Lineare Algebra

Contents

We live in a discrete, digital age.

Übungsstunde Notizen:
- https://n.ethz.ch/~jamatter/
- https://www.felixgbreuer.com/linalg

Nice Blog: https://gregorygundersen.com/blog/tags/la/

Notation:

LGS - Lineare Gleichung System - linear system of equations

$\mathcal{A}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$ - the corresponding Abbildung von the matrix A. Matrix multiplication is often referred to as a function

1 Vector Operations

1.1 Euclidean Norm

$$\boldsymbol{x} \in \mathbb{R}$$
$$|\boldsymbol{x}| = \sqrt{x_1^2 + x_2^2 + ... + x_n^2}$$
$$\boldsymbol{x} \in \mathbb{C}$$
$$|\boldsymbol{x}| = \sqrt{|x_1|^2 + |x_2|^2 + ... + |x_n|^2}$$
$$\text{Where } |x_1| = \sqrt{\text{re}(x_1)^2 + \text{im}(x_1)^2}$$

1.2 Dot / skalar Product

$$\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$$
$$< \boldsymbol{x}, \boldsymbol{y} >= x_1 y_1 + x_2 y_2 + ... + x_n y_n = \boldsymbol{x}^T \boldsymbol{y}$$
$$\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^n$$
$$< \boldsymbol{x}, \boldsymbol{y} >= \overline{x_1} y_1 + \overline{x_2} y_2 + ... + \overline{x_n} y_n = \boldsymbol{x}^H \boldsymbol{y}$$

The dot product can be used to calculate the angle between two vectors:

$$< x, y >= |x||y| \cos\left(\widehat{(x, y)}\right)$$
$$\widehat{(x, y)} = \arccos \frac{< x, y >}{|x||y|}$$
$$x \perp y \text{ is Orthogonal} \Leftrightarrow < x, y >= 0$$

$\frac{<x,y>}{|x||y|} \in [-1, 1]$ is proven by the Cauchy–Schwarz inequality.

1.3 Cross / Vektor product

The cross product is only defined in $\mathbb{R}^3$ and results in a vector orthogonal to both inputs, whose direction can be determined with the right hand rule.

$$\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3$$
$$\boldsymbol{x} \times \boldsymbol{y} = \begin{vmatrix} \boldsymbol{e}^x & \boldsymbol{e}^y & \boldsymbol{e}^z \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix}$$

It can also be used for determining the angle between two vectors in $\mathbb{R}^3$:

$$\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3$$
$$\boldsymbol{x} \times \boldsymbol{y} = |\boldsymbol{x}||\boldsymbol{y}|\boldsymbol{n} \sin \theta$$

($\boldsymbol{n}$ is the unit vector normal to the plane spanned by $\boldsymbol{x}, \boldsymbol{y}$)

2 Fundamentals

2.1 Vectors

LTD: Review what linear algebra is all about: https://en.wikipedia.org/wiki/Linear_map

Lineare kombination - Summe von skalierten Vektoren
Linearly dependent - When a vector can be expressed as a linear combination of the others and thus doesn't add any information to a LGS.
Basis - A set of linearly independent vectors $e_1...e_n$ that span all of space R^n

Matrix multiplication comes from the motivation for an efficient way of representing linear combinations / transformations of space.

2.2 Geometry of an LGS

An LGS can be viewed geometrically (2D/3D) in multiple different ways:

1. A linear combination of vectors (the columns of the matrix), where we are solving for the set of scalars where the superposition of the vectors is equal to the RHS. The columns of the matrix can be viewed as basis vectors of a custom coordinate system, in which we need to find the equivalent of the RHS vector.

2. Alternatively it can be viewed as a set of line / plane equations (where each row is the normal vector to the plane, unsure if the coefficients are meaningful in ax + by=c) and solutions are points / lines of intersection.

2.3 Superposition

TODO: Expand this principle further

In this example, one of the LHS vectors is a linear combination of the other two. This results in the LGS only being able to express vectors in a single plane rather than the entire 3D space (it doesn't contain a 3rd linearly independent basis dimension).

Infinite solutions - if a 3D RHS vector lays in the plane spanned by a 2D basis, there are infinitely many combinations of the two dependent basis vectors that can reach the point.
No solutions - the vector does not lay perfectly on the plane, the LHS vectors lacks a component in its direction and no linear combination of the basis can reach it.

2.4 Line / Plane equations

The solution is the point at which the lines / planes represented by the horizontal equations intersect. There are many possible arrangements which we can visualise, especially in $3D$ space.

Unique solution - Common point of intersection of n non parallel lines / planes.
Infinite solutions - Sheaf of planes or if all lines are the same.
No solution - Not all lines / planes meet at a common point, which is more likely the more equations are introduced into the system. Examples: Parallel lines, trianglular prism from 3 planes.

3 Gaussian Elimination

Method for solving a $m \times n$ system of equations, easy to implement algorithmically and works for all dimensions.

Pivot - Element on the diagonal of an upper matrix that has a non 0 coefficient

Rank - number of non 0 pivots, ie (number of rows - number of Kompatibilitaetsbedingungen) - the number of linearly independent rows / columns - the number of dimensions of the output of a linear transformation.

Row Rank = Column Rank:

$$\text{Rank}(\boldsymbol{A}) = \text{Rank}(\boldsymbol{A^T})$$

Compatibility Conditions - All-0 rows at the bottom of the matrix. If their result is not 0 then there are no solutions for the system. If their result is 0 and the number of equations $\leq$ the number of variables, there are infinite solutions (the corresponding scalar can take any value and the RHS component remains 0).

Free Variables - Any variables not accounted for due to no pivot in their column are called *free variables*. These can be thought of as degrees of freedom, we are free to give them any arbitrary

value and the other variables for that specific solution in the linear combination then depend on these.

"Order is half of the work in maths." - *Dr. Vasile C. Gradinaru*

3.1 Tips

- Never divide / subtract in Gaussian elimination. Simply multiply by $\frac{1}{x}$ or -x and add.

- Switch rows columns carefully **before** carrying out additions.

- **Only** add the row who's pivot is currently being pursued to other rows! Otherwise it is difficult to capture the operation in the elimination matrix (more on this later).

- When switching rows to get existing pivots in the correct place, it is usually best to swap a line with a zero pivot with the row that has the largest pivot in that place.

- *U - Upper (sometimes: R - right) Matrix* - Matrix with 0s under the diagonal

- *L - Lower Matrix* - Matrix with 0s above the diagonal

- *Identity Matrix* - Matrix with 1s on the diagonal, everything else 0.

- *Tridiagonal Matrix* - Matrix with 3 diagonals, and otherwise 0s everywhere

- $\boldsymbol{A}\boldsymbol{x} = 0$ only has the trivial solution $\boldsymbol{x} = 0$, unless $\boldsymbol{A}$ contains free variables.

3.2 Types

Consider a $m \times n$ matrix with rank r:
- **Square** *Regular Matrix* - $r = n$ AND $m = n$ - has exactly one solution for arbitrary RHS; only the trivial solution when homogenous
- **Square** *Singular Matrix (single = strange and uncommon)* - Rank < n - has infinite / no solutions and has infinite non trivial solutions when homogenous
- *Overdetermined* - $m > n$ - An overdetermined LGS only has solutions for very specific RHS values (if the rows are linearly independent) and therefore has no inverse (singular). Solutions can however always be approximated (See Section 17.1.1).

4 Matrices

4.1 Transposed Matrix

For a matrix with notation:

$$i := 1, ..., m \text{ Zeilen}$$
$$j := 1, ..., n \text{ Spalten}$$
$$\boldsymbol{A} = \begin{bmatrix} a_{ij} \end{bmatrix}$$
$$\text{Example: } \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

The transposed matrix A^T is:

$$\boldsymbol{A^T} = \begin{bmatrix} a_{ji} \end{bmatrix}$$
$$\text{Example: } \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$$

This can be thought of as pinning the first elements of each row and letting the rest of the row swing down vertically.

4.1.1 Conjugate Tranpose

$A \in \mathbb{C}^{m \times n}$, A^H - The same as a transposed matrix however each element becomes its complex conjugate $\bar{a}$. In physics, this is often written as $A^\dagger$

A matrix which satisfies $A = A^H$ is called a Hermitian matrix ie. a complex symmetric matrix.

LTD: Investigate thoroughly https://en.wikipedia.org/wiki/Hermitian_adjoint

4.1.2 Properties

- Vectors may be treated like $\mathbb{R}^{n \times 1}, \mathbb{C}^{n \times 1}$ matrices and transposed in the same manner.
- Matrix addition / scalar multiplication is carried out in the same way as vectors.

4.1.3 Identities

$$(AB)^T = B^T A^T$$
$$(AB)^H = B^H A^H$$
$$(A + B)^T = A^T + B^T$$
$$(A + B)^H = A^H + B^H$$

The rank of a matrix is the same as its tranpose:

$$\text{Rank}(A) = \text{Rank}(A^T)$$

4.2 Minor Matrix

This is the matrix formed by removing a certain row and column, useful for computing the determinant in Laplace Expansion.

Consider the following matrix:

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

A certain minor can be written as $M_{i,j}$ where i and j respectively are the row and column number (1-indexed) to remove from the matrix:

$$M_{1,1} = \begin{pmatrix} \square & \square & \square \\ \square & 5 & 6 \\ \square & 8 & 9 \end{pmatrix} = \begin{pmatrix} 5 & 6 \\ 8 & 9 \end{pmatrix}$$

$$M_{3,2} = \begin{pmatrix} 1 & 3 \\ 4 & 6 \end{pmatrix}$$

4.3 Block Matrix

A matrix that can be broken down into submatrices, used when performing Schur decomposition:

As an example, the following matrix can be split into partitions and written as sub matrices:

$$A = \left(\begin{array}{ccc|c} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ \hline 9 & 10 & 11 & 12 \end{array}\right)$$

$$M_1 = \begin{pmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \end{pmatrix}, M_2 = \begin{pmatrix} 4 \\ 8 \end{pmatrix}, M_3 = (9 \ 10 \ 11)$$

$$A = \begin{pmatrix} M_1 & M_2 \\ M_3 & 12 \end{pmatrix}$$

As can be seen, not every element needs to be separated as a submatrix. This is simply chosen so that it represents the subject matter best.

4.4 Adjugate Matrix

TODO: cover adj matrix here https://en.wikipedia.org/wiki/Determinant#Adjugate_matrix

4.5 Matrix Multiplication

Can be thought as the combination of transformations of space.

Two matrices may only be multiplied if they have the same inner dimensions:

$$A_{X \times Y} \times B_{Y \times Z} = C_{X \times Z}$$

Several LGS with the same LHS can be solved simultaneously with matrix multiplication with the inverse of A:

$$X = [x_1, ..., x_n], B = [b_1, ..., b_n]$$

$$A^{-1}X = B$$

$\text{Rank}(AX) = \min(\text{Rank}(A), \text{Rank}(X))$

Matrix multiplication is usually not commutative, however always associative and distributive.

4.6 Inverse

The inverse of a square matrix A is denoted as A^{-1}, which reverses the transformation of space represented by matrix A. Therefore $AA^{-1} = I$.

The inverse can be used to solve a LGS for arbitrary RHS vectors and only exists for square matrices by definition.

Regular, invertable and full rank are synonyms meaning that a matrix has an inverse. Here are some equivalent conditions which show that a matrix $A \in \mathbb{R}^{n \times n}$ is regular:

- A is square
- $\text{Rank}(A) = \text{Rank}(A^T) = n$
- $Ax = b$ is solvable for any b
- $Ax = 0$ only has the trivial solution $x = 0$

IMPORTANT: Just because a matrix is not invertible, does not mean that an LGS it appears in has no solutions! It just cannot be guaranteed to have a single solution for arbitrary RHSs, for example an underdefined system of equations $m < n$ may have infinite solutions and an overdefined system $n < m$ only has a solution for very specific RHSs.

Identities:

$$(AB)^{-1} = B^{-1}A^{-1}$$

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

4.7 Elimination Matrix

Matrix used for tracking the process of Gaussian elimination. The LHS / RHS multiplied by the elimination matrix results in the current state of the elimination!

The elimination matrix can be used with any b to apply the steps of elimination, for example when the first row was multiplied by 2 and added to the 2nd row:

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ 2b_1 + b_2 \\ b_3 \end{pmatrix}$$

It starts as the identity matrix, then the scalar by which another row was multiplied by before adding is written in the position of the currently eliminated variable of the row it was added to.

Important: Keep the elimination matrix lower! This means that for the current column, only the current row who's pivot we are pursuing may be added to other rows. If this doesn't work, use a permutation.

Caution: when swapping rows, do NOT forget adjusting the Elimination Matrix accordingly, by simply swapping all non diagonal values in the rows (this is done in a mathematical manner with Permutation matrices later).

Properties of elimination matrices:
- The inverse of the elimination matrix is itself, but non diagonal values become negative. This makes sense intuitively, when considering the elements of $EE^{-1} = I$.
- Two lower elimination matrices (with no overlapping elements!) multiplied together is the identity matrix with the combination of both lower elements. This means we can combine steps of Gaussian elimination together nicely.

4.8 Permutation Matrix

Orthogonal Matrix used to track the permutation of rows during elimination. This is simply the identity matrix with the corresponding rows swapped.

4.8.1 Properties

$$P_{13} := \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

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

$$\text{Row permutation: } P_{13} \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = \begin{pmatrix} 7 & 8 & 9 \\ 4 & 5 & 6 \\ 1 & 2 & 3 \end{pmatrix}$$

$$\text{Column permutation: } \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} P_{13} = \begin{pmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \\ 9 & 8 & 7 \end{pmatrix}$$

4.9 Calculating the Inverse

The inverse can be calculated through full Gaussian elimination (row-reduced echelon form, ie. with back substitution already carried out so the LHS matrix is the identity matrix) with a RHS of $b = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix}$ and then finding which X results in $X^{-1}b =$ our eliminated original matrix (by simply reading the coefficients of each component of b).

This can be simplified as the so-called **Gauss-Jordan Elimination**. This can be described as the following transformation through regular Gaussian elimination. All operations happen on both sides:

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

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} \rightsquigarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{matrix} \frac{3}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{3}{4} \end{matrix}$$

5 LU Lower Upper (LR Left Right) Decomposition

A matrix can be decomposed into an upper and lower matrix, such that:

$$A = LU$$
$$PA = LU$$

This can be used to decouple the factorization phase from the actual solving phase in Gaussian elimination. When the number of RHS we need to solve for is relatively small and A is extremely large, it is more efficient to carry out LU Zerlegung and the additional steps to solve each system separately rather than to calculate the inverse through Gauss-Jordan elimination.

Proof of $A = LU$:

For steps $1, 2, 3, ..., n$ of Gaussian elimination, the end result of the LHS is an upper matrix.

$$E_n E_{n-1} E_{n-2} ... E_1 A = U$$

The chain of steps (elimination matrices) can be moved to the RHS by multiplying both sides by their inverses due to $EE^{-1} = I$:

$$A = E_n^{-1} E_{n-1}^{-1} E_{n-2}^{-1} ... E_1^{-1} U$$

The chain of elimination steps can of course be represented as a single lower matrix E, therefore:

$$A = E^{-1}U = LU$$

Where E^{-1} is very easy to find (non diagonal elements simply $\times -1$ as mentioned earlier).

Using a combination of Elimination matrices and row + column permutations (these are needed to preserve the diagonal 1s of the resulting elimination matrix), the entire Gaussian elimination process can be encoded as one L matrix.

5.1 Using the LU Decomposition

The decomposed system can then be used to solve for x in the following way:

1. $Ax = b$
2. Decompose into the form $PA = LU$
3. Replacing A as LU, $PLUx = b$
4. $Ux = P^{-1}L^{-1}b$, now x can be solved easily as U is already in reduced form and just requires back substitution

LTD: Other decomposition methods (Cholesky etc)

6 Unitary Matrices

Orthonormal - Normalised orthogonal vectors; ie. a set of vectors that are orthogonal between one another and have length (norm) 1.

Orthonormality can be tested with dot products:

- $< u, v >= 0$ - Orthogonal
- $< u, u >= 1$ - Normal

Unitary matrix A square matrix whose columns are perpendicular to each other (dot product 0) and their Euclidean Norms are 1, hence they have $\det = 1$.

$$Q^H Q = I \Leftrightarrow Q \text{ is Unitary}$$

The condition $Q^H Q = I$ ensures the dot product of every column with it self is 1 (they have length 1) and with other columns is 0 (the columns are orthogonal to each other) due to the 1s along the diagonal and 0s everywhere else in the identity matrix.

Furthermore, their eigenspaces are orthogonal to each other and $\lambda = |1|$ - they are a subset of the normal matrices ($A^H A = A A^H$).

LTD: Investigate unitary matrix applications

6.1 Orthogonal Matrices

Orthogonal matrix - A unitary matrix with only **real elements**. They do not change lengths or angles - ie they only rotate / reflect space.

The inverse of a rotation / reflection of space is logically its transpose (consider the rotation of base vectors to different axis):

$$Q^T Q = I : Q \text{ is Orthogonal}$$
$$\text{Sei P Orthogonal}$$
$$QP \text{ und } PQ \text{ sind auch Orthogonal}$$

Orthogonal matrix multiplication is still generally not commutative.

6.1.1 Properties of orthogonal matrices

Let Q be an orthogonal matrix:

- Preserves lengths:

$$< Qx, Qy >=< x, y >$$
$$|Qx| = |x|$$

- Preserves angles $\widehat{Qx, Qy} = \widehat{x, y}$
- Are always regular $\Leftrightarrow$ columns are linearly independent

6.1.2 Rotation Matrix

Matrix that rotates space by α degrees anticlockwise:[1]

$$R(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

Due to the identity $Q^{-1} = Q^T$ rotation clockwise is:

$$R(\alpha)^T = R(-\alpha) = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix}$$

Alternatively, this can be calculated by inserting $-\alpha$ into the anti-clockwise rotation matrix and applying the trigonometric symmetry identities.

---

[1]Derivation through trigonometry in script by Dr. V Gradinaru

For a rotation in a certain plane of $\mathbb{R}^n$, simply keep all columns and rows unaffected by the rotation as the identity matrix. For example a anti-clockwise rotation in the $x \times y$ plane of $\mathbb{R}^3$:

$$\begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$x \times z$ plane:

$$\begin{pmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{pmatrix}$$

and so on in higher dimensions...

Given's Rotation $G(\varphi) = R(\varphi)$ - Simply a name for a rotation by the angle φ on the plane spanned by two canonical unit vectors in the anti-clockwise direction

Rotations which are not confined to a plane can be achieved through a series of plane rotations multiplied together, still resulting in an orthogonal matrix. Alternatively, this can be found using a change of basis matrix and calculating a Given's rotation in the new basis.

6.1.3 Reflection (Householder) Matrix

The orthogonal Householder matrix Q represents the reflection of space (a vector x which does not lie on the plane) over an arbitrary plane with normal **unit** vector u:

$$|u| = 1$$

$$Q = I - 2uu^T$$

"Fuer die Computer sind alle Zahlen schön" - *Vasile Gradinaru*

7 QU Decomposition

QU Decomposition is a different approach to solving LGSs, where the matrix Q is orthogonal.

Likewise to LU decomposition, it can be used to solve LGSs:

$$Ax = b \Leftrightarrow QUx = b \Leftrightarrow Ux = Q^T b$$

Advantage:
- Reduced rounding errors due to the way computers represent floating point numbers

Disadvantage:
- 3 times as inefficient as LU-Zerlegung

The goal is to find a series of orthogonal transformations, which transform A into an upper matrix when multiplying it. Since orthogonal matrices multiplied together result in an orthogonal matrix, they can easily be combined into one and used to solve the LGS as described above.

7.1 Givens rotations

Considering a vector, for example $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$. We can transform x_2 into 0 by rotating it **clockwise** onto the x-axis, resulting in $\begin{pmatrix} r \\ 0 \end{pmatrix}$ the following applies:

$$r = \sqrt{x_1^2 + x_2^2}$$
$$\cos(\theta) = \frac{x_1}{r}$$
$$\sin(\theta) = \frac{x_2}{r}$$

This works in $\mathbb{R}^n$ dimensions.

Using these formulas with the inverse of a Given's rotation, we can simply replace cos and sin in the matrix with the fractions, targeting the row we don't mind changing and the element we currently want to transform into 0:

$$G(\theta)^{-1} = G(\theta)^T = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

By applying a series of Givens rotations, we can carry out the full Gaussian elimination, ending up with an Upper / Right matrix.

7.2 Householder reflection

The goal is to find a Householder matrix that reflects a column of A onto the axis of the row we are **not** trying to eliminate. As we know the resulting vector, for example $|\boldsymbol{x}| \cdot e_x$, the unit vector u is simply the normed vector between the original and resulting column of A:

$$\boldsymbol{x} \in \mathbb{R}^n$$
$$\boldsymbol{v} = \boldsymbol{x} - |\boldsymbol{x}| \cdot e_n$$
$$\boldsymbol{u} = \frac{\boldsymbol{v}}{|\boldsymbol{v}|}$$

The Householder matrix can then be found as before:

$$\boldsymbol{Q} = \boldsymbol{I} - 2\boldsymbol{u}\boldsymbol{u}^T$$

When targeting inner columns, the reflection matrix should only be in the bottom right corner of the current Q, to avoid ruining our previous progress (the rest of it stays as I. This means that we ignore the upper row(s).

7.3 Which method to choose?

- Givens rotation is great for targeting specfic elements to turn into 0, and it needs a series of rotations to reduce several dimensions at once. It is ideal if there are already several 0s in the column.
- Householder reflections have the power to turn all except one element of a column into 0s at once (reflect a vector in $\mathbb{R}^3$ directly onto the x-axis for example)

8 Linear Vector Spaces

"Es macht Spaß" - *Vasile Gradinaru*

Linear - Lines are mapped to lines after the transformation

$\mathbb{R}^n$ and $\mathbb{C}^n$ are only two examples of many possible vector spaces. Considering the vector space V, the following axioms are defined:

Ein reeller *Vektorraum / linearer Raum* V ist eine Menge mit zwei Operationen:

Addition von Elementen aus V (+):

$$a, b \in V : a + b \in V,$$

Multiplikation mit Skalaren ($\cdot$):

$$\alpha \in \mathbb{R}, a \in V : \alpha \cdot a \in V.$$

Zusätzlich müssen folgende Eigenschaften gelten:
1. Kommutativitätsgesetz: $a + b = b + a$ für alle $a, b \in V$.
2. Assoziativitätsgesetz: $(a + b) + c = a + (b + c)$ für alle $a, b, c \in V$.
3. Es gibt $0 \in V$, sodass $a + 0 = a$ für alle $a \in V$ (dieses $0 \in V$ heisst Nullvektor).
4. Für jedes $a \in V$ gibt es ein $-a \in V$, so dass $a + (-a) = 0$.
5. Kompatibilität mit der Multiplikation mit Skalaren:
   $\alpha(\beta a) = (\alpha\beta)a$ für alle $\alpha, \beta \in \mathbb{R}, a \in V$.
6. Die Addition der Skalare ist distributiv gegen die Multiplikation mit Elementen aus V:
   $(\alpha + \beta)a = \alpha a + \beta b$ für alle $\alpha, \beta \in \mathbb{R}, a \in V$;
   Die Adition in V ist distributiv gegen die Multiplikation mit Skalare:
   $\alpha(a + b) = \alpha a + \alpha b$ für alle $\alpha \in \mathbb{R}, a, b \in V$.
7. Neutralelement für die Multiplikation mit Skalaren: $1 \cdot a = a$ für alle $a \in V$.

Two linear spaces can proved to be equal to one another by proving $A \subseteq B \wedge B \subseteq A \therefore A = B$

8.1 Continuous Differentiable Functions

$$C^s[a, b] := \{f : [a, b] \to \mathbb{R}, f \text{ is continuous and has } s \text{ derivatives}\}$$

Continuous - The function is continuous between [a, b], meaning there are no gaps or jumps in this interval.

Has s derivatives - The derivative exists at all points in the interval $[a, b]$ (ie. it is never a vertical line with an infinite gradient) and so on s times.

The following red function is a member of $C^1[0, 1]$ but not $C^2[0, 1]$, as its first derivative jumps from a positive to a negative value and its derivative therefore doesn't exist at the point of the jump (non-continuous).
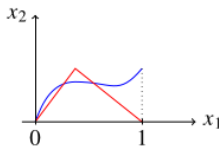
The trigonometric functions have infinite continuous derivatives:

$$\{\sin, \cos\} \in C^\infty[a, b]$$

The larger the value of s, the smaller the set. All functions in C^2 also belong in C^1 and so on...

$$C^\infty \subset \dots \subset C^3 \subset C^2 \subset C^1 \subset C^0$$

This set is a linear vector space, as addition and / or scalar multiplication result in a member of the same set, for example two continuous functions in an interval $[a, b]$ added together still result in a continuous function throughout the same interval.

8.2 Polynomials

$$\mathcal{P}_n := \{\text{polynomials of degree} \leq n - 1\}$$
$$\mathcal{P}_1 \subset \mathcal{P}_2 \subset \mathcal{P}_3 \subset ... \subset \mathcal{P}_\infty$$

For example the polynomial $x^2 + 1$ with degree 2 is a member of $\mathcal{P}_3$ but also all higher sets such as $\mathcal{P}_4$ as the coefficient of x^3 is simply 0.

Addition and **scalar** multiplication are indeed valid operations that result in a member of the same set, therefore it is a linear vector space.

The Taylor Series in Analysis demonstrates how any continuous function in C^0 can be approximated using $\mathcal{P}_n$.

8.3 Other Linear Spaces
- $L^2[a, b] := \left\{ f : [a, b] \to \mathbb{R}, \int_a^b |f|^2 dt \right\}$- Space of quadratically integrable functions
- $\ell := \{a_n \text{ converges}\}$- Space of convergent sequences
- $\ell := \{a_n \text{ converges}\}$- Space of convergent sequences

8.4 Linear Subspace

Sei V ein linearer Raum mit $U \subseteq V$ und $U \neq \emptyset$. Dann heisst U *Unterraum* von V, falls gilt:
1. Wenn $x, \ y \in U$ dann auch $x + y \in U$.
2. Wenn $\alpha \in \mathbb{R}, \ x \in U$ dann auch $\alpha x \in U$.

A linear subspace must also include a zero vector.

$$U := \left\{ \boldsymbol{x} \in \begin{pmatrix} x_1 \\ 0 \end{pmatrix} \in \mathbb{R}^2 \right\}$$

$$U' := \left\{ \boldsymbol{x} \in \begin{pmatrix} x_1 \\ 5 \end{pmatrix} \in \mathbb{R}^2 \right\}$$

U' is not a subspace, as vector addition leads to $\begin{pmatrix} x_{1'} \\ 10 \end{pmatrix}$ which has left the space.

A single contradictory example must be found to prove that something is not a linear subspace.

8.5 Span

The set of all possible linear combinations of elements of a linear space. For example the span of basis vectors of $\mathbb{R}^3$ is the entire $\mathbb{R}^3$, as any point in 3D space can be expressed as linear combination of the basis.

The range of the transformation of space $\mathcal{A}(x) = \boldsymbol{Ax}$ can be expressed as the span of its columns:

$$\mathcal{A} : \mathbb{R}^n \to \mathbb{R}^m$$
$$\text{Range } \mathcal{A} = \text{Span}\{A_1, A_2, ..., A_n\}$$

Erzeugendensystem - Set of vectors which span a vector space V.

Image / Column Space / Range / Codomain - of a transformation can be found by capturing any compatibility conditions in a vector (if there are any), for example of $b_3 - b_1 - b_2 = 0$, the range can be expressed as:

$$\left\{ b_1, b_2 \in \mathbb{R}, \begin{pmatrix} b_1 \\ b_2 \\ b_1 + b_2 \end{pmatrix} \right\}$$

This can also be expressed as a span by breaking it into a linear combination:

$$b_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + b_2 \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = \text{Span} \left\{ \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\}$$

Monome - $p_{n(t)} = t^n$, for example $p_0(t) = 1, p_1(t) = t, p_2(t) = t^2, ...$
The set of monomes is a linear subspace of the polynomials $\mathcal{P}_n$, which can be spanned by $\text{Span}\{p_k \mid k < n\}$

8.6 Kernel (Null-Space / Kern)

Let $A := m \times n$. The kernel is the set of vectors that is transformed to 0 by A.

The kernel can be found by solving the LGS $Ax = 0$ with Gaussian Elimination. This provides infinite solutions for a singular matrix or the null vector for a regular system.

Nullity - Dimensions of the null-space

Each linear space already has two simple subspaces:
- The kernel of its matrix
- The range of it as a transformation

8.7 Basis

Linearly Dependant - A vector in the set is linearly dependent if other vectors in the set can express it as a linear combination. In the case of two vectors:

$$\alpha \in \mathbb{R}$$
$$\boldsymbol{v_1} = \alpha \boldsymbol{v_2}$$

This check can be restructured as a linear combination of several vectors $\boldsymbol{v_n}$, where we find the kernel of them as a combined matrix, meaning the set of scalars which would express each column of the matrix as a linear combination of the other columns:

$$A = \begin{pmatrix} \boldsymbol{v_1} & \boldsymbol{v_2} & ... & \boldsymbol{v_n} \end{pmatrix}, \boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{pmatrix} \in \mathbb{R}^n$$

$$A\boldsymbol{x} = 0$$

This can be rearranged for any columns of A:

$$x_1 \boldsymbol{v_1} + x_2 \boldsymbol{v_2} + ... + x_n \boldsymbol{v_n} = 0$$
$$\boldsymbol{v_1} = \frac{x_2 \boldsymbol{v_2} + ... + x_n \boldsymbol{v_n}}{-x_1}$$

Therefore if only the trivial solution exists for $\boldsymbol{x}$ (A is regular), all vectors $\boldsymbol{v_n}$ are linearly independent. This is most easily checked through elimination.

Basis - A set of linearly independent vectors that spans the entire linear space (minimal Erzeugenden system) and stay completely within the linear space (not allowed to span a parent space as well).

There can be several independent bases in a space, but all bases have the same number of elements (dimension).

Any element in the linear space can be determined by a **unique** linear combination of the basis vectors.

Canonical basis - Bases taken "as canon" (accepted standard) for each space, for example $e_x, e_y, e_z \in \mathbb{R}^3$
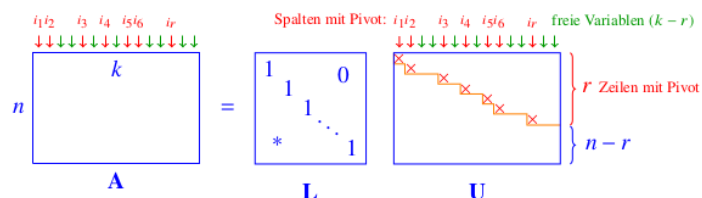
The basis of an erzeugenden system can be found through elimination, where the columns with a pivot are linearly independent of one another and therefore the corresponding columns in A are also a possible basis in the space.

8.7.1 Basis of $\mathcal{P}_n$ are the monomes

Basis of $n \in \mathbb{N}, \mathcal{P}_n = \{p_i = t^i \mid t \in \mathbb{R}, i \in \mathbb{N}_0 < n\}$ - The monomes are linearly independent (proof in script) and span the entire polynomial space.

A trick for proving linear independence of certain items, for example polynomials $q_1, q_2, \dots$ is expressing them in terms of another linear space that is known to be linearly independent, for example monomes and rearranging the basis check equation as follows:

$$q_1 = p_0 + p_2$$
$$q_2 = p_0 - p_2$$
$$q_n = \dots$$

$$x_1 q_1 + x_2 q_2 + \dots = 0$$
$$x_1(p_0 + p_2) + x_2(p_0 - p_2) + \dots = 0$$
$$(x_1 + x_2)p_0 + (x_1 - x_2)p_2 + \dots = 0$$
$$\begin{pmatrix} 1 & 1 & \dots \\ 1 & -1 & \dots \\ \dots & & \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \dots \end{pmatrix} = 0$$

Since the monomes have already been proved to be linearly independent, we can construct the coefficients containing x as an LGS with right hand side 0 and show that this matrix indeed only has the trivial solution (it is regular).

LTD: Continuous differentiable functions as monomes using the Taylor Series sounds like an interesting basis

8.7.2 Wronskian

Another method for checking the linear independence of n differentiable functions $f_1, f_2, \dots f_n$ is by finding the **Wronskian**:

$$W(f_1, f_2, ...f_n)(x) = \det \begin{pmatrix} f_1(x) & f_2(x) & \cdots & f_{n(x)} \\ f_1'(x) & f_2'(x) & \cdots & f_{n(x)}' \\ \vdots & \vdots & \ddots & \vdots \\ f_1^{n-1}(x) & f_2^{n-1}(x) & \cdots & f_{n(x)}^{n-1} \end{pmatrix}$$

Where $f^n(x)$ is the n'th derivative of the function at x.

For only two functions:

$$W(f, g) = fg' - gf'$$

If $W(...)(x) \neq 0 \forall x \in (a, b)$, the functions are linearly independent and suitable as a basis throughout the interval.

8.8 Dimensions

The number of elements in a basis of a space.

Finite Dimensional Vector Space - A finite set of basis vectors spans the entire space.

Examples:
- Finite dimensional: $\mathbb{R}^n, \mathcal{P}_n$ - Dimension = n. (Hence the awkward degree $\leq n - 1$ notation for $\mathcal{P}_n$)
- Non-finite dimensional: $\mathcal{P}, C^k, L^2$

Sometimes non-finite dimensional spaces can be approximated using finite dimensional spaces (Taylor series).

> **Satz 2.3.0.10. Grösse von erzeugenden Systemen in einem endlichdimensionalen Raum**
>
> *Sei V ein linearer Raum mit Dimension n, dann:*
>
> 1) *sind mehr als n Elemente von V linear abhängig.*
>
> 2) *sind weniger als n Elemente von V nicht erzeugend.*
>
> 3) *sind n Elemente von V linear unabhängig genau dann, wenn sie auch erzeugend sind.*

8.9 Fundamental Theorem of Linear Algebra (Gilbert Strang)

For a $n \times k$ matrix $\boldsymbol{A}$ with rank r:

$$\dim(\text{Im}(\boldsymbol{A})) = r$$
$$\dim(\text{Kernel}(\boldsymbol{A})) = k - r = \text{Number of free variables}$$

Transposing the matrix leads to ($\text{Rank}(\boldsymbol{A}) = \text{Rank}(\boldsymbol{A}^T)$):

$$\dim(\text{Im}(\boldsymbol{A}^T)) = r$$
$$\dim(\text{Kernel}(\boldsymbol{A}^T)) = n - r$$

For example, considering a 3x3 matrix P with rank 2:
- The transformation maps any x onto a 2D plane due to only 2 independent vectors in the basis, the image has dimension 2. A solution only exists if the RHS vector is in this plane.
- The kernel of the matrix $Px = 0$ includes an entire line of vectors, meaning that there is 1 free variable (the non-pivot column) that scales along this line.

LTD: Internalise / develop intuition

Orthogonal Spaces - Two linear spaces are orthogonal $U \perp V$ to one another when any two vectors in the spaces are always orthogonal to one another :$< \boldsymbol{u}, \boldsymbol{v} >= 0$

$$\text{Kernel}(A) \perp \text{Im}(A^T)$$
$$\text{Im}(A) \perp \text{Kernel}(A^T)$$

This law can be used to easily decompose any linear space into two orthogonal elements.

8.10 Coordinates

The coordinates of a specific element $x \in \mathbb{R}^n$ formed using the basis $\mathcal{B} = \{v_1, v_2, v_3, ...\}$ are written as:

$$x = x_1 \boldsymbol{v_1} + x_2 \boldsymbol{v_2} + x_3 \boldsymbol{v_3} + ...$$

$$\mathcal{k}_{\mathcal{B}} : \boldsymbol{V} \to \mathbb{R}^3 := \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ ... \end{pmatrix}$$

This mapping transforms a tuple of coordinates into the resulting point in $\mathbb{R}^n$ (commonly accepted canonical basis) using an abstract basis $\mathcal{B}$ so that we can perform operations on elements in different abstract basis as one.

Coordinates can of course be transformed to coordinates of another basis in the same linear space:
- Both coordinates are unique for their underlying basis and both can be mapped to a unique element in a common linear space.
- Therefore each basis vector in the target basis can be represented as a linear combination of the original basis.

8.10.1 Change of Basis

Let us assume we are working in the linear space $\mathbb{R}^n$. We want to determine matrices that transform coordinates between bases $\mathcal{B}_1$ and $\mathcal{B}_2$:
1. Express the target basis $\mathcal{B}_2$ vectors as coordinates in our basis $\mathcal{B}_1$
2. Set these coordinates as columns of a matrix $\boldsymbol{P} : \mathcal{B}_2 \to \mathcal{B}_1$. The order should be consistent with the order of coordinates in $\mathcal{B}_2$.
3. We can now multiply coordinates $x \in \mathcal{B}_2$, $\boldsymbol{Px} = \boldsymbol{b}$. This expresses the same linear combination of $\mathcal{B}_2$ basis vectors, but in coordinates of $\mathcal{B}_1$. $\boldsymbol{b} \in \mathcal{B}_1 = \boldsymbol{x} \in \mathcal{B}_2$
4. To convert from coordinates in $\mathcal{B}_1 \to \mathcal{B}_2$, we can simply use $\boldsymbol{P}^{-1}$

This sheds a new light on transformations. Any square matrix with full rank can be interpreted to represent a change in basis.
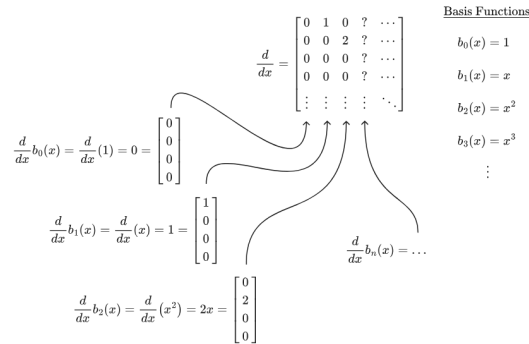
8.11 Linear Transformations

Useful resource: https://www.3blue1brown.com/lessons/abstract-vector-spaces

Here is the formal definition of a linear transformation:

$$\mathcal{F} : X \to Y$$
$$\mathcal{F}(x_1 + x_2) = \mathcal{F}(x_1) + \mathcal{F}(x_2) \forall x_1, x_2 \in X$$
$$\alpha \mathcal{F}(x) = \mathcal{F}(\alpha x) \forall x \in X, \alpha \in \mathbb{R}$$

This therefore means that the origin stays fixed.

For example, derivation is a linear transformation:

$$\frac{d}{dx} = \begin{bmatrix} 0 & 1 & 0 & ? & \cdots \\ 0 & 0 & 2 & ? & \cdots \\ 0 & 0 & 0 & ? & \cdots \\ 0 & 0 & 0 & ? & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Basis Functions

$b_0(x) = 1$

$b_1(x) = x$

$b_2(x) = x^2$

$b_3(x) = x^3$

$\vdots$

$$\frac{d}{dx}b_0(x) = \frac{d}{dx}(1) = 0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\frac{d}{dx}b_1(x) = \frac{d}{dx}(x) = 1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\frac{d}{dx}b_n(x) = \ldots$$

$$\frac{d}{dx}b_2(x) = \frac{d}{dx}(x^2) = 2x = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \end{bmatrix}$$

- Every linear transformation in a finite dimensional space can be represented as matrix multiplication.
- *Isomorphism* - A structure-preserving mapping, ie a bijective linear transformation is a called an isomorphism. The inverse is clearly also an isomorphism. A non-square matrix changes dimensions and thus is not bijective and isomorphic - no inverse.
- *Automorphism* - If the two sets the transformation maps between are the same.
- A linear transformation $\mathcal{F}$ is injective when $\text{Kernel}(\mathcal{F}) = \{0\}$.
- The same properties regarding dimensions sizes regarding basis apply to linear transformations:

Definition 3.2.0.7. Kernel und Bild

Sei $\mathcal{F}$ eine Abbildung, dann können folgende zwei Mengen definiert werden:

Kernel von $\mathcal{F}$:
$\text{Kern}(\mathcal{F}) = \{x \in X, \mathcal{F}x = 0\}$.
Wenn $\mathcal{F}$ eine lineare Abbildung ist, dann ist $\text{Kern}(\mathcal{F})$ ein linearer Unterraum von X.

Bild von $\mathcal{F}$:
$\text{Bild}(\mathcal{F}) = \{y \in Y, \text{ so dass } x \in X \text{ mit } \mathcal{F}x = y\}$.
Wenn $\mathcal{F}$ eine lineare Abbildung ist, dann ist $\text{Bild}(\mathcal{F})$ ein linearer Unterraum von Y.

Satz 3.2.0.8. Kern einer linearen Injektion

Wenn $\mathcal{F} : X \to Y$ eine lineare Abbildung ist, dann gilt:
$\mathcal{F}$ ist injektiv genau dann, wenn $\text{Kern}(\mathcal{F}) = \{0\}$.

Satz 3.2.0.9. Dimensionssatz

Wenn $\mathcal{F} : X \to Y$ eine lineare Abbildung zwischen den beiden endlichdimensionalen Räumen X und Y ist, dann gilt:

$$\dim(\text{Kern}(\mathcal{F})) + \dim(\text{Bild}(\mathcal{F})) = \dim(X).$$


Definition 3.2.0.10. Rang einer linearen Abbildung

Der Rang einer linearen Abbildung $\mathcal{F}$ zwischen zwei endlichdimensionalen Räumen ist

$$\text{Rang}(\mathcal{F}) = \text{Rang}(\mathbf{F}) = \dim(\text{Bild}(\mathbf{F})).$$

A linear transformation is independent of the basis / coordinates used and can be represented with respect to many different bases in the same space. The following diagram highlights these relationships well:

Sei X ein linearer Raum mit den beiden Basen $\mathcal{B}$ und $\widetilde{\mathcal{B}}$, sei Y ein anderer linearer Raum auch mit zwei Basen $\mathcal{C}$ und $\widetilde{\mathcal{C}}$ und sei $\mathcal{F} : X \to Y$ eine lineare Abbildung. Das Diagramm dazu ist:
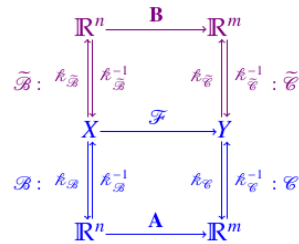
Es gibt also eine neue Matrix $\mathbf{B}$, welche auch die Abbildung $\mathcal{F}$ darstellt, und zwar mit den neuen Basen $\widetilde{\mathcal{B}}$, $\widetilde{\mathcal{C}}$. Da dies ein kommutatives Diagramm ist, können wir die Matrix $\mathbf{B}$ mit den internen Basistransformationen und $\mathbf{A}$ ausrechnen:

$$\mathbf{B} = \underbrace{\hbar_{\widetilde{\mathcal{C}}} \hbar_{\mathcal{C}}^{-1}}_{\mathbf{S}^{-1}} \mathbf{A} \underbrace{\hbar_{\mathcal{B}} \hbar_{\widetilde{\mathcal{B}}}^{-1}}_{\mathbf{T}} = \mathbf{S}^{-1} \mathbf{A}\, \mathbf{T}$$

$$\mathbf{B} = \mathbf{S}^{-1}\mathbf{A}\,\mathbf{T} \iff \mathbf{A} = \mathbf{S}\,\mathbf{B}\,\mathbf{T}^{-1}.$$

$\mathbf{T}$ realisiert den Wechsel von der Basis $\widetilde{\mathcal{B}}$ in die Basis $\mathcal{B}$ im linearen Raum X. D.h. $\mathbf{T}\widetilde{\mathbf{x}} = \mathbf{x}$ für den Koordinatenvektor $\mathbf{x}$ von $x \in X$ in der alten Basis $\mathcal{B}$ und den Koordinatenvektor $\widetilde{\mathbf{x}}$ von x in der neuen Basis $\widetilde{\mathcal{B}}$.

$\mathbf{S}$ realisiert den Wechsel von der alten Basis $\mathcal{C}$ in die neue Basis $\widetilde{\mathcal{C}}$ im linearen Raum Y.

8.11.1 Anti-linear (Conjugate-Linear) Transformation

$$\mathcal{A} : X \to Y$$
$$\mathcal{A}(x_1 + x_2) = \mathcal{A}(x_1) + \mathcal{A}(x_2) \forall x_1, x_2 \in X$$
$$\alpha \mathcal{A}(x) = \mathcal{A}(\overline{\alpha}x) \forall x \in X, \alpha \in \mathbb{C}$$

8.11.2 Linear Form (Linear Functional)

Simply any **linear** transformation that maps a linear space to one dimension $V \to \mathbb{R}$ or $V \to \mathbb{C}$.

LTD: Quaternions

This can be something as simple as $\varphi(x) \to 0$ or the dot product.

8.11.2.1 Riesz Representation Theorem

Let $\mathbb{V}$ be a linear space with a dot product and linear form φ. There exists a unique vector $w \in \mathbb{V}$ (called the Riesz representation) such that:

$$\varphi(v) = \ <w, v> \ \forall v \in \mathbb{V}$$

which can be represented in terms of any orthonormal basis $\{b_1, b_2, ..., b_n\} \in \mathbb{V}$:

$$w = \overline{\varphi(b_1)}b_1 + \overline{\varphi(b_2)}b_2 + ... + \overline{\varphi(b_n)}b_n$$

whereby the conjugation can be ignored if dealing with real linear forms.

This is a cool way of representing any linear form, essentially a regular linear transformation mapping to 1 dimension, as a single vector.

8.11.2.2 Dual Space

The linear vector space V' of all linear forms of a linear vector space V, such that:

$$\varphi, \psi \in V'$$
$$x \in V, \alpha \in \mathbb{R}/\mathbb{C}$$
$$(\varphi + \psi)(x) = \varphi(x) + \psi(x)$$
$$\alpha\varphi(x) = \varphi(\alpha x)$$
$$\alpha\psi(x) = \psi(\alpha x)$$

Each element of V' can be represented as a vector thanks to the Riesz representation theorem.

9 Norms

A norm is a function that transforms any element in a linear space to a positive real number. It must respect the following properties:

Definition 4.1.0.1. Norm

Sei V ein linearer Raum.
Die Funktion $\|\cdot\| : V \to [0, \infty[$ heisst Norm in V, falls sie folgende Eigenschaften erfüllt:

(N1) Aus $\|v\| = 0$ folgt $v = 0$.

(N2) Sei α ein Skalar und $v \in V$ beliebig. Dann gilt: $\|\alpha v\| = |\alpha| \|v\|$.

(N3) Für beliebige $v, w \in V$ gilt: $\|v + w\| \le \|v\| + \|w\|$ (Dreiecksungleichung).

Definition 4.1.0.2. Normierter linearer Raum

Ein linearer Raum V, welcher eine Norm $\|\cdot\|$ besitzt, heisst normierter linearer Raum.

Satz 4.1.0.6. Äquivalenz der Normen

Alle Normen in $\mathbb{R}^d$ sind äquivalent. In anderen Worten:
Seien $\|\cdot\|$ und $\|\|\cdot\|\|$ Normen in $\mathbb{R}^d$, dann gibt es eine Konstante $C \ge 1$, abhängig von der Dimension d, so dass

$$\frac{1}{C}\|\mathbf{v}\| \le \|\|\mathbf{v}\|\| \le C\|\mathbf{v}\| \text{ für alle } \mathbf{v} \in \mathbb{R}^d.$$

A basic example of a norm in $\mathbb{R}$ is the absolute function:

$$x \in \mathbb{R}$$
$$|x|$$

This is the only universal norm in one dimensional spaces. TODO: How is this an isomorphism?
https://en.wikipedia.org/wiki/Norm_(mathematics)

9.1 Taxicab Norm (L^1)

The name comes from the fastest zig zag path a taxi has to drive through in Manhattan's grid based street layout:

$$\|x\|_1 := \sum_{i=1} |x_i|$$

9.2 Euclidean Norm (L^2)

This ubiquitous norm represents the length of a vector and can alternatively be written in vector notation:

$$x \in \mathbb{R}^d$$

$$\|x\|_2 := \sqrt{x^T x}$$

$$x \in \mathbb{C}^d$$

$$\|x\|_2 := \sqrt{x^H x}$$

9.3 p-Norm

$$p \in \mathbb{R} \mid p \geq 1$$

$$\|x\|_p := \left(\sum_{k=1} |x_k|^p \right)^{\frac{1}{p}}$$

All L^n norms can be generalised as the p-norm.

Any vector can be normed to a unit vector with respect to any p norm:

$$x \in \mathbb{R}^d$$
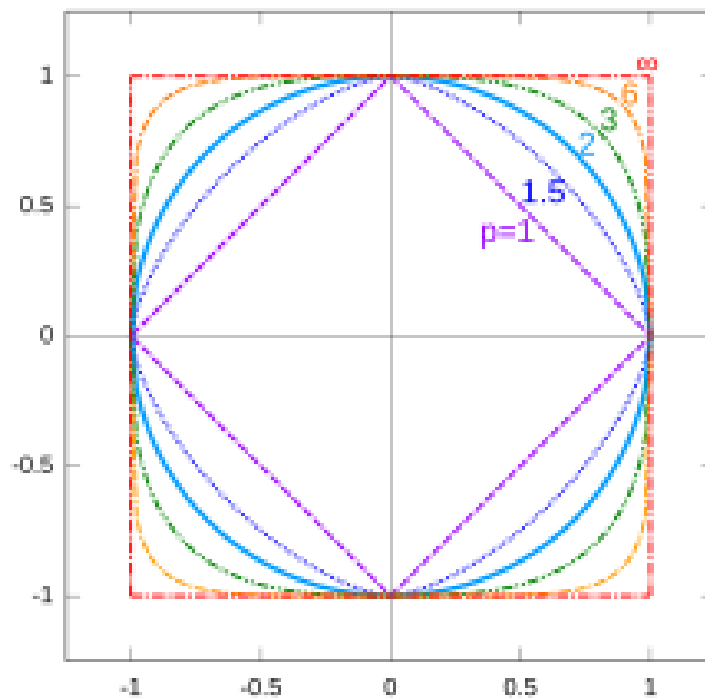
$$e = \frac{x}{\|x\|_p}$$

$$\|e\|_p = 1$$

9.3.1 Maximum Norm (L^∞)

A special case of the p-Norm as p approaches ∞:

$$\|\boldsymbol{x}\|_\infty := \max(|\boldsymbol{x_1}|, |\boldsymbol{x_2}|, ...)$$

The intuition behind this is that the largest element in the vector dominates the sum and therefore this norm effectively zooms in on the largest element.

Here is the unit circle as defined by various p-norms:

Norming a vector with respect to this norm can be useful in proofs (for example the Gershgorin disks, we can prove an inequality for the largest element and ensure every other smaller element satisfies it too).

9.4 Cauchy-Bunjakovski-Schwarz Inequality

This is often regarded as the most important inequality in mathematics, it can be written in many different forms, the 3rd one is very intuitive:

$$v^T w \leq \|v\|\|w\|$$
$$<v, w> \quad \leq \quad <v, v> \cdot <w, w>$$
$$\|v\|\|w\| \cos(\theta) \leq \|v\|\|w\|$$

Where $\|\cdot\|$ is the Euclidean norm. This can be proven by induction.

Both sides are equal $\Leftrightarrow v, w$ are linearly dependent (they point in the same direction and the angle between them is $\{0°, 180°\}$). Due to this, the angle definition of the dot product is actually an application of the inequality used throughout mathematics:

$$\cos(\theta) = \frac{<v, w>}{\|v\|\|w\|}$$

9.5 Matrix Norms

Every norm in $\mathbb{R}^n$ defines a corresponding matrix norm, which is a measure of how much that norm is affected after the matrix is applied as a linear transformation:

$$A \in \mathbb{R}^{m \times n}$$
$$x \in \mathbb{R}^n \mid \|x\|_p = 1$$
$$\|A\|_p := \max(\|Ax\|_p)$$

Here are some examples:

9.5.1 Maximum Column Sum ($\|\cdot\|_1$)

In this case $x = \begin{pmatrix} 0 \\ 0 \\ ... \\ 1 \\ ... \end{pmatrix}$ so that $\|x\|_1 = 1$.

Ax results in each column at a time. When the 1 norm is applied to each column, the sum of the column's elements is returned.

Therefore this simply returns the maximum column sum:

$$A = \begin{bmatrix} 1 & -2 & -3 \\ 2 & 3 & -1 \end{bmatrix} \implies \|A\|_1 = \max\{|1| + |2|, \ |-2| + |3|, \ |-3| + |1|\} = 5.$$

9.5.2 Maximum Row Sum ($\|\cdot\|_\infty$)

The maximum transformation allowed is when $x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ ... \end{pmatrix}$.

Ax results in the sum of each row:

$$A = \begin{bmatrix} 1 & -2 & -3 \\ 2 & 3 & -1 \end{bmatrix} \implies \|A\|_\infty = \max\{|1| + |-2| + |-3|, \ |2| + |3| + |-1|\} = 6.$$

9.5.3 Spectral Norm ($\|\cdot\|_2$)

This is a measure of the maximum factor by which a matrix increases the Euclidean norm (lengths) of vectors in space - an extremely useful result.

It can be calculated as follows:

$$\|A\|_2 = \sigma_{\max}(A) = \sqrt{\lambda_{\max}(A^H A)}$$

If A is normal:

$$\|A\|_2 = |\lambda_{\max}(A)|$$

Therefore, if A is diagonal this is simply the largest absolute diagonal value.

Furthermore, the spectral norm a matrix's inverse can be intuitively calculated as:

$$\left\|A^{-1}\right\|_2 = \frac{1}{\sigma_{\min}(A)} = \frac{1}{\sqrt{\lambda_{\min}(A^H A)}}$$

The intuition stems from the fact that the SVD of **any matrix** A can be rewritten as:

$$AV = U\Sigma$$

where Σ is a diagonal transformation, meaning it only scales space along an orthogonal basis. U and V are unitary and have no effect on lengths, therefore the largest diagonal element (singular value σ_i) of Σ is the maximum increase in the Euclidean norm.

SVD also shows us that the singular values are equal to the magnitude of the eigenvalues in the case of a normal matrix A:

$$A = U\Lambda U^T$$
$$A^T A = U\Lambda^T \Lambda U^T$$
$$\Lambda^T = \Lambda$$
$$\therefore \sigma(A) = \sqrt{\lambda(A^T A)} = \sqrt{\lambda(A^2)}$$
$$= |\lambda(A)|$$

9.5.4 Condition Number

This is a measure of how sensitive an operation is to changes in input and how accurately the inverse of a matrix can be computed:

$$k(A) = \|A\|_2 \left\|A^{-1}\right\|_2 = \frac{\sigma_{\max(A)}}{\sigma_{\min(A)}}$$

For a symmetric matrix A:

$$k(A) = \sqrt{\frac{|\lambda_{\max}(A)|}{|\lambda_{\min}(A)|}}$$

It is purely a property of a matrix based on how much bigger the maximum eigenvalue is than the minimum eigenvalue - ie how much more a matrix stretches vectors in a specific direction than the others - which makes it less "unitary" and more susceptible to rounding errors.

- $k = 1$ - The matrix is said to be well-conditioned
- $k \gg 1$ - Such a matrix is almost singular and is very prone to errors, a tiny change in the vector it is applied to has a drastic effect on the output.

- $k = \infty$ - The matrix has a singular / eigenvalue 0 and part of space is sent to the origin, meaning numerical solutions involving the matrix are very unreliable.

When a matrix comes from the result of a previous calculation, and some of its singular values are very close, but not equal to 0, it can be suspected that this slight deviations arose due to rounding errors and don't represent any meaningful values. Therefore, many Linear Algebra packages offer the option to define a **numerical rank**, which is a limit under which singular values are considered as 0 - their corresponding rank-1 approximations are considered unnecessary. This reduces storage / computational requirements, avoids introducing further rounding errors and often removes useless noise from the data.

LTD: Benchmark condition numbers of different types of matrices.

9.5.5 Entry-Based Matrix Norms

The general $L_{p,q}$ norms ignore the structure of a matrix and treat its entries as one big $m \cdot n$ vector, stemming from the **Frobenius inner product**, a finite dimensional version of the inner (dot) product on Tensor spaces (elements are matrices instead of vectors) $\|A\|_F := \sqrt{<A, A>_F}$.

The $L_{p,q}$ norms are defined as:

$$A \in \mathbb{C}^{m \times n}$$

$$\|A\|_{p,q} := \left(\sum_{j=1}^{n} \left(\sum_{i=1}^{m} |a_{ij}|^p \right)^{\frac{q}{p}} \right)^{\frac{1}{q}}$$

The **Frobenius Norm** is the special $L_{2,2}$ case:

$$\|A\|_F := \sqrt{\sum_{j=1}^{n} \sum_{i=1}^{m} |a_{ij}|^2} = \sqrt{\operatorname{trace}(A^H A)} = \sqrt{\sum_{i=1}^{\min(m,n)} \sigma_i^2(A)}$$

The Frobenius Norm in particular is also an example of a **Schatten norm**:

$$\|A\|_p := \left(\sum_{i=1}^{\min(m,n)} \sigma_i^{p(A)} \right)^{\frac{1}{p}}$$

Another example is the so-called **Nuclear norm**:

$$\|A\|_* := \sum_{i=1}^{\min(m,n)} \sigma_{i(A)}$$

10 Inner Products

In Euclidean space, the dot product is an inner product.

Sei V ein linearer Raum.
Ein *Skalarprodukt* $\langle \cdot, \cdot \rangle$ ist eine Funktion von zwei Variablen in diesem Raum V

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

welche die folgenden drei Eigenschaften erfüllt:

(S1) Die Funktion ist *linear im zweiten Argument*:

$$\langle x, ay + bz \rangle = a\langle x, y \rangle + b\langle x, z \rangle$$

für alle x, y, $z \in V$ und a, b Skalare;

(S2) Die Funktion ist *symmetrisch*:

$$\langle x, y \rangle = \langle y, x \rangle \text{ über } \mathbb{R}$$
$$\langle x, y \rangle = \overline{\langle y, x \rangle} \text{ über } \mathbb{C};$$

(S3) Die Funktion ist *positiv definit*:

$$\langle x, x \rangle \geq 0 \text{ für alle } x \in V$$
$$\langle x, x \rangle = 0 \implies x = 0;$$

falls nur der erste dieser beiden letzten Punkte erfüllt ist,
so sprechen wir von *positiv semi-definit*.

A *positiv semi-definite* operation is not an inner product.

Any operation that satisfies these properties leads to its corresponding norm being defined as:

Man sagt, dass *eine Norm* $\|\cdot\|$ *aus einem Skalarprodukt* $\langle \cdot, \cdot \rangle$ *kommt*, falls

$$\|x\| = \sqrt{\langle x, x \rangle}$$

für alle Elemente x eines linearen Raumes X.

Sei x, $y \in V$, wobei V ein linearer Raum mit einem Skalarprodukt $\langle \cdot, \cdot \rangle$ ist.
Wir sagen, dass x und y orthogonal ($x \perp y$) sind, falls $\langle x, y \rangle = 0$.

$$u \perp v \Leftrightarrow <u, v> = 0$$

10.1 Hilbert Space

Inner Product Space - A vector space with an inner product defined over its members
Complete Space - An inner product space where every Cauchy sequence with members in the space also converges towards a member of the space. This is a rigorous way of saying that the space is not missing any points.

The most obvious example is the Euclidean vector space, with the dot product defined as its inner product. However, the same can be done with other spaces, for example the inner product of the continuous differentiable functions: TODO: Find precise name for this space and explain intuitive similarity to dot product.

10.2 Projectors

Projector - A linear transformation such that repeated application has no effect:

$$P_y P_y \cdots = P_y$$

Thus, projection is idempotent (can be applied arbitrarily many times without changing after the first application).
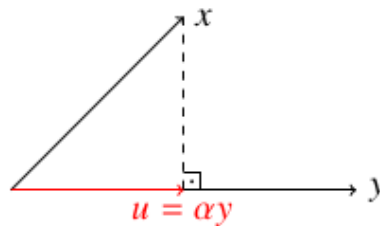
10.2.1 Orthogonal Projection

Grahically, the projection of x onto y looks like this:

Since $(x - u) \perp y$, we can derive the scalar α as:

$$\alpha = \frac{<y, x>}{<y, y>}$$

Thus, we can express the orthogonal projection of a vector onto y as the following function:

$$P_y : \mathbb{V} \to \mathbb{V}$$

$$P_y(x) := \frac{<y, x>}{<y, y>}y$$

10.2.2 Projectors

Projector - Projection represented in matrix form, for example the orthogonal projection in the previous example can also be calculated using a matrix:

$$P_y(x) = \frac{yy^H}{\|y\|^2}x$$

$$= P_y x$$

Projectors have the following properties:
- Every "successful" projection lies along the target $\text{Im}(P_y) = \text{Span}\{y\}$.
- The kernel of every projector is the span of orthogonal vectors $\text{Kern}(P_y) = \text{Span}\{x \mid x \perp y\}$; these are the eigenvectors with eigenvalue 0
- They are therefore non-invertible

10.3 Unit Vectors

In a linear space with dimensions n, n orthogonal unit vectors therefore form a basis in this space, because they are all linearly independent.

The dot product of any vector x with a unit vector is simply the component of x in the direction of the unit vector, which is very useful throughout physics when dealing with components that are not along a predefined axis:

$$x \in \mathbb{R}^n$$
$$< e, x >= \|x\|\|e\| \cos(\theta) = \|x\| \cos(\theta)$$

Furthermore, this implies that any point can be represented as the sum of its projections onto a basis $\mathbb{B}$. This is essentially what the coordinate function defined earlier does:

$$x = \sum_{e \in \mathbb{B}} P_e(x)$$
$$= \sum_{e \in \mathbb{B}} \frac{< e, x >}{< e, e >} e$$

If the basis B uses unit vectors (normal), this simplifies to:

$$= \sum_{e \in \mathbb{B}} \frac{< e, x >}{1} e = \|x\| \cos(\theta) \vec{e}$$

This can also be used to project a vector into a subspace through a smaller dimensional identity matrix looking ahh:

TODO: Lecture 14 & 15 Gilbert Strang, unsure what script means here

11 Gram-Schmidt

LTD: Inner product or dot products?

11.1 Theorem

The span of any set of linearly independent vectors can be spanned by a set of orthonormal vectors.

In other words, any basis can be converted to an orthonormal basis.

It is very computationally advantageous to use an orthonormal basis whenever possible. Many commonly used operations are simplified, for example when projecting an element onto an orthonormal basis as mentioned in the previous chapter.

11.2 Method

We would like to transform the basis $\mathbb{V} = \{v_1, v_2, ..., v_n\} \rightarrow \mathbb{U} = \{e_1, e_2, ..., e_n\}$ where $\mathbb{U}$ is orthonormal.

This is carried out in 2 steps:

1. Orthogonalize the vectors by recursively subtracting the previous n elements of the resulting basis, the first vector can remain unchanged:

$$u_1 = v_1$$
$$u_2 = v_2 - P_{u_1}(v_2)$$
$$u_3 = v_3 - P_{u_1}(v_3) - P_{u_2}(v_3)$$
$$...$$
$$u_k = v_k - \sum_{i=1}^{k-1} P_{u_i}(v_k)$$

2. The resulting vectors are then normalised (this does not affect their orthogonality):

$$e_k = \frac{u_k}{\|u_k\|}$$

The process is visualised in 3D here, showing how subtracting the projected vector results in the orthogonal "missing piece of the triangle" for each projection: https://upload.wikimedia.org/wikipedia/commons/e/ee/Gram-Schmidt_orthonormalization_process.gif

If the **0** vector is output at any stage, the input vectors are linearly dependent.

11.3 Stability

When calculating the recursive method, the resulting vectors are usually not quite orthogonal due to rounding errors, which becomes worse after each iterative subtraction of projections.

This can be alleviated by instead projecting the "orthogonaler" u_{k-1} at each stage of the iterative subtraction:

$$\mathbf{u}_k^{(1)} = \mathbf{v}_k - \text{proj}_{\mathbf{u}_1}(\mathbf{v}_k),$$
$$\mathbf{u}_k^{(2)} = \mathbf{u}_k^{(1)} - \text{proj}_{\mathbf{u}_2}\left(\mathbf{u}_k^{(1)}\right),$$
$$\vdots$$
$$\mathbf{u}_k^{(k-2)} = \mathbf{u}_k^{(k-3)} - \text{proj}_{\mathbf{u}_{k-2}}\left(\mathbf{u}_k^{(k-3)}\right),$$
$$\mathbf{u}_k^{(k-1)} = \mathbf{u}_k^{(k-2)} - \text{proj}_{\mathbf{u}_{k-1}}\left(\mathbf{u}_k^{(k-2)}\right),$$
$$\mathbf{e}_k = \frac{\mathbf{u}_k^{(k-1)}}{\left\|\mathbf{u}_k^{(k-1)}\right\|}$$

This of course results in more computation but significantly reduces rounding errors. Other orthogonalisation algorithms seen in chapter **QR decomposition** using Householder matrices / rotations can have reduced error rates.

11.4 Legendre Polynomials

The most straight-forward bases for $\mathcal{P}_n$ are the monomes, but they are not orthonormal. Using the Gram-Schmidt method, the **Legendre Polynomials** can be derived from them, allowing us to represent any polynomial through an orthonormal basis (for proof see `./exercises/legendre-polynomials.pdf`)

There are many other such sets of polynomials with special names, for example Hermite and Laguerre polynomials.

11.5 QR Decomposition

TODO: Consider moving to QR section, linear vector spaces, norms, dot products, orthogonal, gram-schmidt, qr,

The matrix A with columns $\{v_1, v_2, ..., v_n\}$ can be decomposed into matrices QR because the Gram-Schmidt is an orthogonalization process just like rotation / reflection:

1. Calculate the corresponding orthonormal vectors $\{e_1, e_2, ..., e_n\}$ using the Gram-Schmidt process
2. Each column in A can now be expressed as a linear combination of the new orthonormal basis as projections onto the unit vectors:

$$v_k = \sum_{i=1}^{k} P_{e_i}(v_k) = \sum_{i=1}^{k} <e_i, v_k> e_i$$

3. Therefore this set of linear combinations can be represented as matrix multiplication QR where:

$$A = QR$$

$$= \begin{pmatrix} e_1 & e_2 & ... & e_n \end{pmatrix} \begin{pmatrix} <e_1, v_1> & <e_1, v_2> & ... & <e_1, v_n> \\ 0 & <e_2, v_2> & ... & <e_2, v_n> \\ 0 & 0 & ... & <e_3, v_n> \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & <e_n, v_n> \end{pmatrix}$$

Advantages:
- Straightforward implementation and slightly more computationally efficient

Disadvantages:
- Very numerically unstable compared to other orthogonalization methods

12 Determinants

Definition 6.1.0.1. Multilineare Abbildung

Seien V und W zwei lineare Räume.
$f : V \to W$ ist eine lineare Abbildung falls $f(\lambda x + \mu y) = \lambda f(x) + \mu f(y)$.
$g : V \times V \to W$ ist eine bilineare Abbildung falls g linear in jedem der beiden Argumente ist.

$h : V \times V \times ... \times V \to W$ ist eine multilineare Abbildung falls h linear in jedem Argument ist.

The determinant is multilinear transformation only defined for square matrices:

$$\det : \mathbb{C}^{n \times n} \to \mathbb{C}$$

Definition 6.1.0.3. Determinante

Die Determinante ist eine Funktion

$$\det : \underbrace{\mathbb{R}^n \times \mathbb{R}^n \times ... \times \mathbb{R}^n}_{n} \to \mathbb{R}.$$

mit folgenden Eigenschaften:

(D1) $\det \mathbf{I}_n = 1$,

(D2) $\det$ wechselt das Vorzeichen, wenn zwei Zeilen oder Spalten vertauscht werden (*Antisymmetrie*).

(D3) $\det$ ist linear in jeder Zeile und Spalte:

$$\det \begin{bmatrix} ta & tb \\ c & d \end{bmatrix} = t \cdot \det \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\det \begin{bmatrix} a + \tilde{a} & b + \tilde{b} \\ c & d \end{bmatrix} = \det \begin{bmatrix} a & b \\ c & d \end{bmatrix} + \det \begin{bmatrix} \tilde{a} & \tilde{b} \\ c & d \end{bmatrix}.$$

There is no other function that fulfils these properties.

12.1 Properties

- Adding a linear combination of other rows does not affect the matrix's determinant:

$$A = LU \Rightarrow \det A = \det U$$
$$PA = LU \Rightarrow \det A = \det P \cdot \det U$$

- A zero column or row means the matrix must have $\det \cdot = 0$, it doesn't have full rank.
- The determinant of any **upper** square matrix is the product of its diagonals
- $$\det AB = \det A \cdot \det B$$
- $$\det A^T = \det A$$
$$\det A^H = \overline{\det A}$$
- $$\det A^{-1} = \frac{1}{\det A}$$
- $$\det Q = 1 \vee -1$$

  Orthogonal matrices either rotate or reflect space and are unitary. However, this is no guarantee of orthogonality.
- It is the factor by which the n-dimensional volumes (the volume spanned by n vectors) are scaled after the matrix's transformation is applied (LTD: relate to bivectors).
- A 0 determinant geometrically implies that it transforms the volume to $0 \to$ down a dimension. Information is lost and the transformation cannot be inverted.

These properties are useful for making statements about the singularity of various matrix products.

The determinant can be calculated using a variety of methods, however it is most commonly done alongside Gaussian elimination.

LTD: It can also be used to find the inverse and solve LGS: Cramer's Rule, although this isn't computationally efficient.

12.2 Gaussian Elimination

After the matrix has been eliminated to row-echelon form (upper matrix, pivots haven't been reduced to 1), the determinant is equal to the product of all the diagonal elements.

This makes intuitive sense - if the matrix doesn't have full rank (more columns than pivots) there will be a 0 on the diagonal and hence the diagonal product & determinant are also equal to 0; the matrix is not invertible.

It is perfectly fine to swap rows using a permutation matrix, the resulting determinant is simply $\det P \cdot \det A$. To avoid the need to calculate the permutation matrix's determinant: each time two rows are swapped the sign flips (multiply the determinant by $\times -1$).

The following formulae for 2×2 matrices can be derived from this method:

$$\left| \begin{pmatrix} a & b \\ c & d \end{pmatrix} \right| = ad - bc$$

Which is a useful result when performing Laplace Expansion.

12.3 Laplace Expansion

This method uses the determinant of minor matrices multiplied by signed elements of a row or column, making it useful for calculating low dimensional determinants where certain rows / columns already have many 0s.

1. Choose a row or column with as many 0 elements as possible to minimise the amount of minor determinants we need to calculate.
2. Traverse through the row, multiplying the element of the row by its minor. The signs alternate, starting with positive, and sum all of the individual results

For example when calculating the determinant of a $m \times n$ matrix by traversing the first row:

$$M = \text{Minors of A}$$

$$\det \boldsymbol{A} = \sum_{j=1}^{n} (-1)^{j+1} \cdot \boldsymbol{A_{1,j}} \cdot \det \boldsymbol{M_{1,j}}$$

13 Eigenwerte und Eigenvektoren

Eigenvectors - Vectors which a linear transformation scales onto the original line it spans.
Eigenvalues - The corresponding scalar by which a set of eigenvectors is multiplied by.
Eigenspace - The space containing all eigenvectors for a given eigenvalue - $\text{Span}\{x \mid Ax = \lambda x\}$
Spectrum - The set of eigenvalues of a matrix.

These are very important when analyzing the effects of linear transformations without a standard basis. For example one can find the axis of a rotation of an arbitrary 3D rotation (vectors with eigenvalue 1). LTD: More applications!

Some transformations, for example rotations, have no real, non-zero eigenvalues.

13.1 Properties
• The diagonal elements of an upper square matrix in reduced form are its eigenvalues; algebraic multiplicity can also be easily read from this form. LTD: Find / write proof
• $\det \boldsymbol{A} = \prod \lambda_i$ - The determinant is equal to the product of eigenvalues.
• $\text{trace } \boldsymbol{A} = \sum \lambda_i$ - Sum of diagonal elements of a matrix. This is equal to the sum of eigenvalues.
• Any vector which can be expressed as a linear combination of eigenvector(s) with the same eigenvalue is also also an eigenvector of the same eigenvalue (fundamental rules of linear transformations) - all vectors in the same line have the same eigenvalue. Furthermore, a single eigenvalue may have an $n \geq 1$ dimensional eigenspace - any vector in this space also has the same eigenvalue.

$$\boldsymbol{x} = \alpha \boldsymbol{y} + \beta \boldsymbol{b}$$
$$A(\boldsymbol{x}) = \lambda(\alpha \boldsymbol{y} + \beta \boldsymbol{b})$$
$$= \lambda \boldsymbol{x}$$

• If a matrix has an eigenvalue 0 it must be singular (there is a line of vectors that all become the same zero vector, information is lost and cannot be reversed). The eigenspace for $\lambda = 0$ is simply the null space of the original matrix.
• The inverse of a matrix has the same eigenvectors, and eigenvalues $\frac{1}{\lambda_n}$ (because it brings scaled eigenvectors back to their original length).
• The eigenvalues of repeated transformations A^k are simply powered: $(\lambda_n)^k$ because the same eigenvectors are simply repeatedly scaled.
• $\boldsymbol{A}$ and $\boldsymbol{A^2}$ do **not** necessarily have the same eigenvalues as signs are lost.
• Scalar multiplication of a matrix leads to eigenvalues multiplied by the same scalar - matrices represent linear transformations hence $\alpha \boldsymbol{A}(\boldsymbol{x}) = \boldsymbol{A}(\alpha \boldsymbol{x}) = \alpha \lambda \boldsymbol{x}$.
• Addition / multiplication of matrices does not necessarily mean the eigenvalues can be added / multiplied, algebraic rearrangement can be used to find eigenvalues of a polynomial of a matrix.
• If $\boldsymbol{A}$ is square, $\boldsymbol{A^T}$ has the same pivots and therefore the same spectrum.

- (Scaled) rotation matrices have only complex pairs of eigenvalues, total amount n.

13.2 Finding Eigenvalues / Vectors

The eigenvalues λ and eigenvectors v of a matrix A relate as follows:

$$Av = \lambda v$$

Which can be rearranged using a scalar matrix, allowing us to find the eigenvectors for a given eigenvalue:

$$Av = \lambda I v$$
$$(A - \lambda I)v = 0$$

The infinite non-zero solutions (eigenvectors) for v are only possible when $\det(A - \lambda I) = 0$. Computing this determinant results in a polynomial in terms of λ with degree n which can be solved to find all possible eigenvalues. This is called the **characteristic polynomial**.

- Polynomials can also have complex solutions, hence real matrices can have $\lambda \in \mathbb{C}$, which come in pairs with their conjugates.
- Multiplication by $\times -1$ whilst swapping rows during elimination to find the characteristic polynomial can be ignored, because it is $= 0$ anyway.
- An $n \times n$ matrix has $[1, n]$ real eigenvalues due to the fundamental theorem of algebra.
- *Algebraic Multiplicity* - This tells us how many times a root is repeated. For example the characteristic equation $(\lambda - 1)^2(\lambda - 4) = 0$ has AM 2 for $\lambda = 1$.
- *Geometric Multiplicity* - Dimensions of an eigenspace. If the eigenvectors for an eigenvalue are linearly independent, this is the same as the AM, otherwise it's less:

$$1 \leq \mathrm{GM}(\lambda) \leq \mathrm{AM}(\lambda) \leq n$$

Once we find the eigenvalues of the matrix, the matrix $A - \lambda I$ can be computed for each one and finding the eigenspaces (each set of eigenvectors) simply becomes the task of finding the different nullspaces, for which we have already calculated the reduced form of A whilst finding the characteristic equation:

$$A_{\lambda_1} v = 0$$
$$A_{\lambda_2} v = 0$$
$$\ldots$$

The dimensions of each eigenspace can be calculated thanks to the Fundamental Theorem of Linear Algebra:

$$\dim\left(\mathrm{Kernel}\left(A_{\lambda_i}\right)\right) = \mathrm{Columns} - \mathrm{Rank} = \mathrm{Number\ of\ free\ variables}$$

Where $A_{\lambda_i} = A - \lambda_i I$.

13.2.1 Solving the Characteristic Polynomial

If eigenvectors are needed as well as the eigenvalues, it is best to compute $\det(A - \lambda I)$ using Gaussian elimination (IMPORTANT: This is not equal $\det(A)$), so that the reduced form can be reused in the above eigenvector LGSs.

Handling the λs in elimination can be tricky when all other elements in that column are either 0 or in terms of λ. Sometimes a row can be multiplied / divided by a term involving λ (for example $(\lambda + 2)$) allowing it to eliminate another element in the same column. However, this often results in

characteristic polynomials too complicated to solve (in which case Laplace expansion is often better) and is more useful for inserting λ and finding the eigenspaces.

Vieta's Formulae are useful finding / checking eigenvalues, especially when some are already known / can be deduced from the constant term:

$$ax^3 + bx^2 + cx + k = 0 \Rightarrow -k \text{ is a root}$$

They are as follows:

$$-\frac{b}{a} = \sum \alpha$$
$$\frac{c}{a} = \sum \alpha\beta$$
$$-\frac{d}{a} = \sum \alpha\beta\gamma$$
$$\ldots$$

Where $\sum \alpha$ represents the sum of roots, $\sum \alpha\beta$ the sum of root pairs and so on.

13.3 Gershgorin Disk Theorem

Polynomials of degree $n \geq 5$ have no definite formula for solutions, eigenvalues of matrices with more than 4 columns must be found using numerical methods, expensive and numerically unstable operations.

A matrix norm (measure of how much a transformation affects space) is by definition the maximum scalar by which vector lengths are multiplied. Therefore:

$$|\lambda| \leq \|A\|$$

The Gershgorin Disk Theorem defines an easy-to-compute area in which all eigenvalues can be found:

Definition 7.2.0.11. Gerschgorin Kreis

Sei A eine $n \times n$-Matrix. Für jedes $1 \leq i \leq n$ definieren wir den Gerschgorin Kreis

$$D_i = \{|z - a_{ii} \leq r_i, z \in \mathbf{C}\}, \qquad \text{wobei } r_j = |a_{i1}| + \ldots + |a_{ij-1}| + |a_{ij+1}| + \ldots + |a_{in}|.$$

Das Gerschgorin Gebiet ist die Vereinigung der Gerschgorin Kreise einer Matrix:

$$D_A = \bigcup_{i=1}^{n} D_i \subset \mathbf{C}.$$

Wir sehen, der i-te Gerschgorin Kreis hat das Zentrum in dem i-ten Eintrag a_{ii} der Diagonale von A und Radius r_i gleich mit der Summe der Beträge der i-te Zeile. Das Gerschgorin Gebiet ist nützlich, um die Lage der Eigenwerte einer Matrix zu bestimmen, ohne die Eigenwerte zu berechnen:

Satz 7.2.0.12. Satz von Gerschgorin

Alle Eigenwerte einer Matrix A liegen im Gerschgorin Gebiet $D_A \subset \mathbf{C}$.

In simple terms, all eigenvalues of a matrix can be found in the circles with centers at the diagonal value, such that the radius is less than the absolute sum of all other terms on the row.

This allows us to narrow down the range of values checked in numerical methods, vastly reducing computation.

Each Gershgorin circle of the matrix has a radius smaller than its center position, hence 0 is never an eigenvalue and it must be regular.

14 Diagonal Matrices

Diagonal Matrix - A matrix with 0 entries everywhere except the diagonal.

They are very useful in applications as multiplying by a diagonal matrix only affects the same row:

$$\begin{pmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} a_1 x_1 \\ a_2 x_2 \\ a_2 x_3 \end{pmatrix}$$

which is for example useful when solving linear differential equations. It is also extremely computationally efficient

They are square most of a time, but a non-square diagonal matrix simply takes the form:

$$\begin{pmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \\ 0 & 0 & 0 \end{pmatrix}$$

$\text{diag}(a_1, a_2, \ldots)$ notation can be used to specify a diagonal matrix with diagonal entries in that order.

The inverse of a diagonal matrix D^{-1} can be computed if there are no 0s (full-rank) on the diagonal and the matrix is square. It simply involves replacing diagonal elements with their reciprocal $\frac{1}{x}$.

Scalar Matrix - Diagonal matrix with the same diagonal entries, multiplying by it has the same effect as scalar multiplication.

14.1 Similar Matrices

Two $n \times n$ matrices A and B are similar if there exists an invertible matrix P such that:

$$A = P^{-1}BP$$

Transformations in this form are ubiquitous, for example a transformation which is awkward to calculate in the canonical basis can be written using a change of basis matrix P, the easier to compute equivalent transformation B in the new basis, followed by a return to the original basis P^{-1}.

Intuitively the matrices A and B represent the same transformation in different bases, hence the name **similar**.

As a result of this, similar matrices have **the same eigenvalues** which all have **the same algebraic multiplicity**.

14.2 Diagonalization

A matrix which is similar to a diagonal matrix D can be called **diagonalizable**:

$$A = P^{-1}DP$$

An $n \times n$ matrix A with n linearly independent eigenvectors can be diagonalized by setting those eigenvectors $\{s_1, s_2, ..., s_n\}$ as columns of a matrix S:

$$S = (s_1 \;\; s_2 \;\; ... \;\; s_n)$$

$$S^{-1}AS = \Lambda = \begin{pmatrix} \lambda_1 & 0 & ... & 0 \\ 0 & \lambda_2 & ... & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & ... & \lambda_n \end{pmatrix}$$

$$A = S\Lambda S^{-1}$$

Therefore a similar diagonal matrix can simply be formed using its eigenvalues in any order (as long as S is consistent with this order):

$$D = \mathrm{diag}(\lambda_1, \lambda_2, ...)$$

Those eigenvalues do not need to be unique (although unique eigenvalues guarantee linear independence and diagonalization) - some may have geometric multiplicity > 1. However, the chosen eigenvectors must be independent to ensure S is invertible and represents a basis one can switch to.

When visualising this decomposition in the order $S^{-1}AS = D$, the change of basis matrix S changes the canonical basis to an eigenvector basis. A is then applied, which of course scales the eigenvector basis by λ_i (every vector is also scaled accordingly, as they are simply combinations of the eigenvector basis), after which S^{-1} rotates / reflects space back to the original canonical basis. The basis vectors have been scaled by the eigenvalues, such that the original canonical unit vectors now have length λ_i, forming the diagonal Λ matrix.

Calculating a diagonal form of the matrix (albeit in another basis) can be extremely useful for simplifying calculations, for example calculations where raw access to its eigenvalues is useful, or analyzing the properties of the transformation it represents.

> **Satz 7.2.0.31. Verhältnis zwischen GM und AM entscheidet ob diagonalisierbar oder nicht**
>
> *Wenn die Matrix* **A** *ein Eigenwert besitzt, dessen geometrische Multiplizität strikt kleiner als die algebraische Multiplizität ist, dann ist die Matrix nicht diagonalisierbar.*
> *Wenn für alle Eigenwerte die geometrische Multiplizität gleich der algebraischen Multiplizität ist, dann ist die Matrix diagonalisierbar.*

To be able to chose n independent eigenvectors, $\sum \mathrm{GM} = n$. The fundamental theorem of algebra states $n = \sum \mathrm{AM}$, hence if any eigenvalue's GM < AM it cannot be diagonalized - such eigenvalues are called **defective**.

15 Normal Matrices

Identities in terms of the Hermetian A^H of a matrix will also be assumed to be valid for the real tranpose unless stated otherwise.

$$A \text{ is normal} \Leftrightarrow A^H A = A A^H$$

A useful property of **only** normal matrices is that n eigenvectors **can always be chosen to be orthogonal** to each other:
- If there are n distinct eigenvalues, these eigenspaces are **orthogonal** to one another

- If there are eigenvalues with geometric multiplicity > 1 this is not an issue. For example, all vectors are eigenvectors of the identity matrix's only eigenvalue $\lambda = 1$, and we can choose any n dimensional orthogonal basis without issues to satisfy this statement.

$$\boldsymbol{A} \text{ is normal} \Leftrightarrow \boldsymbol{A} \text{ has n orthogonal eigenvalues} \Rightarrow \boldsymbol{A} \text{ is diagonalizable!}$$

However, the only condition for diagonalizability is **linearly independent** eigenvectors (they do not need to be perfectly orthogonal), hence not all diagonalizable matrices are normal.

Therefore, they are diagonalisable through an ortho(normal) change of basis matrix - the chosen eigenvectors are not just linearly independent but also orthogonal to each other:

$$\boldsymbol{A} = \boldsymbol{Q\Lambda Q^{-1}} = \boldsymbol{Q\Lambda Q^T}$$

Dividing each column by its norm can make the orthogonal matrices orthonormal.

This is an incredibly useful statement for computation - computers love orthonormal and diagonal matrices!

This is be generalised with unitary matrices (entries can be complex) as the so called **Spectral Theorem**:

$$\boldsymbol{A} \text{ is normal} \Rightarrow \boldsymbol{A} = \boldsymbol{U\Lambda U^H}$$

The inverse does **not** always apply - not all diagonalizable matrices are normal!

LTD: Reason for name, investigate optics applications

15.1 Symmetric Matrices
A square matrix may be symmetric across its diagonal:
- *Symmetric* - $\boldsymbol{A^T} = \boldsymbol{A}$
- *Antisymmetric* - $\boldsymbol{A^T} = -\boldsymbol{A}$
- *Hermetian Symmetric* - $\boldsymbol{A^H} = \boldsymbol{A}$

Hence, symmetric matrices are a subset of the normal matrices:

$$\boldsymbol{A^H A} = \boldsymbol{AA} = \boldsymbol{AA^H}$$

15.2 Properties
- The sum of two symmetric matrices is also symmetric
- A^n is also symmetric
- A is symmetric $\Leftrightarrow A^{-1}$ is symmetric
- Since they are normal, eigenvectors can always be chosen to be orthogonal.
- Additionally, **eigenvalues** of a symmetric (including Hermetian symmetric!) matrix are **guaranteed to be real**!
- The number of positive pivots is equal to the number of positive eigenvalues (and vice versa for negatives). This is of similar use as Gershgorin's Circle theorem for estimating eigenvalues of matrices with high order characteristic polynomials.
- All eigenvalues of an antisymmetric matrix are purely imaginary $\lambda = bi \mid b \in \mathbb{R}$, despite this their eigenvectors can still be chosen as orthogonal.
- $< x, Ay >= x^H Ay = x^H A^H y =< Ax, y >$ is valid for a symmetric matrix $A \in \mathbb{R}^{n \times n} \cup \mathbb{C}^{n \times n}$

15.3 Symmetric Positive-Definite Matrices
These are symmetric matrices which have **only strictly-positive** (non-zero) eigenvalues and therefore only strictly positive pivots (and therefore full-rank; **invertible**).

Therefore the determinant (product of eigenvalues / pivots), as well as all determinants of "sub" matrices with $n - k$ dimensions are also positive (we can't only rely on the "total" determinant, it may have two negative eigenvalues).

Another way of checking this is ensuring the dot product of all vectors before and after the transformation is ≥ 0:

$$\forall x \in \mathbb{R}^n \mid < x, Ax >= x^T Ax \geq 0$$
$$< x, Ax >= 0 \Rightarrow x = 0$$

If the zero vector checking condition is not satisfied, the matrix is called **semi positive-definite** - in other words some non-zero vectors are either sent to the zero vector or become orthogonal. It must have at least one 0 eigenvalue.

16 Non-Diagonalizable Decomposition

Regular diagonalization only works for square matrices with n independent eigenvectors. However, there are some slightly less convenient methods which are able to almost diagonalize rectangular / defective matrices.

16.1 Schur Decomposition

All complex **square** matrices (not only normal) are unitarily similar (unitary change of basis matrices) to an upper matrix (may contain complex elements) with its eigenvalues as its pivots.

$$A = QUQ^{-1} = QUQ^H$$

If A is real, it is orthogonally similar to a *block upper matrix*, such that when represented as a block matrix using 1×1 and 2×2 sub matrices, all entries below the diagonal are zero.

The existence of this decomposition is more relevant than using it and it as a useful ingredient in many fundamental proofs.

16.1.1 QU Algorithm

This is an algorithm based on QU decomposition for finding a similar matrix using unitary change of basis vectors, ie computing the Schur decomposition:
1. Compute the QU decomposition: $A = QU$
2. Let $A_{k+1} = UQ$
3. $A_{k+1} = Q^{-1}QUQ = Q^H AQ$
4. Keep iterating until A_{k+1} converges to an upper matrix, the Schur decomposition is then solved and can be rearranged to $A = QA_{k+1}Q^H$

16.2 Jordan Form

This is based on the same principle as regular diagonalisation, with the difference that the eigenvectors which would be columns of S are not required to be linearly independent, hence S would not be guaranteed invertible. The set of eigenvectors are extended through linearly independent **generalized eigenvectors**, allowing them to be always be used as a change of basis matrix.

An almost diagonal, similar *Jordan matrix* can be found for square non-diagonalizable matrices, which is composed of so-called *Jordan Blocks*:

$$J_{\lambda, n} \in \mathbb{R}^{n \times n}$$

$$J_{\lambda, n} = \begin{pmatrix} \lambda & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda \end{pmatrix}$$

which consist of an eigenvector of the original matrix on the diagonals, 1s directly above each diagonal element (superdiagonal) and 0s everywhere else.

- They can also be formed with 1s under the diagonal (subdiagonal), in which case all of the following information still applies.
- These blocks can also have dimensions 1×1, meaning that that particular eigenvalue has the same AM and GM (and its block appears AM / GM times).

Consider the **square** matrix $A \in \mathbb{R}^{n \times n}$, which has eigenvalues $\lambda_1, \lambda_2, ..., \lambda_k$. There is an invertible matrix S such that:

$$A = S \, \text{diag}\left(J_{\lambda_1, \, \text{AM}(\lambda_1)}, J_{\lambda_2, \, \text{AM}(\lambda_2)}, ..., J_{\lambda_k, \, \text{AM}(\lambda_k)}\right) S^{-1}$$

Here is an example of a Jordan Matrix, where eigenvalue λ_1 has AM 3:

$$\begin{pmatrix} \boxed{\begin{matrix} \lambda_1 & 1 & \\ & \lambda_1 & 1 \\ & & \lambda_1 \end{matrix}} & & & & \\ & \boxed{\begin{matrix} \lambda_2 & 1 \\ & \lambda_2 \end{matrix}} & & & \\ & & \boxed{\lambda_3} & & \\ & & & \ddots & \\ & & & & \boxed{\begin{matrix} \lambda_n & 1 \\ & \lambda_n \end{matrix}} \end{pmatrix}$$

The decomposition is not unique but there is only one possible S for a specific order of Jordan Blocks along the diagonal.

We can read the following properties about A by looking at its corresponding Jordan Matrix:
- All of its eigenvalues are along the diagonal
- The total number of times an eigenvalue appears on the diagonal indicates its algebraic multiplicity (sum of dimensions of all blocks belonging to that eigenvalue).
- The number of discrete Jordan blocks for a specific eigenvalue indicates its geometric multiplicity.

16.2.1 Finding S

Generalized Eigenvectors - These are a sequence of linearly independent vectors for an eigenvalue λ generated by starting with a real eigenvector v_1 and calculated iteratively using:

$$(A - \lambda I)v_{n+1} = v_n$$

Where v_{n+1} is the next generalized eigenvector and can be found using Gaussian elimination.

1. Find the characteristic polynomial of A and its eigenvalues + their AMs. Calculate the GM of each eigenvalue by calculating the nullity of $A - \lambda I$ (n - rank)
2. Construct the set of Jordan Blocks accordingly and place them along the diagonal to form the Jordan Matrix.

3. For each column with **no super- / subdiagonal 1** in the Jordan Matrix, we can simply set a linearly independent eigenvector in the corresponding column of S and move on.
4. For each column **with a super- / subdiagonal 1**, the eigenspace lacks enough linearly independent vectors and we have to create a new generalized eigenvector based on the previous column eigenvector / generalized eigenvector in the block.
5. Once S is complete, calculate its inverse using Gauss-Jordan elimination and assemble the final decomposition: $A = SAS^{-1}$

16.3 Singular Value Decomposition

Singular value decomposition (SVD) allows us to "diagonalize" **all** matrices using unitary change of basis matrices, with the downside that the "change of basis" matrices are different and therefore need to be calculated twice:

$$A \in \mathbb{R}^{m \times n}, U \in \mathbb{R}^{m \times m}, V \in \mathbb{R}^{n \times n}$$

$$\Sigma \in \mathbb{R}^{m \times n} = \text{diag}(\sigma_1, \sigma_2, ..., \sigma_p), p = \min(m, n), \sigma_1 \geq \sigma_2 \geq ... \geq \sigma_p \geq 0$$

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

There are multiple (but finite) possible decompositions for a given matrix. This depends on order of singular values on the diagonal and their underlying geometric multiplicy.

This can be rearranged to:

$$AV = U\Sigma$$
$$\therefore AVx = U\Sigma x$$

Which leads to a **beautiful** geometric interpretation:

Any matrix A can be modified to behave as a diagonal matrix Σ (only scales along orthogonal basis vectors, no shearing), followed by a rotation / reflection U, by rotating space before applying the transformation A (change the basis) with the unitary matrix V.
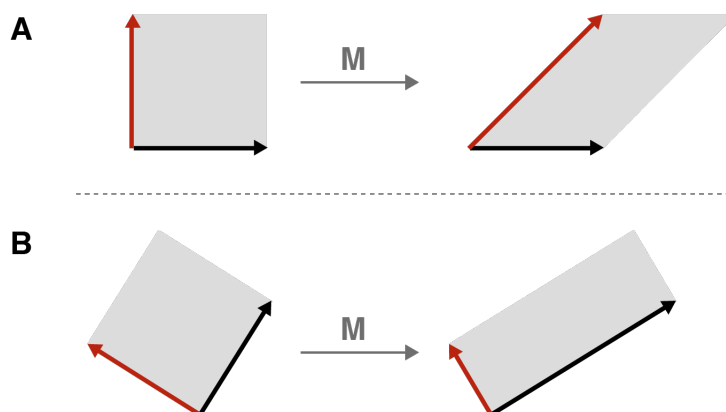

Alternatively, the decomposition can be written as a weighted sum of rank-1 representation of the matrix, which is useful for data compression:

$$A \in \mathbb{R}^{m \times n}, u_i \in \mathbb{R}^{m \times 1}, v_i \in \mathbb{R}^{n \times 1}$$

$$A = \sum_{i=1}^{r} \sigma_i u_i (v_i)^H$$

Where $r = \text{rank}(A)$ (the number of non-zero singular values), u_i is the corresponding column of U and v_i is the corresponding column of the matrix V (**pre-transpose**, or alternatively the i-th row) such that $v^T_{\ i}$ has dimensions $1 \times n$.

Here are some examples of the dimensions involved in SVD:
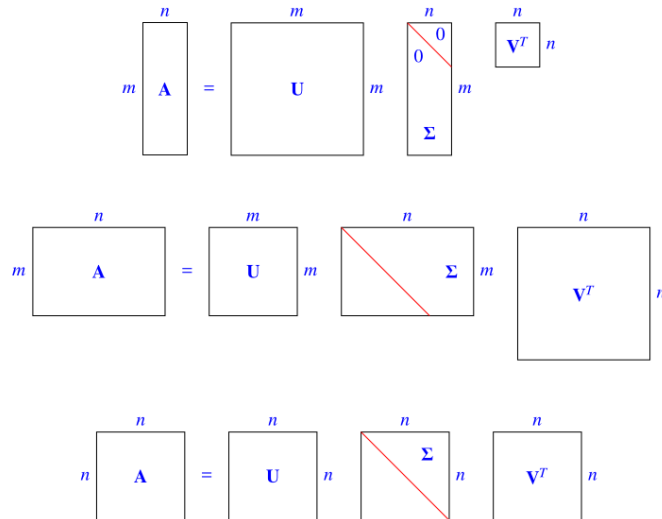
16.3.1 Properties

$$A \in \mathbb{R}^{m \times n}$$

The SVD has strong connections to Strang's Fundamental Theorem of Linear Algebra:
- $AV = U\Sigma$ shows that the left singular vectors (columns of U) are in the column space (image) of A
- Rearranging to $A^H U = V\Sigma$ shows that the right singular vectors (columns of V) are in the row space of A
- Let $r := \text{Rank}(A)$. To span the entire row / column space, V and U only need r vectors but have dimensions $n \times n$ and $m \times m$ respectively. Therefore the remaining $n - r$ **right** singular vectors are a basis for $\text{Kernel}(A)$.
- The remaining $m - r$ **left** singular vectors are a basis for $\text{Kernel}(A^H)$. The orthogonality properties discussed in Section 8.9 apply to these spaces as usual.
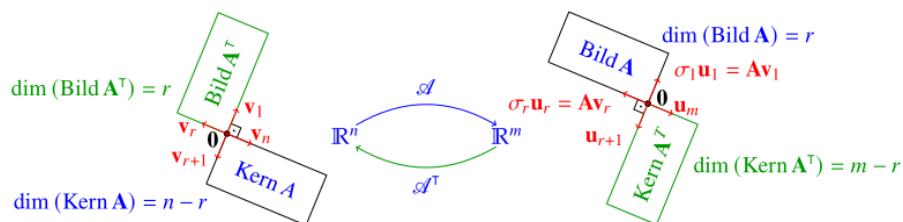
16.3.2 Calculating SVD

SVD is based on the fact that **any** matrix multiplied with its transpose results in one of 2 square, **symmetric** matrices with the **same eigenvalues** and which can be diagonalised!

U and V can be isolated as follows:

$$A^H A = (U\Sigma V^H)^H (U\Sigma V^H) = V\Sigma^H U^H U\Sigma V^H$$
$$= V\Sigma^H \Sigma V^H = V\Sigma^2 V^H$$
$$AA^H = (U\Sigma V^H)(U\Sigma V^H)^H = U\Sigma V^H V\Sigma^H U^H$$
$$= U\Sigma\Sigma^H U^H = U\Sigma^2 U^H$$

Showing us that U and V are simply orthonormal eigenvectors of multiplications with the transpose (the so-called **singular vectors**). These identities can be used to check which singular vectors belong to U and V if this is not obvious from the dimensions.

Singular values - σ_i - **Positive** square roots of $\lambda(A^H A) = \lambda(AA^H)$, usually listed in descending order. Recall that the maximum singular value is the spectral norm $\|.\|_2$ of a matrix.
- The rank of A is equal to the number of **non-zero** singular values because $\text{Rank}(A) = \text{Rank}(\Sigma)$, unitary transformations do not affect rank
- If a matrix is normal, $\sigma = |\lambda|$:

$$A = U\Lambda U^H$$
$$A^H A = (U\Lambda^H \Lambda U^H)$$
$$\Lambda^H = \Lambda$$
$$\therefore \sigma(A) = \sqrt{\lambda(A^H A)}$$
$$= \sqrt{\lambda(A^2)} = \sqrt{\lambda(SD^2 S^{-1})} = |\lambda(D)|$$
$$= |\lambda(A)|$$

Hence to calculate the SVD of a matrix A:
1. Calculate $A^H A$ and AA^H
2. Find the eigenvalues of the simpler product (they are the same for both) and use them to find both sets of orthogonal eigenvectors (singular vectors)
3. Arrange them in order of descending singular value as the columns of U and V respectively ($AA^H \to U, A$ **and** U **come before** and $A^H A \to V, A$ **and** V **come after**)
4. Norm the columns of U and V if needed so their $\|.\|_2 = 1$
5. Σ has the singular values in descending order along its diagonal
6. It is often quicker to find U using $U = \Sigma^{-1} AV$ if Σ is invertible (strictly positive singular values).

17 Applications
Finally, we can enjoy the wide variety of real world uses for linear algebra!

17.1 Linear Regression
This is one the fundamental algorithms in machine learning used for finding the best (this can have different interpretations) linear relationship between n independent variables and a dependent variable(s). In other words, the dependent variable(s) is / are assumed to be a linear combination of the inputs and we are trying to determine the best linear combination for a set of sample points.

Most real-world applications are expected to depend on not only one but several quantifiable inputs, for example the likelihood of getting cancer depending on several lifestyle choices.

This most often uses the **least squares** approach for reducing error, which is carried out using techniques in linear algebra

Non-linear relationships can also be found using the multivariable minima through means of calculus, maybe a good LTD topic...

LTD: Generalise for multiple dependent variables (https://en.wikipedia.org/wiki/General_linear_model) essentially performing multiple linear regressions simultaneously

17.1.1 Linear Least Squares

This is a method for finding the best approximation of an overdetermined system of equations:

$$Xa \approx y$$
$$Xa - y \to 0$$
$$\therefore \min \|Xa - y\|_2$$

Applying it to the context of linear regression, we suspect that there is a linear relationship between n independent variables $x \in \mathbb{R}^n$ and the dependent scalar $y \in \mathbb{R}$ such that:

$$m \in \mathbb{R}^n, c \in \mathbb{R}$$
$$m^T x + c = y$$

This linear combination + constant can be written in an easily-extendible matrix form as the proposed answer vector $a \in \mathbb{R}^{n+1}$ and input matrix $X \in \mathbb{R}^{1 \times (n+1)}$:

$$a = \begin{pmatrix} c \\ m \\ ... \end{pmatrix}$$
$$X = \begin{pmatrix} 1 & x & ... \end{pmatrix}$$

Such that:

$$Xa = c \cdot 1 + <x, m> = y$$

To create a measure of error for an answer vector a, we can represent the **residual** for a proposed answer vector a and an actual input (inside X) and output y_1:

$$Xa - y_1 = r$$

Let us consider we have s samples available to optimise our answer vector against. We can extend X with each sample $\{x_1, x_2, ..., x_s\}$ and subtract the corresponding sample outputs $y = \begin{pmatrix} y_1 \\ y_2 \\ ... \\ y_s \end{pmatrix}$ to calculate the set of residuals $r \in \mathbb{R}^s$:

$$X = \begin{pmatrix} 1 & x_1 & ... \\ 1 & x_2 & ... \\ ... & ... & ... \\ 1 & x_s & ... \end{pmatrix}$$
$$Xa - y = r$$

We can now convert this set of residuals into an unsigned scalar representing how "bad" the current answer vector a is using a norm, usually the Euclidean L_2 norm:

$$r = \|r\|$$

Our goal is to find the answer vector a to minimise:

$$\|Xa - y\|$$

for a given sample set encoded as X and y.

Of course in the vast majority of cases there is no perfect solution such that:

$$r = \|\boldsymbol{Xa} - \boldsymbol{y}\| = 0$$
$$\boldsymbol{Xa} = \boldsymbol{y}$$

In other words, $\boldsymbol{y}$ is probably not in $\text{Im}\boldsymbol{X}$. However, there are several techniques available for approximating the closest answer vector $\boldsymbol{a} \in \text{Im}\boldsymbol{X}$.

17.1.1.1 Orthogonal Projection Method

The current goal is to find the best answer $\boldsymbol{a}$ such that $\boldsymbol{Xa} = \boldsymbol{y}$. We can express $\vec{y}$ as the sum of the closest vector in the image $\boldsymbol{p} \in \text{Im}\boldsymbol{X}$ and another vector $\boldsymbol{c} \in \text{Kern } \boldsymbol{A}^T$ which we know are orthogonal to one another thanks to the Fundamental Theorem of Linear Algebra:

$$\boldsymbol{Xa} = \boldsymbol{y} = \boldsymbol{p} + \boldsymbol{c}$$
$$\boldsymbol{p} \perp \boldsymbol{c}$$

Their orthogonality means that $\boldsymbol{c}$ is the shortest possible error from the real $\vec{y}$ possible. We can find $\boldsymbol{p}$ by simply projecting $\boldsymbol{y}$ onto the image subspace of $\boldsymbol{X}$.

Thus we can now find a solution for answer vector $\boldsymbol{a}$, because $\boldsymbol{p}$ is guaranteed to be in the image of the matrix $\boldsymbol{X}$:

$$\boldsymbol{Xa} = \boldsymbol{p}$$

This is the optimal solution to our regression problem :)

To summarise the entire process, we can find the optimal solution for $\boldsymbol{a}$ using the following formula (derived based on the discussed method), $\boldsymbol{X}$ and the corresponding $\boldsymbol{y}$ samples are given:

$$\boldsymbol{X}^T \boldsymbol{Xa} = \boldsymbol{X}^T \boldsymbol{y}$$

However, this is still not guaranteed to have a solution, namely when $\boldsymbol{X}$ doesn't have full rank (can happen quite often for perfectly reasonable samples). Furthermore, it is very sensitive to rounding errors.

17.1.1.2 QR-Decomposition Method

This method solves the rounding error issues, although it **still only works if $\boldsymbol{X}$ has full rank** (the upper reduced $\boldsymbol{R}$ matrix has the same rank as $\boldsymbol{X}$ and needs to be invertible to deliver a single solution). Wasteful data cleaning would have to be done to ensure these methods are fit for use, therefore why singular value decomposition is used much more commonly in practice.

We can rewrite $\boldsymbol{Xa} = \boldsymbol{y}$ using the QR decomposition of $\boldsymbol{X}$:

$$\boldsymbol{X} = \boldsymbol{QR}$$
$$\boldsymbol{QRa} = \boldsymbol{y}$$
$$\boldsymbol{Ra} = \boldsymbol{Q}^H \boldsymbol{y}$$

Assuming the matrix $\boldsymbol{X}$ has rank n, only the first n rows contain are non-0 in the $\boldsymbol{R}$ matrix, and therefore only the first n rows of the RHS matter when solving for $\boldsymbol{a}$. In block form this looks something like this:

$$\begin{pmatrix} \boldsymbol{R}\,|_n \\ 0 \end{pmatrix} \boldsymbol{a} = \begin{pmatrix} (\boldsymbol{Q}^H \boldsymbol{y})\,|_n \\ r_{n+1} \\ \dots \\ r_s \end{pmatrix}$$

Where $\cdot|_n$ represents the first n rows of a matrix / vector.

We can now solve $R|_n \, a = (Q^H y)|_n$ The rest of the elements of $Q^H y$, named $r_{n+1}, ..., r_s$, represent the residuals of the optimal solution - their norm shows exactly what the lowest possible error is for the sample. TODO: Apply method and check what the residuals look like in practice

LTD: The same technique can be used to find a polynomial of degree n solution TODO: Bespiel 5.1.0.4, think about how this corresponds to curved surfaces in 3D space.

17.1.1.3 SVD / The Pseudoinverse

This powerful method works regarding the rank of the matrix X.

$$X = U\Sigma V^H$$
$$U\Sigma V^H a = y$$
$$a = V\Sigma^+ U^H y$$
$$X^+ := V\Sigma^+ U^H$$

Where X^+, Σ^+ denote the so-called **Pseudoinverses**. Professor Gradinaru uses dagger notation $X^\dagger$.

The Pseudoinverse of a diagonal matrix, for example Σ^+, is simply the reciprocal of non-zero values along the diagonal followed by tranposing the matrix (to satisfy SVD dimensions).

Intuitively, 0 singular values represent the least important rank 1 approximations of the matrix hence they can be "ignored" whilst taking the inverse of the matrix's SVD, creating the optimal least squares solution inverse.

A **unique** (no matter the order of the singular values) Pseudoinverse exists for **every matrix** and is calculated using the SVD $A^+ := V\Sigma^+ U^H$ as seen above and exhibits the following properties:
- If A is invertible $A^{-1} = A^+$
- $(A^+)^+ = A$
- $AA^+ A = A$
- $(A^+ A)^T = A^+ A$
- $(AA^+)^T = AA^+$

17.2 Optimization

17.2.1 Quadratic Forms

A quadratic form is a multivariable polynomial who's terms are all degree 2, for example:

$$f(x_1, x_2) = a(x_1)^2 + 2bx_1 x_2 + c(x_2)^2$$

We can represent such equations using the following operation with a symmetric matrix:

$$A = \begin{pmatrix} a & b \\ b & c \end{pmatrix}, x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$
$$x^T A x = f(x_1, x_2)$$

If A is positive-definite, it satisfies the properties of an inner product:

$$\forall x_1, x_2 \in \mathbb{R}, f(x_1, x_2) \geq 0$$
$$f(0, 0) = 0$$

This shows us that $f(x_1, x_2)$ has a global minimum at $(0, 0)$. This can be generalized by introducing translational parameters to the quadratic form, allowing us to find the minimum of an n-dimensional quadratic form.

17.2.2 Generic Multivariable Functions

Furthermore, linear algebra allows us to define easy to calculate criteria for checking if a point is a local minimum / maximum of a multivariable function.

Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable within a specific domain of our interest. We can check if a point is critical if all of the partial derivatives are equal to 0:

$$\begin{pmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} \\ ... \\ \frac{\partial f(\boldsymbol{x})}{\partial x_n} \end{pmatrix} = \boldsymbol{0}$$

To check if it's a local minima, the second partial derivatives must all be greater than 0. Modelling them as a matrix:

$$\begin{pmatrix} \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_1 \partial x_j} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_j} \end{pmatrix} \geq \boldsymbol{0}$$

This matrix's and the first derivative vector's entries can be calculated using numerical methods. The vector check is straightforward; we can check if the 2nd derivative condition is satisfied by rearranging it as an inner product $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}$, performing Gaussian elimination on $\boldsymbol{A}$ and ensuring it is **semi positive-definite**.
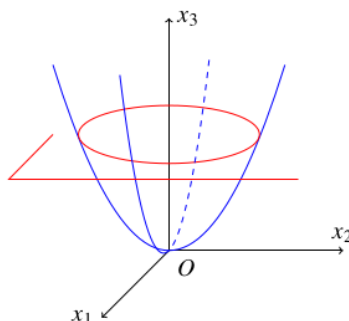
LTD: Define similar criteria for multi-variate (n dimensional output) functions

17.3 Contour Lines

A contour line and the corresponding level set (set of points for which a function gives the same output) can be determined with the help of linear algebra:

$$L_{c(f)} = \{\boldsymbol{x} \mid f(\boldsymbol{x}) = c\}$$

17.3.1 Quadratic Form

As seen in the optimization examples, a quadratic form can be expressed using a symmetric positive-definite matrix $\boldsymbol{A}$:

$$f(x_1, x_2) = \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}$$

As A is symmetric and therefore normal, the contour line at $f(x_1, x_2) = c$ can be diagonalized (thanks to the spectral theorem):

$$c = f(x_1, x_2) = x^T A x = x^T Q D Q^{-1} x$$
$$= x^T Q D Q^T x$$

The diagonalization now allow us to make better sense of the contour line in a helpful basis, which in the case of a quadratic form is an n-dimensional ellipse. Let $\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$ be axes in our new basis:

$$y = Q^T x$$
$$c = y^T D y$$
$$= (y_1 \ y_2) \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$
$$= \lambda_1 (y_1)^2 + \lambda_2 (y_2)^2$$

Therefore the radii of the n-dimensional ellipse (contour line) are $\frac{c}{\sqrt{\lambda_n}}$.

17.4 Linear Differential Equations

Extremely useful in physics! I will just cover the ways linear algebra can be used to solve them, see analysis notes for more background.

17.4.1 First-order

TODO: https://tutorial.math.lamar.edu/Classes/DE/Linear.aspx , 7.1 The solution satisfies an equation in the form:

$$\dot{y}(t) + a()y(t) =$$

17.5 Markov Matrices

TODO: Method for finding steady state of probability based change using eigenvectors and special form of matrix with columns adding to 1 (all possibilities covered) and elements >0 (no negative probabilities). Name city movement example / maybe find some physics related one.

17.6 Discrete Fourier Transform and FFT

The change-of-basis nature of the discrete Fourier transform makes a matrix implementation the obvious choice and allows us to recognize the optimized FFT variant of the algorithm. See `../../ semester-2/mathematische-methoden/notes.pdf` for explanation of how the formula works:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi \frac{k}{N} n}$$

Multiplying a sample vector $S \in \mathbb{C}^{N \times 1}$ by the following **Fourier matrix** performs the complete analysis:

$$F_{k, n \in \mathbb{Z}_{[0, N-1]}} = \frac{e^{-i2\pi \frac{k}{N} n}}{\sqrt{N}}$$

Here is an example of a DFT matrix for 8 samples:

$$W = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \frac{1-i}{\sqrt{2}} & -i & \frac{-1-i}{\sqrt{2}} & -1 & \frac{-1+i}{\sqrt{2}} & i & \frac{1+i}{\sqrt{2}} \\ 1 & -i & -1 & i & 1 & -i & -1 & i \\ 1 & \frac{-1-i}{\sqrt{2}} & i & \frac{1-i}{\sqrt{2}} & -1 & \frac{1+i}{\sqrt{2}} & -i & \frac{-1+i}{\sqrt{2}} \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & \frac{-1+i}{\sqrt{2}} & -i & \frac{1+i}{\sqrt{2}} & -1 & \frac{1-i}{\sqrt{2}} & i & \frac{-1-i}{\sqrt{2}} \\ 1 & i & -1 & -i & 1 & i & -1 & -i \\ 1 & \frac{1+i}{\sqrt{2}} & i & \frac{-1+i}{\sqrt{2}} & -1 & \frac{-1-i}{\sqrt{2}} & -i & \frac{1-i}{\sqrt{2}} \end{bmatrix}$$

The resulting vector $FS = T \in \mathbb{C}^{N \times 1}$ contains the complex number representing the amplitude and phase shift each bin frequency.

17.6.1 Properties

- The Fourier matrix is unitary. The scalar multiplication $\frac{1}{\sqrt{N}}$ makes each column have norm 1 (do not forget complex conjugation in the complex inner product $\|c\| = \sqrt{c^H c}$) and all columns are orthogonal to each other $< a, b >= 0$.
- The inverse discrete Fourier transform can be performed by taking the inverse of the matrix $F^{-1} = F^H = \overline{F}$, simply the complex conjugate of each element as it is also symmetric.

17.6.2 Cooley-Tukey FFT

This famous algorithm is based on maximizing the number of 0s (skipped operations) in the Fourier matrix thanks to a set of permutations and recognizing the symmetry of roots of unity. It reduces the number of operations from N^2 to $N \log N$ transforming digital sampling (alongside the development of high frequency ADCs).

Operation - A single operation involves a complex multiplication followed by a complex addition. For example multiplying a $\mathbb{C}^{N \times 1}$ vector by a $\mathbb{C}^{N \times N}$ matrix involves N operations for computing each element of the resulting vector, hence N^2 in total during the normal DFT.

The rows of a Fourier matrix are simply the k'th roots of unity $z^N = 1$:

$$e^{in\frac{2\pi}{N}}$$

Multiplication by a fundamental root of unity ($n = 1$) twice is equal to the fundamental root of unity for $z^{\frac{N}{2}} = 1$:

$$\left(e^{i\frac{2\pi}{N}}\right)^2 = e^{i2\frac{2\pi}{N}}$$

Therefore with some slight adjustment, a N dimensional Fourier matrix can be calculated using a Fourier matrix half the size + some permutation and correction factors, reducing half of the Fourier matrix to 0s (operations we can skip):

$$F_{64} = \begin{pmatrix} I & D \\ I & -D \end{pmatrix} \begin{pmatrix} F_{32} & 0 \\ 0 & F_{32} \end{pmatrix} C$$

Where C is a **circulant matrix** (LTD: Describe properties) which rearranges the order of the signal samples and D has the roots of unity along its diagonal. Furthermore, the circulant matrix can be "hardwired" into the application and does not require any significant computation. This decomposition can be iteratively carried out, further reducing the dimensions of the Fourier matrices and the number of operations.

Gleichzeitig diagonalisierbar => AB = BA and alle EW haben AM 1, sie haben dieselbe eigenbasis

17.7 Compression

17.7.1 Low-rank Approximation
Recall that all matrices can be decomposed to:

$$A \in \mathbb{R}^{m \times n}, u_i \in \mathbb{R}^{m \times 1}, v_i \in \mathbb{R}^{n \times 1}$$

$$A = \sum_{i=1}^{r} \sigma_i u_i (v_i)^H$$

Storing the elements of the SVD of a matrix requires less space if the following inequality holds:

$$r(m + n + 1) < m \cdot n$$

Rearranging, SVD storage is advantageous for matrices where:

$$r < \frac{m \cdot n}{m + n + 1}$$

Most matrices holding data (rather than some transformation with heavily related elements) satisfy this inequality, making SVD a good choice for storing images. Furthermore, we can set a numerical rank k and compress an image by only keeping the first x columns of U and V, which are the heaviest weighted and most "important" components of the entire decomposition. This is a better approximation of the matrix than simpler kinds of lossy compression, for example taking the average values of submatrices to reduce dimension.

Here are some nice demonstrations of SVD compression: https://dmicz.github.io/machine-learning/svd-image-compression/

Working with matrices in the SVD format is more tricky, however many algorithms can be adjusted to handle matrices in this format (Numerical Methods course).

17.7.1.1 Eckart-Young Theorem
The SVD of a matrix cut-off at numerical rank k (A_k) is the best posible approximation at this rank limit:

$$\|A - A_k\| = \min_{\text{Rank}(B) \leq k} \|A - B\| = \sigma_{k+1}$$

Where σ_{k+1} is the max singular value of the discarded terms (definition of Spectral norm) and represents the smallest possible error of the approximation. This has been proven for the Spectral, Frobenius and Nuclear norms.

17.8 Polar Decomposition
A square, invertible matrix can be converted into a "polar form" of a stretch followed by a rotation, just like complex numbers can be expressed as $z = |z| \operatorname{cis}(\theta)$:

$$A = U\Sigma V^H = UV^H (V\Sigma V^H) = QS$$

Where Q is an orthogonal rotation matrix and S is a symmetric positive-definite stretching matrix.

18 Upcoming
LTD:
- PCA
- 4D, Flatland trick, hyperplane, cube etc

- Jacobian