Vectors

Kindergarten: vector $= (v_1, \ldots, v_n)$, $v_i \in \mathbb{R}$
Quantum kindergarten: $v_i \in \mathbb{C}$

Mathematics: **vector space** (linear space) is defined by the axioms:

For vectors u, v, w and numbers $a, b \in \mathbb{R}$ or $\mathbb{C}$ (a field[a] in general):

$$
\begin{aligned}
u + (v + w) &= (u + v) + w \\
u + v &= v + u \\
\exists \text{ null vector } 0 : v + 0 &= v \\
\exists \text{ opposite vector } -v : v + (-v) &= 0 \\
a(bv) &= (ab)v \\
1v &= v \\
a(u + v) &= au + av \\
(a + b)v &= av + bv
\end{aligned}
$$

} as in $\mathbb{R}$

Notation: v, $\boldsymbol{v}$, $\vec{v}$ (real in 2D, 3D), $\underline{v}$, $|v\rangle$ (“ket”), v_i[b]

[a] česky komutativní těleso

[b] formally, v_i is one element, but the logic is the same as for f vs. $f(x)$

Linear dependence

A set of nonzero vectors $v^{(i)}$, $i = 1..m$, is **linearly dependent** if there is a null linear combination with at least one of a_i nonzero:

$$\sum a_i v^{(i)} = 0$$

A linearly independent set of vectors such that any vector (of given space) can be expressed as its linear combination is called a **basis**

$$v = \sum v_i b^{(i)}$$

mmpc1.mw
mmpc1/1a-lindep.py
mmpc1/1b-lindep.py
mmpc1/1c-basis.py

Example. Are the following vectors in $\mathbb{R}^4$ linearly dependent?

$$(1, 2, 3, 4)^\mathsf{T}, (1, -2, 3, -4)^\mathsf{T}, (1, 0, 1, 0)^\mathsf{T}$$

no

Example. Are the following vectors in $\mathbb{C}^2$ linearly dependent?

$$\begin{pmatrix} i \\ 1 \end{pmatrix}, \begin{pmatrix} 1+i \\ 1-i \end{pmatrix}$$

yes

Example. Consider a linear space of functions of $x \in [0, 2\pi]$ with basis $\{1, \cos(x), \cos(2x), \cos(3x), \ldots\}$. Can function $\cos^2(x)$ be expressed in this basis?

yes: $\{1/2, 0, 1/2, 0, 0, 0, \ldots\}$

Scalar (inner, dot) product

We need a richer structure!

Kindergarten: scalar product $\vec{u}\cdot\vec{v}=\sum u_i v_i$

Mathematics: (u,v) is a number (real, complex) obeying axioms:

$$\begin{aligned}
(u,v) &= (v,u)^* \ (^* = \text{complex conjugate})\\
(au,v) &= a^*(u,v) \quad \text{(in physics)} \quad \Rightarrow \quad (u,av)=a(u,v)\\
&= a(u,v) \quad \text{(in mathematics)} \ \Rightarrow \quad (u,av)=a^*(u,v)\\
(u+v,w) &= (u,w)+(v,w)\\
(u,u) &\geq 0\\
(u,u)=0 &\Rightarrow u=0 \text{ (null vector)}
\end{aligned}$$

Notation: $u^{\mathsf{T}}v$, $u^{\mathsf{T}}\cdot v$, $u^{\dagger}v$,[c] (u,v), $\langle u,v\rangle$, $\vec{u}\cdot\vec{v}$, $\boldsymbol{u}\cdot\boldsymbol{v}$, $\langle u|v\rangle$ (bra-ket), $u_i v^i$ (covector-vector)
$\cdot$ (real spaces) or | (complex spaces) = sum over a pair of indices

Definition: If $(u,v)=0$, vectors u, v are **perpendicular**

$(u,u)^{1/2}=|u|=\|u\|=$ norm[d]

[c] symbol † = transpose + complex conjugate = adjoint = Hermitean (Hermitian) conjugate

[d] similar space with a norm only (and complete) = Banach space; under some conditions $(u,v)=(|u+v|^2-|u-v|^2)/4$

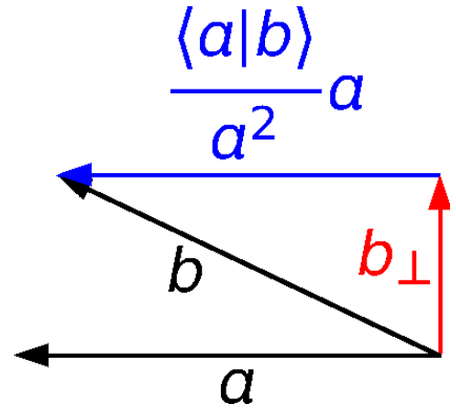
(Cauchy–)Schwarz inequality

Dot-product in $\mathbb{R}^n$: $\vec{x}\cdot\vec{y} = |\vec{x}||\vec{y}|\cos\theta \equiv xy\cos\theta \le xy$.

For nonzero a, b (zero cases are trivial):[e]

$$b_\perp = b - \frac{\langle a|b\rangle}{a^2}a \quad\Rightarrow\quad \langle a|b_\perp\rangle = \langle a|b - \frac{\langle a|b\rangle}{a^2}a\rangle = \langle a|b\rangle - \frac{\langle a|b\rangle}{a^2}\langle a|a\rangle = 0$$

$$b = \frac{\langle a|b\rangle}{a^2}a + b_\perp$$



Pythagoras[f]:

$$b^2 = \left(\frac{|\langle a|b\rangle|}{a^2}\right)^2 a^2 + b_\perp^2 \ge \left(\frac{|\langle a|b\rangle|}{a^2}\right)^2 a^2 = \frac{|\langle a|b\rangle|^2}{a^2}$$

$$a^2b^2 \ge |\langle a|b\rangle|^2 \quad\Rightarrow\quad \boxed{|a||b| \ge |\langle a|b\rangle|} \overset{\mathbb{R}}{\ge} \langle a|b\rangle$$

|| has two meanings:
|complex number|
and |vector|

⇒ **triangle inequality** (in $\mathbb{R}$)

$$|a+b| \le |a| + |b| \quad \text{or} \quad |x-z| \le |x-y| + |y-z|$$

i.e., $|a-b|$ is a **metric**.

[e] Common shortcut: $a^2 \equiv |a|^2 = \langle a|a\rangle$; symbol a^2 is less appropriate in complex spaces, better use $|a|^2$

[f] In complex spaces: $\langle b|a\rangle^* = \langle a|b\rangle$ and for scalar $c \in \mathbb{C}$ it holds $|ca|^2 = \langle ca|ca\rangle = c^*c\langle a|a\rangle = |c|^2|a|^2$

Hilbert space

Hilbert space = linear space with a scalar product which is:

- complete (any Cauchy sequence[g] converges in the (u,u) metric)
  analogy: rational numbers have "holes": irrational numbers
- usually also separable (it contains a countable dense subset
  $\Rightarrow$ there is a countable basis)

Example of incomplete space: finite linear combinations of: {
$(1,0,0,\ldots)$,
$(0,1,0,\ldots)$,
$(0,0,1,\ldots)$,
$\vdots$ }

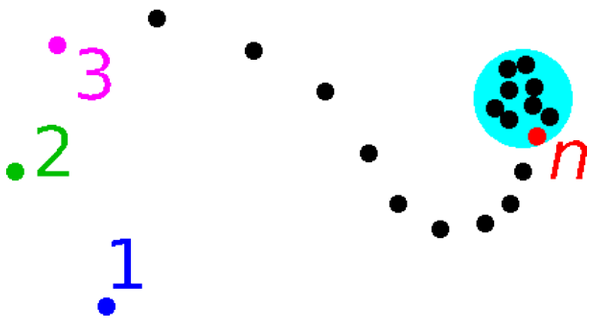
Loosely: "no vector is missing"
"it is not too big" or "there are no problems with using infinite sums"

Any finite vector space is a Hilbert space.

Example. Wavefunction is a vector of a Hilbert space, $\int |\psi(\boldsymbol{\tau})|^2 \mathrm{d}\boldsymbol{\tau}$ must be finite[h]. The scalar product is:

$$\langle\phi|\psi\rangle = \int \phi(\boldsymbol{\tau})^* \, \psi(\boldsymbol{\tau}) \mathrm{d}\boldsymbol{\tau}$$

n bosons: $\boldsymbol{\tau} \in \mathbb{R}^{3n}$, n fermions (chemistry): $\boldsymbol{\tau} \in (\mathbb{R} \times \{\alpha, \beta\})^{3n}$



[g] Sequence $\{\nu_i\}_{i=1}^{\infty}$ is Cauchy if $\forall d > 0 \; \exists n : |\nu_j - \nu_i| < d \; \forall i,j > n$, $\rightarrow\rightarrow\rightarrow$
"sequences of vectors whose elements get arbitrarily close to one another."

[h] For bound states , cf. the de Broglie free-space "matter waves" . . .

Orthogonal and orthonormal bases

Orthogonal basis = all vectors are perpendicular.

Orthonormal basis = also normalized:

$$b^{(i)} \cdot b^{(j)} = \delta_{ij}$$

Components of v in an orthonormal basis:

$$v_i = v \cdot b^{(i)} \;\Rightarrow\; v = \sum v_i b^{(i)} = (v_1, \ldots, v_n)_b$$

Scalar product:

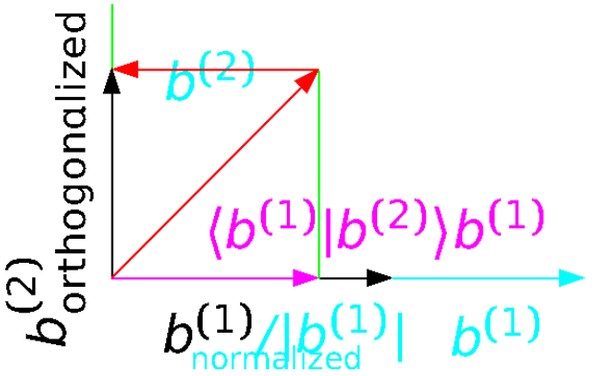
$$u \cdot v = \sum u_i v_i$$

Scalar product in $\mathbb{C}$ in physics

$$\langle u|v\rangle = \sum u_i^* v_i$$

Gram–Schmidt orthogonalization

A general basis $b^{(i)}$ can be orthogonalized
by the Gram–Schmidt algorithm:



$$b^{(1)} := b^{(1)}/|b^{(1)}|$$

$$b^{(2)} := b^{(2)} - \langle b^{(1)}|b^{(2)}\rangle b^{(1)},\ b^{(2)} := b^{(2)}/|b^{(2)}|$$

$$b^{(3)} := b^{(3)} - \langle b^{(1)}|b^{(3)}\rangle b^{(1)} - \langle b^{(2)}|b^{(3)}\rangle b^{(2)},\ b^{(3)} := b^{(3)}/|b^{(3)}|$$

":=" means "assign to" as in some programming languages (also "=" or "<-")

Bases used in a Hilbert space are usually orthogonal or orthonormal

Example. Find all orthonormal bases $\{b^{(1)}, b^{(2)}\}$ in $\mathbb{C}^2$ for $b^{(1)} = (1, i)/\sqrt{2}$

$$\left\langle \begin{pmatrix} 1 \\ i \end{pmatrix} \middle| \begin{pmatrix} x \\ y \end{pmatrix} \right\rangle = x - iy \stackrel{!}{=} 0 \quad \Rightarrow \quad x = iy \quad \Rightarrow \quad b^{(2)} = \frac{c}{\sqrt{2}} \left| \begin{pmatrix} i \\ 1 \end{pmatrix} \right\rangle, |c| = 1$$

Remember complex conjugate in the dot product, $i^* = -i$

Another notation: $\left| \begin{matrix} 1 \\ i \end{matrix} \right\rangle^\dagger = \left\langle \begin{matrix} 1 \\ -i \end{matrix} \right|$

more examples:
mmpc1.mw
mmpc1/1*.py

Linear forms

Linear form f (linear operator) assigns a number $f(v) \in \mathbb{R}$ (or $\mathbb{C}$) to a vector.

Axioms: for linear forms f, g, number a, and a vector v:

$$\begin{aligned} (f+g)(v) &= f(v)+g(v) \\ f(av) &= af(v) \end{aligned}$$

For finite n one can write (in infinite-dimension spaces there may be continuity problems):

$$f(u) = \sum_{i=1}^{n} f_i u_i$$

Otherwise in Hilbert spaces linear form ≈ scalar product:

$$f(v) = \sum f_i v_i = (f^*, v)$$

Linear form in Euclidean spaces (in some context) = covector, dual vector, covariant vector ("normal" vector = contravariant vector). Usually:

- vector = column vector,
- covector = row vector (transposed) f^{T}, inverse transformation if a basis changes

Scalar product then is: $f(u) = f^{\mathsf{T}} \cdot u = f^{\mathsf{T}} u = f^i u_i$ (Einstein summation convention).

In complex Hilbert spaces $^{\mathsf{T}} \to ^{\dagger}$.

Covector example

Example. Force $\vec{F}$ = covector, path $\mathrm{d}\vec{s}$ = vector.

$$\vec{F} = -\vec{\nabla} U, \quad \mathrm{d}W = \vec{F} \cdot \mathrm{d}\vec{s}$$

Units: $[\vec{F}]$ = energy/length, $[\mathrm{d}\vec{s}]$ = length.

If length unit changes from m to cm, $\mathrm{d}\vec{s}$ multiplies $100\times$, but (if the energy unit remains the same) $\vec{F}$ multiplies $0.01\times$.

Linear algebra in software:

Maple

In package LinearAlgebra,
operator "." is used for dot product:
`covector.vector`
`rows.columns` (in matrix multiplication)

^+ = transposition
^∗ = Hermitean conjugate

vector = column vector (^+ needed)

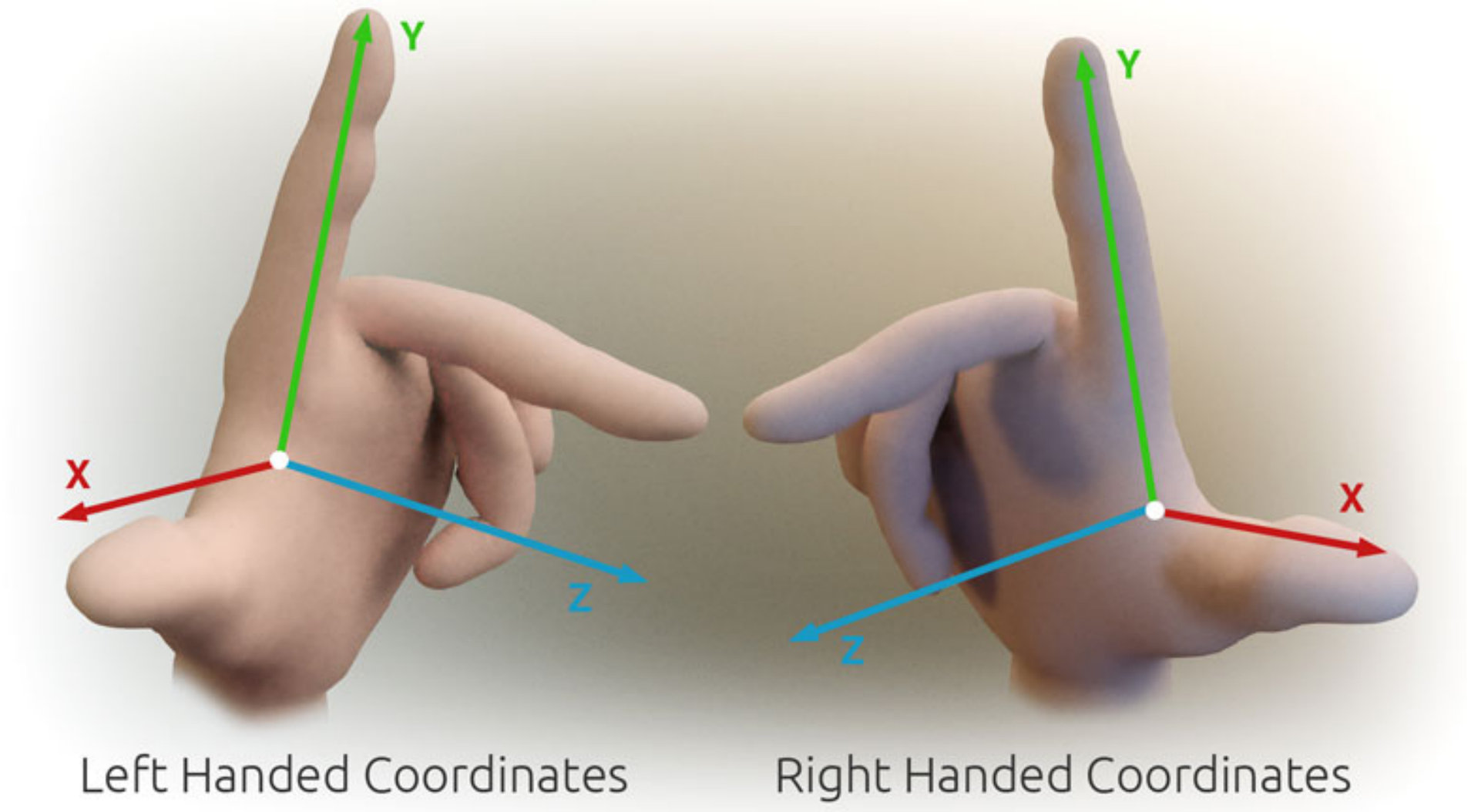
Python (numpy)

real spaces: numpy.dot(,) = @ = dot product
complex spaces: numpy.vdot(,)
both: norm = numpy.linalg.norm()

transposition: M.T
Hermitean conjugate: M.H (for Matrix M)
M.conj().T (for array M[a])
column: A[:,col] (line vector)

[a] not recommended: less efficient implementation

3D: Right- and left-handed coordinate system



credit: Wikipedia

Right-handed: math, science, technology (Maple default), most Python libraries (mplot3d)

Left-handed: 3D image processing (Micro$oft Direct 3D (part of DirectX), PovRay)

Matrices

Square matrix $n \times n$, for instance:

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}$$

may represent:

- matrix of coefficients of a set of n of linear equations for n unknowns:

$$\sum_j A_{ij} x_j = b_i \quad \text{or} \quad A \cdot x = b \quad \text{or} \quad Ax = b \quad \text{or} \quad |\hat{A}|x\rangle = |b\rangle$$

- linear transformation (map, operator) $\mathbb{R}^n \to \mathbb{R}^n$ or $\mathbb{C}^n \to \mathbb{C}^n$

$$x_i \to \sum_j A_{ij} x_j \quad \text{or} \quad x \to A \cdot x \quad \text{or} \quad x \to Ax \quad \text{or} \quad |x\rangle \to |\hat{A}|x\rangle$$

- matrix of coefficients of a quadratic form $\mathbb{R}^n \to \mathbb{R}$ or $\mathbb{C}^n \to \mathbb{C}$

$$x_i \to \sum_{ij} x_i A_{ij} x_j \quad \text{or} \quad x \to x^{\mathsf{T}} \cdot A \cdot x \quad \text{or} \quad x \to x^{\mathsf{T}} A x \quad \text{or} \quad |x\rangle \to \langle x|\hat{A}|x\rangle$$

- a quadratic tensor; e.g., of pressure or small deformation

Matrices

Notation:

mmpc1.mw
mmpc1/1g-invmat.py

- In quantum theory often denoted as $\hat{A}$.
- Other habits (e.g., as tensors): $\overleftrightarrow{A}$, $\underline{\underline{A}}$.
- $A \cdot x$ is less common than Ax;
  in the bra-ket notation $A|x\rangle$ or $|Ax\rangle$ or $|A|x\rangle$.
- Vectors u and co-vectors u^{T} or $u^{\dagger} \equiv \langle u|$ ("bra") should be distinguished.

Matrices in infinite-dimension spaces are infinite = linear operators.

If the set of equations $A \cdot x = b$ can be solved $\forall b$, then A is called **regular**. The solution is then:

$$x = A^{-1} \cdot b$$

where A^{-1} = **inverse matrix**, $A \cdot A^{-1} = A^{-1} \cdot A = \delta$, and $\delta = \mathrm{diag}(1, 1, \ldots)$ = **unit matrix**, identity matrix, in coordinates Kronecker delta, also written as E, $\mathbb{1}$, $\boldsymbol{I}$, I, $\overleftrightarrow{I}$, etc.

Examples. Invert matrices:

a) $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$, **b)** $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$

$\begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}$ **(q** , $\begin{pmatrix} 1/3 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ **(e**

Determinant

Determinant of a square matrix A is the number defined as a sum over all $n!$ permutations p of indices $\{1, 2, \ldots, n\}$:

$$\det A = \sum_p \operatorname{sign}(p) \prod A_{i,p(i)}$$

where the signnature of $p = \operatorname{sign}(p) = (-1)^{\text{number of transpositions in } p}$.
Alternative definitions: using elementary transformations / multiplicative property / row expansion.

It holds $\det A \neq 0$ for a regular matrix.

It holds

$$\det(A \cdot B) = \det(A) \det(B), \quad \det(A^{-1}) = \frac{1}{\det A} \text{ (for regular } A\text{).}$$

The determinant of a diagonal or triangular matrix = product of the numbers on the diagonal.

Example. Calculate a) $\operatorname{sign}(2, 3, 1)$, b) $\operatorname{sign}(n, n-1, n-2, \ldots, 2, 1)$

a) 1, b) $(-1)^{n(n-1)/2}$ (= 1 for $n \equiv 0, 3 \bmod 4$ and -1 otherwise)

For completeness – the permanent of A is

$$\operatorname{perm} A = \sum_p \prod A_{i,p(i)}.$$

Unitary matrix

Orthogonal[i] (in $\mathbb{R}^n$) or **unitary** (in $\mathbb{C}^n$) matrix is a square matrix for which:

$$U^{\mathsf{T}} \cdot U = \delta \quad \text{or} \quad U^{\dagger} \cdot U = \delta,$$

or in coordinates

$$\sum_j U^{\mathsf{T}}_{ij} U_{jk} = \sum_j U_{ji} U_{jk} = \delta_{ik} \quad \text{or} \quad \sum_j U^{\dagger}_{ij} U_{jk} = \sum_j U^{*}_{ji} U_{jk} = \delta_{ik}.$$

- Columns U_{*i} can be treated as coordinates of an orthonormal basis (in other orthonormal basis), i.e., a (matrix of) unitary transformation.
- U is regular: $U^{-1} = U^{\dagger}$.
- $|\det U| = 1$ (in $\mathbb{C}$); in $\mathbb{R}$ this means that $\det U = \pm 1$.
- A unitary matrix transforms an orthonormal basis to an orthonormal basis.
- Linear map $x \to U \cdot x$ “preserves angles”, in $\mathbb{R}$ it can be interpreted as:
  - rotation in $\mathbb{R}^n$ (for $\det U = 1$),
  - rotation and reflection in $\mathbb{R}^n$ (for $\det U = -1$).

mmpc1.mw
mmpc1/1h-unitary.py

[i] term “orthonormal” is not used

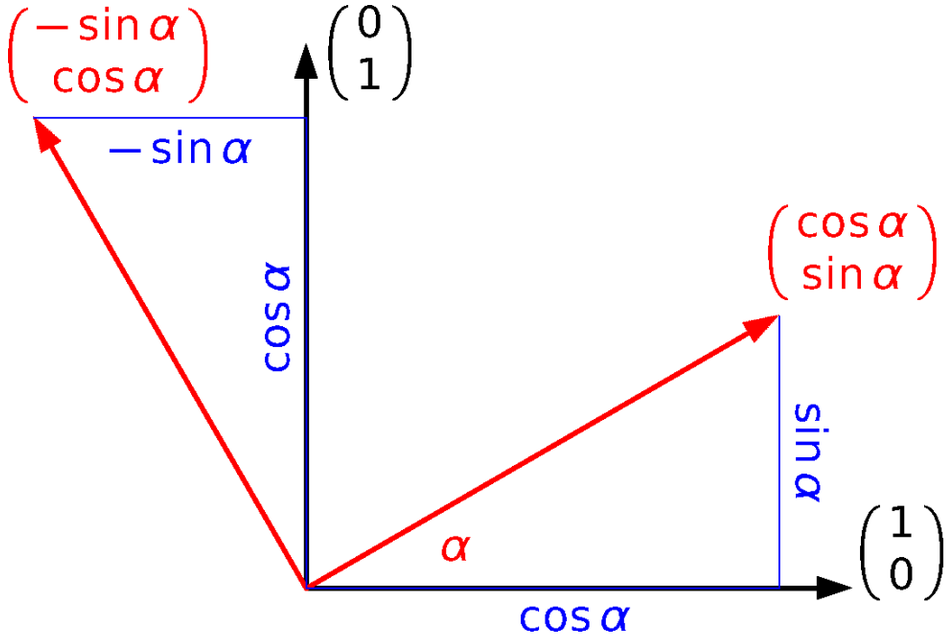
Matrix of rotation

Matrix of rotation by oriented angle $+\alpha$ in 2D:

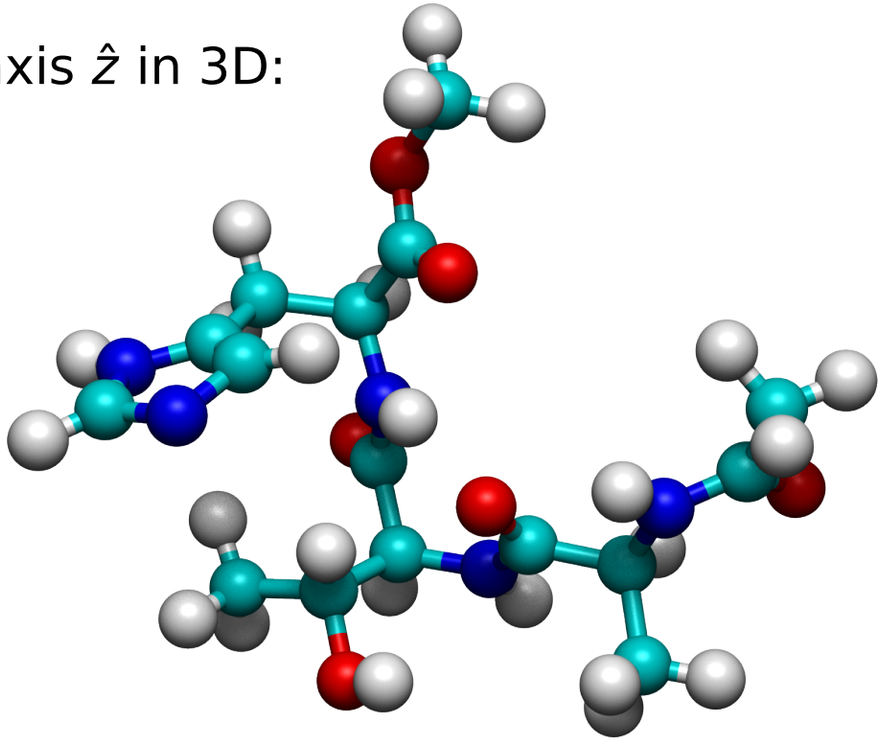
$$\begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}.$$

Matrix of rotation by angle α around axis $\hat{z}$ in 3D:

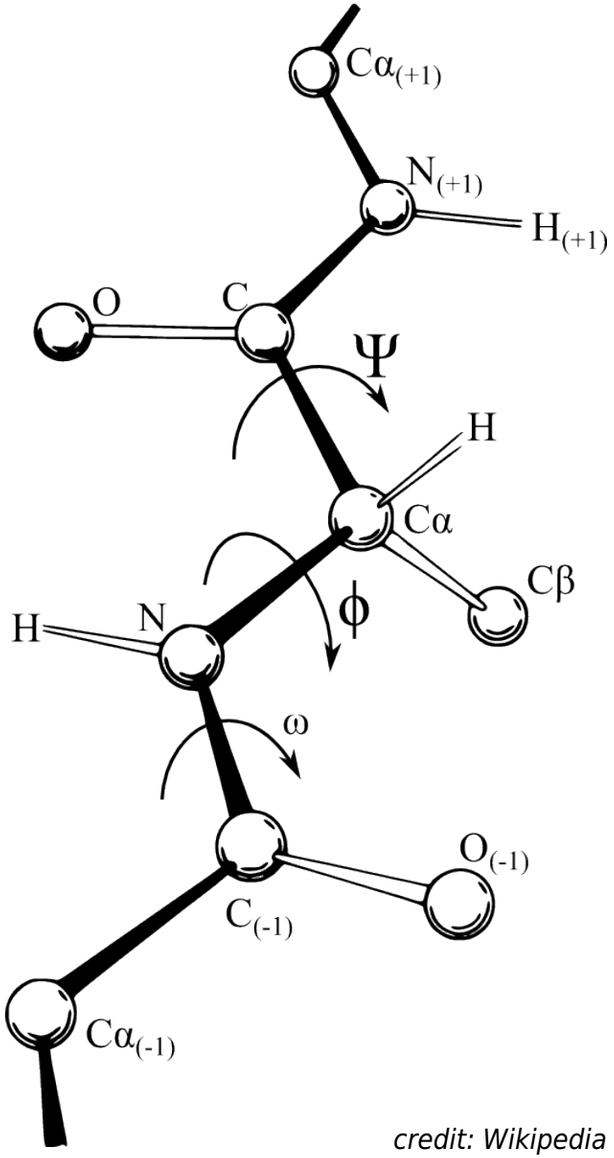
$$\begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$



mmpc1.mw
mmpc1/1i-2drot.py



Internal coordinates:

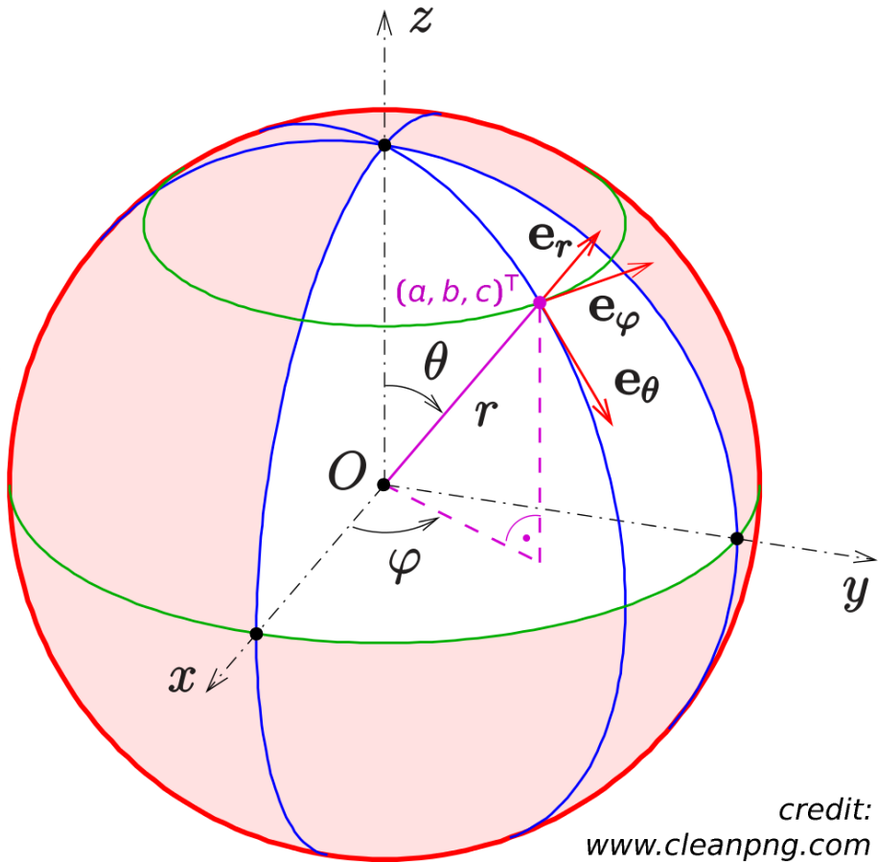


credit: Wikipedia

Example

Write a matrix of rotation by angle α around vector $(a, b, c)^T$

- Use spherical coordinates:
  $(a, b, c) = (r\sin\theta\cos\varphi, r\sin\theta\sin\varphi, r\cos\theta)$
  reverse: $r = \sqrt{a^2 + b^2 + c^2}, \theta = \arccos(c/r), \varphi = \arctan(b, a)$
  Maple: Overloaded function $\arctan(b, a) = \arctan(b/a) + k\pi$, where k is such integer that $\varphi = \arctan(b, a)$ is in the correct quadrant
  Fortran, C, Python: `atan2(,)`
  x86 (assembler): FPATAN
- Compose from right (= in the order it is applied to a vector):
  R_1^{-1} = rotation by $-\varphi$ around $\hat{z}$
  R_2^{-1} = rotation by $-\theta$ around $\hat{y}$
  R_3 = rotation by α around $\hat{z}$
  R_2 = rotation by θ around $\hat{y}$
  R_1 = rotation by φ around $\hat{z}$
- Final rotation matrix: $R = R_1 \cdot R_2 \cdot R_3 \cdot R_2^{-1} \cdot R_1^{-1}$



credit:
www.cleanpng.com

mmpc1.mw
mmpc1/1j-3drot.py