Methods in Computational Science - Introduction (ch.1-4)

Johan Hoffman

Methods in Scientific Computing

- Computing in science, and the science of computing
- Theory- and data-driven computational modeling and simulation
- Modern hardware and software platforms

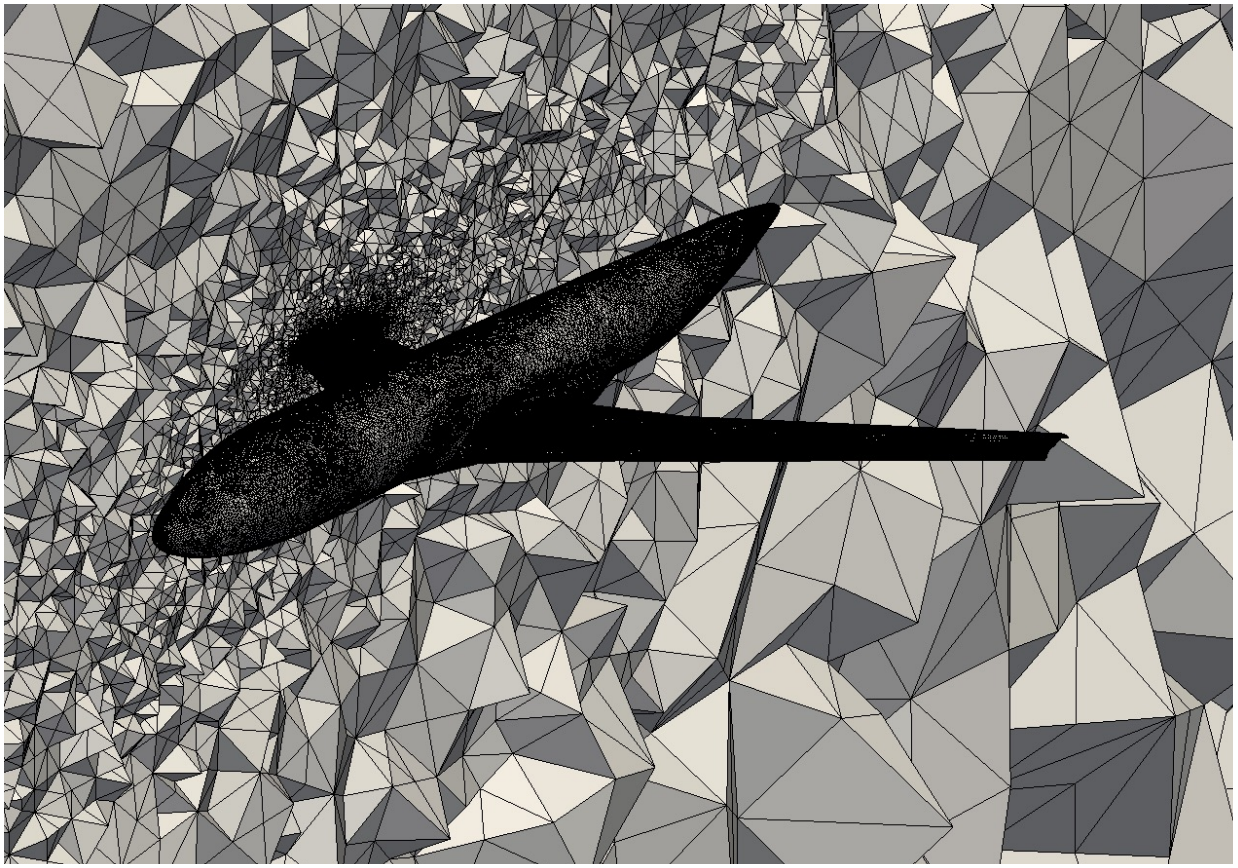
Three key algorithms

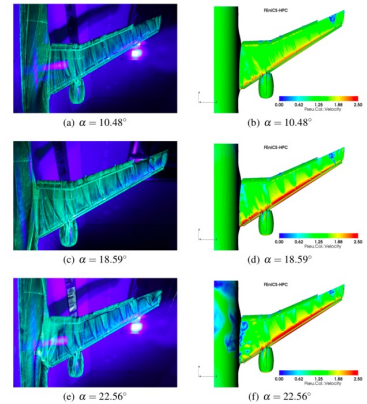
- Time stepping
- Convolution
- Matrix-vector product

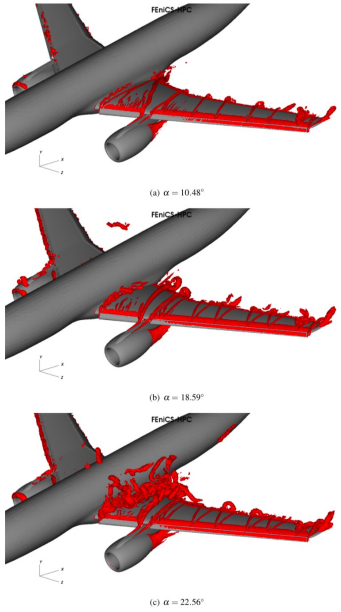
Methods in Scientific Computing

- Book webpage (SIAM)
  https://my.siam.org/Store/Product/viewproduct/?ProductId=39300058
- GitHub repo (slides, Google Colab/Jupyter notebooks)
  https://github.com/johanhoffman/methods-in-computational-science
- Johan Hoffman
  https://www.kth.se/profile/jhoffman

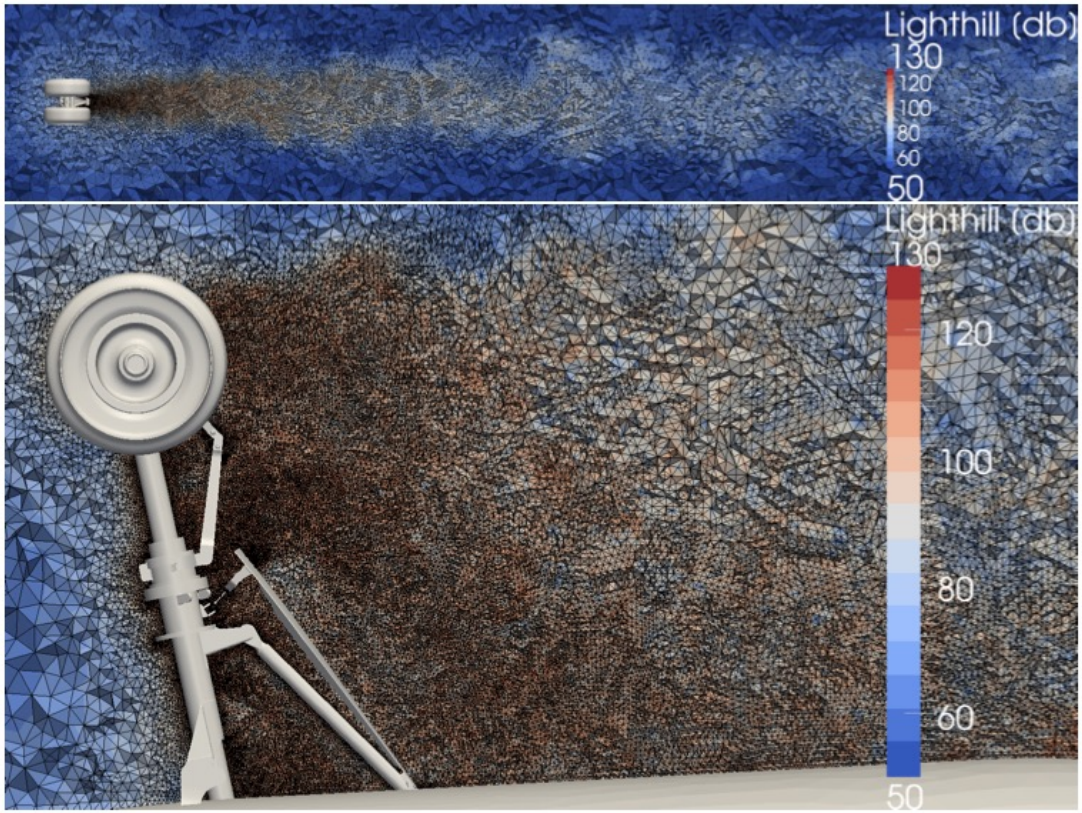
Simulation of turbulent flow

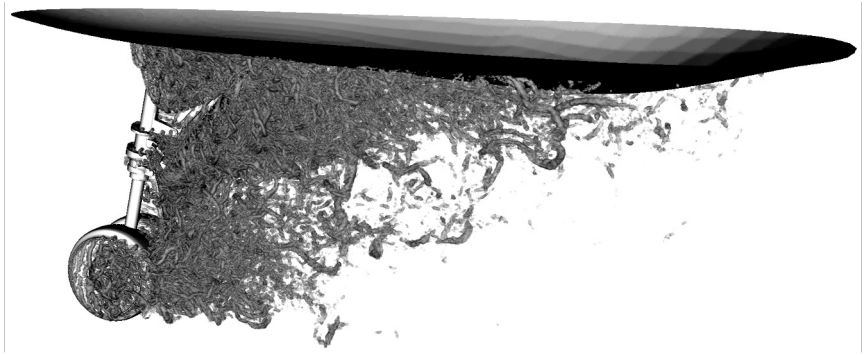




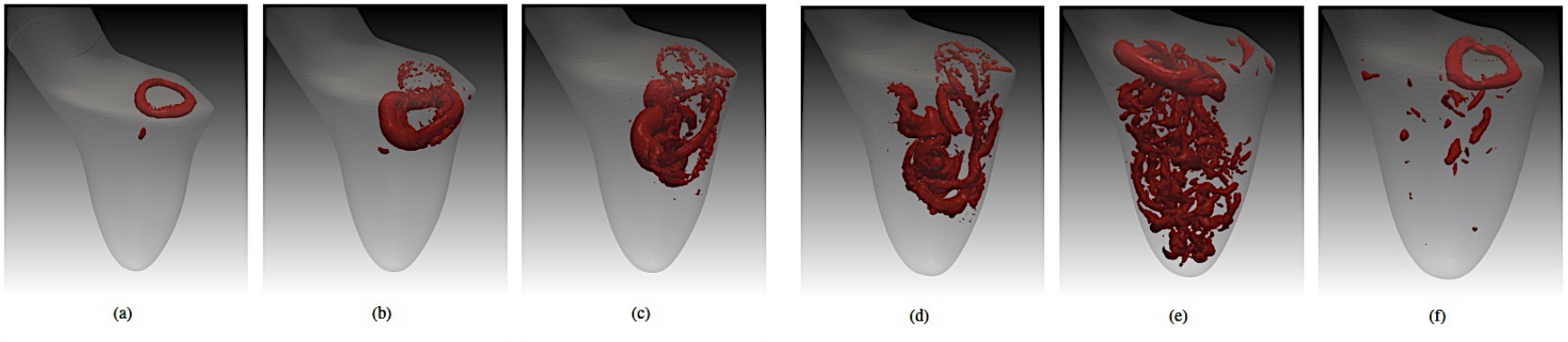


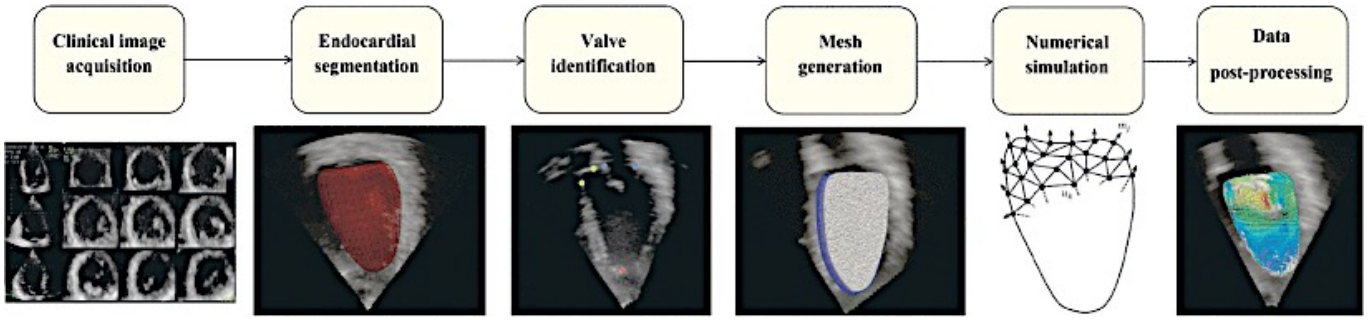
Simulation of turbulent flow





Blood flow simulation and data analysis





Time-stepping algorithm

ALGORITHM 3.6. (u, t) = time_step(T, u0, t0, dt).
Input: state transition function **T**, old state **u0**, old time **t0**, time step **dt**.
Output: new state **u**, new time **t**.

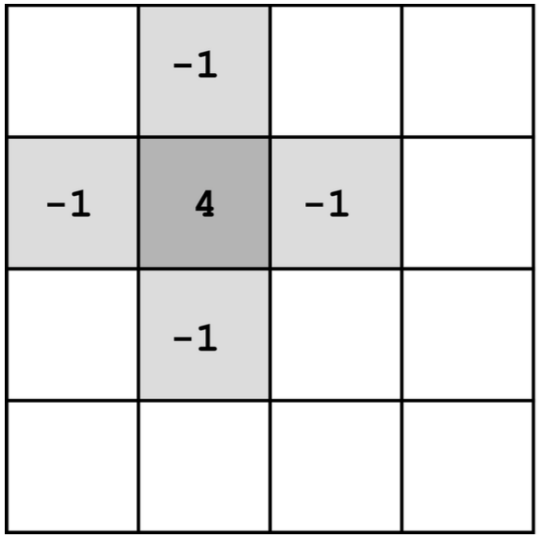
```
1: u = T(u0, t0, dt)
2: t = t0 + dt
3: return u, t
```

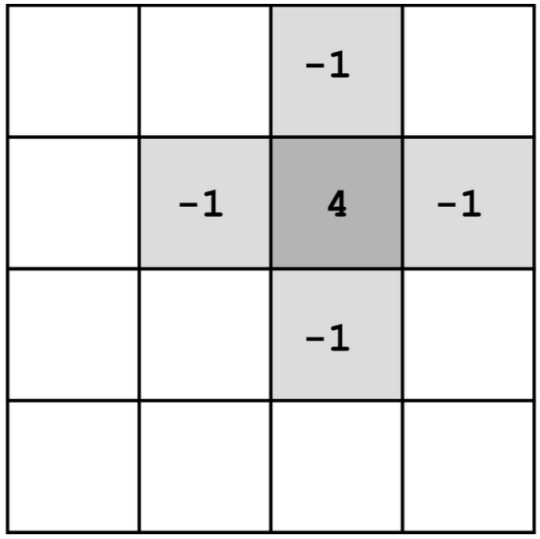
Convolution (filtering)

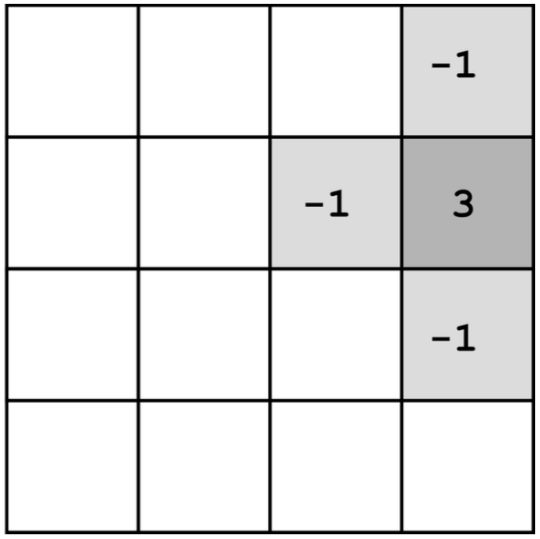
Convolution (filtering)

$$S = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \qquad B = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Convolution (filtering)





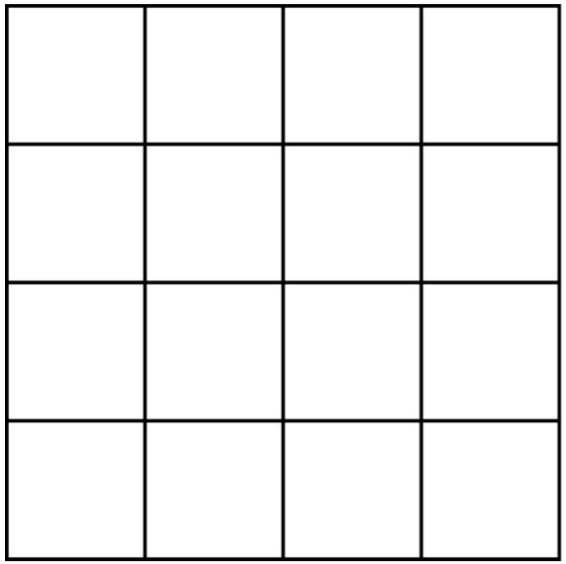


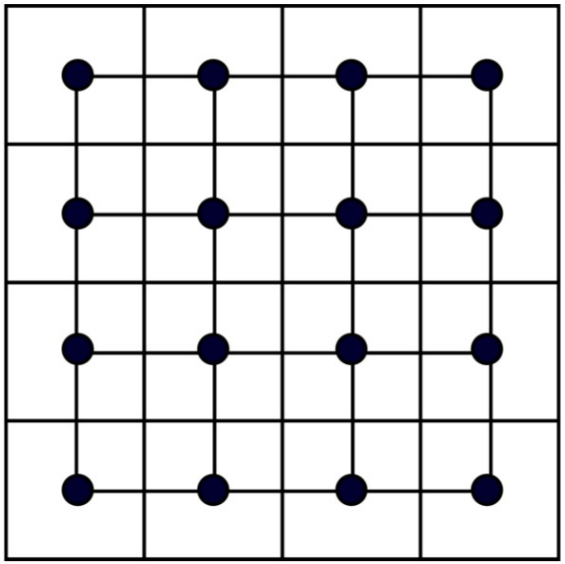
Convolution (filtering)

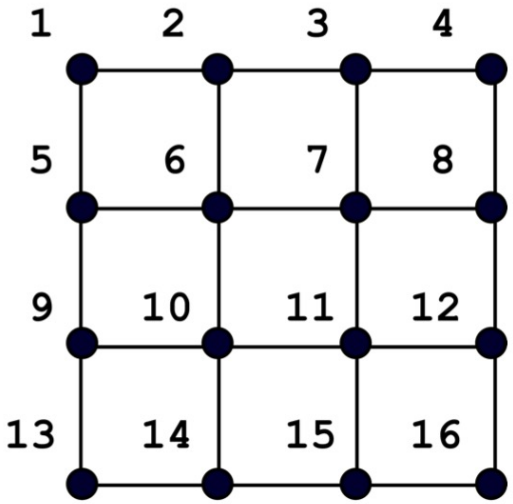
ALGORITHM 3.5. filtered_image = convolution(image, kernel).
Input: an array of pixel intensities **image**, and a **kernel**.
Output: an array of modified pixel intensities **filtered_image**.

```
1: for i=0:length(image)-1 do
2:     neighbors = kernel.get_neighbors(i)
3:     filtered_image[i] = 0
4:     for j=0:length(neighbor_pixels)-1 do
5:         filtered_image[i] = filtered_image[i] + kernel.weights(j) * image[neighbors[j]]
6:     end for
7: end for
8: return filtered_image
```

Vectorization







Convolution kernel -> sparse matrix

$$
\begin{bmatrix}
2 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 2 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 2 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 2
\end{bmatrix}
$$

Matrix-vector product

ALGORITHM 3.2. b = matrix_vector_product(A, x).
Input: an nxn matrix **A** and an n vector **x**.
Output: the matrix-vector product n vector **b** = Ax.

```
1: b = 0
2: for i=0:n-1 do
3:     for j=0:n-1 do
4:         b[i] = b[i]+A[i,j]*x[j]
5:     end for
6: end for
7: return  b
```

High performance computing

Distributed multi-core processors

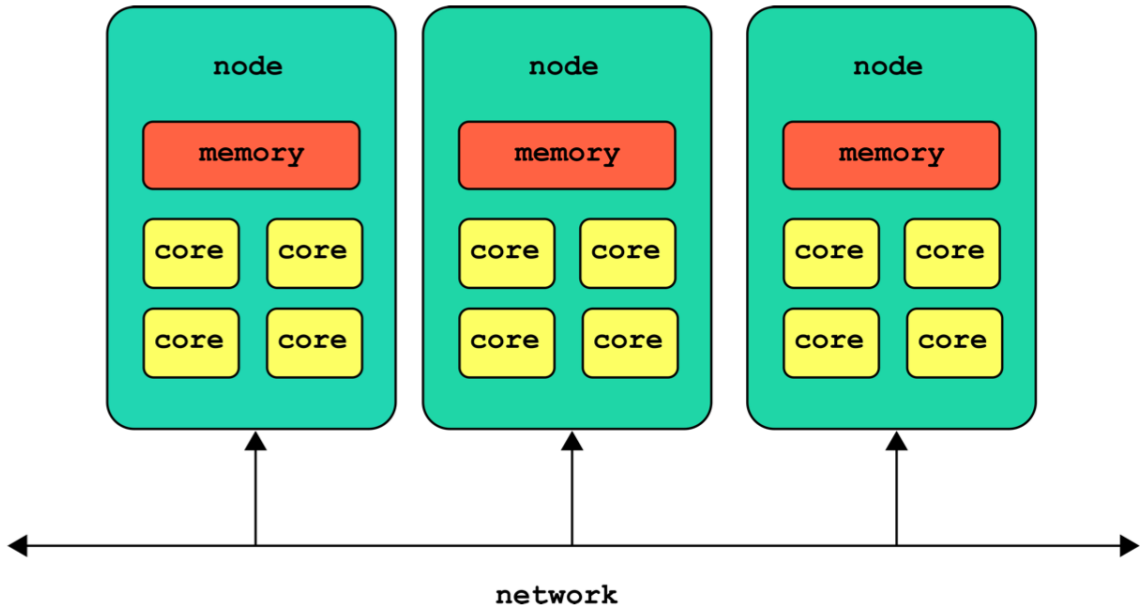


Figure 4.2. *A typical supercomputer architecture with multiple distributed multi-core processors (nodes) which have access to a shared memory, connected through a high speed interconnect network.*

GPU accelerator

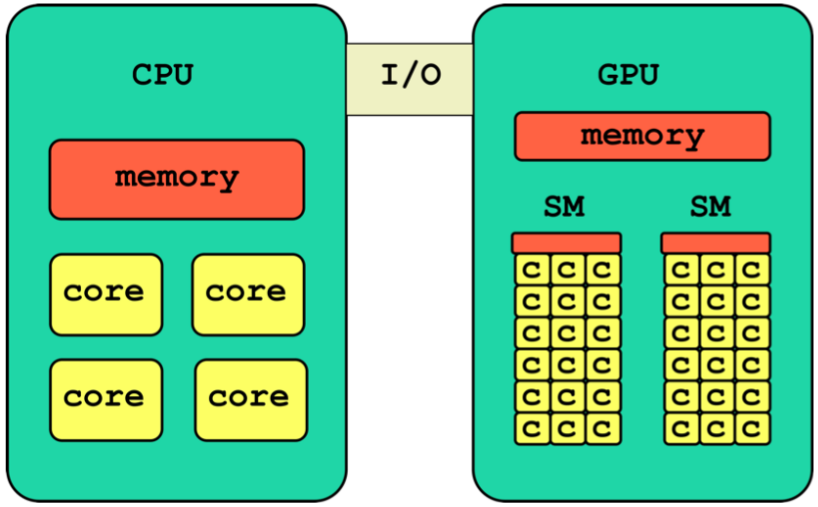


Figure 4.3. *A typical architecture for GPGPU, a CPU host connected to a GPU device. The CPU is a general purpose processor with multiple cores ($\sim$ 10), whereas the GPU functions as an accelerator with many cores ($\sim$ 10 000). The GPU memory is organized in a hierarchy from small local memory for each core, to a shared memory for cores on the same SM, and a global memory for the whole GPU device.*

MPI message passing interface

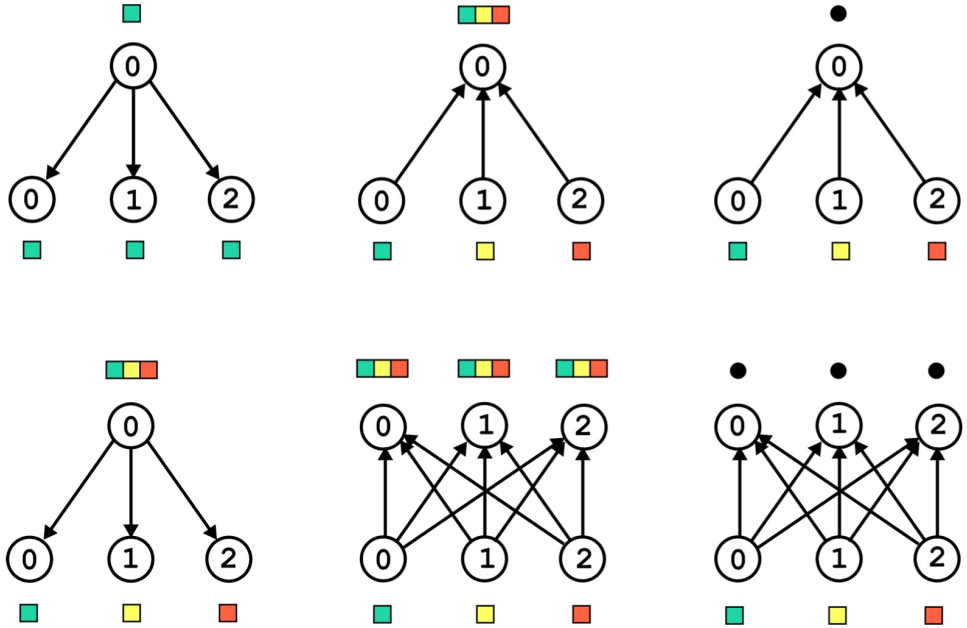


Figure 4.1. *MPI collective communication (left to right):* **MPI_Bcast()** *and* **MPI_Scatter()**, *data broadcast vs a scattered array from a root,* **MPI_Gather()** *and* **MPI_Allgather()**, *gather an array to a root vs to all processes,* **MPI_Reduce()** *and* **MPI_Allreduce()**, *reduction of data to a scalar gathered on the root vs sent to all processes. The squares represent data arrays whereas the dots indicate scalar data.*

ALGORITHM 4.4. z = mpi_vector_addition(x ,y).
Input: two arrays **x** and **y** of length n.
Output: an array **z** which holds the sum of the two arrays x and y.

```
1: MPI_Scatter(x, y, no_local_elements)
2: for i=0:no_local_elements-1 do
3:     z[i] = x[i] + y[i]
4: end for
5: MPI_Gather(z)
6: return z
```

ALGORITHM 4.5. z = mpi_scalar_product(x ,y).
Input: two arrays **x** and **y** of length n.
Output: the scalar product **z** of the two arrays x and y.

```
1: MPI_Scatter(x, y, no_local_elements)
2: z = 0
3: for i=0:no_local_elements-1 do
4:     z = z + x[i]*y[i]
5: end for
6: MPI_Allreduce(z, sum)
7: return z
```

Complexity of algorithms

Algorithms can be broken down into a number of basic operations, elementary arithmetic operations such as summation, subtraction, multiplication and division; and logical, comparison and assignment operations. The computational cost of an algorithm in terms of the number of basic operations is referred to as the *time complexity*, or just *complexity*, of the algorithm. To measure the complexity of an algorithm we can use the asymptotic operation count in all the steps of the algorithm performed for input data of size n, expressed by the *Big O notation*. If the exact operation count of an algorithm is $f(n)$, then

$$f(n) = \mathcal{O}(g(n)),$$

if and only if there exist numbers M and n_0 such that $|f(n)| \leq Mg(n)$, for all $n \geq n_0$. We also say that $f(n)$ and $g(n)$ are *of the same order*, which we write $f(n) \sim g(n)$.

Accuracy and stability of algorithms

The standard approximation of a real number in a computer is a *floating point number*, which takes the form

$$significand \times base^{exponent}$$

where the *significand* and the *exponent* are integers, representing the precision and range of the floating point system, and the *base* an integer greater than or equal to 2. If the base is 10 we have a *decimal floating point number* and if the base is 2 a *binary floating point number*.

The floating point number $fl(x)$ is only an approximation of the real number x it seeks to represent, subject to the *round-off error* $fl(x) - x$, often expressed as the *relative error*

$$\epsilon = (fl(x) - x)/x.$$

Accuracy and stability of algorithms

Example 3.2. The sum of the two floating point numbers 1.0×10^8 and 1.0×10^2, each with two significant digits, has seven significant digits,

$$1.0 \times 10^8 + 1.0 \times 10^2 = 1.000001 \times 10^8,$$

and this difference between two numbers with seven significant digits have two signifiant digits

$$1.000011 \times 10^6 - 1.000001 \times 10^6 = 0.000010 \times 10^6 = 1.0 \times 10^1.$$

The multiplication

$$4.5 \times 10^1 \cdot 4.5 \times 10^1 = 2.025 \times 10^3$$

doubles the number of significant digits and triples the exponent, and the division

$$1.0 \times 10^2 / 3.0 \times 10^2 = 0.333... \times 10^0,$$

leads to a infinite number of significant digits.

Vector spaces (chapter 1)

Vector spaces

We define a *vector space*, or *linear space*, over a field F as a set V which is closed under the two binary operations of *vector addition* and *scalar multiplication*, in other words,

$$x, y \in V \Rightarrow x + y \in V,$$
$$x \in V, \alpha \in F \Rightarrow \alpha x \in V.$$

The elements of V are the *vectors*, and the elements of F the *scalars*. Vector addition and scalar multiplication satisfy the algebraic rules of associativity, commutativity and distributivity, and it must exist a zero vector $0 \in V$, defined by

$$v + 0 = v, \quad \forall v \in V,$$

and an inverse vector $-v \in V$, such that $v + (-v) = v - v = 0$.

The Euclidian space R^n

Example 1.1 (The Euclidian space R^n). We define the Euclidian space R^n as the real vector space of *column vectors*

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{pmatrix} = (x_1, ..., x_i, ..., x_n)^T$$

where each of the n components is a real number $x_i \in R$. Here $(x_1, ..., x_i, ...x_n)$ is a *row vector*, and the superscript T denotes the *transpose* operation which changes a row vector into a column vector. The two vector space operations are defined by component-wise addition and multiplication, for $\alpha \in R$ and $x, y \in R^n$,

(i) $x + y = (x_1 + y_1, ..., x_n + y_n)^T$,

(ii) $\alpha x = (\alpha x_1, ..., \alpha x_n)^T$.

The Euclidian space R^2

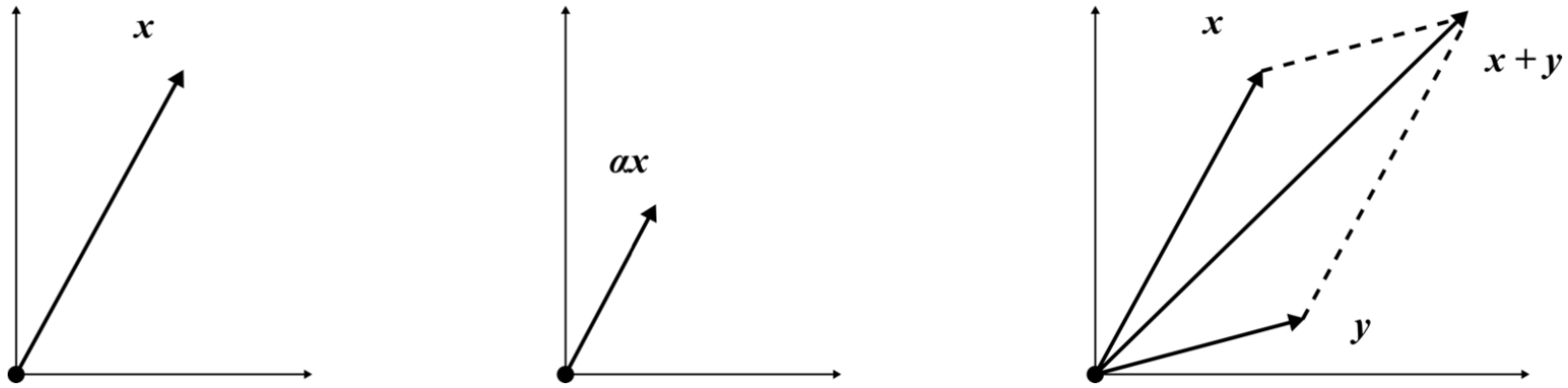


Figure 1.2. *Geometric interpretation of a vector $x \in R^2$ where origo is marked by a dot in the coordinate system (left), scalar multiplication αx with $\alpha = 0.5$ (center), and vector addition $x + y$ (right).*

Functions on vector spaces

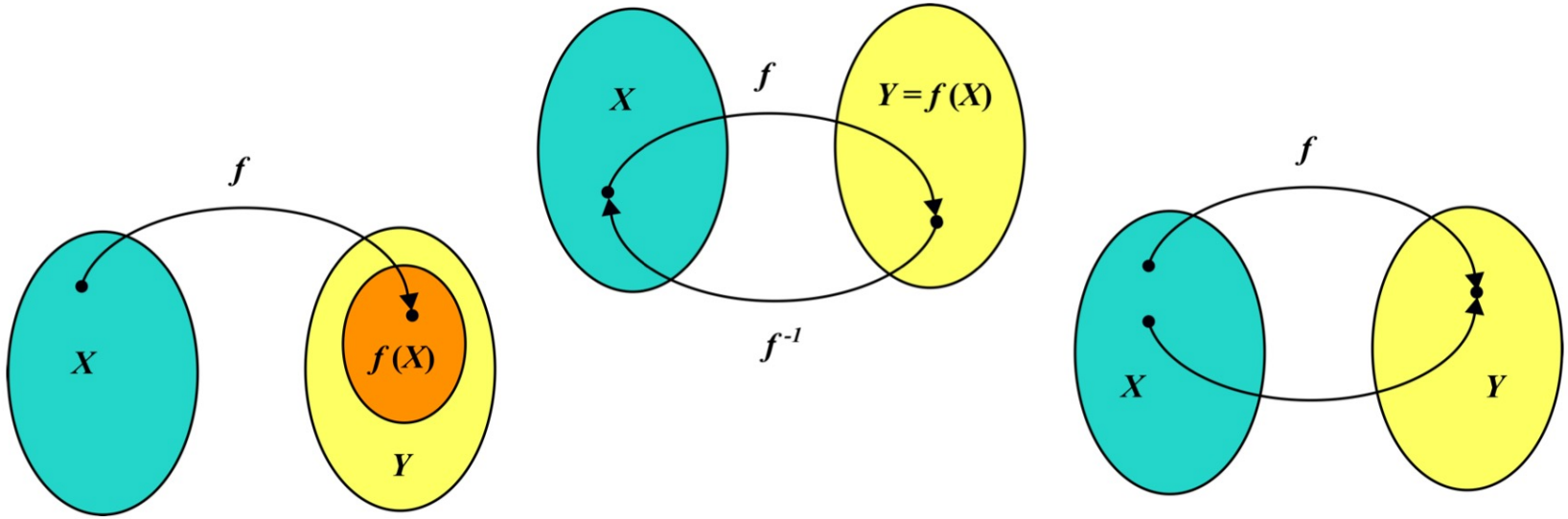


Figure 1.1. *From left to right we show illustrations of: a function $f : X \to Y$ with domain X (blue), codomain Y (yellow) and range $f(X)$ (orange); a bijective function with its inverse $f^{-1} : Y \to X$; and a surjective but not injective function (right).*

Linear, bilinear and quadratic forms

Our main focus is functions for which the domain is a vector space, and of specific interest are functions that map vectors to the scalar field, also referred to as *functionals*. For a real vector space V, a function $L : V \to R$ is a *linear form*, or *linear functional*, if

(i) $L(x + y) = L(x) + L(y), \quad \forall x, y \in V,$
(ii) $L(\alpha x) = \alpha L(x), \quad \forall x \in V, \alpha \in R.$

Similarly, a function $a : V \times V \to R$ is a *bilinear form* if it is linear in both arguments, so that for all $x, y, z \in V$ and $\alpha, \beta \in R$,

(i) $a(\alpha x + \beta y, z) = \alpha a(x, z) + \beta a(y, z),$
(ii) $a(x, \alpha y + \beta z) = \alpha a(x, y) + \beta a(x, z).$

Each bilinear form also generates an associated *quadratic form* $q : V \to R$,

$$q(x) = a(x, x), \quad \forall x \in V.$$

Norm

To measure the size of vectors in a general vector space V we introduce *norms*. A norm is a functional $\|\cdot\| : V \to R$ which satisfies the following conditions for all $x, y \in V$ and $\alpha \in R$,

(i) $\|x\| \geq 0$,
(ii) $\|\alpha x\| = |\alpha| \|x\|$,
(iii) $\|x + y\| \leq \|x\| + \|y\|$,
(iv) $\|x\| = 0 \Leftrightarrow x = 0$,

where (iii) is the *triangle inequality*. If only conditions (i)-(iii) are satisfied, the function is a *seminorm*. A *normed vector space* is a vector space on which a norm is defined.

Metric

A *metric* on a vector space V is a function $d : V \times V \to [0, \infty)$ which measures the distance between two vectors in V, defined by

(i) $d(x, y) \geq 0$,
(ii) $d(x, y) = 0 \Leftrightarrow x = y$,
(iii) $d(x, y) = d(y, x)$,
(iv) $d(x, z) \leq d(x, y) + d(y, z)$,

for all $x, y, z \in V$. If $d(x, y) = d(x + z, y + z)$ the metric is *translation invariant*. In a normed vector space a translation invariant metric can be defined by

$$d(x, y) = \|x - y\|.$$

Inner product

A bilinear form $(\cdot, \cdot) : V \times V \to R$ is an inner product if it is *symmetric* and its associated quadratic form is *positive definite*,

(i) $(x, y) = (y, x)$,

(ii) $(x, x) \geq 0$,

(iii) $(x, x) = 0 \Leftrightarrow x = 0$,

for all $x, y \in V$. An *inner product space* is a vector space on which an inner product is defined. Each inner product induces an associated norm through its quadratic form

$$\|x\| = (x, x)^{1/2}$$

Cauchy-Schwarz inequality

Theorem 1.5 (Cauchy-Schwarz inequality). *If $\|\cdot\|$ is the induced norm of the inner product $(\cdot,\cdot)$ in the inner product space V, then*

$$|(x,y)| \leq \|x\|\|y\|, \quad \forall x, y \in V.$$

Proof. Let $s \in R$ be an arbitrary parameter, then

$$0 \leq \|x+sy\|^2 = (x+sy, x+sy) = \|x\|^2 + 2s(x,y) + s^2\|y\|^2.$$

The right hand side defines a function $f(s) = \|x\|^2 + 2s(x,y) + s^2\|y\|^2$, which we seek to minimize to make the inequality as sharp as possible. It follows that we should set the parameter to the critical point $s = -(x,y)/\|y\|^2$ for which $f'(s) = 0$ and $f''(s) \geq 0$, hence, $f(s)$ minimal.

Euclidian inner product and norm

Example 1.6. R^n is an inner product space with the *Euclidian inner product*, also referred to as the *scalar product* or *dot product*, defined by

$$(x, y) = x \cdot y = x_1 y_1 + ... + x_n y_n,$$

for $x, y \in R^n$. The scalar product induces the l^2 norm $\|x\|_2 = (x, x)^{1/2}$, and we often drop the subscript for the l^2 norm in R^n with the understanding that $\|x\| = \|x\|_2$.

L^p norms

The l^2 norm is not the only way to measure a vector $x \in R^n$. In fact, we can define a whole family of related norms that we refer to as l^p norms. For $1 \leq p < \infty$, we define the l^p norm by

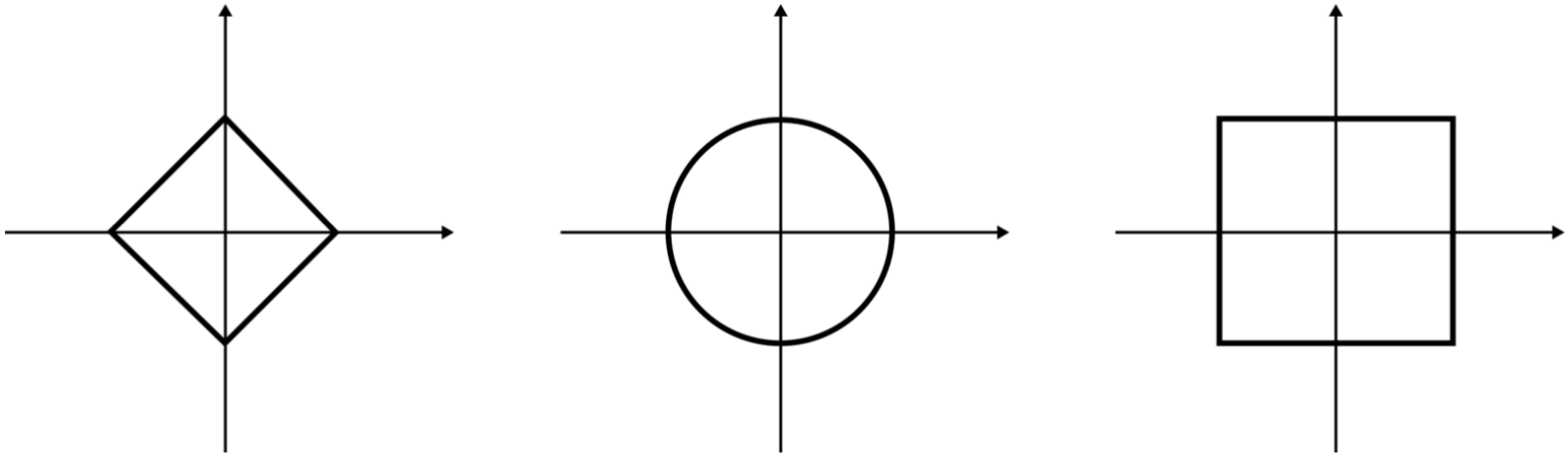
$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p},$$

and for $p = \infty$,

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|.$$

L^p norms

To illustrate the differences between the l^p norms, in Figure 1.3 we plot the corresponding *unit circles* in R^2 $\{x \in R^2 : \|x\|_p = 1\}$ for $p = 1, 2, \infty$.



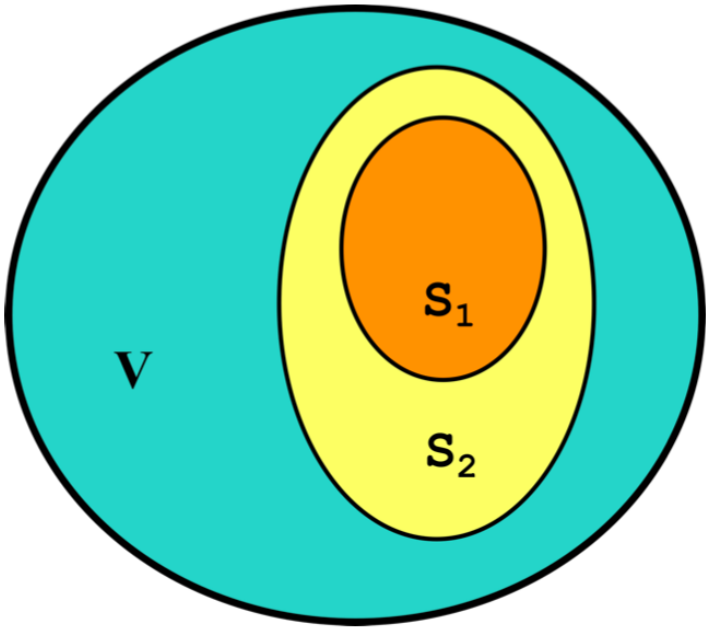
Subspace

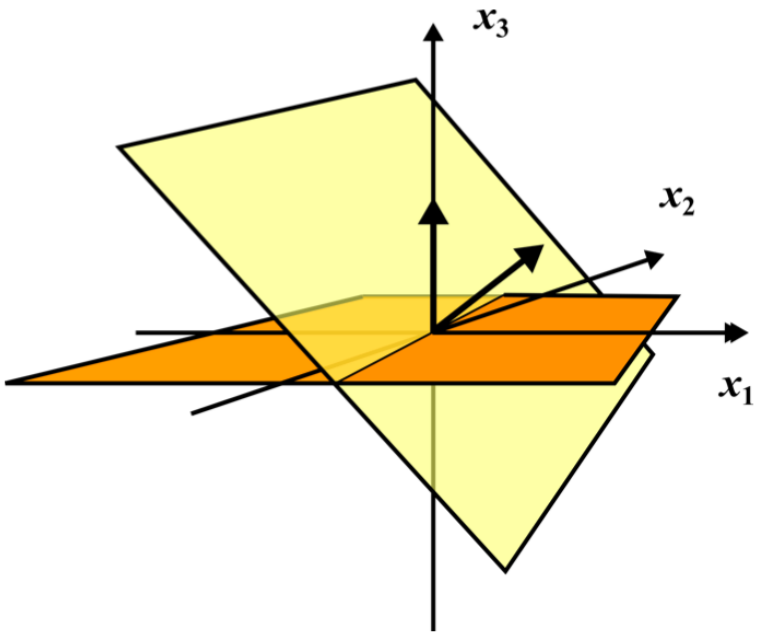
A *subspace* of a vector space V is a subset $S \subset V$ such that S together with the vector space operations in V define a vector space in its own right.

Example 1.11. In the Euclidian space R^3 a linear equation defines a plane, and the following planes define subspaces of R^3,

$$S_1 = \{x \in R^3 : x_3 = 0\},$$
$$S_2 = \{x \in R^3 : ax_1 + bx_2 + cx_3 = 0, \text{ for } a, b, c \in R\}.$$

Subspace





Linear combination

Now let $W = \{w_i\}_{i=1}^n$ be a finite set of vectors in V, then the sum

$$\sum_{i=1}^{n} \alpha_i w_i,$$

with scalar coefficients $\alpha_i \in R$, is referred to as a *linear combination* of the vectors $w_i \in W$.

All possible linear combinations of the vectors in W define a subspace

$$S = \{v \in V : v = \sum_{i=1}^{n} \alpha_i w_i,\ \alpha_i \in R,\ w_i \in W\},$$

the *linear span* or *linear hull* of W, and we say that the vector space S is *spanned* by W,

$$S = \text{span}(W) = \text{span}(\{w_i\}_{i=1}^n) = \langle w_1, ..., w_n \rangle.$$

Basis

$$B = \{b_i\}_{i=1}^{n}$$

is *linearly independent* if the only linear combination which is zero is the one where all scalar coefficients are zero, that is,

$$\sum_{i=1}^{n} \alpha_i b_i = 0 \Leftrightarrow \alpha_i = 0, \quad \forall i = 1, ..., n.$$

If the set B is linearly independent and all $v \in V$ can be expressed as a linear combination of the vectors in B,

$$v \in V \Leftrightarrow v = \sum_{i=1}^{n} \alpha_i b_i,$$

then B is a *basis* for V and $\alpha_i \in R$ are the *coordinates* of v with respect to the basis B. We denote the *dimension* of V by $\dim(V)$, defined as the number of vectors in the basis B.

Standard basis

Example 1.13 (The standard basis). The vector space R^n is spanned by the *standard basis*

$$\{e_1, ..., e_n\} = \{(1, 0, ..., 0)^T, ..., (0, ..., 0, 1)^T\}.$$

Therefore, $\dim(R^n) = n$ and any $x \in R^n$ can be expressed as

$$x = \sum_{i=1}^{n} x_i e_i.$$

We refer to the coordinates $x_i \in R$ as the *Cartesian coordinates*, and if nothing else is stated, the components of a vector $x \in R^n$ are the Cartesian coordinates of that vector.

Orthogonality

The inner product provides a method to extend the concept of measuring angles between vectors, from the Euclidian plane to a general inner product space V. Specifically, two vectors $x, y \in V$ are *orthogonal* if

$$(x, y) = 0.$$

If a vector $v \in V$ is orthogonal to all vectors s in a subspace $S \subset V$,

$$(v, s) = 0, \quad \forall s \in S,$$

then v is said to be orthogonal to S. We denote by $S^{\perp}$ the *orthogonal complement* of S in V,

$$S^{\perp} = \{v \in V : (v, s) = 0, \, \forall s \in S\}.$$

Orthogonality

For a finite dimensional vector space V, we have that $(S^{\perp})^{\perp} = S$, and any $v \in V$ can be decomposed into two orthogonal components $v = s_1 + s_2$, such that $s_1 \in S$ and $s_2 \in S^{\perp}$, which we can also write as

$$V = S \oplus S^{\perp}.$$

The only vector in V that is an element of both S and $S^{\perp}$ is the zero vector, and the dimension of $S^{\perp}$ is equal to the *codimension* of the subspace S in V,

$$\dim(S^{\perp}) = \dim(V) - \dim(S).$$

Analogous to a plane in R^3, we define a *hyperplane* in R^n as the orthogonal complement of the one dimensional vector space of *normal vectors* in R^n. Hence, the codimension of a hyperplane is one.

Orthogonal projection

The *orthogonal projection* of a vector x on another vector y is the scaled vector βy, with

$$\beta = (x, y)/\|y\|^2,$$

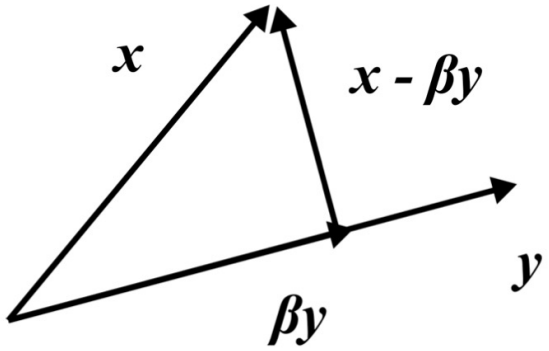
where $\|\cdot\| = (\cdot,\cdot)^{1/2}$ is the norm induced by the inner product in V. We note that the difference between the two vectors is orthogonal to y,

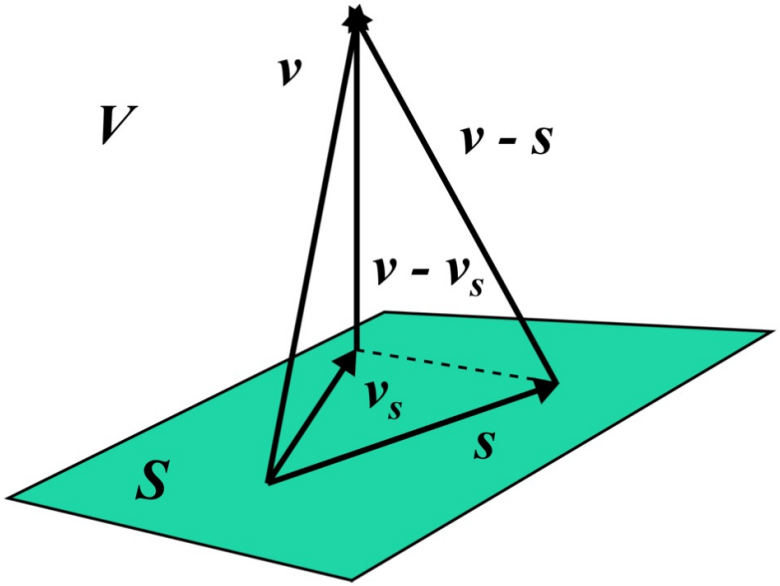
$$(x - \beta y, y) = 0.$$

Analogously, the orthogonal projection of a vector $v \in V$ on the subspace $S \subset V$ is a vector $v_s \in S$, such that

$$(v - v_s, s) = 0, \quad \forall s \in S.$$

Orthogonal projection





Orthogonal projection

Theorem 1.16 (Optimality of orthogonal projection). *The orthogonal projection $v_s \in S$, defined by*

$$(v - v_s, s) = 0, \quad \forall s \in S, \tag{1.7}$$

is the optimal approximation of $v \in V$ in $S \subset V$, in the sense that

$$\|v - v_s\| \leq \|v - s\|, \quad \forall s \in S, \tag{1.8}$$

for $\|\cdot\| = (\cdot,\cdot)^{1/2}$ the norm induced by the inner product in V.

Orthogonal projection

Proof. For any vector $s \in S$ we have that

$$\|v - v_s\|^2 = (v - v_s, v - v_s) = (v - v_s, v - s) + (v - v_s, s - v_s) = (v - v_s, v - s),$$

since $s - v_s \in S$, and therefore $(v - v_s, s - v_s) = 0$ by equation (1.7). The result then follows from Cauchy-Schwarz inequality and division by $\|v - v_s\|$,

$$\|v - v_s\|^2 = (v - v_s, v - s) \leq \|v - v_s\|\|v - s\| \Leftrightarrow \|v - v_s\| \leq \|v - s\|.$$

Outlook: Banach and Hilbert spaces

Now consider a normed vector space V with norm $\|\cdot\|_V$. A sequence $\{x_i\}_{i=1}^{\infty} \subset V$ is a *Cauchy sequence* if the distances between the vectors become arbitrary small as the indices increase,

$$\lim_{i\to\infty} \|x_j - x_i\|_V = 0,$$

for $j > i$. If each Cauchy sequence $\{x_i\}_{i=1}^{\infty}$ converges to a vector $x \in V$,

$$\lim_{i\to\infty} \|x - x_i\|_V = 0,$$

we say that V is *complete*, or that V is a *Banach space*. Analogously, a *Hilbert space* is a complete inner product space.

Outlook: Dual vector space

A linear functional $L : V \to R$ is *bounded* if it exists a finite real number $M > 0$, such that

$$|L(y)| \leq M\|y\|, \quad \forall y \in V.$$

The set of all bounded linear functionals L on V form a vector space V', the *dual vector space* of V, with vector addition and scalar multiplication defined pointwise for each $y \in V$, so that for $L_1, L_2 \in V'$ and $\alpha \in R$,

$$(L_1 + L_2)(y) = L_1(y) + L_2(y),$$
$$(\alpha L_1)(y) = \alpha L_1(y).$$

Outlook: Riesz representation theorem

If V is a Hilbert space with inner product $(\cdot, \cdot)$, then each $x \in V$ defines a linear functional

$$L_x(\cdot) = (x, \cdot),$$

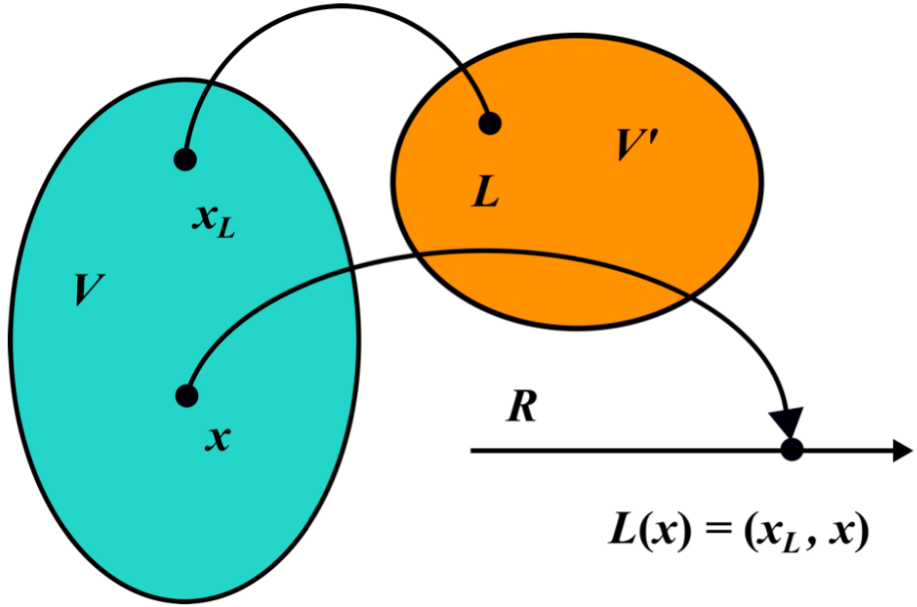
which is bounded with $M = \|x\|$ by the Cauchy-Schwarz inequality.

Theorem 1.20 (Riesz representation theorem). *For a Hilbert space V with norm $\|\cdot\|$ and inner product $(\cdot, \cdot)$, every bounded linear functional $L \in V'$ can be expressed as*

$$L(\cdot) = (x_L, \cdot)$$

for a unique $x_L \in V$, the Riesz representer of L, and $\|L\| = \sup_{\substack{y \in V \\ y \neq 0}} \frac{|L(y)|}{\|y\|} = \|x_L\|.$

Outlook: Riesz representation theorem



Linear transformations (chapter 2)

Linear transformations

A function $f : R^n \to R^m$ defines a *linear transformation* if

(i) $f(x+z) = f(x) + f(z)$,
(ii) $f(\alpha x) = \alpha f(x)$,

for all $x, z \in R^n$ and $\alpha \in R$. If $m = 1$ the function f is a linear functional, and if $m = n$ we refer to f as a *linear operator* on R^n. In the standard basis $\{e_1, ..., e_n\}$ we can express the ith component of the vector $y = f(x) \in R^m$ as

$$y_i = f_i(x) = f_i\left(\sum_{j=1}^{n} x_j e_j\right) = \sum_{j=1}^{n} x_j f_i(e_j),$$

where $f_i : R^n \to R$ for all $i = 1, ..., m$.

Matrix

$$
\begin{aligned}
y_1 &= a_{11}x_1 + \ldots + a_{1n}x_n \\
&\vdots \\
y_m &= a_{m1}x_1 + \ldots + a_{mn}x_n
\end{aligned}
$$

with $a_{ij} = f_i(e_j)$. Note that the coefficients depend on the specific basis used to express the vector x. Here we used the standard basis, but with another basis the coefficients would be different. We can also write (2.1) as $y = Ax$, where A is an $m \times n$ *matrix*,

$$
A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}.
$$

Matrix-vector product

The *matrix-vector product* Ax is an m-dimensional column vector $y = (y_i)$, where

$$y_i = \sum_{j=1}^{n} a_{ij} x_j, \quad i = 1, ..., m.$$

With $a_{:j}$ the jth column of A, an m-dimensional column vector, we can express the matrix-vector product as a linear combination of the set of column vectors $\{a_{:j}\}_{j=1}^{n}$,

$$y = Ax = \sum_{j=1}^{n} x_j a_{:j}$$

Matrix-vector product

$$y = Ax = \sum_{j=1}^{n} x_j a_{:j},$$

$$\left[\begin{array}{c} \\ y \\ \\ \end{array}\right] = \left[\begin{array}{c|c|c|c} & & & \\ a_{:1} & a_{:2} & \cdots & a_{:n} \\ & & & \end{array}\right] \left[\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_n \end{array}\right] = x_1 \left[\begin{array}{c} \\ a_{:1} \\ \\ \end{array}\right] + x_2 \left[\begin{array}{c} \\ a_{:2} \\ \\ \end{array}\right] + \ldots + x_n \left[\begin{array}{c} \\ a_{:n} \\ \\ \end{array}\right]$$

Fundamental theorem of linear algebra

Theorem 2.2 (Fundamental theorem of linear algebra, part I). *For an $m \times n$ matrix A, the following relations hold between the four fundamental subspaces,*

(i) $R^n = \textit{range}(A^T) \oplus \textit{null}(A)$,
(ii) $R^m = \textit{range}(A) \oplus \textit{null}(A^T)$,
(iii) $n = \textit{rank}(A^T) + \dim(\textit{null}(A))$,
(iv) $m = \textit{rank}(A) + \dim(\textit{null}(A^T))$.

$$\text{range}(A) = \{y \in R^m : y = Ax,\ x \in R^n\}$$
$$\text{null}(A) = \{x \in R^n : Ax = 0\}$$

Vector spaces of matrices

It is useful to also define vector spaces of matrices. The vector space $R^{m\times n}$ is defined as the set of real valued $m \times n$ matrices, together with the component-wise vector space operations

$$A+B=\begin{bmatrix} a_{11}+b_{11} & \cdots & a_{1n}+b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1}+b_{m1} & \cdots & a_{mn}+b_{mn} \end{bmatrix}, \quad \alpha A=\begin{bmatrix} \alpha a_{11} & \cdots & \alpha a_{1n} \\ \vdots & \ddots & \vdots \\ \alpha a_{m1} & \cdots & \alpha a_{mn} \end{bmatrix},$$

for all $A, B \in R^{m\times n}$ and $\alpha \in R$. Note that in the context of the vector space $R^{m\times n}$ the matrices are the vector elements, not to be confused with the column vectors of the vector space R^n.

$$\|A\|_p = \max_{\substack{x \in R^n \\ x \neq 0}} \frac{\|Ax\|_p}{\|x\|_p} = \max_{\substack{x \in R^n \\ \|x\|_p = 1}} \|Ax\|_p$$

Matrix-matrix product

The standard *matrix-matrix product* $B = AC$ is a matrix in $R^{m \times n}$, defined for two matrices $A \in R^{m \times l}$ and $C \in R^{l \times n}$ by

$$b_{ij} = \sum_{k=1}^{l} a_{ik} c_{kj},$$

$$\left[\begin{array}{c|c|c} & & \\ b_{:1} & \cdots & b_{:n} \\ & & \end{array}\right] = \left[\begin{array}{c|c|c} & & \\ a_{:1} & \cdots & a_{:l} \\ & & \end{array}\right] \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{l1} & \cdots & c_{ln} \end{bmatrix}$$

$$= \left[\begin{array}{c|c|c} & & \\ c_{11} \begin{bmatrix} a_{:1} \end{bmatrix} + \ldots + c_{l1} \begin{bmatrix} a_{:l} \end{bmatrix} & \cdots & c_{1n} \begin{bmatrix} a_{:1} \end{bmatrix} + \ldots + c_{ln} \begin{bmatrix} a_{:l} \end{bmatrix} \\ & & \end{array}\right]$$

Inner and outer product

$$v^T w = \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = v_1 w_1 + ... + v_n w_n = (v, w)$$

$$vw^T = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \begin{bmatrix} w_1 & \cdots & w_n \end{bmatrix} = \begin{bmatrix} v_1 w_1 & \cdots & v_1 w_n \\ \vdots & & \vdots \\ v_m w_1 & & v_m w_n \end{bmatrix}$$

Linear operators in R^n

A square matrix $A \in R^{n \times n}$ defines a linear operator, or *linear map*, on R^n,

$$x \mapsto Ax,$$

by the operations of matrix-vector product and scalar multiplication,

$$\begin{aligned} A(x+y) &= Ax + Ay, \quad & x, y \in R^n, \\ A(\alpha x) &= \alpha Ax, & x \in R^n, \alpha \in R. \end{aligned}$$

We distinguish between a linear map and an *affine map*, a linear map composed with a translation, corresponding to multiplication by a matrix A followed by addition of a vector $t \in R^n$,

$$x \mapsto Ax + t.$$

Linear operators in R^n

The adjoint matrix A^T also defines a linear operator on R^n, and the relation of the matrix A to its adjoint A^T constitutes an important characteristic. A square matrix A is said to be *normal* if $A^T A = AA^T$, *symmetric*, or *self-adjoint*, if $A^T = A$, and *skew-symmetric* if $A^T = -A$. Clearly a symmetric matrix A is normal, and if A also satisfies the following inequality

$$(Ax, x) > 0,$$

for all nonzero $x \in R^n$, we say that A is a *symmetric positive definite* matrix. The significance of a symmetric positive definite matrix A is that the bilinear form $a(x, y) = (Ax, y)$ defines an inner product in R^n, and hence the quadratic form $q(x) = a(x, x)$ a norm in R^n.

Linear operators in R^n

Example 2.3. The matrix

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

is symmetric positive definite, since for $x = (x_1, x_2) \neq 0$,

$$(Ax, x) = x^T Ax = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1^2 + x_2^2 + (x_1 - x_2)^2 > 0.$$

Linear operators in R^n

Example 2.4. The matrix

$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

is skew-symmetric, since $A^T = -A$, and normal, because

$$A^T A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = AA^T.$$

Linear operators in R^n

$$\operatorname{trace}(A) = \sum_{i=1}^{n} a_{ii}.$$

$$\det(A) = \sum_{j=1}^{n} a_{ij} C_{ij} = \sum_{i=1}^{n} a_{ij} C_{ij}, \quad C_{ij} = (-1)^{i+j} M_{ij},$$

where each *minor* M_{ij} is the determinant of the $(n-1) \times (n-1)$ matrix obtained by removing row i and column j from the matrix A. For a 3×3 matrix expanded in terms of the first row,

$$\det(A) = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

$$= a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31}).$$

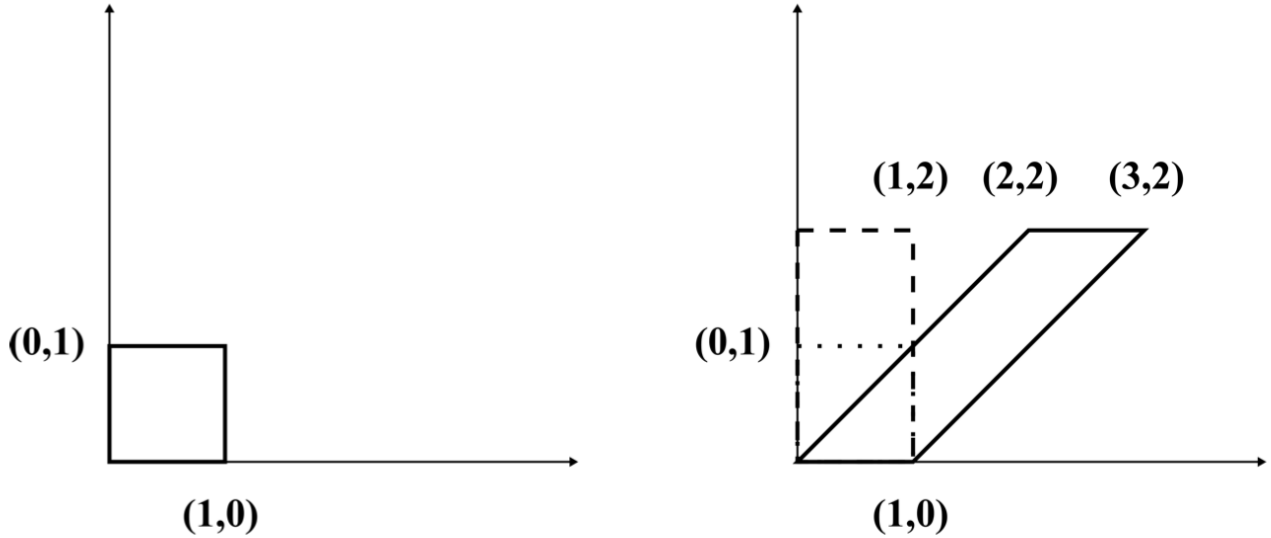
Matrix factorization

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix} \tag{2.5}$$

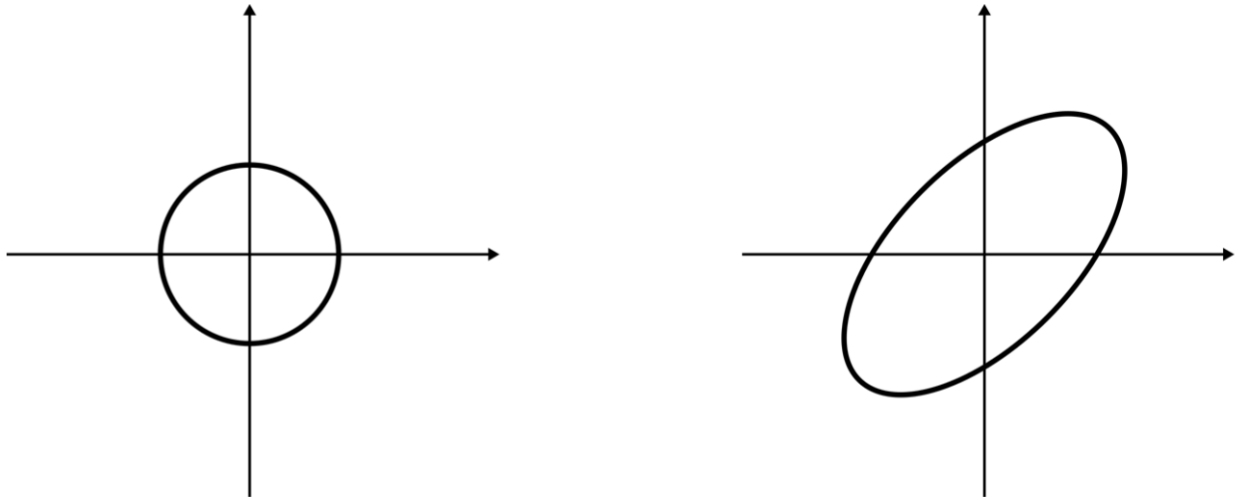
Example 2.7. The matrix (2.5) can be factorized into $A = A_2 A_1$, with $\det(A_1) = 2$ and $\det(A_2) = 1$, where A_1 is a stretch transformation and A_2 a shear transformation,

$$A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}. \tag{2.7}$$

Matrix factorization



Matrix factorization



System of linear equations

Theorem 2.8 (Inverse matrix). *A square matrix $A \in R^{n \times n}$ is said to be nonsingular if there exists an inverse matrix A^{-1}, and the following statements are equivalent:*

(i) *A has an inverse A^{-1},*
(ii) $\det(A) \neq 0$,
(iii) *rank*$(A) = n$,
(iv) *range*$(A) = R^n$,
(v) *null*$(A) = \{0\}$.

Specifically, a symmetric positive definite matrix A is nonsingular since null$(A) = \{0\}$.

Example 2.9 (System of linear equations). A system of linear equations can be stated as the matrix equation $Ax = b$, for A a given square matrix, b a given vector, and x the unknown solution. If A is nonsingular we obtain the solution as $x = A^{-1}b$.

Orthogonal matrix

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

It follows that a symmetric orthogonal matrix is its own inverse, and that $Q^T Q = I$, or in matrix form with $q_{:j}$ the columns of Q,

$$\left[\begin{array}{c} q_{:1}^T \\ \hline q_{:2}^T \\ \hline \vdots \\ \hline q_{:n}^T \end{array}\right] \left[\begin{array}{c|c|c|c} & & & \\ q_{:1} & q_{:2} & \cdots & q_{:n} \\ & & & \end{array}\right] = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}.$$

Hence, the columns $q_{:j}$ form an orthonormal basis since $(q_{:i}, q_{:j}) = \delta_{ij}$, with the *Kronecker delta function* defined by

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

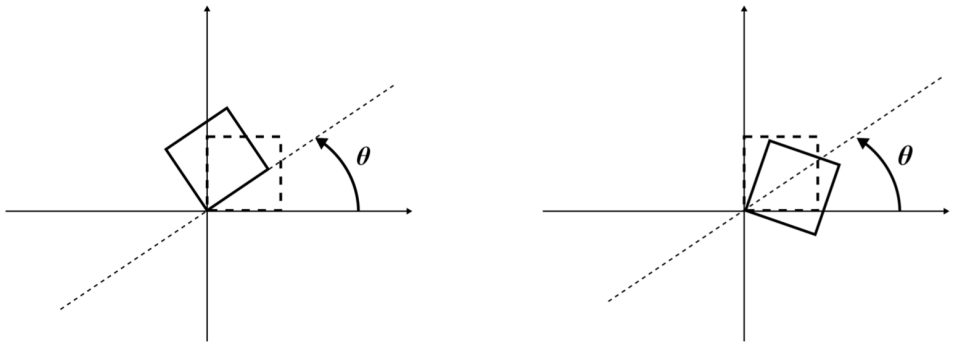
Orthogonal matrix

Multiplication by an orthogonal matrix preserves the inner product of two vectors $x, y \in R^n$,

$$(Qx, Qy) = (Qx)^T Qy = x^T Q^T Qy = x^T y = (x, y),$$

and therefore also the norm of a vector,

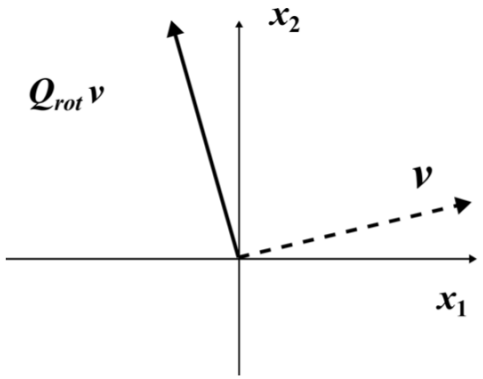
$$\|Qx\| = (Qx, Qx)^{1/2} = (x, x)^{1/2} = \|x\|.$$

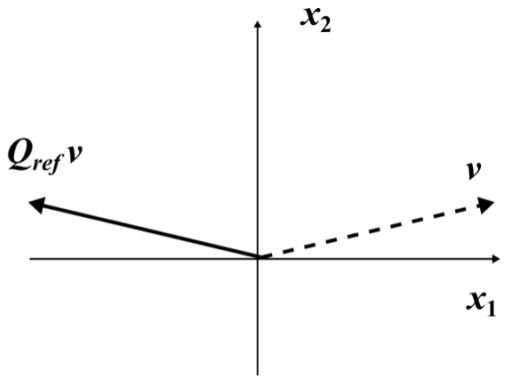


Givens rotation and reflection in R^2

$$Q_{rot}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$Q_{ref}(\theta) = \begin{bmatrix} \cos(2\theta) & \sin(2\theta) \\ \sin(2\theta) & -\cos(2\theta) \end{bmatrix}$$





Projectors

A *projection matrix*, or *projector*, is a square matrix P which is defined by the property that

$$P^2 = P,$$

where P^2 denotes the matrix-matrix product PP. Therefore, for all vectors $v \in \text{range}(P)$,

$$Pv = v,$$

since v is of the form $v = Px$ for some x, and hence $Pv = P^2x = Px = v$. From the definition of a projector, in the case $v \notin \text{range}(P)$, the projection error $v - Pv \in \text{null}(P)$, since

$$P(v - Pv) = Pv - P^2v = 0.$$

Projectors

It follows that the matrix $I - P$ is also a projector, the *complementary projector* to P, because

$$(I - P)^2 = I - 2P + P^2 = I - P,$$

and the ranges and nullspaces of the two projectors are related as

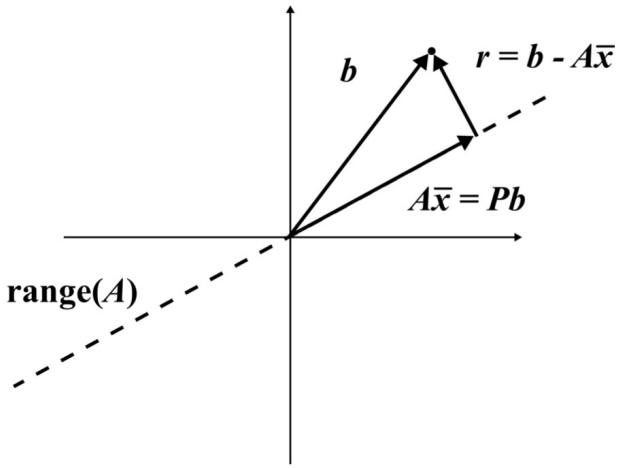
$$\text{range}(I - P) = \text{null}(P), \tag{2.15}$$
$$\text{range}(P) = \text{null}(I - P). \tag{2.16}$$

Least squares problem

Example 2.16 (Least squares problems). In a least squares problem we seek a solution to the system of linear equations defined by the matrix equation

$$Ax = b,$$

for a rectangular $m \times n$ matrix A with $m > n$, where $b \in R^m$ and $x \in R^n$. Hence, there are



Least squares problem

Theorem 2.2 (Fundamental theorem of linear algebra, part I). *For an $m \times n$ matrix A, the following relations hold between the four fundamental subspaces,*

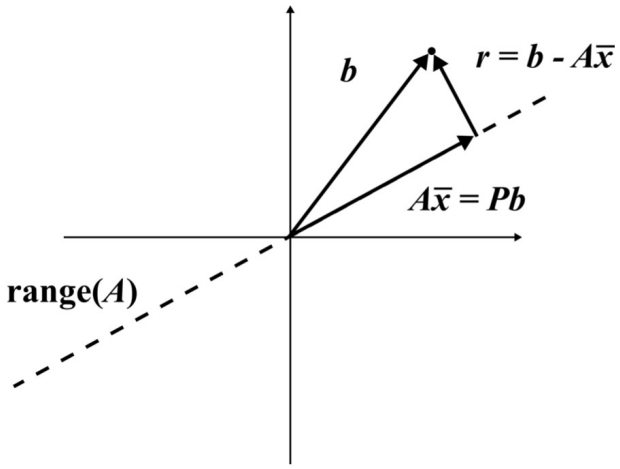
(i) $R^n = \textit{range}(A^T) \oplus \textit{null}(A)$,
(ii) $R^m = \textit{range}(A) \oplus \textit{null}(A^T)$,
(iii) $n = \textit{rank}(A^T) + \dim(\textit{null}(A))$,
(iv) $m = \textit{rank}(A) + \dim(\textit{null}(A^T))$.

$$\text{range}(A) = \{y \in R^m : y = Ax,\ x \in R^n\}$$
$$\text{null}(A) = \{x \in R^n : Ax = 0\}$$

Least squares problem

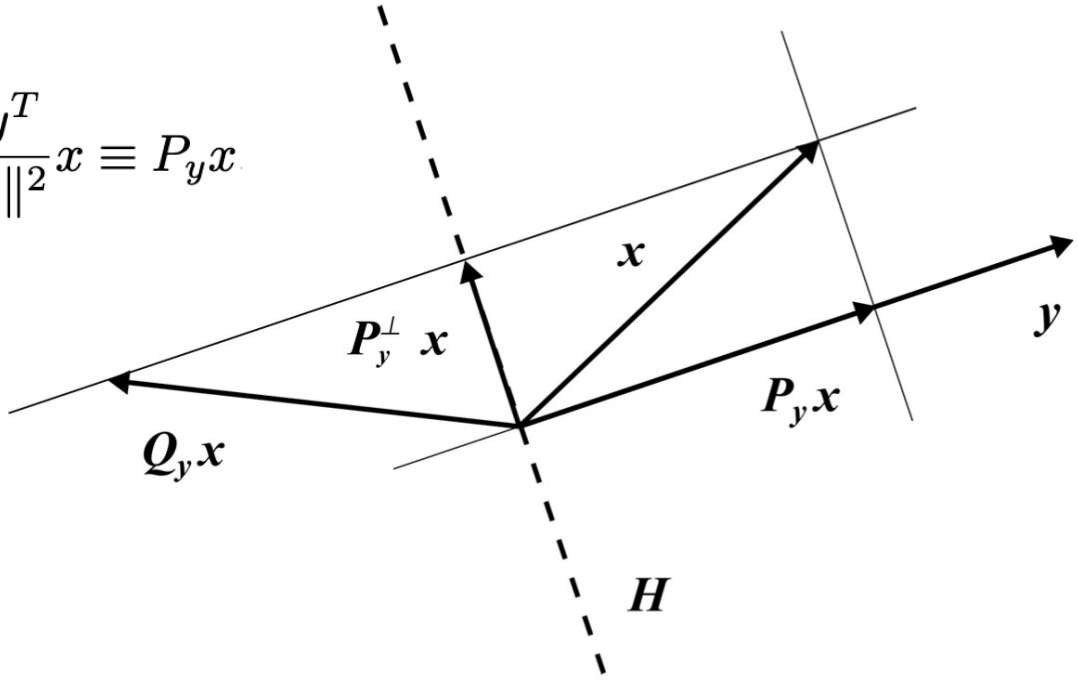
Now recall that $\text{range}(A)^{\perp} = \text{null}(A^T)$, so by Theorem 2.2 we have that $A^T(b - A\bar{x}) = 0$, which corresponds to the condition that the residual should be orthogonal to all column vectors of A with respect to the standard inner product in R^m. This condition is also referred to as the *normal equations*

$$A^T A\bar{x} = A^T b$$



Projection and hyperplane reflection

$$\frac{(x,y)}{\|y\|^2}y = \frac{y(y,x)}{\|y\|^2} = \frac{y(y^Tx)}{\|y\|^2} = \frac{yy^T}{\|y\|^2}x \equiv P_y x$$



If $\|y\| = 1$, then $P_y = yy^T$, $P_y^{\perp} = I - yy^T$ and $Q_y = I - 2yy^T$