtf going on here ![[Pasted image 20250414122252.png|500]] in 2nd tutorial

Express recursive sequence as non-recursive using eigenvalues
- For x_n recursive (*e.g.* $x_{n+1}=x_n+x_{n-1}$, $x_0=0$, $x_1=1$)
- Find A such that $[x_{n+1},x_n,\dots]^T=A[x_n,x_{n-1},\dots]^T$ (*e.g.* $[x_{n+1},x_n]^T=\begin{bmatrix}1 & 1\\ 1 & 0\end{bmatrix}[x_n,x_{n-1}]^T$)
- Find **initial vector** $I=[\dots,x_1,x_0]^T$ such that $[x_{n+1},x_n,\dots]^T=A^nI$ *(e.g. $[x_{n+1},x_n]^T=A^n[1,0]^T$)
- Find **eigenvalues/eigenvectors** of A, and use $A\mathbf{u}=\lambda\mathbf{u}\implies A^n\mathbf{u}=\lambda^n\mathbf{u}$ to write I as linear combination of eigenvectors
- Substitute that linear combination to get x_n as function of n alone

Positive (semi-)definite symmetric matrices
- Consider symmetric $A\in\mathbb{R}^{n\times n}$, i.e. $A=A^T$
- A is positive-definite **iff** $x^TAx>0$ for all $x\ne0$
- A is positive-definite **iff** all its eigenvalues are **strictly positive**
- A is positive-definite => all its diagonals are **strictly positive**
- A is positive-definite => $\max(A_{ii},A_{jj})>|A_{ij}|$, i.e. **strictly larger coefficient** on the diagonals
- A is positive-definite => all **upper-left submatrices** are *also* positive-definite
- **Sylvester's criterion**: A is positive-definite **iff** all **upper-left submatrices** have strictly positive determinant
- A is positive semi-definite **iff** $x^TAx\ge0$ for all x
- A is positive semi-definite **iff** all its eigenvalues are **non-negative**
- A is positive semi-definite => all its diagonals are **non-negative**
- A is positive semi-definite => $\max(A_{ii},A_{jj})\ge|A_{ij}|$, i.e. **no coefficient larger** than on the diagonals
- A is positive semi-definite => all **upper-left submatrices** are *also* positive semi-definite
- A is positive semi-definite => it has a [[tutorial 4#Cholesky Decomposition|Cholesky Decomposition]]
- For any $M\in\mathbb{R}^{m\times n}$, MM^T and M^TM are symmetric and **positive semi-definite**

Singular Value Decomposition (SVD) & Singular Values
- **Singular Value Decomposition** of $A\in\mathbb{R}^{m\times n}$ is *any decomposition* of the form $A=USV^T$, where
- [[tutorial 1#Orthogonality concepts|Orthogonal]] $U=[\mathbf{u}_1|\dots|\mathbf{u}_m]\in\mathbb{R}^{m\times m}$ and $V=[\mathbf{v}_1|\dots|\mathbf{v}_n]\in\mathbb{R}^{n\times n}$
- $S=\mathrm{diag}_{m\times n}(\sigma_1,\dots,\sigma_p)$ where $p=\min(m,n)$ and $\sigma_1\ge\dots\ge\sigma_p\ge0$
- $\sigma_1,\dots,\sigma_p$ are **singular values** of A,
- ∗ (*Positive*) **singular values** are (*positive*) **square-roots of eigenvalues** of AA^T or A^TA
- ∗ i.e. $\sigma_1^2,\dots,\sigma_p^2$ are **eigenvalues** of AA^T or A^TA
- ∗ $\|A\|_2=\sigma_1$ (*link to [[tutorial 1#Matrix norms|matrix norms]]*)
- Let $r=\mathrm{rk}(A)$, then number of strictly positive **singular values** is r
- i.e. $\sigma_1\ge\dots\ge\sigma_r>0$ and $\sigma_{r+1}=\dots=\sigma_p=0$
- $A=\sum_{i=1}^r\sigma_i\mathbf{u}_i\mathbf{v}_i^T$
- **SVD** is *similar* to [[tutorial 1#Eigen-values/vectors|spectral decomposition]], except it always exists
- If $n\le m$ then work with $A^TA\in\mathbb{R}^{n\times n}$:
- ∗ Obtain eigenvalues $\sigma_1^2\ge\dots\ge\sigma_n^2\ge0$ of A^TA
- ∗ Obtain **orthonormal** eigenvectors $\mathbf{v}_1,\dots,\mathbf{v}_n\in\mathbb{R}^n$ of A^TA (apply **normalization** e.g. **Gram-Schmidt**!!!! to **eigenspaces** E_{σ_i})
- ∗ $V=[\mathbf{v}_1|\dots|\mathbf{v}_n]\in\mathbb{R}^{n\times n}$ is [[tutorial 1#Orthogonality concepts|orthogonal]] so $V^T=V^{-1}$
- ∗ $r=\mathrm{rk}(A)$ = no. of strictly +ve σ_i
- ∗ Let $\mathbf{u}_i=\frac{1}{\sigma_i}A\mathbf{v}_i$ then $\mathbf{u}_1,\dots,\mathbf{u}_r\in\mathbb{R}^m$ are **orthonormal** (*therefore linearly independent*)
- · The [[tutorial 1#Orthogonality concepts|orthogonal compliment]] of $\mathrm{span}\{\mathbf{u}_1,\dots,\mathbf{u}_r\}$ => $\mathrm{span}\{\mathbf{u}_1,\dots,\mathbf{u}_r\}^\perp=\mathrm{span}\{\mathbf{u}_{r+1},\dots,\mathbf{u}_m\}$
- · Solve for unit-vector $\mathbf{u}_{r+1}$ s.t. it is orthogonal to $\mathbf{u}_1,\dots,\mathbf{u}_r$
- · Then solve for unit-vector $\mathbf{u}_{r+2}$ s.t. it is orthogonal to $\mathbf{u}_1,\dots,\mathbf{u}_{r+1}$
- · And so on… [[#Tricks Computing orthonormal vector-set extensions|see this for better methods]]
- ∗ $U=[\mathbf{u}_1|\dots|\mathbf{u}_m]\in\mathbb{R}^{m\times m}$ is [[tutorial 1#Orthogonality concepts|orthogonal]] so $U^T=U^{-1}$
- ∗ $S=\mathrm{diag}_{m\times n}(\sigma_1,\dots,\sigma_n)$, **AND DONE!!!**
- If $m<n$ then let $B=A^T$
- ∗ apply above method to B => $B=A^T=USV^T$
- ∗ $A=B^T=VS^TU^T$

Tricks: Computing orthonormal vector-set extensions
- You have **orthonormal** vectors $\mathbf{u}_1,\dots,\mathbf{u}_r\in\mathbb{R}^m$ => need to **extend** to **orthonormal** vectors $\mathbf{u}_1,\dots,\mathbf{u}_m\in\mathbb{R}^m$
- Special case => two 3D vectors => use **cross-product** => $a\times b\perp a,b$
- Extension via standard basis $\mathbf{I}_m=[\mathbf{e}_1|\dots|\mathbf{e}_m]$ using [[tutorial 1#Gram-Schmidt method to generate orthonormal basis from any linearly independent vectors|(tweaked) GS]]:
- **Choose candidate vector**: just work through $\mathbf{e}_1,\dots,\mathbf{e}_m$ sequentially starting from $\mathbf{e}_1$ => denote the current candidate $\mathbf{e}_k$
- **Orthogonalize**: Starting from $j=r$ going to $j=m$ with each iteration => with current orthonormal vectors $\mathbf{u}_1,\dots,\mathbf{u}_j$
- ∗ Notice $\langle\mathbf{u}_1,\dots,\mathbf{u}_j\rangle$ is
- ∗ Compute $\mathbf{w}_{j+1}=\mathbf{e}_k-\sum_{i=1}^j(\mathbf{e}_k\cdot\mathbf{u}_i)\mathbf{u}_i=\mathbf{e}_k-\sum_{i=1}^j(\mathbf{u}_i)_k\mathbf{u}_i$
- ∗
- ∗ **NOTE:** $\mathbf{e}_k\cdot\mathbf{u}_i=(\mathbf{u}_i)_k$ i.e. k-th component of $\mathbf{u}_i$
- · Can rewrite as $\mathbf{w}_{j+1}=\mathbf{e}_k-U_j[(\mathbf{u}_1)_k,\dots,(\mathbf{u}_j)_k]^T=\mathbf{e}_k-[\mathbf{u}_1|\dots|\mathbf{u}_j][(\mathbf{u}_1$
- · The above matrix form can be more convenient to calculate with
- ∗ If $\mathbf{w}_{j+1}=0$ then $\mathbf{e}_k\in\mathrm{span}\{\mathbf{u}_1,\dots,\mathbf{u}_j\}$ => discard $\mathbf{w}_{j+1}$, choose next candidate $\mathbf{e}_{k+1}$, try this step again
- **Normalize**: $\mathbf{w}_{j+1}\ne0$ so compute unit vector $\mathbf{u}_{j+1}=\hat{\mathbf{w}}_{j+1}$
- **Repeat**: keep repeating the above steps, now with **new** orthonormal vectors $\mathbf{u}_1,\dots,\mathbf{u}_{j+1}$

SVD Application: Principal Component Analysis (PCA)
- Assume $A_{\mathrm{uncentered}}\in\mathbb{R}^{m\times n}$ represent m **samples** of n-**dimensional data** (*with* $m\ge n$)
- **Data centering**: subtract mean of each column from that column's elements
- Let the **resulting matrix** be $A\in\mathbb{R}^{m\times n}$, who's **columns** have **mean zero**
- **PCA** is done on **centered** data-matrices like A:
- SVD exists i.e. $A=USV^T$ and $r=\mathrm{rk}(A)$
- Let $A=[\mathbf{r}_1;\dots;\mathbf{r}_m]$ be rows $\mathbf{r}_1,\dots,\mathbf{r}_m\in\mathbb{R}^n$ => each row *corresponds to* a sample
- Let $A=[\mathbf{c}_1|\dots|\mathbf{c}_n]$ be columns $\mathbf{c}_1,\dots,\mathbf{c}_n\in\mathbb{R}^m$ => each column *corresponds to* one dimension of the data
- Let $X_1,\dots,X_n$ be **random variables** where each X_i *corresponds* to column $\mathbf{c}_i$
- i.e. each X_i *corresponds* to i-th component of data
- i.e. random vector $X=[X_1,\dots,X_n]^T$ models the data $\mathbf{r}_1,\dots,\mathbf{r}_m$
- **Co-variance matrix** of X is $\mathrm{Cov}(A)=\frac{1}{m-1}A^TA$ => $(A^TA)_{ij}=(A^TA)_{ji}=\mathrm{Cov}(X_i,X_j)$
- $\mathbf{v}_1,\dots,\mathbf{v}_r$ (*columns of* V) are **principal axes** of A
- Let $w\in\mathbb{R}^n$ be some unit-vector => let $\alpha_j=\mathbf{r}_j\cdot w$ be the **projection/coordinate** of sample $\mathbf{r}_j$ onto w
- **Variance (Bessel's correction)** of $\alpha_1,\dots,\alpha_m$ is $\mathrm{Var}_w=\frac{1}{m-1}\sum_j\alpha_j^2=\frac{1}{m-1}w^T\left(\sum_j\mathbf{r}_j^T\mathbf{r}_j\right)w=\frac{1}{m-1}w^TA$
- **First (principal) axis defined** => $w_{(1)}=\arg\max_{\|w\|=1}w^TA^TAw=\arg\max_{\|w\|=1}(m-1)\mathrm{Va}$
- i.e. $w_{(1)}$ the direction that maximizes variance Var_w, i.e. maximizes variance of **projections on line $\mathbb{R}w_{(1)}$
- $\sigma_1\mathbf{u}_1,\dots,\sigma_r\mathbf{u}_r$ (*columns of* US) are **principal components/scores** of A
- Recall: $A=\sum_{i=1}^r\sigma_i\mathbf{u}_i\mathbf{v}_i^T$ with $\sigma_1\ge\dots\ge\sigma_r>0$, so that relates principal axes and principal components
- **Data compression**: If $\sigma_1\gg\sigma_2$ then **compress** A by projecting in direction of principal component =>

$\approx \sigma_1 u_1 v_1^T$

Generalised Eigenvectors

- **TODO: this seems low-priority, do when have time**
- gen-eigenvectors
- jordan chains (common cases)
  https://www.youtube.com/watch?v=aTh6peJfAQQ&list=PLJMXxxxxx...gQ0RW5&index=3
- JNF, form
- some tips on how to solve common cases
- JNF decomposition and basis of generalized eigenvectors

General: visualizing transformations of matrices

- TODO: do when have time -> where standard basis-vectors map to
- TODO: rotations, reflections, scaling, shearing, etc

Cholesky Decomposition

- Consider **positive (semi-)definite** $A \in \mathbb{R}^{n \times n}$
- **Cholesky Decomposition** is $A = LL^T$ where L is lower-triangular
  - For positive semi-definite => **always exists**, but **non-unique**
  - For positive-definite => **always *uniquely* exists** s.t. diagonals of L are positive
- Finding a Cholesky Decomposition:
  - Compute LL^T and solve $A = LL^T$ by matching terms
  - For square roots always pick positive
  - If there is **exact solution** then **positive-definite**
  - If there are **free variables** at the end, then **positive semi-definite**
  - ∗i.e. the decomposition is a **solution-set** parameterized on **free variables**
  - ∗e.g. $\begin{bmatrix}1&1&1\\1&1&1\\1&1&2\end{bmatrix} = LL^T$ where $L = \begin{bmatrix}1&0&0\\1&0&0\\1&c&\sqrt{1-c^2}\end{bmatrix}, c \in [0,1]$
- If $A = LL^T$ you can use [[#Forward/backward substitution|forward/backward substitution]] to **solve equations**
  - For $Ax = b$ => let $y = L^T x$
  - Solve $Ly = b$ by forward substitution to **find** y
  - Solve $L^T x = y$ by backward substitution to **find** x
- For $n = 3$ => $L = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33}\end{bmatrix}$

$$LL^T = \begin{bmatrix} l_{11}^2 & l_{11}l_{21} & l_{11}l_{31} \\ l_{11}l_{21} & l_{21}^2 + l_{22}^2 & l_{21}l_{31}+l_{22}l_{32} \\ l_{11}l_{31} & l_{21}l_{31}+l_{22}l_{32} & l_{31}^2+l_{32}^2+l_{33}^2 \end{bmatrix}$$

Forward/backward substitution

- **Forward substitution**: for lower-triangular $L = \begin{bmatrix}\ell_{1,1} & & 0\\ \vdots & \ddots & \\ \ell_{n,1} & \cdots & \ell_{n,n}\end{bmatrix}$
  - For $Lx = b$, just **solve** the first row $\ell_{1,1}x_1 = b_1 \implies x_1 = \frac{b_1}{\ell_{1,1}}$ and **substitute down**
  - Then **solve** the second row $\ell_{2,1}x_1 + \ell_{2,2}x_2 = b_2 \implies x_2 = \frac{b_2 - \ell_{2,1}x_1}{\ell_{2,2}}$ and **substitute down**
  - ...and so on until all x_i are solved
- **Backward substitution**: for upper-triangular $U = \begin{bmatrix}u_{1,1} & \cdots & u_{1,n}\\ & \ddots & \vdots \\ 0 & & u_{n,n}\end{bmatrix}$
  - For $Ux = b$, just **solve** the last row $u_{n,n}x_n = b_n \implies x_n = \frac{b_n}{u_{n,n}}$ and **substitute up**
  - Then **solve** the second-to-last row $u_{n-1,n-1}x_{n-1} + u_{n-1,n}x_n = b_{n-1} \implies x_{n-1} = \frac{b_{n-1} - u_{n-1,n}x_n}{u_{n-1,n-1}}$ and **substitute up**
  - ...and so on until all x_i are solved

Thin QR Decomposition w/ Gram-Schmidt (GS)

- Consider **full-rank** $A = [a_1 | \dots | a_n] \in \mathbb{R}^{m \times n}$ $(m \ge n)$, i.e. $a_1, \dots, a_n \in \mathbb{R}^m$ are linearly independent
- Apply [[tutorial 1#Gram-Schmidt method|Gram-Schmidt method]] to generate orthonormal basis from any linearly independent vectors|GS]] i.e. $q_1, \dots, q_n \leftarrow GS(a_1, \dots, a_n)$ to build **ONB** $(q_1, \dots, q_n) \in \mathbb{R}^m$ for $C(A)$
- **For exams**: more efficient to compute as $u_{j+1} = a_j - Q_j c_j$
  1) Gather $Q_j = [q_1 | \dots | q_j] \in \mathbb{R}^{m \times j}$ **all-at-once**
  2) Compute $c_j = [q_1 \cdot a_{j+1}, \dots, q_j \cdot a_{j+1}]^T \in \mathbb{R}^j$ **all-at-once**
  3) Compute $Q_j c_j \in \mathbb{R}^m$, and subtract from a_{j+1} **all-at-once**
- Can now rewrite $a_j = \sum_{i=1}^{j} (q_i \cdot a_j) q_i = Q_j c_j$
- Choose $Q = Q_n = [q_1 | \dots | q_n] \in \mathbb{R}^{m \times n}$, notice its [[tutorial 1#Orthogonality concepts|semi-orthogonal]] since $Q^T Q = I_n$
- Notice => $a_j = Q_j c_j = Q[q_1 \cdot a_j, \dots, q_j \cdot a_j, 0, \dots, 0]^T = Qr_j$
- Let $R = [r_1 | \dots | r_n] \in \mathbb{R}^{n \times n}$ => $A = QR = Q\begin{bmatrix} q_1^T a_1 & \dots & q_1^T a_n \\ & \ddots & \vdots \\ 0 & & q_n^T a_n \end{bmatrix}$, notice its [[tutorial 1#Properties of matrices|upper-triangular]]

Full QR Decomposition

- Consider **full-rank** $A = [a_1 | \dots | a_n] \in \mathbb{R}^{m \times n}$ $(m \ge n)$, i.e. $a_1, \dots, a_n \in \mathbb{R}^m$ are linearly independent
- Apply [[#Thin QR Decomposition w/ Gram-Schmidt (GS)|thin QR decomposition]] to obtain:
  - ONB $(q_1, \dots, q_n) \in \mathbb{R}^m$ for $C(A)$
  - Semi-orthogonal $Q_1 = [q_1 | \dots | q_n] \in \mathbb{R}^{m \times n}$ and upper-triangular $R_1 \in \mathbb{R}^{n \times n}$, where $A = Q_1 R_1$
- [[tutorial 3#Tricks Computing orthonormal vector-set extensions|Compute basis extension]] to obtain remaining $q_{n+1}, \dots, q_m \in \mathbb{R}^m$, where $(q_1, \dots, q_m)$ is **ONB** for $\mathbb{R}^m$
  - Notice $(q_{n+1}, \dots, q_m)$ is **ONB** for $C(A)^\perp = \ker(A^T)$
  - Let $Q_2 = [q_{n+1} | \dots | q_m] \in \mathbb{R}^{m \times (m-n)}$, let $Q = [Q_1 | Q_2] \in \mathbb{R}^{m \times m}$, let $R = [R_1 ; 0_{m-n}] \in \mathbb{R}^{m \times n}$
- Then **full QR decomposition** is $A = QR = [Q_1 | Q_2]\begin{bmatrix}R_1 \\ 0_{m-n}\end{bmatrix} = Q_1 R_1$
  - Q is **orthogonal**, i.e. $Q^{-1} = Q^T$, so its a basis transformation
  - $\mathrm{proj}_{C(A)} = Q_1 Q_1^T$, $\mathrm{proj}_{C(A)^\perp} = Q_2 Q_2^T$ are [[tutorial 1#Projection properties|orthogonal projections]] **onto** $C(A)$, $C(A)^\perp = \ker(A^T)$ *respectively*
  - Notice: $QQ^T = I_m = Q_1 Q_1^T + Q_2 Q_2^T$
- **Generalizable** to $A \in \mathbb{C}^{m \times n}$ by changing transpose to conjugate-transpose
  - Inner product $x^T y$ => $x^\dagger y$
  - Orthogonal matrix $U^{-1} = U^T$ => unitary matrix $U^{-1} = U^\dagger$
  - ∗For orthogonal $U = [u_1 | \dots | u_k] \in \mathbb{R}^{m \times k}$ => $\mathrm{proj}_U = UU^T$ projects **onto** $C(U)$
  - ∗For unitary $U = [u_1 | \dots | u_k] \in \mathbb{C}^{m \times k}$ => $\mathrm{proj}_U = UU^\dagger$ projects **onto** $C(U)$
  - And so on...

Lines and hyperplanes in Euclidean space $\mathbb{E}^n (= \mathbb{R}^n)$

- Consider **standard Euclidean space** $\mathbb{E}^n (= \mathbb{R}^n)$
  - with standard basis $(e_1, \dots, e_n) \in \mathbb{R}^n$
  - with standard origin $0 \in \mathbb{R}^n$
- A **line** $L = \mathbb{R}n + c$ is *characterized* by direction $n \in \mathbb{R}^n$ $(n \ne 0)$ and offset from origin $c \in L$
  - It is customary that:
    - ∗n is a **unit vector**, i.e. $\|n\| = \|\hat{n}\| = 1$
    - ∗$c \in L$ is **closest point to origin**, i.e. $c \perp n$
  - If $c \ne \lambda n$ => L **not** vector-subspace of $\mathbb{R}^n$
    - ∗i.e. $0 \notin L$, i.e. L doesn't go through the origin
    - ∗L is affine-subspace of $\mathbb{R}^n$
  - If $c = \lambda n$, i.e. $L = \mathbb{R}n$ => L is vector-subspace of $\mathbb{R}^n$
    - ∗i.e. $0 \in L$, i.e. L goes through the origin
    - ∗L has $\dim(L) = 1$ and orthonormal basis (ONB) $(\hat{n})$
- A **hyperplane** P is *characterized* by normal $n \in \mathbb{R}^n$ $(n \ne 0)$ and offset from origin $c \in P$
  - It *represents* an $(n-1)$**-dimensional slice** of the n**-dimensional space**
    - ∗**Points** are hyperplanes for $n = 1$
    - ∗**Lines** are hyperplanes for $n = 2$
    - ∗**Planes** are hyperplanes for $n = 3$
  - It is customary that:
    - ∗n is a **unit vector**, i.e. $\|n\| = \|\hat{n}\| = 1$
    - ∗$c \in P$ is **closest point to origin**, i.e. $c = \lambda n$
  - ∗With those => $P = \{x \in \mathbb{R}^n \mid x \cdot n = \lambda\}$
  - If $c \cdot n \ne 0$ => P **not** vector-subspace of $\mathbb{R}^n$
    - ∗i.e. $0 \notin P$, i.e. P doesn't go through the origin
    - ∗P is affine-subspace of $\mathbb{R}^n$
  - If $c \cdot n = 0$, i.e. $P = (\mathbb{R}n)^\perp$ => P is vector-subspace of $\mathbb{R}^n$
    - ∗i.e. $0 \in P$, i.e. P goes through the origin
    - ∗P has $\dim(P) = n - 1$
  - Notice $L = \mathbb{R}n$ and $P = (\mathbb{R}n)^\perp$ are orthogonal compliments, so:
    - $\mathrm{proj}_L = \hat{n}\hat{n}^T$ is orthogonal projection **onto** L (**along** P)
    - $\mathrm{proj}_P = \mathrm{id}_{\mathbb{R}^n} - \mathrm{proj}_L = I_n - \hat{n}\hat{n}^T$ is orthogonal projection **onto** P (**along** L)
    - $L = \mathrm{im}(\mathrm{proj}_L) = \ker(\mathrm{proj}_P)$ and $P = \ker(\mathrm{proj}_L) = \mathrm{im}(\mathrm{proj}_P)$
    - $\mathbb{R}^n = \mathbb{R}n \oplus (\mathbb{R}n)^\perp$, i.e. all vectors $v \in \mathbb{R}^n$ uniquely decomposed into $v = v_L + v_P$

Reflection w.r.t. hyperplanes and Householder Maps

- Two points $x, y \in \mathbb{E}^n$ are **reflections** w.r.t hyperplane $P = (\mathbb{R}n)^\perp + c$ if:
  1) The translation $\vec{xy} = y - x$ is **parallel** to normal n, i.e. $\vec{xy} = \lambda n$
  2) Midpoint $m = 1/2(x + y) \in P$ **lies** on P, i.e. $m \cdot n = c \cdot n$
- Suppose $P_u = (\mathbb{R}u)^\perp$ goes through the origin with unit normal $u \in \mathbb{R}^n$
- **Householder matrix** $H_u = I_n - 2uu^T$ is reflection w.r.t. hyperplane P_u
  - Recall: let $L_u = \mathbb{R}u$
  - ∗$\mathrm{proj}_{L_u} = uu^T$ and $\mathrm{proj}_{P_u} = I_n - uu^T$ => $H_u = \mathrm{proj}_{P_u} - \mathrm{proj}_{L_u}$
  - ∗**Visualize** as preserving component in P_u, then flipping component in L_u
  - H_u is involutory, orthogonal and symmetric, i.e. $H_u = H_u^{-1} = H_u^T$

Modified Gram-Schmidt

- Go check [[tutorial 1#Gram-Schmidt method to generate orthonormal basis from any linearly independent vectors|Classical GM]] first, as this is just an alternative computation method
- Let $P_{\perp q_j} = I_m - q_j q_j^T$ be **projector** onto [[tutorial 5#Lines and hyperplanes in Euclidean space $ mathbb{E} {n}(= mathbb{R} {n})$|hyperplane]] $(\mathbb{R}q_j)^\perp$, i.e. [[tutorial 5#Lines and hyperplanes in Euclidean space $ mathbb{E} {n}(= mathbb{R} {n})$|orthogonal compliment]] of line $\mathbb{R}q_j$
- Notice: $P_{\perp j} = I_m - Q_j Q_j^T = \prod_{i=1}^{j}(I_m - q_i q_i^T) = \prod_{i=1}^{j} P_{\perp q_i}$
  - ∗[[tutorial 1#Column-wise & row-wise matrix/vector ops|Outer-product sum equivalence]] => $Q_j Q_j^T = [q_1 | \dots | q_j][q_1^T; \dots; q_j^T] = \sum_{i=1}^{j} q_i q_i^T$
  - ∗For $i \ne k$, => $\prod_{i=1}^{j}(I_m - q_i q_i^T) = I_m - \sum_{i=1}^{j} q_i q_i^T = I_m - Q_j Q_j^T$
- Re-state: $u_{j+1} = (I_m - Q_j Q_j^T) a_{j+1}$ =>
  $u_{j+1} = \left(\prod_{i=1}^{j} P_{\perp q_i}\right) a_{j+1} = (P_{\perp q_j} \cdots P_{\perp q_1}) a_{j+1}$
- **Projectors** $P_{\perp q_1}, \dots, P_{\perp q_j}$ are iteratively applied to a_{j+1}, removing its components along q_1, then along q_2, *and so on...*
- Let $u_k^{(j)} = \left(\prod_{i=1}^{j} P_{\perp q_i}\right) a_k$, i.e. a_k *without* its components along $q_1, \dots, q_j$
- Notice: $u_j = u_j^{(j-1)}$, thus $q_j = \hat{u}_j = u_j^{(j-1)}/r_{jj}$ where $r_{jj} = \|u_j^{(j-1)}\|$
- Iterative step: $u_k^{(j)} = \left(P_{\perp q_j}\right) u_k^{(j-1)} = u_k^{(j-1)} - \left(q_j \cdot u_k^{(j-1)}\right) q_j$
  - i.e. each **iteration** j of MGS computes $P_{\perp q_j}$ *(and projections under it)* **in one go**
- At **start** of iteration $j \in 1..n$ we have ONB $q_1, \dots, q_{j-1} \in \mathbb{R}^m$ and residual $u_j^{(j-1)}, \dots, u_n^{(j-1)} \in \mathbb{R}^m$
  - Compute $r_{jj} = \|u_j^{(j-1)}\|$ => $q_j = u_j^{(j-1)}/r_{jj}$
  - For each $k \in (j+1)..n$ compute $r_{jk} = q_j \cdot u_k^{(j-1)}$ => $u_k^{(j)} = u_k^{(j-1)} - r_{jk} q_j$
  - We have **next ONB** $(q_1, \dots, q_j)$ and **next residual** $u_{j+1}^{(j)}, \dots, u_n^{(j)}$
  - NOTE: for $j = 1$ => $q_1, \dots, q_{j-1} = \emptyset$, i.e. we don't have any yet
- By **end** of iteration $j = n$, we have **ONB** $(q_1, \dots, q_n) \in \mathbb{R}^m$ of n**-dim subspace** $U_n = \mathrm{span}(a_1, \dots, a_n)$

$$A = [a_1 | \dots | a_n] = [q_1 | \dots | q_n] \begin{bmatrix} r_{11} & \dots & r_{1n} \\ & \ddots & \vdots \\ 0 & & r_{nn}\end{bmatrix} = QR$$

corresponds to [[tutorial 5#Thin QR Decomposition w/ Gram-Schmidt (GS)|thin QR decomposition]]
- Where $A \in \mathbb{R}^{m \times n}$ is full-rank, $Q \in \mathbb{R}^{m \times n}$ is semi-orthogonal, and $R \in \mathbb{R}^{n \times n}$ is upper-triangular

Classical vs. Modified Gram-Schmidt (for thin QR)

- These algorithms both compute [[tutorial 5#Thin QR Decomposition w/ Gram-Schmidt (GS)|thin QR decomposition]] ![[Pasted image 20250418034701.png|400]] ![[Pasted image 20250418034855.png|400]]
- Classical GS => j-th column of Q and the j-th column of R
- Modified GS => j-th column of Q and the j-th row of R
- Both have **flop (floating-point operation)** count of $O(2mn^2)$
  - NOTE: **Householder method** has $2(mn^2 - n^3/3)$ **flop** count, but better numerical properties
- Recall: $Q^\dagger Q = I_n$ => check for loss of orthogonality with $\|I_n - Q^\dagger Q\| = \text{loss}$
  - **Classical GS** => $\|I_n - Q^\dagger Q\| \approx \mathrm{Cond}(A)^2 \epsilon_{mach}$
  - **Modified GS** => $\|I_n - Q^\dagger Q\| \approx \mathrm{Cond}(A) \epsilon_{mach}$
  - NOTE: **Householder method** has $\|I_n - Q^\dagger Q\| \approx \epsilon_{mach}$

Multivariate Calculus

- Consider $f : \mathbb{R}^n \to \mathbb{R}$ => **when clear** write j**-th component** of input as j instead of x_j
- **Level curve** w.r.t. to $c \in \mathbb{R}$ is all points s.t. $f(x) = c$
  - Projecting level curves onto $\mathbb{R}^n$ gives **contour-map** of f
- n_k-**th order partial derivative** w.r.t i_k of ..., of n_1-**th order partial derivative** w.r.t i_1 of f is:

$$\frac{\partial^{n_k+\dots+n_1}}{\partial x_{i_k}^{n_k} \cdots \partial x_{i_1}^{n_1}} f = \partial_{i_k}^{n_k} \cdots \partial_{i_1}^{n_1} f = f_{i_1 \cdots i_k}^{(n_1, \dots, n_k)} = \left(f_{i_1 \cdots i_k}^{(n_1, \dots, n_k)}\right)$$

  - Overall, its an N**-th order partial derivative** where $N = \sum_k n_k$
- $\nabla f = [\partial_1 f, \dots, \partial_n f]^T$ is gradient of f => $(\nabla f)_i = \frac{\partial f}{\partial x_i}$
  - $\nabla^T f = (\nabla f)^T$ is transpose of ∇f, i.e. $\nabla^T f$ is row vector
- $D_u f(x) = \lim_{\delta \to 0} \frac{f(x + \delta u) - f(x)}{\delta}$ **directional-derivative** of f
  - It is rate-of-change in direction u, where $u \in \mathbb{R}^n$ is unit-vector
  - $D_u f(x) = \nabla f(x) \cdot u = \|\nabla f(x)\| \|u\| \cos(\theta)$ => $D_u f(x)$ **maximized** when $\cos\theta = 1$
  - i.e. when x, u are parallel => hence $\nabla f(x)$ is direction of **max.** rate-of-change
- $H(f) = \nabla^2 f = J(\nabla f)^T$ is the **Hessian** of f => $H(f)_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$
- f has **local minimum** at x_{loc} if there's radius $r > 0$ s.t. $\forall x \in B[r; x_{loc}]$ we have $f(x_{loc}) \le f(x)$
- f has **global minimum** x_{glob} if $\forall x \in \mathbb{R}^n$ we have $f(x_{glob}) \le f(x)$
- A local minimum satisfies optimality conditions:
  - ∗$\nabla f(x) = 0$, e.g. for $n = 1$ its $f'(x) = 0$
  - ∗$\nabla^2 f(x)$ is positive-definite, e.g. for $n = 2$ its $f''(x) > 0$
- Interpret $F : \mathbb{R}^n \to \mathbb{R}^m$ as m functions $F_i : \mathbb{R}^n \to \mathbb{R}$ *(one per output-component)*
  - $J(F) = [\nabla^T F_1; \dots; \nabla^T F_m]$ is **Jacobian matrix** of F => $J(F)_{ij} = \frac{\partial F_i}{\partial x_j}$

Conditioning

- A **problem** is some $f : X \to Y$ where X, Y are normed vector-spaces
  - A problem **instance** is f with fixed input $x \in X$, shortened to **just** "problem" *(with $x \in X$ implied)
  - δx is **small perturbation** of x => $\delta f = f(x + \delta x) - f(x)$
- A problem (instance) is:
  - ∗**Well-conditioned** if all **small** δx lead to **small** δf, i.e. if κ is **small** *(e.g. 1, 10, 10^2)*
  - ∗**Ill-conditioned** if some **small** δx lead to **large** δf, i.e. if κ is **large** *(e.g. 10^6, 10^{16})*
- ***Absolute* condition number** $\mathrm{cond}(x) = \hat{\kappa}(x) = \hat{\kappa}$ of f **at** x is
  - $\hat{\kappa} = \lim_{\delta \to 0} \sup_{\|\delta x\| \le \delta} \frac{\|\delta f\|}{\|\delta x\|}$ => for most problems simplified to $\hat{\kappa} = \sup_{\delta x} \frac{\|\delta f\|}{\|\delta x\|}$
  - If Jacobian $J_f(x)$ exists then $\hat{\kappa} = \|J_f(x)\|$, where matrix norm $\|\cdot\|$ induced by norms on X and Y
- ***Relative* condition number** $\kappa(x) = \kappa$ of f **at** x is
  - $\kappa = \lim_{\delta \to 0} \sup_{\|\delta x\| \le \delta} \left(\frac{\|\delta f\|}{\|f(x)\|} \Big/ \frac{\|\delta x\|}{\|x\|} \right)$ => for most problems simplified to $\kappa = \sup_{\delta x}\left(\frac{\|\delta f\|}{\|f(x)\|} \Big/ \frac{\|\delta x\|}{\|x\|} \right)$
  - If Jacobian $J_f(x)$ exists then $\kappa = \frac{\|J_f(x)\|}{\|f(x)\| / \|x\|}$
  - More important than $\hat{\kappa}$ for *numerical analysis*
- ***Matrix* condition number** $\mathrm{Cond}(A) = \kappa(A) = \|A\| \|A^{-1}\|$
  - => comes up so often that has its own name
  - $A \in \mathbb{C}^{m \times m}$ is well-conditioned if $\kappa(A)$ is **small**, ill-conditioned if **large**
  - $\kappa(A) = \kappa(A^{-1})$ and $\kappa(A) = \kappa(\gamma A)$
  - If $\|\cdot\| = \|\cdot\|_2$ then $\kappa(A) = \frac{\sigma_1}{\sigma_m}$
- For $A \in \mathbb{C}^{m \times n}$, the problem $f_A(x) = Ax$ has $\kappa = \|A\| \frac{\|x\|}{\|Ax\|}$ => if A^{-1} exists then $\kappa \le \mathrm{Cond}(A)$
- If $Ax = b$ problem of finding x given b is just $f_{A^{-1}}(b) = A^{-1}b$ => $\kappa = \|A^{-1}\| \frac{\|b\|}{\|x\|} \le \mathrm{Cond}(A)$
- For $b \in \mathbb{C}^m$, the problem $f_b(A) = A^{-1}b$ *(i.e. finding x in $Ax = b$)* has $\kappa = \|A\| \|A^{-1}\| = \mathrm{Cond}(A)$

Stability

- Given a problem $f : X \to Y$, an **algorithm** for f is $\tilde{f} : X \to Y$
  - $\tilde{f}$ is **computer implementation**, so inputs/outputs are FP
  - Input $x \in X$ is first rounded to $\mathrm{fl}(x)$, i.e. $\tilde{f}(x) = \tilde{f}(\mathrm{fl}(x))$
  - $\tilde{f}$ cannot be **continuous** *(for the most part)*
  - **Absolute error** => $\|\tilde{f}(x) - f(x)\|$, **relative error** => $\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|}$
- $\tilde{f}$ is **accurate** if $\forall x \in X$, $\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O(\epsilon_{mach})$
- $\tilde{f}$ is **stable** if $\forall x \in X$, $\exists \tilde{x} \in X$ s.t. $\frac{\|\tilde{f}(x) - f(\tilde{x})\|}{\|f(\tilde{x})\|} = O(\epsilon_{mach})$ and $\frac{\|\tilde{x} - x\|}{\|x\|} = O(\epsilon_{mach})$
  - i.e. nearly the right answer to nearly the right question
  - **outer-product** is stable
- $\tilde{f}$ is **backwards stable** if $\forall x \in X$, $\exists \tilde{x} \in X$ s.t. $\tilde{f}(x) = f(\tilde{x})$ and $\frac{\|\tilde{x} - x\|}{\|x\|} = O(\epsilon_{mach})$
  - i.e. exactly the right answer to nearly the right question, a **subset of stability**
  - $\oplus, \ominus, \otimes, \oslash$, **inner-product**, back-substitution w/ triangular systems, are backwards stable
  - If **backwards stable** $\tilde{f}$ and f has condition number $\kappa(x)$ then relative error $\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = O(\kappa(x)\epsilon_{mach})$
- Accuracy, stability, backwards stability are **norm-independent** for fin-dim X, Y

Big-O meaning for numerical analysis

- In complexity analysis $f(n) = O(g(n))$ as $n \to \infty$
- But in numerical analysis $f(\epsilon) = O(g(\epsilon))$ as $\epsilon \to 0$, i.e. $\limsup_{\epsilon \to 0} \|f(\epsilon)\| / \|g(\epsilon)\| < \infty$
  - i.e. $\exists C, \delta > 0$ s.t. $\forall \epsilon$, we have $0 < \|\epsilon\| < \delta \implies \|f(\epsilon)\| \le C\|g(\epsilon)\|$
- $O(g)$ is set of functions $\{f : \limsup_{\epsilon \to 0} \|f(\epsilon)\| / \|g(\epsilon)\| < \infty\}$
- **Smallness** partial order $O(g_1) \le O(g_2)$ defined by set-inclusion $O(g_1) \subseteq O(g_2)$
  - i.e. as $\epsilon \to 0$, $g_1(\epsilon)$ goes to zero **faster** than $g_2(\epsilon)$
  - *Roughly* same hierarchy as complexity analysis but **flipped** *(some break pattern)*
  - ∗e.g. $\dots, O(\epsilon^3) < O(\epsilon^2) < O(\epsilon) < O(1)$
- **Maximum**: $O(\max(|g_1|, |g_2|)) = O(g_2) \iff O(g_1) \le O(g_2)$
  - ∗e.g. $O(\max(\epsilon^k, \epsilon)) = O(\epsilon)$
- Using functions $f_1, \dots, f_n$ let $\Phi(f_1, \dots, f_n)$ be formula **defining some function**
  - Then $\Phi(O(g_1), \dots, O(g_n))$ is the class of functions $\{\Phi(f_1, \dots, f_n) : f_1 \in O(g_1), \dots, f_n \in O(g_n)\}$
  - ∗e.g. $\epsilon^{O(1)} = \{\epsilon^{f(\epsilon)} : f \in O(1)\}$
- General case: $\Phi_1(O(f_1), \dots, O(f_m)) = \Phi_2(O(g_1), \dots, O(g_n))$ means $\Phi_1(O(f_1), \dots, O(f_m)) \subseteq \Phi_2(O(g_1), \dots, O(g_n))$
  - ∗e.g. $\epsilon^{O(1)} = O(k^\epsilon)$ means $\{\epsilon^{f(\epsilon)} : f \in O(1)\} \subseteq O(k^\epsilon)$ *not necessarily true*
- Special case: $f = \Phi(O(g_1), \dots, O(g_n))$ means $f \in \Phi(O(g_1), \dots, O(g_n))$
  - ∗e.g. $(\epsilon + 1)^2 = \epsilon^2 + O(\epsilon)$ means $\epsilon \mapsto (\epsilon + 1)^2 \in \{\epsilon^2 + f \mid f \in O(\epsilon)\}$ *not necessarily true*
- Let $f_1 = O(g_1)$, $f_2 = O(g_2)$ and let $k \ne 0$ be a constant
  - $f_1 f_2 = O(g_1 g_2)$ and $f \cdot O(g) = O(fg)$
  - $f_1 + f_2 = O(\max(|g_1|, |g_2|))$ => **if** $g_1 = g_2 = g$ **then** $f_1 + f_2 = O(g)$
  - $O(|k| \cdot g) = O(g)$

Floating-point numbers

- Consider **base/radix** $\beta \ge 2$ *(typically 2)* and **precision** $t \ge 1$ *(24 or 53 for IEEE single/double precisions)*
- **Floating-point numbers** are discrete subset $\mathbb{F} = \{(-1)^S (m/\beta^t)\beta^e \mid 1 \le m \le \beta^t, S \in \mathbb{B}, m, e \in \mathbb{Z}\}$
  - S is sign-bit, m/β^t is **mantissa**, e is **exponent** *(8-bit for single, 11-bit for double)*
  - Equivalently, can restrict to $\beta^{t-1} \le m \le \beta^t - 1$ for unique m and e
  - $\mathbb{F} \subset \mathbb{R}$ is idealized *(ignores over/underflow)*, so is countably infinite and self-similar *(i.e. $\mathbb{F} = \beta\mathbb{F}$)*
  - For all $x \in \mathbb{R}$ there exists $\mathrm{fl}(x) \in \mathbb{F}$ s.t. $|x - \mathrm{fl}(x)| \le \epsilon_{mach}|x|$
  - ∗Equivalently $\mathrm{fl}(x) = x(1 + \delta)$, $|\delta| \le \epsilon_{mach}$
- **Machine epsilon** $\epsilon_{machine} = \epsilon_{mach} = \frac{1}{2}\beta^{1-t}$ is maximum relative gap between FPs
  - Half the gap between 1 and next largest FP
  - $2^{-24} \approx 5.96 \times 10^{-8}$ and $2^{-53} \approx 10^{-16}$ for single/double
- **FP arithmetic**: let $*, \circledast$ be real and floating counterparts of arithmetic operation
  - For $x, y \in \mathbb{F}$ we have $x \circledast y = \mathrm{fl}(x * y) = (x * y)(1 + \epsilon)$, $|\delta| \le \epsilon_{mach}$
  - ∗Holds for ***any*** arithmetic operation $\circledast = \oplus, \ominus, \otimes, \oslash$
  - Complex floats implemented pairs of real floats, so above applies complex ops as-well
  - ∗Caveat: $\epsilon_{mach} = \frac{1}{2}\beta^{1-t}$ must be scaled by factors on the order of $2^{3/2}, 2^{5/2}$ for $\otimes, \oslash$ respectively
- $(x_1 \circledast \dots \circledast x_n) = (x_1 * \dots * x_n) + \sum_{i=1}^{n} x_i \left(\sum_{j=i}^{n} \delta_j\right)$, $|\delta_j| \le \epsilon_{mach}$
  - $(x_1 \circledast \dots \circledast x_n) = (x_1 * \dots * x_n)(1 + \epsilon)$, $\epsilon \le 1.06(n-1)\epsilon_{mach}$
  - $\mathrm{fl}\left(\sum x_i y_i\right) = \sum x_i y_i (1 + \epsilon_i)$ where $1 + \epsilon_i = (1 + \delta_i) \times (1 + \eta_i) \cdots (1 + \eta_n)$ and $|\delta_i|, |\eta_i| \le \epsilon_{mach}$
  - ∗$1 + \epsilon_i \approx 1 + \delta_i + (\eta_i + \dots + \eta_n)$
  - ∗$|\mathrm{fl}(x^T y) - x^T y| \le \sum |x_i y_i| |\epsilon_i|$
  - ∗Assuming $n\epsilon_{mach} \le 0.1$ => $|\mathrm{fl}(x^T y) - x^T y| \le \phi(n)\epsilon_{mach}|x|^T|y|$, where $|x|_i = |x_i|$ is vector and $\phi(n)$ is small function of n
  - **Summing a series** is more stable if terms added in order of increasing magnitude
- For **FP matrices**, let $|M|_{ij} = |M_{ij}|$, i.e. matrix $|M|$ of absolute values of M
  - $\mathrm{fl}(\lambda A) = \lambda A + E$, $|E|_{ij} \le |\lambda A|_{ij}\epsilon_{mach}$
  - $\mathrm{fl}(A + B) = (A + B) + E$, $|E|_{ij} \le |A + B|_{ij}\epsilon_{mach}$
  - $\mathrm{fl}(AB) = AB + E$, $|E|_{ij} \le n\epsilon_{mach}(|A||B|)_{ij} + O(\epsilon_{mach}^2)$
- ***Taylor series*** about $a \in \mathbb{R}$ gives $f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(a)}{k!}(x - a)^k + O\left((x - a)^{n+1}\right)$ as $x \to a$
  - Need $a = 0$ => $f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(0)}{k!} x^k + O(x^{n+1})$ as $x \to 0$
  - e.g. $(1 + \epsilon)^p = \sum_{k=0}^{n} \binom{p}{k}\epsilon^k + O(\epsilon^{n+1}) = \sum_{k=0}^{n} \frac{p!}{k!(p-k)!}\epsilon^k + O(\epsilon^{n+1})$ as $\epsilon \to 0$

Elementary Matrices

- Identity $I_n = [e_1 | \dots | e_n] = [e_1; \dots; e_n]$ has elementary vectors $e_1, \dots, e_n$ for rows/columns
- **Row/column switching**: permutation matrix P_{ij} obtained by switching e_i and e_j in I_n *(same for rows/columns)*
  - Applying P_{ij} **from left** will switch rows, **from right** will swap columns
  - $P_{ij} = P_{ij}^T = P_{ij}^{-1}$, i.e. applying twice will **undo** it
- **Row/column scaling**: $D_i(\lambda)$ obtained by scaling e_i by λ in I_n *(same for rows/columns)*
  - Applying D_i **from left** will scale rows, **from right** will scale columns
  - $D_i(\lambda) = \mathrm{diag}(1, \dots, \lambda, \dots, 1)$ has all **diagonal** properties apply, e.g. $D_i(\lambda)^{-1} = D_i(\lambda^{-1})$
- **Row addition**: $L_{ij}(\lambda) = I_n + \lambda e_i e_j^T$ performs $R_i \leftarrow R_i + \lambda R_j$ when applying **from left**
  - $\lambda e_i e_j^T$ is zeros except for λ **in** (i, j)**-th entry**
  - $L_{ij}(\lambda)^{-1} = L_{ij}(-\lambda)$ both triangular matrices

LU factorization w/ Gaussian elimination

- [[tutorial 1#Representing EROs/ECOs as transformation matrices|Recall that]] you can represent **EROs** and **ECOs** as transformation matrices R, C respectively
- LU **factorization** => finds $A = LU$ where L, U are lower/upper triangular *respectively*
- **Naive Gaussian Elimination** performs $[I_m \mid A \mid I_n] \Rightarrow [R^{-1} \mid U \mid I_n]$ to get $AI_n = R^{-1}U$ using only row addition
  - R^{-1}, i.e. **inverse EROs** in reversed order, is **lower-triangular** so $L = R^{-1}$
  - ![[Pasted image 20250419051217.png|400]]
- The **pivot element** is simply diagonal entry $u_{kk}^{(k-1)}$
  - fails if $u_{kk}^{(k-1)} \approx 0$
  - $\tilde{L}\tilde{U} = A + \delta A$, $\frac{\|\delta A\|}{\|L\| \cdot \|U\|} = O(\epsilon_{mach})$ only **backwards stable** if $\|L\| \cdot \|U\| \approx \|A\|$
  - Work required: $\sim \frac{2}{3}m^3$ flops $\sim O(m^3)$
  - Solving $Ax = LUx$ is $\sim \frac{2}{3}m^3$ flops *(back substitution is $O(m^2)$)*
  - **NOTE:** Householder triangularisation requires $\sim \frac{4}{3}m^3$
- **Partial pivoting** computes $PA = LU$ where P is a permutation matrix => $PP^T = I$, i.e. its orthogonal
  - For each column j, finds largest entry and row-swaps to make it new pivot => P_j
  - Then performs normal elimination on that column => L_j
  - Result is $L_{m-1}P_{m-1} \dots L_2 P_2 L_1 P_1 A = U$, where $L_{m-1}P_{m-1} \dots L_2 P_2 L_1 P_1 = L'_{m-1} \dots L'_1 P_{m-1} \dots P_1$
  - Setting $L = (L'_{m-1} \dots L'_1)^{-1}$, $P = P_{m-1} \dots P_1$ gives $PA = LU$
- ![[Pasted image 20250420092322.png|450]]
  - Work required: $\sim \frac{2}{3}m^3$ flops $\sim O(m^3)$ results in $l_{ij} \le 1$ so $\|L\| = O(1)$
  - Stability depends on **growth-factor** $\rho = \frac{\max_{i,j}|u_{ij}|}{\max_{i,j}|a_{ij}|}$ => for partial pivoting $\rho \le 2^{m-1}$
  - $\|U\| = O(\rho\|A\|)$ => $\tilde{L}\tilde{U} = \tilde{P}A + \delta A$, $\frac{\|\delta A\|}{\|A\|} = O(\rho\epsilon_{machine})$ => only **backwards stable** if $\rho = O(1)$
- **Full pivoting** is $PAQ = LU$ finds largest entry in **bottom-right submatrix**
  - Makes it **pivot** with row/column swaps before normal elimination
  - Very expensive $O(m^3)$ search-ops, **partial pivoting** only needs $O(m^2)$

Systems of Equations: Iterative Techniques

- Let $A, R, G \in \mathbb{R}^{n \times n}$ where G^{-1} exists => **splitting** $A = G + R$ helps iteration
  - $Ax = b$ rewritten as $x = Mx + c$ where $M = -G^{-1}R$; $c = G^{-1}b$
  - Define $f(x) = Mx + c$ and sequence $x^{(k+1)} = f(x^{(k)}) = Mx^{(k)} + c$ with starting point $x^{(0)}$
  - **Limit** of (x_k) is fixed point of f => unique fixed point of f is **solution** to $Ax = b$
  - If $\|\cdot\|$ is consistent norm and $\|M\| < 1$ then (x_k) converges for any $x^{(0)}$ *(because Cauchy-completeness)*
  - ∗For splitting, we want $\|M\| < 1$ and easy to compute M; c
  - ∗**Stopping criterion** usually the relative residual $\frac{\|b - Ax^{(k)}\|}{\|b\|} \le \epsilon$
- Assume A's **diagonal is non-zero** *(w.l.o.g. permute/change basis if isn't)* then $A = D + L + U$
  - Where D is diagonal of A, L, U are strict **lower/upper triangular** parts of A
- **Jacobi Method**: $G = D$; $R = L + U$ => $M = -D^{-1}(L + U)$; $c = D^{-1}b$
  - $x_i^{(k+1)} = \frac{1}{A_{ii}}\left(b_i - \sum_{j \ne i} A_{ij} x_j^{(k)}\right)$ => $x_i^{(k+1)}$ only needs b_i; $x^{(k)}$; A_{i*} => row-wise parallelization
- **Gauss-Seidel (G-S) Method**: $G = D + L$; $R = U$ => $M = -(D + L)^{-1}U$; $c = (D + L)^{-1}b$
  - $x_i^{(k+1)} = \frac{1}{A_{ii}}\left(b_i - \sum_{j=1}^{i-1} A_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} A_{ij} x_j^{(k)}\right)$
  - Computing $x_i^{(k+1)}$ needs b_i; $x^{(k)}$; A_{i*} and $x_j^{(k+1)}$ for $j < i$ => lower storage requirements
- **Successive over-relaxation (SOR)**: $G = \omega^{-1}D + L$; $R = (1 - \omega^{-1})D + U$ => $M = -(\omega^{-1}D + L)^{-1}((1 - \omega^{-1})D + U)$; $c = -(\omega^{-1}D + L)^{-1}b$
  - $x_i^{(k+1)} = \frac{\omega}{A_{ii}}\left(b_i - \sum_{j=1}^{i-1} A_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} A_{ij} x_j^{(k)}\right) + (1 - \omega)x_i^{(k)}$ for **relaxation factor** $\omega > 1$
- If A is **strictly row diagonally dominant** then Jacobi/Gauss-Seidel methods converge
  - A is **strictly row diagonally dominant** if $|A_{ii}| > \sum_{j \ne i}|A_{ij}|$
- If A is positive-definite then **G-S** and **SOR** $(\omega \in (0, 2))$ converge

Break up matrices into (uneven blocks)

- e.g. symmetric $A \in \mathbb{R}^{n \times n}$ can become $A = \left[\begin{array}{c|c} a_{11} & b^T \\ \hline b & C \end{array}\right]$, then perform proofs on that

Catchup: metric spaces and limits

- Metrics obey these axioms
  - $d(x, x) = 0$
  - $x \ne y \implies d(x, y) > 0$
  - $d(x, y) = d(y, x)$
  - $d(x, z) \le d(x, y) + d(y, z)$
- For metric spaces, **mix-and-match** these infinite/finite limit definitions:
  - $\lim_{x \to +\infty} f(x) = +\infty \iff \forall r \in \mathbb{R}, \exists N \in \mathbb{N}, \forall x > N : f(x) > r$
  - $\lim_{x \to p} f(x) = L \iff \forall \epsilon > 0, \exists \delta > 0, \forall x \in A : 0 < d_X(x, p) < \delta \implies d_Y(f(x), L) < \epsilon$
- **Cauchy sequences**, i.e. $\forall \epsilon > 0, \exists N \in \mathbb{N}, \forall m, n \ge N : d(a_m, a_n) < \epsilon$, converge in **complete spaces**
- You can manipulate matrix limits much **like in real analysis**, e.g. $\lim_{n \to \infty}(A^n B + C) = \left(\lim_{n \to \infty} A^n\right) B + C$
- Turn **metric limit** $\lim_{n \to \infty} x_n = L$ into **real limit** $\lim_{n \to \infty} d(x_n, L) = 0$ to leverage real analysis
- Bounded **monotone sequences** converge in $\mathbb{R}$
- Sandwich theorem for limits in $\mathbb{R}$ => pick easy upper/lower bounds
- $\lim_{n \to \infty} r^n = 0 \iff |r| < 1$ and $\lim_{n \to \infty} \sum_{i=0}^{n} ar^i = \frac{a}{1 - r} \iff |r| < 1$

Eigenvalue Problems: Iterative Techniques

- If A is [[tutorial 1#Properties of matrices|diagonalizable]] then [[tutorial 1#Eigen-values/vectors|eigen-decomposition]] $A = X\Lambda X^{-1}$
  - **Dominant** λ_1; x_1 are such that $|\lambda_1|$ is strictly largest for which $Ax = \lambda x$
  - **Rayleigh quotient** for Hermitian $A = A^\dagger$ is $R_A(x) = \frac{x^\dagger Ax}{x^\dagger x}$
  - ∗Eigenvectors are stationary points of R_A
  - ∗$R_A(x)$ is closest to being like eigenvalue of x, i.e. $R_A(x) = \mathrm{argmin}_\alpha \|Ax - \alpha x\|_2$
  - ∗$R_A(x) - R_A(v) = O(\|x - v\|^2)$ as $x \to v$ where v is eigenvector
- **Power iteration**: define sequence $b^{(k+1)} = \frac{Ab^{(k)}}{\|Ab^{(k)}\|}$ with initial $b^{(0)}$ s.t. $\|b^{(0)}\| = 1$
  - Assume **dominant** λ_1; x_1 exist for A, and that $\mathrm{proj}_{x_1}(b^{(0)}) \ne 0$
  - Under above assumptions, $\mu_k = R_A(b^{(k)}) = \frac{b^{(k)\dagger}Ab^{(k)}}{b^{(k)\dagger}b^{(k)}}$ converges to **dominant** λ_1
  - (b_k) converges to some **dominant** x_1 associated with λ_1 => $\|Ab^{(k)}\|$ converges to $|\lambda_1|$
  - If $\mathrm{proj}_{x_1}(b^{(0)}) = 0$ then (b_k); μ_k converge to second **dominant** λ_2; x_2 instead
  - If **no dominant** λ_1 *(i.e. multiple eigenvalues of maximum $|\lambda|$)* then (b_k) will converge to linear combination of their corresponding eigenvectors
  - Slow convergence if **dominant** λ_1 not "very dominant"
  - $\|b^{(k)} - \alpha_k x_1\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$ for **phase factor** $\alpha_k \in \{-1, 1\}$, it may alternate if $\lambda_1 < 0$
  - ∗$\alpha_k = \frac{(\lambda_1)^k c_1}{|\lambda_1|^k |c_1|}$ where $c_1 = x_1^\dagger b^{(0)}$ and assuming $b^{(k)}$; x_1 are normalized
- $(A - \sigma I)$ has **eigenvalues** $\lambda - \sigma$ => power-iteration on $(A - \sigma I)$ has $\frac{\lambda_2 - \sigma}{\lambda_1 - \sigma}$
  - Eigenvector guess => estimated eigenvalue
- **Inverse (power-)iteration**: perform power iteration on $(A - \sigma I)^{-1}$ to get $\lambda_{1,\sigma}$ **closest** to σ
  - $(A - \sigma I)^{-1}$ has eigenvalues $(\lambda - \sigma)^{-1}$ so power iteration will yield **largest** $(\lambda_{1,\sigma} - \sigma)^{-1}$
  - i.e. will yield **smallest** $\lambda_{1,\sigma} - \sigma$, i.e. will yield $\lambda_{1,\sigma}$ **closest** to σ
  - $\|b^{(k)} - \alpha_k x_{1,\sigma}\| = O\left(\left|\frac{\lambda_{1,\sigma} - \sigma}{\lambda_{2,\sigma} - \sigma}\right|^k\right)$ where $x_{1,\sigma}$ corresponds to $\lambda_{1,\sigma}$ and $\lambda_{2,\sigma}$ is 2nd-closest to σ
  - Efficiently compute eigenvectors for **known eigenvalues** σ
  - Eigenvalue guess => estimated eigenvector
  - ![[Pasted image 20250420131643.png|300]]
  - Can reduce matrix inversion $O(m^3)$ to $O(m^2)$ by pre-factorization

Nonlinear Systems of Equations: Iterative Techniques

- [[tutorial 6#Multivariate Calculus|Recall]] that $\nabla f(x)$ is direction of **max.** rate-of-change $|\nabla f(x)|$
- Search for stationary point by **gradient descent**: $x^{(k+1)} = x^{(k)} - \alpha\nabla f(x^{(k)})$ for step length α
- A is positive-definite solving $Ax = b$ and $\min_x f(x) = \frac{1}{2}x^T Ax - x^T b$ are equivalent
  - Get iterative methods $x^{(k+1)} = x^{(k)} - \alpha^{(k)}p^{(k)}$ for step length $\alpha^{(k)}$ and directions $p^{(k)}$
- **Conjugate gradient (CG) method**: if $A \in \mathbb{R}^{n \times n}$ also symmetric then $\langle u, v \rangle_A = u^T Av$ is an inner-product
  - GC chooses $p^{(k)}$ that are conjugate w.r.t. A, i.e. $\langle p^{(i)}, p^{(j)} \rangle_A = 0$ for $i \ne j$
  - And chooses $\alpha^{(k)}$ s.t. **residuals** $r^{(k)} = -\nabla f(x^{(k)}) = b - Ax^{(k)}$ are orthogonal
  - ∗$k = 0$ => $p^{(0)} = -\nabla f(x^{(0)}) = r^{(0)}$
  - ∗$k \ge 1$ => $p^{(k)} = r^{(k)} - \sum_{i < k} \frac{\langle p^{(i)}, r^{(k)} \rangle_A}{\langle p^{(i)}, p^{(i)} \rangle_A} p^{(i)}$
  - ∗$\alpha^{(k)} = \mathrm{argmin}_\alpha f(x^{(k)} + \alpha p^{(k)}) = \frac{p^{(k)} \cdot r^{(k)}}{\langle p^{(k)}, p^{(k)} \rangle_A}$
  - Without rounding errors, CG converges in $\le n$

iterations

* Similar to to [[tutorial 1#Gram-Schmidt method to generate orthonormal basis from any linearly independent vectors|Gram-Schmidt]] *(different inner-product)*
* $\langle \mathbf{p}^{(0)}, \dots, \mathbf{p}^{(n-1)} \rangle$ and $\langle \mathbf{r}^{(0)}, \dots, \mathbf{r}^{(n-1)} \rangle$ are bases for $\mathbb{R}^n$

QR Algorithm to find Schur decomposition $A = QUQ^\dagger$

- Any $A \in \mathbb{C}^{m \times m}$ has **Schur decomposition** $A = QUQ^\dagger$
  - Q is unitary, i.e. $Q^\dagger = Q^{-1}$ and upper-triangular U
  - Diagonal of U contains **eigenvalues** of A
- ![[Pasted image 20250420135506.png|300]]
- For $A \in \mathbb{R}^{m \times m}$ each iteration $A^{(k)} = Q^{(k)} R^{(k)}$ produces orthogonal ${Q^{(k)}}^T = {Q^{(k)}}^{-1}$
- So $A^{(k+1)} = R^{(k)} Q^{(k)} = ({Q^{(k)}}^T Q^{(k)}) R^{(k)} Q^{(k)} = {Q^{(k)}}^T A^{(k)} Q^{(k)}$ means $A^{(k+1)}$ is **similar** to $A^{(k)}$
  - Setting $A^{(0)} = A$ we get $A^{(k)} = \tilde{Q}^{(k)T} A \tilde{Q}^{(k)}$ where $\tilde{Q}^{(k)} = Q^{(0)} \dots Q^{(k-1)}$
- Under certain conditions **QR algorithm** converges to **Schur decomposition**
- We can **apply shift** $\mu^{(k)}$ at iteration k => $A^{(k)} - \mu^{(k)} I = Q^{(k)} R^{(k)};\ A^{(k+1)} = R^{(k)} Q^{(k)} + \mu^{(k)} I$
  - If **shifts** are good eigenvalue estimates then last column of $\tilde{Q}^{(k)}$ converges quickly to an **eigenvector**
  - Estimate $\mu^{(k)}$ with Rayleigh quotient => $\mu^{(k)} = (A_k)_{mm} = \tilde{\mathbf{q}}_m^{(k)T} A \tilde{\mathbf{q}}_m^{(k)}$ where $\tilde{\mathbf{q}}_m^{(k)}$ is m-th column of $\tilde{Q}^{(k)}$