Linear Algebra Recap

1. Linear Systems (Purpose of LinAlg)
2. Matrix Algebra & Properties
3. Solutions: Inverse, Cholesky, LU, QR, SVD
4. Vectorspaces
5. Matrix Properties: Norm, Orthogonal, Determinant
6. Eigenvalues & Eigenvectors
7. Special Matrices: Square, Symmetric, Normal, Orthogonal
   Triangular, Diagonal, Nilpotent, Skew-Symmetric, Positive Defin
   Idempotent
8. Covariance, Correlation, Stochastic
9. Powers and Exponentials

Linear Systems (Purpose of Linear Algebra)

$$A \cdot x = b$$

$$\begin{cases} x+y=3 \\ 2x-y=0 \end{cases} \Rightarrow \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

$\rightarrow x=1,\ y=2$

Python Animation: What is a Matrix transformation

Why:

- Approximation of reality, Taylor expansion: $f(x+\delta) \approx f(x) + Df(x)\delta$
- Control systems: $\dot{x} = Ax + Bu$
- Signal processing / kernals
- Machine Learning: backpropagation (gradient decent), neural networks, PCA
- Rotations / Transformations: Robotics, Graphics, Computer vision
- Optimizations
- Compression

Properties

$\begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$

$x+y=3$
$2x-y=0$

$\rightarrow x=1,\ y=2$

One unique solution

unknown variables = independent equations

$\text{rank}(A) = n = m$

n = number of equations
m = number of unknowns

$\begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \end{bmatrix}$

$x+y=3$
$2x+2y=6 \rightarrow x+y=3$

$\Rightarrow x = 3-y$

infinitely many solutions

unknow variables > independent equations

$\text{rank}(A) < m$
$n < m$
$\text{Kern}(A) = m - \text{rank} = 1$

$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

$x+y=1$
$x+y=2$

no solution

inconsistent system

$\begin{bmatrix} 1 & 1 \\ 2 & -1 \\ 1 & 3 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 6 \end{bmatrix}$

$x+y=3$
$2x-y=0$
$x+3y=6$

$\text{ran}(A) > m$
overdetermined

$\text{rank}(A)$ = number of independent equations

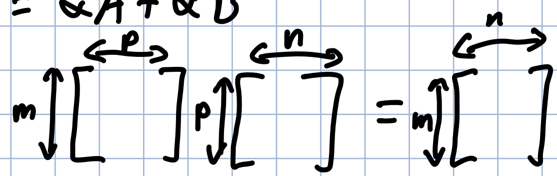
$\dim(\text{image}(A)) = \text{rank}(A)$: which b are reachable / Possible outcome

Solve with Least Square, QR or SVD

(control system: reachable states)

$\dim(\ker(A)) = m - \text{rank}(A)$: vector mapped to 0
$Ax = 0$
(eigenspace for eigenvalue 0)
(what the matrix forgets, information lost)
(you change input, but output stays same)
((control system: uncontrollable modes)
(ML: redundant features)

Allowed operations and Properties

- $A+B$ iff same shape $(A+B)_{ij} = A_{ij} + B_{ij}$ | $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 1+1 & 2+2 \\ 3+3 & 4+4 \end{bmatrix}$
- $\alpha \cdot A$ scales each entry $2 \cdot \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$
  $\rightarrow \alpha(A+B) = \alpha A + \alpha B$
- Multiplication 

only allowed if same colums of first as rows of second matrix.
- $AB \neq BA$
- $A(BC) = (AB)C$, $A(B+C) = AB + AC$
- $ABCx$ ⇒ order is right to left, we apply A onto (BCx), apply C onto x etc.
- Transpose $(A^T)_{ij} = A_{ji}$, $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$, $A^T = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$, Hermitian for complex
A^H $\begin{pmatrix} 1+i & 2+i \\ 3+i & 4+i \end{pmatrix} \rightarrow \begin{pmatrix} 1-i & 3-i \\ 2-i & 4-i \end{pmatrix}$
$\rightarrow (AB)^T = B^T A^T$; $(AB)^H = B^H A^H$

Determinant : how much the matrix scales the input.

- $\det(AB) = \det(A) \cdot \det(B)$
- $\det(A^T) = \det(A)$
- iff $\det(A) \neq 0 \Rightarrow A$ is invertible, $\det(A^{-1}) = \frac{1}{\det(A)}$

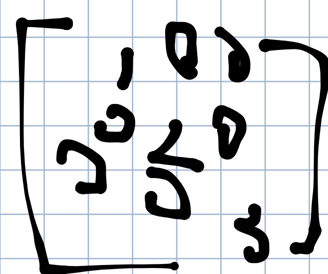
Trace : addition of all eigen values
- $tr(A) = \sum_i A_{ii}$ (all diagonal entries) $tr\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = 1+4$

Special matrices

- Square: $n \times n$, input output are in same space
- Symmetric: $A^T = A$, diagonizable, real Evals, eigendecompostion
- Normal: $A^TA = AA^T$: Orthogonal eigen basis
- Orthogonal/unitary (complex): $Q^TQ = I$, $Q^T = Q^{-1}$: Norm-perserving, det ± 1, EVals on unit circle
- Triangular: $\begin{bmatrix} * & * & * \\ 0 & * & * \\ & & * \end{bmatrix}$: EVal on diagonal, det $= \prod \lambda$, Easy A^k

- Diagonal: only diagonal entries, easy powers. inverse, exponentials
- Nilpotent $A^k = 0$, Eigenvalues are all 0, finite e^{At}

- Scalar Matrix: $\lambda \cdot I$
- Skew-symmetric: $A^T = -A$



- Positive definite: $x^TAx > 0 \; \forall x \neq 0$
- Idempotent $A^2 = A$
- Singular: not invertible

- Stochastic: nonnegative and rows or colums sum to 1

- Sparse: many zeros

Inverse

$$A \cdot A^{-1} = I, \quad A^{-1}A^{-1} = A, \quad (AB)^{-1} = B^{-1}A^{-1}, \quad (A^T)^{-1} = (A^{-1})^T, \quad (\alpha A)^{-1} = \frac{1}{\alpha} A^{-1}$$

Inverse

$$\begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

$$\begin{vmatrix} x + y = 3 \\ 2x - y = 0 \end{vmatrix}$$

$\rightarrow x = 1, \; y = 2$

One unique solution

$Ax = b$

Inverse: $A^{-1} \cdot A = I = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$

$A^{-1}Ax = A^{-1}b \Rightarrow x = A^{-1}b$

Square Matrix:
- $\det(A) \neq 0$, 0 is not an Eigenvalue
- $\mathrm{rank}(A) = n$
- row and colums are linear independent
- $Ax = 0$ only for $x = 0$
- $Ax = b$ has a unique solution for every b

Computing inverse

linverse of 2X2

$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$

$A^{-1} = \frac{1}{ad - cb} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$

Gaussian elimination

$[A \mid I]$

$\downarrow$ row operations

$[I \mid A^{-1}]$

Adjugate

$\frac{1}{\det(A)} \mathrm{adj}(A)$

$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$

$A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} +\det(d) & -\det(c) \\ -\det(b) & +\det(a) \end{pmatrix}^T$

$A = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \quad \det A = -1 - 2 \cdot 1 = -3$

$A^{-1} = -\frac{1}{3} \begin{pmatrix} -1 & -1 \\ -2 & 1 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 1 & 1 \\ 2 & -1 \end{pmatrix}$

$= \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & -\frac{1}{3} \end{bmatrix}$

$\left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 2 & -1 & 0 & 1 \end{array}\right] -2a$

$\left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 0 & -3 & -2 & 1 \end{array}\right] : -3$

$\left[\begin{array}{cc|cc} 1 & 1 & 1 & 0 \\ 0 & 1 & \frac{2}{3} & -\frac{1}{3} \end{array}\right] - b$

$\left[\begin{array}{cc|cc} 1 & 0 & \frac{1}{3} & \frac{1}{3} \\ 0 & 1 & \frac{2}{3} & -\frac{1}{3} \end{array}\right]$

$x = \frac{1}{3} \begin{pmatrix} 1 & 1 \\ 2 & -1 \end{pmatrix} \begin{pmatrix} 3 \\ 0 \end{pmatrix}$

$= \frac{1}{3} \begin{pmatrix} 3 \\ 6 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \begin{matrix} x = 1 \\ y = 2 \end{matrix}$

Problems:

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 1.0001 \end{pmatrix} \quad b = \begin{pmatrix} 2 \\ 2.0001 \end{pmatrix}$$

$$A^{-1} = \begin{pmatrix} 10001 & -10000 \\ -10000 & 10000 \end{pmatrix} \quad x = A^{-1} b = \begin{pmatrix} 10001 \cdot 2 - 10000 \cdot 2.0001 \\ -10000 \cdot 2 + 10000 \cdot 2.0001 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Computer calculates in bits

→ rounding problems as intermediate numbers get very large

Condition number $\kappa(A) = \|A\| \, \|A^{-1}\|$ is very large $\qquad \kappa_2(A) = \frac{\sigma_{max}}{\sigma_{min}}$

small errors get magnified.

$$\sqrt{a^2 + b^2}$$

SPD \

Direct dense solvers (square $n \times n$)

Method	Time (≈ flops)	Extra memory (beyond storing A)	When / Why (short)
Cholesky ($A = LL^T$)	≈ $(1/3)\cdot n^3$	≈ $\frac{1}{2}\cdot n^2$ (store L)	SPD matrices → fastest and stable; use if A symmetric positive definite.
LU (with partial pivoting)	≈ $(2/3)\cdot n^3$	≈ n^2 (L/U in-place)	General square systems; default for many RHS (factor once).
QR (Householder)	≈ $(2/3)\cdot n^3$	≈ n^2 (if forming Q) or less (economy)	Numerically stable; use for least-squares or when orthogonality is needed.
SVD (full)	≈ $4\cdot n^3$ (order-of-magnitude)	≈ $2\cdot n^2$ (U and V)	Ill-conditioned or rank-deficient problems; gives rank/conditioning/pseudoinverse.
Explicit inverse (compute A^{-1})	≈ 2–3$\cdot n^3$	≈ n^2 (store A^{-1})	Theoretical only — avoid in practice (wastes work, less stable).

Rectangular / large / sparse / iterative methods

Method	Time per op	Extra memory	When / Why (short)
QR (m×n, economy)	≈ **O(m n²)** (m ≥ n)	≈ **O(m n)** or **O(n²)** if full Q	Least-squares (overdetermined) with good stability; avoid forming A^TA.
SVD (m×n, full)	≈ **O(m n²)** (constant bigger than QR)	≈ **O((m+n)·min(m,n))** (store U,V)	Best for rank detection, pseudoinverse; expensive for large dense matrices.
Truncated SVD (rank k)	≈ **O(m n k)** (or randomized ≈ O(m n log k))	≈ **O(k(m+n))**	Low-rank approximation / compression — big space/time savings for k≪n.
Conjugate Gradient (CG)	per iter ≈ **O(nnz(A))**	**O(nnz(A)) + O(n)** (few vectors)	Large sparse SPD; scalable; needs preconditioning.
GMRES	per iter ≈ **O(nnz(A))**	**O(k·n)** (k = iterations before restart)	Large sparse non-symmetric; memory grows with iterations (use restarted GMRES).
LSQR / LSMR	per iter ≈ **O(nnz(A))**	**O(nnz(A)) + O(n+m)**	Sparse least-squares; robust for large-scale problems.

Normal Equations $A^TAx = A^Tc$ fast

only if A is well conditioned

$\kappa(A^TA) = \kappa(A)^2$!

Cholesky

If A is positive definite

$A = LL^T$, $Ax = b \to L(L^Tx) = b \to$ $\begin{matrix} L^Tx = y \\ Ly = b \end{matrix}$ solve

$L = \begin{bmatrix} 2 & 0 & 0 \\ 6 & 1 & 0 \\ -8 & 5 & 3 \end{bmatrix}$, $y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$ $b = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

$\Rightarrow$ $2y_1 = 1$

$6y_1 + y_2 = 2$

$-8y_1 + 5y_2 + 3y_3 = 3$

$$L_{kk} = \sqrt{A_{kk} - \sum_{s=1}^{k-1} L_{ks}^2}$$

For $i > k$, $L_{ik} = \left(A_{ik} - \sum_{s=1}^{k-1} L_{is} L_{ks}\right) / L_{kk}$

```
for k in range(n):
    L[k,k] = sqrt(A[k,k] - sum(L[k,0:k]**2))
    for i in range(k+1, n):
        L[i,k] = (A[i,k] - dot(L[i,0:k], L[k,0:k])) / L[k,k]
```

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 5 & 6 & 7 \\ 3 & 6 & 8 & 9 \\ 4 & 7 & 9 & 10 \end{pmatrix} = \begin{pmatrix} a & & & \\ b & e & & \\ c & f & h & \\ d & g & i & j \end{pmatrix} \begin{pmatrix} a & b & c & d \\ & e & f & g \\ & & h & i \\ & & & j \end{pmatrix} = \begin{pmatrix} a^2 & & & \\ ab & b^2+e^2 & & \\ ac & bc+ef & c^2+f^2+h^2 & \\ ad & bd+eg & cd+fg+hi & d^2+g^2+i^2+j^2 \end{pmatrix}$$

Naïve pure-Python/NumPy (educational, not optimal for performance):

```python
import numpy as np

def cholesky_lower(A):
    A = np.array(A, dtype=float)
    n = A.shape[0]
    L = np.zeros_like(A)
    for k in range(n):
        s = np.dot(L[k,:k], L[k,:k])
        L[k,k] = np.sqrt(A[k,k] - s)
        for i in range(k+1, n):
            s = np.dot(L[i,:k], L[k,:k])
            L[i,k] = (A[i,k] - s) / L[k,k]
    return L
```

Using NumPy / SciPy (production):

```python
import numpy as np
L = np.linalg.cholesky(A)      # returns lower-triangular L s.t. A = L @ L.T
# Solve Ax = b:
y = np.linalg.solve(L, b)
x = np.linalg.solve(L.T, y)
```

Or SciPy:

```python
from scipy.linalg import cho_factor, cho_solve
c, lower = cho_factor(A)       # calls LAPACK dpotrf
x = cho_solve((c, lower), b)
```

LDL for symmetric matrices.

values on diagonal D don't need to be $\sqrt{A}$ $L = \begin{pmatrix} 1 & & 0 \\ L_{21} & 1 & \\ L_{31} & L_{32} & 1 \end{pmatrix}$, $D = \begin{pmatrix} d_1 & & 0 \\ 0 & d_2 & \\ & & d_3 \end{pmatrix}$

```ruby
for k = 1..n:
    D[k] = A[k,k] - sum(L[k,1:k-1]^2 * D[1:k-1])
    for i = k+1..n:
        L[i,k] = (A[i,k] - sum(L[i,1:k-1] * L[k,1:k-1] * D[1:k-1])) / D[k]
```

- Use symmetric pivoting in indefinite cases (Bunch–Kaufman) to maintain numerical stability.

Sampling Gaussian

if $x \sim N(\mu, A^{-1})$ we can sample $z \sim N(0, I)$

$x = \mu + L^{-T} z$ if A is covariance LL^T

$\rightarrow$ $x = \mu + Lz$

LU - decomposition

$Ax = b$, $LUx = b$ or $P^T LUx = b$ P(is permutation matrix)

if a pivot $=0$ you need to swap it, that's why some don't have LU but need $P^T LU$.

(usually swap for largest pivot)

LU (with partial pivoting)	≈ (2/3)·n³	≈ n² (L/U in-place)	General square systems; default for many RHS (factor once).

Solving:

$$Ly = Pb, \quad Ux = y$$

LU-steps:

$$A = \begin{pmatrix} 2 & 3 & 1 \\ 4 & 7 & 7 \\ -2 & 4 & 5 \end{pmatrix} \qquad P_0 = I, \quad L_0 = I, \quad U_0 = A$$

$$\begin{array}{ccc} P & U & L \end{array}$$

$$\left[\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}\;\middle|\;\begin{matrix} 2 & 3 & 1 \\ 4 & 7 & 7 \\ -2 & 4 & 5 \end{matrix}\;\middle|\;\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}\right]$$

$$\left[\begin{matrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{matrix}\;\middle|\;\begin{matrix} 4 & 7 & 7 \\ 2 & 3 & 1 \\ -2 & 4 & 5 \end{matrix}\;\middle|\;\begin{matrix} 1 & & \\ & 1 & \\ & & 1 \end{matrix}\right]$$

$$\left[\begin{matrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{matrix}\;\middle|\;\begin{matrix} 4 & 7 & 7 \\ 0 & -0{,}5 & -2{,}5 \\ 0 & 7{,}5 & 8{,}5 \end{matrix}\;\middle|\;\begin{matrix} 1 & 0 & 0 \\ 0{,}5 & 1 & 0 \\ -0{,}5 & 0 & 1 \end{matrix}\right]$$

$$\left[\begin{matrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{matrix}\;\middle|\;\begin{matrix} 4 & 7 & 7 \\ 0 & 7{,}5 & 8{,}5 \\ 0 & -0{,}5 & -2{,}5 \end{matrix}\;\middle|\;\begin{matrix} 1 & 0 & 0 \\ 0{,}5 & 1 & 0 \\ -0{,}5 & 0 & 1 \end{matrix}\right]$$

$$\left[\begin{matrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{matrix}\;\middle|\;\begin{matrix} 4 & 7 & 7 \\ 0 & 7{,}5 & 8{,}5 \\ 0 & 0 & -\frac{29}{15} \end{matrix}\;\middle|\;\begin{matrix} 1 & 0 & 0 \\ 0{,}5 & 1 & 0 \\ -0{,}5 & -\frac{1}{15} & 1 \end{matrix}\right]$$

$$\begin{array}{ccc} = & = & = \\ P & U & L \end{array}$$

python

```python
x = np.linalg.solve(A, b)
```

python

```python
from scipy.linalg import solve

x = solve(A, b)
```

python

```python
import numpy as np
from scipy.linalg import lu, lu_solve

A = np.array([[2, 3, 1],
              [4, 7, 7],
              [-2, 4, 5]], dtype=float)

b = np.array([1, 2, 3], dtype=float)

# PLU decomposition
P, L, U = lu(A)

# Solve Ax = b
# First apply permutation to b
Pb = P @ b

# Forward + backward substitution
from scipy.linalg import solve_triangular

y = solve_triangular(L, Pb, lower=True)
x = solve_triangular(U, y)

print("Solution x =", x)
```

QR-decomposition

QR (Householder)	≈ (2/3)·n^3	≈ n^2 (if forming Q) or less (economy)	Numerically stable; use for least-squares or when orthogonality is needed.

Rectangular / large / sparse / iterative methods

Method	Time per op	Extra memory	When / Why (short)
QR (m×n, economy)	≈ O(m n^2) (m ≥ n)	≈ O(m n) or O(n^2) if full Q	Least-squares (overdetermined) with good stability; avoid forming A^TA.

$Ax = b$, $A = QR$ where Q is orthogonal ($Q^TQ = I$)
R is upper triangular

$A \in \mathbb{R}^{m \times n}$: Works for overdetermined systems
Solves least square $\|Ax - b\|_2$ minimizes. Linear Regression

- Used for Eigenvalue algos (schur)

Solve: $QRx = b$, $Q^TQRx = Q^Tb$ $\qquad Q^T = Q^{-1}$

$Rx = Q^Tb$

Computation: (1. Gram-Schmidt 2. Householder 3. Givens Rotation)

1. Modified Gram Schmidt:

(Gram-Schmidt to find orthonormal basis of vectors)

$$w_i' = v_i - \sum_{k=1}^{i-1} (v_i, w_k) \cdot w_k$$

$$w_i = \frac{w_i'}{\sqrt{(w_i', w_i')}}$$

$$A = \begin{array}{c} \begin{matrix} v_1 & v_2 & v_3 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \end{array}$$

$$\overset{r_{11}}{\|v_1\|_2} = \sqrt{1^2+1^2+0^2} = \sqrt{2}$$

$$w_1' = v_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \quad w_1 = \frac{v_1}{\sqrt{2}} = \underline{\underline{\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}}}$$

$$w_2' = v_2 - \overset{r_{12}}{(v_2, w_1)} \cdot w_1 \qquad \overset{r_{12}}{(v_2, w_1)} = \left(\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}\right)$$

$$= \frac{1}{\sqrt{2}}(1\cdot 1 + 0 + 0) = \frac{1}{\sqrt{2}}$$

$$= v_2 - \frac{1}{\sqrt{2}} w_1$$

$$= \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} - \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 0{,}5 \\ -0{,}5 \\ 1 \end{pmatrix}$$

$$w_2 = \frac{w_2'}{\|w_2'\|_2} \qquad \overset{r_{22}}{\|w_2'\|_2} = \sqrt{\frac{1}{4}+\frac{1}{4}+1} = \frac{\sqrt{6}}{2}$$

$$= \frac{\frac{1}{2}(1\ -1\ 2)^\top}{\sqrt{6}/2}$$

$$= \underline{\underline{\frac{1}{\sqrt{6}}\begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}}}$$

$$w_3' = v_3 - \overset{r_{13}}{(v_3, w_1)} w_1 - \overset{r_{23}}{(v_3, w_2)} w_2$$

$$\overset{r_{13}}{(v_3, w_1)} = \frac{1}{\sqrt{2}}$$

$$\overset{r_{23}}{(v_3, w_2)} = \frac{1}{\sqrt{6}}$$

$$= \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} - \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} - \frac{1}{6}\begin{pmatrix} 1 \\ -1 \\ 2 \end{pmatrix}$$

$$= \frac{2}{3}\begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}$$

$$w_3 = \frac{w_3'}{\|w_3'\|_2} \qquad \overset{r_{33}}{\|w_3'\|_2} = \sqrt{3\cdot\frac{4}{9}} = \frac{2}{\sqrt{3}}$$

$$= \frac{1}{\sqrt{3}}\begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}$$

$$Q = \begin{pmatrix} w_1 & w_2 & w_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{3}} \\ 0 & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{3}} \end{pmatrix}$$

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{pmatrix} = \begin{pmatrix} \sqrt{2} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & \frac{\sqrt{2}}{6} & \frac{1}{\sqrt{6}} \\ 0 & 0 & \frac{2}{\sqrt{3}} \end{pmatrix}$$

Solve: $QRx = b$, $Q^T Q R x = Q^T b$ $\qquad Q^T = Q^{-1}$

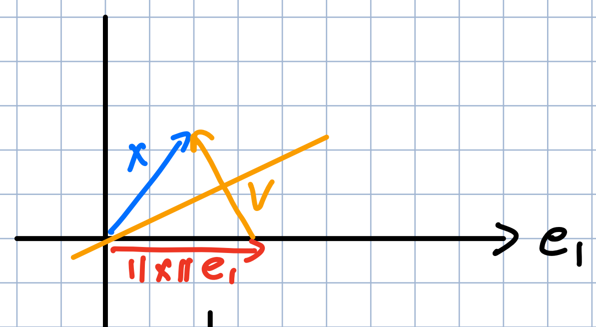
$Rx = Q^T b$ $\qquad \left| \; y = Q^T b, \; Rx = y \right.$

Substitutions

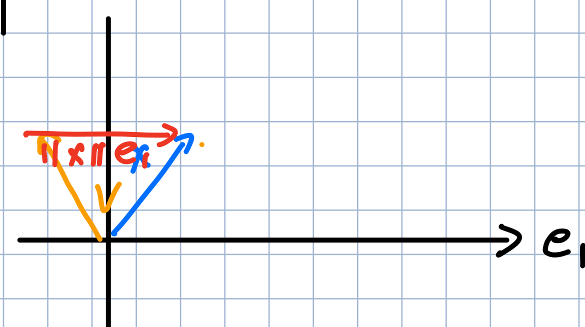
2. Householder

- use reflections to zero out entries below the diagonal
- reflections are orthogonal and numerically stable

$H = I - 2\frac{vv^T}{v^T v}$, $\quad Hx = \pm \|x\| e_1$



$A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ we want $R = \begin{pmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \end{pmatrix}$



$\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \to \begin{pmatrix} x \\ 0 \\ 0 \end{pmatrix}$

$\|x\| = \sqrt{2} \qquad \alpha = -\|x\| = -\sqrt{2}$

$v = x - \alpha e_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} - (-\sqrt{2}) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1+\sqrt{2} \\ 1 \\ 0 \end{pmatrix}$

$H_1 = I - 2\frac{vv^T}{v^T v}$ $\qquad v^T v = (1+\sqrt{2})^2 + 1^2 + 0^2 = 4 + 2\sqrt{2}$ $\qquad vv^T = \begin{pmatrix} (1+\sqrt{2})^2 & (1+\sqrt{2}) & 0 \\ (1+\sqrt{2}) & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

$$H_1 \approx \begin{pmatrix} -0{,}707 & -0{,}707 & 0 \\ -0{,}707 & 0{,}707 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$A^{(1)} = H_1 A = \begin{pmatrix} -\sqrt{2} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} -\sqrt{2} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 1 \end{pmatrix} \quad \text{we want} \quad \begin{pmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \end{pmatrix}$$

$$x = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 1 \end{pmatrix} \qquad \|x\| = \sqrt{\tfrac{1}{2} + 1^2} = \sqrt{\tfrac{3}{2}}$$

$$\alpha = -\sqrt{\tfrac{3}{2}}$$

$$v_2 = x - \alpha e_2 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 1 \end{pmatrix} - -\sqrt{\tfrac{3}{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} + \sqrt{\frac{3}{2}} \\ 1 \end{pmatrix}$$

$$v_2^T v_2 = 3 + \sqrt{3}$$

$$v_2 v_2^T = \begin{pmatrix} \left(\frac{1}{\sqrt{2}} + \sqrt{\frac{3}{2}}\right)^2 & \frac{1}{\sqrt{2}} + \sqrt{\frac{3}{2}} \\ \frac{1}{\sqrt{2}} + \sqrt{\frac{3}{2}} & 1 \end{pmatrix}$$

$$H_2 = \begin{pmatrix} 1 & 0\,0 \\ 0 & \tilde{H}_2 \\ 0 & \end{pmatrix} \qquad \tilde{H}_2 = I_2 - 2\frac{v_2 v_2^T}{v_2^T v_2} \approx \begin{pmatrix} -0{,}577 & -0{,}816 \\ -0{,}816 & 0{,}577 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & -0{,}577 & -0{,}816 \\ 0 & -0{,}816 & 0{,}577 \end{pmatrix}$$

$$R = H_2 A^{(1)} \quad (H_2 H_1 A) \qquad Q = H_1 H_2$$

$$= \begin{pmatrix} -\sqrt{2} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & -\sqrt{\frac{3}{2}} & -\frac{1}{\sqrt{6}} \\ 0 & 0 & \frac{2}{\sqrt{3}} \end{pmatrix}$$

$$= \begin{pmatrix} -0{,}707 & -0{,}707 & 0 \\ -0{,}707 & 0{,}707 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -0{,}577 & -0{,}816 \\ 0 & -0{,}816 & 0{,}577 \end{pmatrix}$$

Python Script to visualize

Solve: $QRx = b$, $Q^T Q R x = Q^T b$ $\qquad Q^T = Q^{-1}$

$Rx = Q^T b$ $\quad$ | $y = Q^T b$, $Rx = y$ |

Substitutions

3. Givens Rotation

- We do a 2D rotation to make one entry zero. Repeating for all sub diagonal

$$G(i,j,\theta) = \begin{pmatrix} 1 & & & \\ & \ddots & c & -s \\ 0 & & s & c & \ddots & 1 \end{pmatrix}$$

$c = \cos\theta$
$s = \sin\theta$

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix}$$

$ca - sb = r$, $sa + cb = 0$

$s = -\frac{b}{a} c$

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

$a = 1$
$b = 1$

$r = \sqrt{1^2 + 1^2} = \sqrt{2}$

$c = a/r \simeq 0{,}707$

$s = -b/r \simeq -0{,}707$

$$G_{12} = \begin{pmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0{,}707 & 0{,}707 & 0 \\ -0{,}707 & 0{,}700 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$A^{(1)} = G_{12} A = \begin{pmatrix} 1{,}414 & 0{,}707 & 0{,}707 \\ 0 & -0{,}707 & 0{,}707 \\ 0 & 1 & 1 \end{pmatrix}$$

$a = -0{,}707$
$b = 1$

$r \simeq 1{,}225$ $\quad c = a/r \quad s = -b/r$

$$G_{23} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -0{,}577 & 0{,}816 \\ 0 & -0{,}816 & -0{,}577 \end{pmatrix}$$

$$R = G_{23} G_{12} A = G_{23} A^{(1)} = \begin{pmatrix} 1.414 & 0{,}707 & 0{,}707 \\ 0 & 1.225 & 0{,}408 \\ 0 & 0 & 1.155 \end{pmatrix} \qquad Q = G_{12}^T G_{23}^T$$

Python Script to visualize

Solve: $QRx = b$, $Q^TQRx = Q^Tb$ $Q^T = Q^{-1}$

$Rx = Q^Tb$ | $y = Q^Tb$, $Rx = y$ |
Substitutions

SVD - Singular Value Decomposition

SVD (full)	≈ **4·n³** (order-of-magnitude)	≈ **2·n²** (U and V)	Ill-conditioned or rank-deficient problems; gives rank/conditioning/pseudoinverse.
SVD (m×n, full)	≈ **O(m n²)** (constant bigger than QR)	≈ **O((m+n)·min(m,n))** (store U,V)	Best for rank detection, pseudoinverse; expensive for large dense matrices.

$A \in \mathbb{R}^{m\times n}$ $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$, $\Sigma = diag(\sigma_1 \dots \sigma_n)$

$A = U\Sigma V^T$: orthogonal change of basis → axis-aligned scaling → orthogonal change of basis

- $\sigma_i^2 = \lambda_i(A^TA)$ singular values are sqr of eigenvalues of A^TA
  - A^TA is symmetric positive semidefinite, $\lambda \geq 0$
- Symmetric Matrices $\xrightarrow{SVD}$ $A = U\Sigma U^T$ (eigendecomposition)

Moore-Penrose Pseudoinverse : $A^+ = V\Sigma^+U^T$ $\Sigma^+ = diag(1/\sigma_1 \dots 1/\sigma_r)$

- $Av^{(i)} = s_iU^{(i)}$; $U\overset{\sigma}{\Sigma}\overset{I}{\cancel{V^TV}} = s_iU$
- $A^Tu^{(i)} = s_iv^{(i)}$; $V\Sigma^TU^TU^{(i)} = s_iv^{(i)}$

- $A^TA = US\underbrace{V^TV}_{I}S^TU^T = USS^TU^T = US^2U^T$

- Low-Rank Approximation : $A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$, Properties:

- Column space: span of first r columns in U
- Row space: span of first r columns in V
- Nullspace: columns of V where $\sigma_i = 0$
- Left nullspace: columns of U corresponding to $\sigma_i = 0$

Computation:

1. A^TA or AA^T (depending which is easier)
2. Compute eigenvalues of A^TA ($\det(A^TA - \lambda I) = 0$)
3. Singular values ($\sigma_i = \sqrt{\lambda_i}$, sorted large to small)
4. Compute right singular vectors ($(A^TA - \lambda_i I)v_i = 0$), normalize, orthonormalize $\to V$
5. Compute U: $u_i = \frac{1}{\sigma_i} A v_i$

$A = \begin{pmatrix} 1 & 0 \\ 2 & 1 \\ 0 & 1 \end{pmatrix}$

1. $A^TA = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 2 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 5 & 2 \\ 2 & 2 \end{pmatrix}$

2. $\det(A^TA - \lambda I) = 0$, $\det\begin{pmatrix} 5-\lambda & 2 \\ 2 & 2-\lambda \end{pmatrix} = (5-\lambda)(2-\lambda) - 4 = \lambda^2 - 7\lambda + 6 \overset{!}{=} 0$

   $\lambda_1 = 6$, $\lambda_2 = 1$

3. $\sigma_1 = \sqrt{6}$, $\sigma_2 = 1$ (sort by size $\sigma > \sigma > \sigma > 0$)

4. $\lambda = 6$:

   $(A^TA - 6I)v = 0$, $\begin{pmatrix} -1 & 2 \\ 2 & -4 \end{pmatrix} v = 0$, $v_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ $\|v_1\| = \sqrt{5}$

   $v_1 = \frac{1}{\sqrt{5}} \begin{pmatrix} 2 \\ 1 \end{pmatrix}$

   $\lambda = 1$ $(A^TA - I)v = 0$ $\begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix} v = 0$, $v_2 = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$ $\|v_2\| = \sqrt{5}$

   $v_2 = \frac{1}{\sqrt{5}} \begin{pmatrix} -1 \\ 2 \end{pmatrix}$

$$\Rightarrow V = (v_1\ v_2) = \frac{1}{\sqrt{5}}\begin{pmatrix}2 & -1\\ 1 & 2\end{pmatrix}$$

5. $u_i = \frac{1}{\sigma_i} A v_i$

a) $\sigma = \sqrt{6}$, $v_1 = \frac{1}{\sqrt{5}}\begin{pmatrix}2\\1\end{pmatrix}$

$$u_i = \frac{1}{\sqrt{6}}\begin{pmatrix}1 & 0\\ 2 & 1\\ 0 & 1\end{pmatrix}\cdot\frac{1}{\sqrt{5}}\begin{pmatrix}2\\1\end{pmatrix} = \frac{1}{\sqrt{30}}\begin{pmatrix}2\\5\\1\end{pmatrix}$$

b) $\sigma = 1$, $v_2 = \frac{1}{\sqrt{5}}\begin{pmatrix}-1\\2\end{pmatrix}$

$$u_2 = 1\cdot\begin{pmatrix}1 & 0\\ 2 & 1\\ 0 & 1\end{pmatrix}\cdot\frac{1}{\sqrt{5}}\begin{pmatrix}-1\\2\end{pmatrix} = \frac{1}{\sqrt{5}}\begin{pmatrix}-1\\0\\2\end{pmatrix}$$

c) because $A \in 3\times 2$ we need 1 more u

we need sth orthogonal to u_1 and u_2 we can use gramschmidt or cross product

cross product: $u_3 = u_1 \times u_2 = \frac{1}{\sqrt{30}}\begin{pmatrix}2\\5\\1\end{pmatrix}\times\begin{pmatrix}-1\\0\\2\end{pmatrix}\frac{1}{\sqrt{5}} = \frac{1}{\sqrt{150}}\begin{pmatrix}10\\-5\\5\end{pmatrix} = \begin{pmatrix}\frac{2}{\sqrt{6}}\\ -\frac{1}{\sqrt{6}}\\ \frac{1}{\sqrt{6}}\end{pmatrix}$

Find: $U = \begin{pmatrix}\frac{2}{\sqrt{30}} & -\frac{1}{\sqrt{5}} & \frac{2}{\sqrt{6}}\\ \sqrt{\frac{5}{6}} & 0 & -\frac{1}{\sqrt{6}}\\ \frac{1}{\sqrt{30}} & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{6}}\end{pmatrix}$ $\Sigma = \begin{pmatrix}\sqrt{6} & 0\\ 0 & 1\\ 0 & 0\end{pmatrix}$ $V^T = \frac{1}{\sqrt{5}}\begin{pmatrix}2 & 1\\ -1 & 2\end{pmatrix}$

Solve: $Ax = b$

$$\Rightarrow U\Sigma V^T x = b,\ \overset{I}{\cancel{U^T U}}\Sigma V^T x = U^T b,\ x = V\Sigma^+ U^T b,\ x = A^+ b$$

Python SVD script.

Vector Spaces

Column Space: Output space $\mathbb{R}^m$, what outputs A can produce

Row Space: Input space $\mathbb{R}^n$, which input direction actually affects output

Nullspace: Input space $\mathbb{R}^n$, which inputs get Killed/canceled out

Left Nullspace: Output space $\mathbb{R}^m$, which outputs are impossible/unobservable

n : unknown variables, m : Equations

SVD:

$$A = U \Sigma V^T$$

input $x \xrightarrow{V^T}$ input coordinates $\xrightarrow{\Sigma}$ scaled modes $\xrightarrow{U}$ output b

$$x \qquad V^T = \begin{bmatrix} — V_1 \\ — V_r \\ — V_{null} \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots \; \sigma_r & \\ & & 0_{\ddots} \end{bmatrix} \qquad U = \begin{bmatrix} | & | & | \\ U_1 & U_r & U_{null} \end{bmatrix} \qquad b$$

weight of u_i

$$\text{Output } b \quad \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} | & | & | \\ U_1 & U_r & U_{null} \end{bmatrix} \begin{bmatrix} \sigma_1 x' \\ \sigma_r y' \\ 0 \end{bmatrix} = \begin{bmatrix} \sigma_1 & & \\ & \ddots \; \sigma_r & \\ & & 0_{\ddots} \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} — V_1 \\ — V_r \\ — V_{null} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Left nullspace

Null space

Column Space

Row space

$r = \text{rank}(A)$

$\sigma > 0$: active directions

$\sigma = 0$: Killed directions

Control systems: $\dot{x} = Ax + Bu,\ y = Cx$

Controllable vs uncontrollabe

Observable vs unobservable

and how much.

Vector Space definition:

Rules: $x, y, z \in V \quad \forall \lambda, \mu \in \mathbb{R}$

- $x + y = y + x$
- $(x+y)+z = x+(y+z)$
- $\exists$ "0" $\in V \Rightarrow x + "0" = x$
- $\forall x \in V \; \exists$ "$-x$" $\in V \Rightarrow \quad x + "-x" = "0"$
- $\mu(\lambda x) = (\mu\lambda) x$
- $(\mu + \lambda)x = \mu x + \lambda x, \quad \mu(x+y) = \mu x + \mu y$
- $I \cdot x = x$

Properties:

$a, b \in V, \; \alpha \in \mathbb{R}$

- $b + a = a \Rightarrow b = 0$
- $\alpha \cdot 0 = 0 \in V$
- $0 \cdot a = 0 \in V$
- $(-1) a = -a$

Norm

$\|\cdot\|$ assigns a non-negative length to a vector in a vector space

$\|v\| \geq 0, \quad \|v\| = 0 \Leftrightarrow v = 0$

measures vector length

$\|\alpha v\| = |\alpha| \cdot \|v\|$

$\|v + w\| \leq \|v\| + \|w\|$

Euclidean norm:

$\sqrt{x_1^2 \cdots x_n^2} = \sqrt{x^T x}$

$\|x\|_\infty = \max\{|x_1| \cdots |x_n|\}$

$\|x\|_p = (|x_1|^p + \cdots |x_p|^p)^{1/p}$

Inner Product

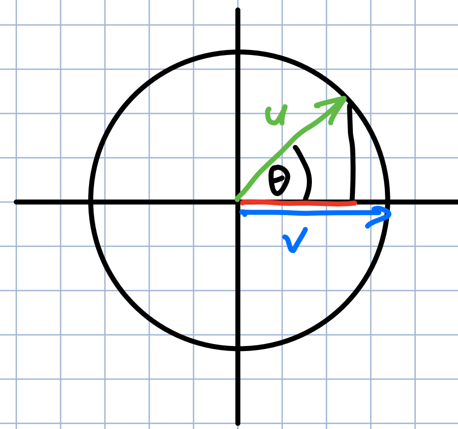
$\langle \cdot , \cdot \rangle \quad V \times V \to \mathbb{R}$

definition:

- linearity: $\langle \alpha u + \beta v, w \rangle = \alpha \langle u, w \rangle + \beta \langle v, w \rangle$
- symmetry: $\langle u, v \rangle = \langle v, u \rangle$
- positive definiteness: $\langle v, v \rangle \geq 0$, $\langle v, v \rangle = 0 \Leftrightarrow v = 0$

measures how aligned two vectors are

$\langle u, v \rangle = \|u\|\|v\| \cos\theta$, $\cos\theta = \dfrac{\langle u, v \rangle}{\|u\|\|v\|}$



$\langle u, v \rangle = u^T v$

weighted:

$\langle u, v \rangle_W = u^T W v$

Orthogonal

$u \perp v \Longleftrightarrow \langle u, v \rangle = 0$

(Orthonormal basis $\rightarrow$ $\langle e_i, e_j \rangle = \delta_{ij}$ (konecker delta $\begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases}$)

Functions:

$\langle f, g \rangle = \int f(t)\, g(t)\, dt$

Properties:

Orthogonal

means "perpendicular" or "independent" from each other

- $\langle u \cdot v \rangle = 0$
- $\int f(x) g(x) dx = 0$
- uncorrelated
- $Q^T Q = I$ (orthonormal) $\rightarrow Q^{-1} = Q^T$

- $\|Qx\| = \|x\|$ (perserves length and angles)

Determinant

Volume scaling factor of the linear map

properties:

- $\det(AB) = \det(A) \cdot \det(B)$
- $\det(A^T) = \det(A)$
- $\det(I) = 1$, if A orthogonal $\det(A) = \pm 1$
- $\det(\alpha A) = \alpha^n \det(A)$
- row swap $\Rightarrow$ $\det \cdot (-1)$
- A invertible $\Leftrightarrow$ $\det(A) \neq 0$, $\det(A^{-1}) = \det(A)^{-1}$
- $A^{-1} = \frac{1}{\det(A)} \operatorname{adj}(A)$
- Eigenvalues: $\det(A) = \prod_{i=1}^{n} \lambda_i$
- Singular values: $|\det(A)| = \prod_{i=1}^{n} \sigma_i$
- Exponential: $\det(e^A) = e^{tr(A)}$
- log: $\log \det(A) = tr(\log A)$ (A is positive definite)
- Multilinearity: (used in computer vision)
  - Additivity: $\det(v_1, a+b, \cdots v_n) = \det(v_1, a, \cdots v_n) + \det(v_1, b, \cdots v_n)$
  - Homogenity: $\det(v_1, k v_i, \cdots v_n) = k \cdot \det(v_1, v_i \cdots v_n)$
  - linear dependent row: $\det = 0$

Calculations:

- Gauss / LU

$$\det(A) = \underset{\text{(sign of row swaps)}}{\det(P)} \prod_{i=1}^{n} U_{ii} \quad \left((-1)^{\text{row swaps}} \cdot \text{all the values on diagonal} \right)$$

- Eigen decomposition for symmetric (diagonalizable) matrices:

  if $A = Q \Lambda Q^{-1}$, $\det(A) = \det(\Lambda) = \prod \lambda_i$

- Block triangular

  $$A = \begin{pmatrix} A_{11} & x \\ 0 & A_{22} \end{pmatrix}, \quad \det(A) = \det(A_{11}) \cdot \det(A_{22})$$

- Matrix determinant lemma

  $$\det(A + uv^T) = \det(A) \cdot (1 + v^T A^{-1} u)$$

- Block determinant (schur complement)

  $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ iff A invertible

  $$\det(M) = \det(A) \cdot \det(D - CA^{-1}B)$$

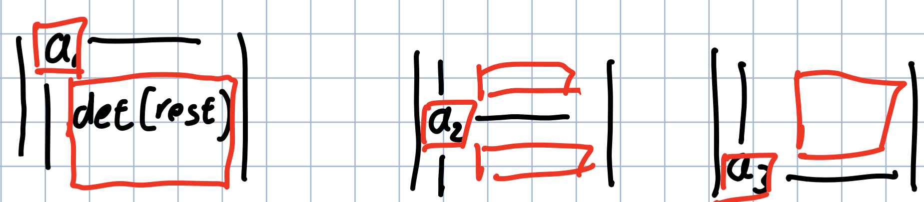
  - Probability ($x \sim N(\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix})$ $\mathrm{cov}(x_2 | x_1) = \Sigma_{22} - \Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12}$)

- 2X2 matrix

  $$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad \det(A) = ad - cb$$

- Laplace

  $$\begin{bmatrix} + & - & + \\ - & + & - \\ + & - & + \end{bmatrix} \quad \det \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} = a_1 \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix} - a_2 \begin{vmatrix} b_1 & c_1 \\ b_3 & c_3 \end{vmatrix} + a_3 \begin{vmatrix} b_1 & c_1 \\ b_2 & c_2 \end{vmatrix}$$



Eigen Values & Eigen Vectors

Definition:

- $Av = \lambda v$
- $(A - \lambda I)v = 0 \rightarrow \lambda$ is EVal iff $\det(A - \lambda I) = 0$

Eigenvectors are directions that A simply scales by λ (Eigenvalues)

- $\det(A) = \prod_{i=1}^{n} \lambda_i$
- $tr(A) = \sum_{i=1}^{n} \lambda_i$
- iff triangular A. EVals are diagonal entries
- $B = S^{-1}AS$ have same eigenvalues
- if A is real and has complex Evals λ, $\bar{\lambda}$ is also an EVal
- Symmetric: $A = A^T$, $A = Q \Lambda Q^T$
- Normal matrix: $AA^* = A^*A$
- SVD: $\sigma_i = |\lambda_i|$ iff A normal and diagonizable

Computation:

- Characteristic polynomial

$p_A(\lambda) = \det(A - \lambda I)$

$\det(A - \lambda I) = 0$

$$A = \begin{pmatrix} 0 & 0 & 6 \\ 1 & 0 & -11 \\ 0 & 1 & 6 \end{pmatrix} \qquad A - \lambda I = \begin{pmatrix} -\lambda & 0 & 6 \\ 1 & -\lambda & -11 \\ 0 & 1 & 6-\lambda \end{pmatrix} \qquad \det(A-\lambda I) = -\lambda \cdot (-\lambda \cdot (6-\lambda)) +$$

$$-\lambda \cdot -(1 \cdot -11) + -1 \cdot -1 \cdot 6 = -\lambda^3 + 6\lambda^2 - 11\lambda + 6$$

$$tr(A) = 6 = \Sigma \lambda_i$$

$$\lambda^3 - 6\lambda^2 + 11\lambda - 6 = (\lambda - 1)(\lambda - 2)(\lambda - 3)$$

EVal: $\lambda_1 = 1$, $\lambda_2 = 2$, $\lambda_3 = 3$

Algebraic multiplicity of λ: are all 1 in this example.

how many times λ appears as a root.

E Vec:

$\lambda = 1$

$$A - 1 \cdot I = \begin{pmatrix} -1 & 0 & 6 \\ 1 & -1 & -11 \\ 0 & 1 & 5 \end{pmatrix} \Rightarrow \begin{cases} -x + 6z = 0 \\ x - y - 11z = 0 \\ y + 5z = 0 \end{cases} \quad \begin{array}{l} x = 6z \\ 6z - 11z = y, \ -5z = y \\ y = -5z \ \checkmark \end{array}$$

we choose $z = 1$ $\quad V_1 = \begin{pmatrix} 6 \\ -5 \\ 1 \end{pmatrix}$

$\lambda = 2$

$$A - 2I = \begin{pmatrix} -2 & 0 & 6 \\ 1 & -2 & -11 \\ 0 & 1 & 4 \end{pmatrix} \Rightarrow \begin{cases} -2x + 6z = 0 \\ x - 2y - 11z = 0 \\ y + 4z = 0 \end{cases} \quad \begin{array}{l} x = 3z \\ y = -4z \\ y = -4z \ \checkmark \end{array}$$

we choose $z = 1$ $\quad V_2 = \begin{pmatrix} 3 \\ -4 \\ 1 \end{pmatrix}$

$\lambda = 3$

$$A - 3I = \begin{pmatrix} -3 & 0 & 6 \\ 1 & -3 & -11 \\ 0 & 1 & 3 \end{pmatrix} \Rightarrow \begin{cases} -3x + 6z = 0 \\ x - 3y - 11z = 0 \\ y + 3z = 0 \end{cases} \quad \begin{array}{l} x = 2z \\ y = -3z \\ y = -3z \ \checkmark \end{array}$$

choose $z = 1$ $\quad V_3 = \begin{pmatrix} 2 \\ -3 \\ 1 \end{pmatrix}$

Geometric multiplicity of λ

dimension of eigenspace $\ker(A-\lambda I)$

(How many Evec we have)

$$A=\begin{pmatrix}2&0&0\\0&2&1\\0&0&3\end{pmatrix} \quad \lambda_1=2 \quad \lambda_2=2 \quad \lambda_3=3$$

$AM(2)=2$, $AM(3)=1$

$A-2I=\begin{pmatrix}0&0&0\\0&0&1\\0&0&1\end{pmatrix} \Rightarrow \begin{matrix}0=0\\x_3=0\\x_3=0\end{matrix} \Rightarrow$ we can feely choose x_1, x_2

$\Rightarrow \begin{pmatrix}1\\0\\0\end{pmatrix}\begin{pmatrix}0\\1\\0\end{pmatrix}$ $GM=2$

iff $\begin{pmatrix}0&1&0\\0&0&0\\0&0&1\end{pmatrix} \Rightarrow \begin{matrix}x_2=0\\0=0\\x_3=0\end{matrix} \Rightarrow$ only choose x_1

$\Rightarrow \begin{pmatrix}1\\0\\0\end{pmatrix}$ $GM=1$

$$\text{EVal}=(3,2,1)$$

$$\text{EVec}=\begin{pmatrix}2&3&6\\-3&-4&-5\\1&1&1\end{pmatrix}$$

if $AM=GM$ then A is diagonizable $A=V\Lambda V^{-1}$

and it's called easy.

Schur Decomposition:

For square matrices there exists orthogonal Q and a quasi-upper-triangle T such that $A=QTQ^T$

- Eigenvalues on diagonal
- if A is normal (symmetric) then T is diagonal

→ easier matrix exponential, matrix function, lyapunov, log etc.

→ $f(A)=U f(T) U^*$

Use QR to get Schur

```python
#Schur decomposition
T: ndarray[_Shape, dtype[Any]] | NDArray[lon…, Q = schur(A, output="real")

print("Orthogonal matrix Q:")
print(Q)

print("\nSchur matrix T (upper triangular):")
print(T)
```

0.0s

```
Orthogonal matrix Q:
[[ 0.762001  0.639035  0.104828]
 [-0.635001  0.705601  0.314485]
 [ 0.127    -0.306204  0.943456]]

Schur matrix T (upper triangular):
[[ 1.       -4.088311 11.595824]
 [ 0.        2.       -5.461005]
 [ 0.        0.        3.      ]]
```

$A_0 = A$

$A_k = Q_k R_k$

$A_{k+1} = R_k Q_k$

$Q = Q_0 Q_1 Q_2$

$A = Q T Q^*$

See Python Script Characteristic Polynomial

Special Matrices

- **Square:** $n \times n$, input output are in same space

- **Symmetric:** $A^T = A$, diagonizable, real Evals, eigendecomposition can do LDL
  - real Eigen values
  - Orthogonal Eigenvectors with Diagonal Eigenvalues
    $A = Q \Lambda Q^T$
  - Spectral decomposition
    $A = \sum_{i=1}^{n} \lambda_i u_i u_i^T$
  - SVD = Eigendecomposition with $\sigma_i = |\lambda_i|$
  - Functions:
    $f(A) = Q f(\Lambda) Q^T$
  - Inverse: $A^{-1} = Q \Lambda^{-1} Q^T$
  - low rank approximation: $A_k = \sum_{i=1}^{k} \lambda_i u_i u_i^T$
  - Sylvester's law of inertia: $S^T A S$ positive/negative/zero Eval invariant

- Powers: $A^k = Q \Lambda^k Q^T$

- PSD

  can do cholesky decomposition (LL)

  - Covariance matrices, graph laplacian, Hessian in optimization
  - SVD = Eigendecomposition
  - Square root: $A^{1/2} = Q \Lambda^{1/2} Q^T$

- **Normal:** $A^TA = AA^T$: Orthogonal eigen basis
  - Cases:
    - Hermitian: $A = A^{*\,\text{or}\,T}$ Eigenvalues real
    - Unitary: $U^*U = I$ Eigenvalues on unit circle
    - skew Hermitian: $A^* = -A$ Eigenvalues pure imaginary
    - diagonal
    - rotation matrix, complex diagonals
  - $A = U \Lambda U^*$
  - SVD $\sigma_i = |\lambda_i|$
  - A^k is normal
  - $A^* = U \bar{\Lambda} U^*$
  - Schur = diagonal

- **Orthogonal / Unitary (complex)**

  $Q^TQ = I$, $Q^T = Q^{-1}$: Norm-perserving, det ± 1, EVals on unit circle

  - condition number $= 1$
  - perserve length and orthogonality
  - $Q^{-1} = Q^T$
  - Cases:
    - Householder

- Givens rotation
- Product of reflectors / rotations
- Cayley transform (skew-symmetric A) $A^T = -A$ $\quad Q = (I-A)(I+A)^{-1}$ if EVal $\neq -1$
  - $e^A \approx (I - \frac{1}{2}A)(I + \frac{1}{2}A)^{-1}$
  - $e^{\Delta t A} \approx (I - \frac{\Delta t}{2}A)(I + \frac{\Delta t}{2}A)^{-1}$
  - approximates $e^A = I + A + \frac{1}{2}A^2 + \frac{1}{6}A^3 \cdots$ to $I + A + \frac{1}{2}A^2 + O(A^3)$
  - Padé (1,1) approximation $\left(f(x) \simeq \frac{P_m(x)}{Q_n(x)}\right)$

- Triangular : $\begin{bmatrix} x & x & x \\ 0 & x & x \\ & & x \end{bmatrix}$: EVal on diagonal, $\det = \prod \lambda$, Easy A^k

- EVal are its diagonal entries
- Det $= \prod T_{ii}$
- Invertibility : iff all diagonal are nonzero, inverse is also a triangular
- closed under multiplication / addition : $T^1 +/\times T^2 = T^3$
- T^T switches type upper / lower
- only need forward / backward substitution for $Lx = b$ solve $O(n^2)$
- $\log\det(T) = \Sigma_i \log(T_{ii})$
- Exponents :
  - $E = e^T$, $TE = ET$, $E = \sum_{k=0}^{\infty} \frac{T^k}{k!}$ Parlett's recurrences

$$\begin{pmatrix} \lambda_1 & a & b \\ 0 & \lambda_2 & c \\ 0 & 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} e^{\lambda_1} & E_{12} & E_{13} \\ 0 & e^{\lambda_2} & E_{23} \\ 0 & 0 & e^{\lambda_3} \end{pmatrix} = \begin{pmatrix} e^{\lambda_1} & E_{12} & E_{13} \\ & e^{\lambda_2} & E_{23} \\ & & e^{\lambda_3} \end{pmatrix} \begin{pmatrix} \lambda_1 & a & b \\ 0 & \lambda_2 & c \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

$$TE_{12} = \lambda_1 E_{12} + a e^{\lambda_2} = a e^{\lambda_1} + E_{12}\lambda_2 \;,\; E_{12} = a\frac{e^{\lambda_1} - e^{\lambda_2}}{\lambda_1 - \lambda_2}$$

$$E_{23} = c\frac{e^{\lambda_2} - e^{\lambda_3}}{\lambda_2 - \lambda_3}$$

$i < k$

$$E_{ii} = e^{T_{ii}}$$

$$(\lambda_i - \lambda_k) E_{i,k} = T_{i,k}(e^{\lambda_i} - e^{\lambda_k}) + \sum_{n=i+1}^{k-1} (T_{i,n}E_{n,k} - E_{i,n}T_{n,k}) \quad E_{13} = \frac{b(e^{\lambda_1} - e^{\lambda_3}) + aE_{23} - cE_{12}}{(\lambda_1 - \lambda_3)}$$

- Decomposition Trick

$T = D + N$ (Diagonal + Nilpotent)

if $DN = ND \rightarrow (D+N)^k = \sum_{i=0}^{k} \binom{k}{i} D^{k-i} N^i$ best when $N^n = 0$

A^{10}, $A = \begin{pmatrix} 2 & 1 \\ 0 & 2 \end{pmatrix}$, $A = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = 2I + N$

$A^k = (2I+N)^k = \binom{k}{0} 2I^k + \binom{k}{1}(2I)^{k-1} N + \cancel{\binom{k}{2}(2I)^{k-2} N^2 + \cdots}$ as $N^{2+} = 0$

$A^{10} = 2^{10}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + 10 \cdot 2^9 \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1024 & 5120 \\ 0 & 1024 \end{pmatrix}$

- Jordan Normal Form

$J = \begin{pmatrix} \lambda & 1 & & \\ & \lambda & 1 & \\ & & \lambda & 1 \\ & & & \ddots \end{pmatrix}$ $\quad J^k_{ij} = \binom{k}{j-i} \lambda^{k-(j-i)}$

- **Diagonal:** only diagonal entries, easy powers, inverse, exponentials

  - Inverse: $D^{-1} = \mathrm{diag}\left(\frac{1}{d_1} \cdots \frac{1}{d_n}\right)$
  - Power: $D^k = \mathrm{diag}(d_1^k \cdots d_n^k)$
  - Exponent: "
  - $A^k = PD^kP^{-1}$
  - Jordan Form: if not diagonalizable $J = D + N$

- **Nilpotent:** $A^k = 0$, Eigenvalues are all 0, finite e^{At}

  - $\mathrm{rank}(N) < n$, $\mathrm{rank}(N^k)$ decreases
  - Jordan block $\begin{pmatrix} 0 & 1 & & \\ & 0 & 1 & \\ & & 0 & 1 \\ & & & \ddots \end{pmatrix}$
  - Never invertible but $(I-N)^{-1} = I + N + N^2 \cdots + N^{k-1}$, $N^k = 0$

- Exponentials $e^N = \sum_{k=0}^{\infty} \frac{N^k}{k!} = I + N + \frac{N^2}{2!} + \frac{N^3}{3!}$
  e^N is always invertible $(e^N)^{-1} = e^{-N}$
- Jordan-Chevalley $A = S + N$ (S diagonalizable)
- $A = PNP^{-1}$ if $N^k = 0 \to A^k = 0$
- $x' = Nx$, $x(t) = e^{Nt}x(0) \to$ growth is polynomial, marginally stable
- Directed acyclic graph (behaves like it, move from state $1 \to 2$ but never back)

Python animation

- Skew-Symmetric: $A^T = -A$

properties:

- diagonal $= 0$
- rank is even
- odd dimension $\Rightarrow$ singular
- Eigenvalues are imaginary or 0
- Normal $A^TA = AA^T$
- $(e^A)^Te^A = I$, $e^A \in O(n)$ (orthogonal n-dim) if $\det(e^A) = 1$ $e^A \in SO(n)$ special orthogonal
- Exponent $\to$ orthogonal matrix
- Rotation: A represents Axis of rotation, e^{At} is actual rotation
- A^{2k} is symmetric and negative semidefinite
- A^{2k+1} is skew-symmetric
- invertible only if n is even and $\text{rank}(A) = n$
- $x^TAx = 0$ for all x
- Pfaffian: $\det(A) = Pf(A)^2$

Orthogonal decomposition (Frobenius Inner Product)

$$M = \underbrace{\tfrac{1}{2}(M + M^T)}_{\text{symmetric}} + \underbrace{\tfrac{1}{2}(M - M^T)}_{\text{skew-symmetric}}$$

- (Symmetric) Positive Definite: $x^TAx > 0 \ \forall x \neq 0$

$x^TAx > 0 \ \forall x \neq 0$, $A^T = A$ (because $A = \underbrace{\tfrac{1}{2}(A+A^T)}_{S} + \cancel{\tfrac{1}{2}(A-A^T)}^{"0"}$)

Properties:

- invertible, A^{-1} also PD
- $\det(A) > 0$
- $tr(A)\ \sum \lambda_i > 0$
- all EVals > 0
- Orthogonally diagonalizable $A = Q\Lambda Q^T$
- can do cholesky LL^T
- $A^k = Q \ diag(\lambda_i^k)\ Q^T$
- Inverse "
- Exponential/Log/Fraction "
- Optimization, iff Hessian is PD $\Rightarrow$ global minimizer

- Idempotent: $A^2 = A$

$A^2 = A \quad \rightarrow \quad A(A-I) = 0$

Properties

- every eigenvalue satisfies $\lambda^2 = \lambda \rightarrow \lambda \in \{0, 1\}$
- characteristic poly $\chi_A(\lambda) = \lambda^{n-r}(\lambda-1)^r$
- diagonalizable
- rank(A) = nr of Eval = 1, $tr(A) = rank$
- $\det(A) = 0$ unless $A = I$
- $(I - A)$ is also indempotent
- $x = Ax + (I-A)x$, $Ax \in Im(A)$, $(I-A)x \in ker(A)$

- $p(A) = p(0)(I - A) + p(1)A$ (probability)
- $A^k = A$
- $e^A = e^0(I-A) + e^1 A = I + (e-1)A$
- $f(A) = f(0)(I-A) + f(1)A$

Covariance, Correlation, Stochastic

$Var(X) = Cov(X,X) = E[(X-\mu)(X-\mu)^T]$

μ = mean

$\sigma = \sqrt{Var(x)}$ σ^2 = Variance

σ = standard deviation

$Cov(X,Y) = E[XY] - E[X]E[Y]$

$\Sigma_{ij} = E[(X_i - \mu_i)(X_j - \mu_j)]$

$\Sigma_{ii} = Var(X_i)$

$Var(2X+Y)$, $z = \begin{bmatrix} X \\ Y \end{bmatrix} \to a = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

$= a^T \Sigma a = (2\ 1)\begin{bmatrix} 1 & 0.5 \\ 0.5 & 2 \end{bmatrix}\begin{pmatrix} 2 \\ 1 \end{pmatrix}$ $\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 2 \end{bmatrix}$

$= 8$

Properties:
- Symmetric
- Positive semidefinite

Correlation:

$$\rho(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(X)\,Var(Y)}}, \quad R = D^{-\frac{1}{2}} \Sigma D^{-\frac{1}{2}}$$

$-1 \le \rho \le 1$

diagonals $= 1$

$\Sigma = \begin{bmatrix} 4 & 3 \\ 3 & 9 \end{bmatrix}$

$D = \begin{bmatrix} 4 & 0 \\ 0 & 9 \end{bmatrix}$

$D^{-1/2} = \begin{bmatrix} \frac{1}{\sqrt{4}} & 0 \\ 0 & \frac{1}{\sqrt{9}} \end{bmatrix}$

$R = D^{-\frac{1}{2}} \Sigma D^{-\frac{1}{2}}$

$= \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{bmatrix}$

Gaussian Distribution: $X \sim N(\mu, \Sigma)$

Density: $e^{\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)}$

Conditioning / Schur $\quad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \quad \Sigma_{1|2} = \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}$

9. Powers and Exponentials

Powers Definition

$A^k = \underbrace{A \cdot A \cdots A}_{k}$ Only for square Matricies

$A^0 = I$

$A^{-k} = (A^{-1})^k$

Properties:

- $A^m A^n = A^{m+n}$
- $(A^m)^n = M^{mn}$
- $(cA)^n = c^n A^n$
- $A^k B^k \neq (AB)^k$ unless $AB = BA$ (non-commutativity)

Diagonals:

- $D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$, $D^k = \begin{bmatrix} \lambda^k & 0 \\ 0 & \lambda^k \end{bmatrix}$
- $A = PDP^{-1}$, $A^k = PD^kP^{-1}$

Eigenvalues

- if λ is an eigenvalue of A, then λ^k is Eval of A^k

Jordan Form

- If $A = PJP^{-1}$, where $J = \begin{bmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & \lambda \end{bmatrix}$

$\rightarrow J^k = \begin{bmatrix} \lambda^k & k\lambda^{k-1} & \binom{k}{2}\lambda^{k-2} \\ 0 & \lambda^k & k\lambda^{k-1} \\ 0 & 0 & \lambda^k \end{bmatrix}$

Nilpotent

$N^m = 0$ for some m

Identity: $I^k = I$

Projection: $P^2 = P \Rightarrow P^k = P$

Involution: $A^2 = I \Rightarrow A^{2k} = I, \ A^{2k+1} = A$

Exponentials:

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

(Always converges, only for square matrices)

Use Case: solution for linear systems

$$\frac{d}{dt} x(t) = A\, x(t)$$

Sol: $x(t) = e^{At} x(0)$

Properties:

- $e^0 = I, \ e^I \neq eI$
- $e^{tA} = \sum_{k=0}^{\infty} \frac{(tA)^k}{k!}$ (skalar multiplication)
- $(e^A)^{-1} = e^{-A}$
- If $A = PBP^{-1}$, $e^A = Pe^B P^{-1}$
- Non Commutativity: $e^{A+B} \neq e^A e^B$

  except $AB = BA$, $e^{A+B} = e^A e^B$
- Diagonal: $e^D = diag(e^{\lambda_1} \cdots e^{\lambda_n})$
- If $A = PDP^{-1}$, $e^A = P e^D P^{-1}$
- Jordan: $e^{Jt} = e^{\lambda t} \begin{bmatrix} 1 & t & t^2/2 \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix}$
- Nilpotent: $N^k = 0$, $e^N = I + N + \frac{N^2}{2!} \cdots \frac{N^{k-1}}{(k-1)!}$
- Differential:

$$\frac{d}{dt} e^{At} = Ae^{At} = e^{At}A$$

- Skew symmetric: $A^T = -A \Rightarrow e^A$ is orthogonal
- Symmetric: $A^T = A \Rightarrow e^A$ is SPD
- $P^2 = P \Rightarrow e^P = I + (e-1)P$

- Padé - Approx:
  - $e^A \approx (I - \frac{1}{2}A)(I + \frac{1}{2}A)^{-1}$
  - $e^{\Delta t A} \approx (I - \frac{\Delta t}{2} A)(I + \frac{\Delta t}{2} A)^{-1}$

- BCH: $e^A e^B = e^C$?

$[A,B] = AB - BA$

$$C = A + B + \frac{1}{2}[A,B] + \frac{1}{12}[A,[A,B]] - \frac{1}{12}[B,[A,B]]$$