Summary

Inhalt

[1. Linear Algebra Basics 2](#_Toc52262519)

[1.1 Scalar, points and vectors 2](#_Toc52262520)

[1.2 Vector addition and subtraction 2](#_Toc52262521)

[1.3 Vector space 2](#_Toc52262522)

[1.4 Linear independence 2](#_Toc52262523)

[1.5 Affine space 3](#_Toc52262524)

[1.6 Dot and cross products 3](#_Toc52262525)

[1.6.1 Dot-product 3](#_Toc52262526)

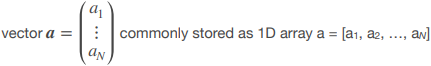
[1.6.2 Cross-product 4](#_Toc52262527)

[1.7 Matrices 4](#_Toc52262528)

[1.7.1 Matrix Multiplication and Inverse 4](#_Toc52262529)

# 1. Linear Algebra Basics

## 1.1 Scalar, points and vectors

* Scalars, vectors, matrices and tensors
  + Individual numerical measures or numeric data elements can be combined into data records such as vectors and matrices
* Basic types denoting points and directions in space in N-dimensions
  + N-tuple of values, common column vector math notation
  + 
* A vector is a quantity with orientation and magnitude
  + High-dimensional data vector (e.g. velocity or force in physics)
* Multiway data arrays for 2D matrices or general d *N*D data tensors

## 1.2 Vector addition and subtraction

* Vector addition is diagonal of parallelogram
  + Head-to-tail rule of placing vectors
* Scalar multiplication and component-wise vector addition
  + Scalarmultiplication *b=s\*a* is equivalent to adding a together s times
* Subtraction is equivalent to adding a negative vector
  + Vector negation is inversion of direction
* Vectors are equal if of same length and direction
  + Subtract to a zero vector

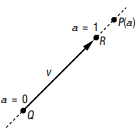
## 1.3 Vector space

* In a vector space V, addition and multiplication satisfy certain mathematical rules, such as being:
  + Closed: *u+v ∈ V, ∀ u,v ∈ V*
  + Commutative: *u + v = v + u*
  + Associative: *u + (v + w) = (u + v) + w*
  + Distributive: *α(u + v) = αu + αv , (α + β)u = αu + βu* for scalars *α,β*
  + Zero vector: *u + 0 = u*
  + Additive inverse: *u + (-u) = 0*
* A vector space only contains vectors as elements!

## 1.4 Linear independence

* A linear combination of *N* vectors *v1 … vn* is defined as *u = a1\*v1+…+an\*vn*
  + Linearly independent if *u=0* only if all *ai=0*
  + The dimension is defined by the largest number of linearly independent vectors
* *n* linearly independent vectors *v1…vn* form a basis of an *n*-dimensional vector space *V*
  + *∀u ∈ V, ∃β1…N ∈ ℝ u = β1 \* v1 + … + βN \* vN*
* A cartesian coordinate system is defined by a set of orthonormal basis vectors
* Given by the standard Cartesian N-dimensional basis vectors *vi=1,…N = (δi1,…, δij , …, δiN) T* with components *δij*
  + For 3D we get the expected *v1 = (1,0,0) v2 = (0,1,0) v3 = (0,0,1)*

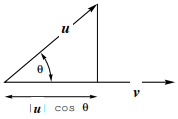
## 1.5 Affine space

* An affine space contains points in addition to scalars and vectors
  + Points are locations and different from vectors
  + Operations: point-vector addition and point-point subtraction
    - Add a vector to a point to get a new point
    - Subtract two points to get the vector in between them
  + No addition of points or multiplication of points
    - Cannot scale a point by scalar multiplication
* The line *L(t): P = Q + t⋅(R–Q)* is an affine space
  + 
* A line, a plane or a volume in 2D or 3D space represents an affine (sub-) space

## 1.6 Dot and cross products

### 1.6.1 Dot-product

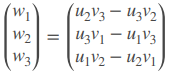
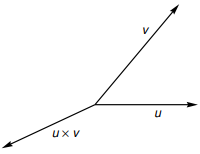
* Orthogonal projection of *u* onto *v* given by *|u| ⋅ cos θ = u ◦ v / |v|*



* Given two *N*-D vectors *u* and *v* the inner- or dot-product ‘◦’ is defined as u◦v = u1 v1 +...+ uNvN
  + Operation on two vectors with scalar result
* The dot-product is
  + Symmetric: *u ◦ v = v ◦ u*
  + Non-degenerate: *v ◦ v = 0 ⇔ v = 0*
  + Bilinear: *v ◦ (u + αw) = v ◦ u + α (v ◦ w)*

### 1.6.2 Cross-product

* Given two *n*-D vectors *u* and *v* the cross-product *w = u × v* is defined as

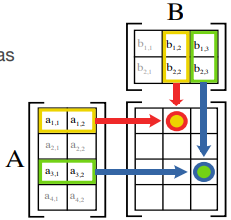
 

* The cross-product *w = u × v* is perpendicular to *u* and *v*
  + Defines right-handed vector system
* Relation to angle between vectors: *|u × v| = |sin θ| ⋅ |u|⋅|v|*
* Vectors *u* and *v* are collinear if *u × v = 0*
* Length *|u × v|* is area of parallelogram spanned by *u* and *v*

## 1.7 Matrices

* A matrix is a two-dimensional array of numbers
  + *MxN* grid of values organized in *M* rows and *N* columns
    - 2D-matrix is commonly stored as a 2D array = [[…],[…]]
* Special cases:
  + Square: M=N
  + Diagonal: all *aij = 0* for *i ≠ j*
  + Identity: diagonal and all *aii=1*
  + Zero/ null matrices: all entries 0
* Two matrices A and B are equal if *aij= bij*
* The sum of two matrices A and B is *cij = aij + bij*
* The scalar multiple *C= s\*A* is *cij = s ⋅ aij*

### 1.7.1 Matrix Multiplication and Inverse

* Is defined component-wise on its elements
* *MxN* matrix *A* and *NxP* matrix *B* we get an *MxP* matrix *C*
  +  *Cij= aix◦bxj=ai1\*b1j+…+ain\*bnj*
  + Matrix multiplication is not commutative *AB ≠ BA*
  + Matrix multiplication is distributive *A(B+C) = AB + AC*
* The inverse *A-1* has the property that *A-1A=AA-1=I*
  + *I* being the identity matrix
  + Can be found by Gaussian elimination in general
  + Inverse only exists for square matrices with non-zero determinant

## 1.8 Transpose