14. Some applications

3-space

In the vector space $X = \mathbb{R}^3$, the standard inner product is also called the **dot product**, because of the customary notation

$$y^t x = \langle x, y\rangle =: x \cdot y, \qquad x, y \in \mathbb{R}^3.$$

In this most familiar vector space, another vector 'product' is of great use, the so-called **cross product** $x \times y$. It is most efficiently defined implicitly, i.e., by

$$(14.1) \qquad (x \times y) \cdot z := \det[x, y, z], \qquad \forall x, y, z \in RR^3.$$

From (13.13) (see also page 163), we work out that

$$\det[x, y, z] = (x(2)y(3) - x(3)y(2))z(1) + (x(3)y(1) - x(1)y(3))z(2) + (x(1)y(2) - x(2)y(1))z(3),$$

hence

$$x \times y = (x(2)y(3) - x(3)y(2), x(3)y(1) - x(1)y(3), x(1)y(2) - x(2)y(1)).$$

Given what you already know about determinants, the definition (14.1), though implicit, makes all the basic facts about the cross product immediate:

(i) *The cross product $x \times y$ is linear in its two arguments, x and y.*

(ii) *The cross product $x \times y$ is* **alternating**, *meaning that $y \times x = -(x \times y)$.*

(iii) Perhaps most importantly, *$x \times y$ is a vector perpendicular to both x and y.*

(iv) *$x \times y = 0$ if and only if $[x, y]$ is not 1-1.*

Indeed, if $[x, y]$ is 1-1, then we can always extend it to a basis $[x, y, z]$ for $\mathbb{R}^3$, and then $(x \times y)^t z$ is not zero, hence then $x \times y \neq 0$. If $[x, y]$ fails to be 1-1, then, for any z, $[x, y, z]$ fails to be 1-1, hence then, necessarily, $x \times y = 0$.

So, assuming that $[x, y]$ is 1-1, we can compute the *unit* vector

$$u := (x \times y)/\|x \times y\|,$$

and so conclude that

$$\|x \times y\|_2^2 = \det[x, y, x \times y] = \|x \times y\| \det[x, y, u].$$

In other words,

(v) *the Euclidean length of $x \times y$ gives the (unsigned) area of the parallelepiped spanned by x and y.*

This also holds when $[x, y]$ fails to be 1-1 since then that area is zero.

When $[x, y]$ is 1-1, then there are exactly two unit vectors (or, directions) perpendicular to the plane $\operatorname{ran}[x, y]$ spanned by x and y, namely $u := (x \times y)/\|x \times y\|$ and $(y \times x)/\|y \times x\| = -u$, with u the choice that makes $\det(x, y, u)$ positive. If you imagine the thumb of your *right* hand to be x, and the pointer of that hand to be y, then the middle finger, bent to be perpendicular to both thumb and pointer, would be pointing in the direction of $x \times y$. For that reason, any basis $[x, y, z]$ for $\mathbb{R}^3$ with $\det[x, y, z] > 0$ is said to be **right-handed**.

14.1 Relate the standard choice $(x_2, -x_1)$ for a vector perpendicular to the 2-vector x to the above construction.

14.2 Give a formula for an n-vector $x_1 \times \cdots \times x_{n-1}$ that is perpendicular to the $n-1$ n-vectors $x_1, \ldots, x_{n-1}$ and whose Euclidean length equals the (unsigned) volume of the parallelepiped spanned by the vectors $x_1, \ldots, x_{n-1}$.

Rotation in 3-space

A particularly useful transformation of 3-space is counter-clockwise rotation by some angle θ around some given axis-vector a. Let $R = R_{\theta,a}$ be this rotation. We are looking for a computationally efficient way to represent this map.

This rotation leaves its **axis**, i.e., $\mathrm{ran}[a]$, pointwise fixed, and rotates any vector in the plane $H := a^\perp$ counterclockwise θ radians. In other words, with

$$p = q + r, \quad q := P_{[a]}p, \quad r := p - q,$$

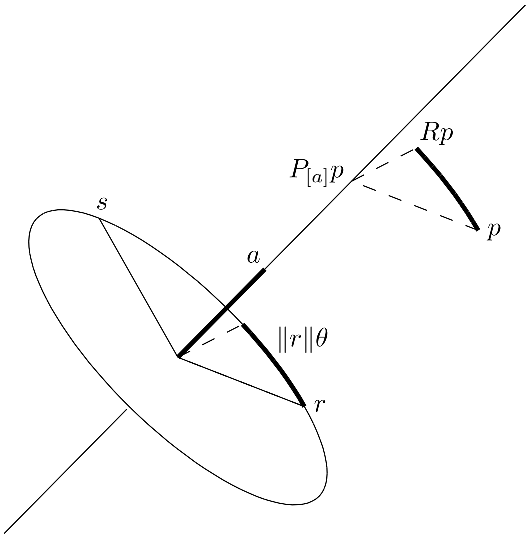
we have

$$Rp = q + Rr,$$

by the linearity of the rotation. To compute Rr, let s be the vector in H obtained by rotating r counterclockwise $\pi/2$ radians. Then

$$Rr = \cos(\theta)r + \sin(\theta)s,$$

and that's it.



(14.2) Figure. Rotation of the point p counterclockwise θ radians around the axis spanned by the vector a. The orthogonal projection r of p into the plane H with normal a, together with its rotation s counterclockwise $\pi/2$ radians around that axis, serve as a convenient orthogonal coordinate system in H.

It remains to construct s, and this is traditionally done with the aid of the **cross product**

$$a \times r := (a_2r_3 - a_3r_2, a_3r_1 - a_1r_3, a_1r_2 - a_2r_1)$$

of a with r. Check the chapter on determinants for the source of this vector. It is easy to verify directly that $a \times r$ is a vector perpendicular to a, hence also to r since, after all, $r \times a = -(a \times r)$, and that, for $a \perp r$, we have $\|a \times r\|^2 = \|a\|^2\|r\|^2$. Hence, assuming without loss that a is normalized, we now know that $a \times r$ is in the plane H and perpendicular to r and of the same length as r. Of the two vectors in H that have this property, it also happens to be the one obtained from r by a $(\pi/2)$-rotation that appears counterclockwise when looking down on H from the side that the vector a points into. (Just try it out.)

The calculations can be further simplified. The map

$$r \mapsto a \times r$$

is linear and, by inspection, $a \times a = 0$. Since a is normalized by assumption, we compute

$$r = p - (a^{\mathrm{t}} p)a,$$

hence

$$a \times r = a \times p.$$

So, altogether

$$Rp = (a^{\mathrm{t}} p)a + \cos(\theta)(p - (a^{\mathrm{t}} p)a) + \sin(\theta)(a \times p) = \cos(\theta)p + (1 - \cos(\theta))(a^{\mathrm{t}} p)a + \sin(\theta)(a \times p).$$

This is the formula that is most efficient for the calculation of Rp. However, if the matrix for $R = R\,\mathrm{id}_3$ (with respect to the natural basis) is wanted, we read it off as

$$R = \cos(\theta)\,\mathrm{id}_3 + (1 - \cos(\theta))[a][a]^{\mathrm{t}} + \sin(\theta)(a\times),$$

with

$$a\times := \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

the matrix for the linear map $r \mapsto a \times r$.

Markov Chains

Recall our example of a random walk on some graph. There we were interested in the matrices M^k, $k = 1, 2, 3, \ldots$, with the entries of the square matrix M all nonnegative and all entries in any particular row adding up to 1. In other words, $M \geq 0$ and $Me = e$, with

$$e := (1, 1, \ldots, 1).$$

In particular, $1 \in \mathrm{spec}(M)$. Further, since $\|M\|_\infty = 1$, we conclude from (13.1) that $\rho(M) \leq 1$. Hence, 1 is an absolutely largest eigenvalue for M. Assume, in addition, that M is irreducible. This is certainly guaranteed if $M > 0$. Then, by the Perron-Frobenius theory, 1 is a nondefective eigenvalue of M, and is the unique absolutely largest eigenvalue. By (11.10)Theorem, this implies that M is convergent. In fact, since 1 is a nondefective simple eigenvalue of M with corresponding eigenvector e, there is a basis $V = [e, W]$, with W a basis for $\mathrm{ran}(M - \mathrm{id})$, hence

$$MV = [e, MW] = V\,\mathrm{diag}(1, B)$$

for some B with $\rho(B) < 1$. Therefore,

$$M^k V = V\,\mathrm{diag}(1, B^k) \xrightarrow[k\to\infty]{} V\,\mathrm{diag}(1, 0).$$

In other words,

$$\lim_{k\to\infty} M^k = eu^{\mathrm{t}},$$

with $M^{\mathrm{t}} u = u$, i.e., u is an eigenvector of M^{t} belonging to the eigenvalue 1. In particular, all rows of M^k converge to this particular nonnegative vector whose entries sum to 1.

An example from CAGD

In Computer-Aided Geometric Design, one uses repeated corner-cutting to refine a given polygon into a smooth curve of approximately the same shape. The best-known example is the **Chaikin algorithm**. This algorithm consists in applying repeatedly, until satisfied, the following step:

input: the vertices $x_1, x_2, \ldots, x_n, x_{n+1} := x_1 \in \mathbb{R}^2$ of a closed polygon.
for $j = 1 : n$**, do:** $y_{2j-1} \leftarrow (3x_j + x_{j+1})/4$; $y_{2j} \leftarrow (x_j + 3x_{j+1})/4$; **enddo**

output: the vertices $y_1, y_2, \ldots, y_{2n}, y_{2n+1} := y_1 \in \mathbb{R}^2$ of a closed polygon that is inscribed into the input polygon.

In other words,

$$[y_1, \ldots, y_{2n}] = [x_1, \ldots, x_n]C_n,$$

with C_n the $n \times (2n)$-matrix

$$C_n := \begin{bmatrix} 3 & 1 & 0 & 0 & 0 & 0 & \cdots & 1 & 3 \\ 1 & 3 & 3 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 3 & 3 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 3 & 1 \end{bmatrix} /4.$$

It is possible to show that, as $k \to \infty$, the polygon with vertex sequence

$$[x_1^{(k)}, \ldots, x_{2^k n}^{(k)}] := [x_1, \ldots, x_n]C_n C_{2n} \cdots C_{2^k n}$$

converges to a smooth curve, namely the curve

$$t \mapsto \sum_j x_j B_2(t - j),$$

with B_2 a certain smooth piecewise quadratic function, a so-called quadratic B-spline (whatever that may be).

Here, we consider the following much simpler and more radical corner-cutting:

$$[y_1, \ldots, y_n] = [x_1, \ldots, x_n]A,$$

with

$$(14.3) \qquad A := \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 1 \end{bmatrix} /2.$$

In other words, the new polygon is obtained from the old by choosing as the new vertices the midpoints of the edges of the old.

Simple examples, hand-drawn, quickly indicate that the sequence of polygons, with vertex sequence

$$[x_1^{(k)}, \ldots, x_n^{(k)}] := [x_1, \ldots, x_n]A^k$$

seem to shrink eventually into a point. Here is the analysis that this is, in fact, the case, with that limiting point equal to the average, $\sum_j x_j/n$, of the original vertices.

(i) The matrix A, defined in (14.3), is a **circulant**, meaning that each row is obtainable from its predecessor by shifting everything one to the right, with the right-most entry in the previous row becoming the left-most entry of the current row. All such matrices have eigenvectors of the form

$$u_\lambda := (\lambda^1, \lambda^2, \ldots, \lambda^n),$$

with the scalar λ chosen so that $\lambda^n = 1$, hence $\lambda^{n+1} = \lambda$. For our A, we compute

$$Au_\lambda = (\lambda^n + \lambda^1, \lambda^1 + \lambda^2, \ldots, \lambda^{n-1} + \lambda^n)/2.$$

Hence, if $\lambda^n = 1$, then

$$Au_\lambda = \frac{1+\lambda}{2\lambda} u_\lambda.$$

(ii) The equation $\lambda^n = 1$ has exactly n distinct solutions, namely the n **roots of unity**

$$\lambda_j := \exp(2\pi \mathrm{i} j/n) = \omega^j, \quad j = 1{:}n.$$

Here,

$$\omega := \omega_n := \exp(2\pi\mathrm{i}/n)$$

is a **primitive nth root of unity**. Note that

$$\overline{\omega} = 1/\omega.$$

Let

$$V = [v_1, \ldots, v_n] := [u_{\lambda_1}, \ldots, u_{\lambda_n}]$$

be the column map whose jth column is the eigenvector

$$v_j := (\omega^j, \omega^{2j}, \ldots, \omega^{nj})$$

of A, with corresponding eigenvalue

$$\mu_j := \frac{1+\lambda_j}{2\lambda_j} = (\omega^{-j} + 1)/2, \quad j = 1{:}n.$$

Since these eigenvalues are distinct, V is 1-1 (by (10.8)Lemma), hence V is a basis for $\mathbb{C}^n$. In particular,

$$A = V \operatorname{diag}(\ldots, \mu_j, \ldots) V^{-1}.$$

(iii) It follows that

$$A^k = V \operatorname{diag}(\ldots, \mu_j^k, \ldots) V^{-1} \xrightarrow[k\to\infty]{} V \operatorname{diag}(0, \ldots, 0, 1) V^{-1}$$

since $|\mu_j| < 1$ for $j < n$, while $\mu_n = 1$. Hence

$$\lim_{k\to\infty} A^k = v_n V^{-1}(n, :).$$

(iv) In order to compute $V^{-1}(n, :)$, we compute $V^{\mathrm{c}} V$ (recalling that $\overline{\omega^r} = \omega^{-1}$):

$$(V^{\mathrm{c}} V)(j,k) = {v_j}^{\mathrm{c}} v_k = \sum_{r=1}^{n} \omega^{-rj}\, \omega^{rk} = \sum_{r=1}^{n} \omega^{(k-j)r}.$$

That last sum is a geometric series, of the form $\sum_{r=1}^{n} \nu^r$ with $\nu := \omega^{k-j}$, hence equals n in case $k = j$, and otherwise $\nu \neq 1$ and the sum equals $(\nu^{n+1} - \nu)/(\nu - 1) = 0$ since $\nu^n = 1$, hence $\nu^{n+1} - \nu = 0$. It follows that

$$V^{\mathrm{c}} V = n \,\mathrm{id}_n,$$

i.e., $V/\sqrt{n}$ is *unitary*, i.e., an o.n. basis for $\mathbb{C}^n$. In particular, $V^{-1} = V^{\mathrm{c}}/n$, therefore

$$V^{-1}(n,:) = {v_n}^{\mathrm{c}}/n.$$

(v) It follows that

$$\lim_{k\to\infty} A^k = (1/n) v_n {v_n}^{\mathrm{c}},$$

with

$$v_n = (1, 1, \ldots, 1).$$

Consequently,

$$\lim_{k\to\infty} [\ldots, x_j^{(k)}, \ldots] \;=\; \sum_j x_j/n \; {v_n}^{\mathrm{c}} \;=\; [\ldots, \sum_j x_j/n, \ldots],$$

i.e., the rank-one matrix all of whose columns equal the average $\sum_j x_j/n$ of the vertices of the polygon we started out with.

Tridiagonal Toeplitz matrix

Circulants are a special case of *Toeplitz* matrices, i.e., of matrices that are constant along diagonals. Precisely, the matrix A is **Toeplitz** if

$$A(i,j) = a_{i-j}, \quad \forall i, j,$$

for some sequence $(\ldots, a_{-2}, a_{-1}, a_0, a_1, a_2, \ldots)$ of appropriate domain. Circulants are special in that the determining sequence a for them is periodic, i.e., $a_{i+n} = a_i$, all i, if A is of order n.

Consider now the case of a **tridiagonal** Toeplitz matrix A. For such a matrix, only the (main) diagonal and the two next-to-main diagonals are (perhaps) nonzero; all other entries are zero. This means that only a_{-1}, a_0, a_1 are, perhaps, nonzero, while $a_i = 0$ for $|i| > 1$. If also $a_{-1} = a_1 \neq 0$, then the circulant trick, employed in the preceding section, still works, i.e., we can fashion some eigenvectors from vectors of the form $u_\lambda = (\lambda^1, \ldots, \lambda^n)$. Indeed, now

$$(Au_\lambda)_j = \begin{cases} a_0\lambda + a_1\lambda^2 & \text{for } j = 1; \\ a_1\lambda^{j-1} + a_0\lambda^j + a_1\lambda^{j+1} & \text{for } j = 2{:}n{-}1; \\ a_1\lambda^{n-1} + a_0\lambda^n & \text{for } j = n. \end{cases}$$

Hence,

$$Au_\lambda = (a_1/\lambda + a_0 + a_1\lambda) u_\lambda - a_1(e_1 + \lambda^{n+1} e_n).$$

At first glance, this doesn't look too hopeful since we are after eigenvectors. However, recall that, for a unimodular λ, i.e., for $\lambda = \exp \mathrm{i}\varphi$ for some real φ, we have $1/\lambda = \overline{\lambda}$, hence

$$Au_{\overline{\lambda}} = (a_1/\lambda + a_0 + a_1\lambda) u_{\overline{\lambda}} - a_1(e_1 + \overline{\lambda^{n+1}} e_n).$$

It follows that, by choosing λ as an $(n+1)$st root of unity, i.e.,

$$\lambda = \lambda_j := \exp(2\pi\mathrm{i}j/(n+1)), \quad j = 1{:}n,$$

and setting

$$v_j := (u_\lambda - u_{\overline{\lambda}})/(2\mathrm{i}) = (\sin(2\pi kj/(n+1)) : k = 1{:}n),$$

we obtain

$$Av_j = \mu_j v_j$$

with

$$\mu_j := a_0 + a_1(\lambda_j + \overline{\lambda_j}) = a_0 + 2a_1 \cos(2\pi j/(n+1)).$$

Since we assumed that $a_1 \neq 0$, these n numbers μ_j are pairwise distinct, hence $V =: [v_1, \ldots, v_n]$ is 1-1 by (10.8)Lemma, hence a basis for $\mathbb{C}^n$. In fact, since V maps $\mathbb{R}^n$ to $\mathbb{R}^n$, V is a basis for $\mathbb{R}^n$. Hence if both a_0 and a_1 are real, then also each μ_j is real and then, A is diagonable even over $\mathbb{F} = \mathbb{R}$.

Linear Programming

In Linear Programming, one seeks a minimizer for a *linear* **cost function**

$$x \mapsto c^{\mathrm{t}}x$$

on the set

$$F := \{x \in \mathrm{I\!R}^n : Ax \le b\}$$

of all n-vectors x that satisfy the m **linear constraint**s

$$A(i,:)^{\mathrm{t}}x \le b_i, \quad i = 1{:}m,$$

with $c \in \mathrm{I\!R}^n$, $A \in \mathrm{I\!F}^{m\times n}$, $b \in \mathrm{I\!R}^m$ given. Here and below, for $y, z \in \mathrm{I\!R}^m$,

$$y \le z \;:=\; z - y \in \mathrm{I\!R}^m_+ := \{u \in \mathrm{I\!R}^m : 0 \le u_j, j = 1{:}m\}.$$

The set F, also called the **feasible set**, is the intersection of m halfspaces, i.e., sets of the form

$$H(a,b) := \{x \in \mathrm{I\!R}^n : a^{\mathrm{t}}x \le b\}.$$

Such a **halfspace** consists of all the points that lie on that side of the corresponding **hyperplane**

$$h(a,b) := \{x \in \mathrm{I\!R}^n : a^{\mathrm{t}}x = b\}$$

that the **normal** a of the hyperplane points away from; see (2.4)Figure, or (14.4)Figure.

Here is a simple example: Minimize

$$2x_1 + x_2$$

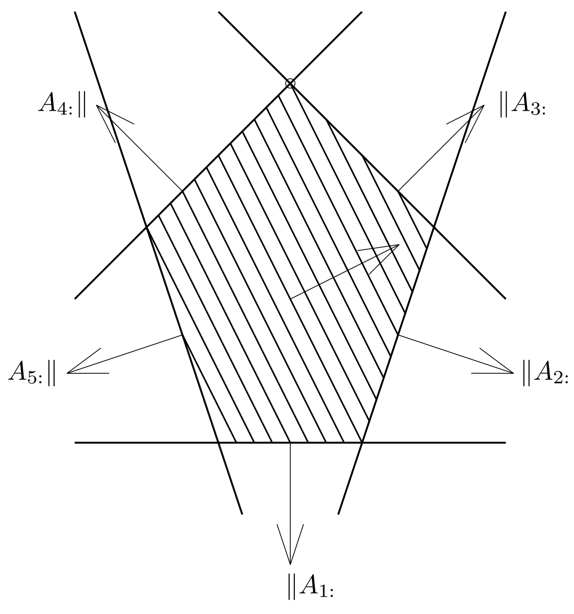
over all $x \in \mathrm{I\!R}^2$ for which

$$x_2 \ge -2, \quad 3x_1 - x_2 \le 5, \quad x_1 + x_2 \le 3$$
$$x_1 - x_2 \ge -3, \quad 3x_1 + x_2 \ge -5.$$

In matrix notation, and more uniformly written, this is the set of all $x \in \mathrm{I\!R}^2$ for which $Ax \le b$ with

$$A := \begin{bmatrix} 0 & -1 \\ 3 & -1 \\ 1 & 1 \\ -1 & 1 \\ -3 & -1 \end{bmatrix}, \quad b := \begin{bmatrix} 2 \\ 5 \\ 3 \\ 3 \\ 5 \end{bmatrix}.$$

In this simple setting, you can visualize the set F by drawing each of the hyperplanes $h(A_{i:}, b_i)$ along with a vector pointing in the same direction as its normal vector, $A_{i:}$; the set F lies on the side that the normal vector points away from; see (14.4)Figure.



(14.4) Figure. The feasible set for five linear constraints in the plane, as filled out by some level lines of the cost function. Since the gradient of the cost function is shown as well, the location of the minimizer is clear.

□

In order to provide a handier description for F, one introduces the so-called **slack variables**

$$y := b - Ax;$$

earlier, we called this the *residual*. With their aid, we can describe F as

$$F = \{x \in \mathbb{R}^n : \exists y \in \mathbb{R}^m_+ \text{ s.t. } (x, y, -1) \in \operatorname{null}[A,\ \mathrm{id}_m, b]\},$$

and use elimination to obtain a concise description of $\operatorname{null}[A,\ \mathrm{id}_m, b]$.

For this, assume that A is 1-1. Then, each column of A is bound, hence is also bound in $[A,\ \mathrm{id}, b]$. Therefore, after n steps of the (3.2)Elimination Algorithm applied to $[A,\ \mathrm{id}, b]$, we will arrive at a matrix B, with the same nullspace as $[A,\ \mathrm{id}, b]$, and an n-vector `nbs` (with $\mathtt{nbs}(k)$ the row used as pivot row for the kth unknown or column, all k), for which

$$B(\mathtt{nbs}, 1{:}n)$$

is upper triangular with nonzero diagonals while, with `bs` the rows not yet used as pivot rows,

$$B(\mathtt{bs}, 1{:}n) = 0.$$

Further, since the next m columns of $[A,\ \mathrm{id}_m, b]$ have nonzero entries in these pivot rows `nbs` only in columns $n + \mathtt{nbs}$, the other columns, i.e., columns $n + \mathtt{bs}$, will remain entirely unchanged. It follows that

$$B(\mathtt{bs}, n + \mathtt{bs}) = \mathrm{id}_{m-n}.$$

Therefore, after dividing each of the n pivot rows by their pivot element and then using each pivot row to eliminate its unknown also from all other pivot rows, we will arrive at a matrix, still called B, for which now

$$B([\mathtt{nbs}, \mathtt{bs}], \mathtt{bound}) = \mathrm{id}_m$$

with

$$\mathtt{bound} := [1{:}n, n + \mathtt{bs}]$$

the bound columns of $[A,\ \mathrm{id}, b]$.

For our particular example, the matrix B will be reached after just two steps:

$$[A,\ \mathrm{id}, b] = \begin{bmatrix} 0 & -1 & 1 & 0 & 0 & 0 & 0 & 2 \\ 3 & -1 & 0 & 1 & 0 & 0 & 0 & 5 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 3 \\ -1 & 1 & 0 & 0 & 0 & 1 & 0 & 3 \\ -3 & -1 & 0 & 0 & 0 & 0 & 1 & 5 \end{bmatrix} \to \begin{bmatrix} 0 & -1 & 1 & 0 & 0 & 0 & 0 & 2 \\ 0 & -4 & 0 & 1 & -3 & 0 & 0 & -4 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 3 \\ 0 & 2 & 0 & 0 & 1 & 1 & 0 & 6 \\ 0 & 2 & 0 & 0 & 3 & 0 & 1 & 14 \end{bmatrix}$$

$$\to \begin{bmatrix} 0 & 0 & 1 & 0 & 1/2 & 1/2 & 0 & 5 \\ 0 & 0 & 0 & 1 & -1 & 2 & 0 & 8 \\ 1 & 0 & 0 & 0 & 1/2 & -1/2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1/2 & 1/2 & 0 & 3 \\ 0 & 0 & 0 & 0 & 2 & -1 & 1 & 8 \end{bmatrix} =: B,$$

with $\mathtt{nbs} = [3, 4]$, and $\mathtt{bs} = [1, 2, 5]$.

It follows that

$$\mathtt{free} := [n + \mathtt{nbs}, n + m + 1]$$

gives the free columns of $[A,\ \mathrm{id}, b]$. In particular, we can freely choose $y_{\mathtt{nbs}}$, i.e., the slack variables associated with the n pivot rows, and, once they are chosen, then x as well as the bound slack variables, $y_{\mathtt{bs}}$, are uniquely determined by the requirement that $(x, y, -1) \in \operatorname{null} B$.

This suggests eliminating x altogether, i.e., using the pivot rows $B(\mathtt{nbs}, :)$ to give

$$x = B(\mathtt{nbs}, \mathtt{end}) - B(\mathtt{nbs}, n + \mathtt{nbs})y_{\mathtt{nbs}},$$

(with `end` being `MATLAB`'s convenient notation for the final row or column index) and, with that, rewrite the cost function in terms of $y_{\mathtt{nbs}}$:

$$y_{\mathtt{nbs}} \mapsto c^{\mathrm{t}}B(\mathtt{nbs}, \mathtt{end}) - c^{\mathrm{t}}B(\mathtt{nbs}, n + \mathtt{nbs})y_{\mathtt{nbs}}.$$

Correspondingly, we simplify the working array B in the following two ways:

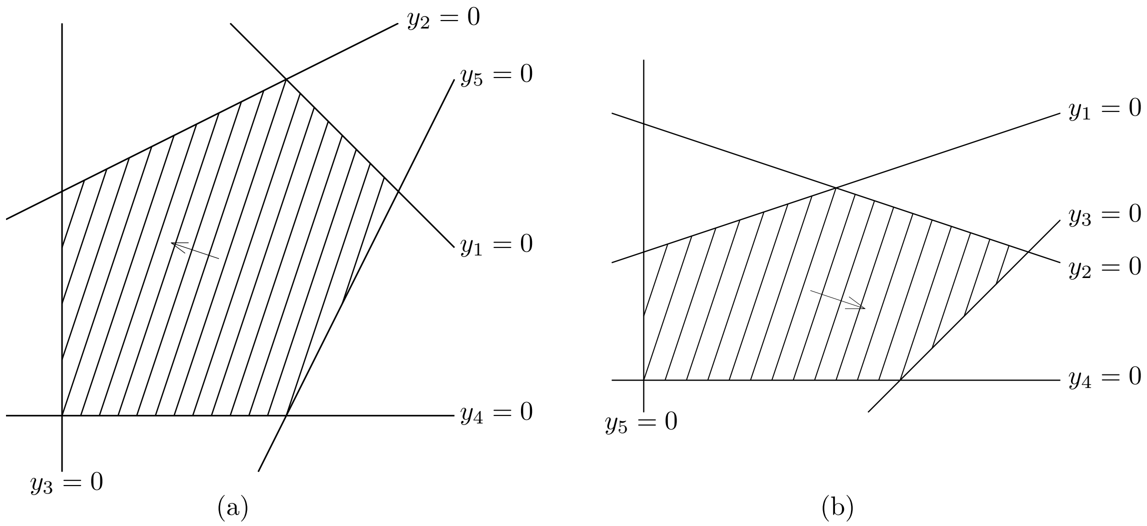
(i) We append the row $B(m + 1, :) := c^{\mathrm{t}}B(\mathtt{nbs}, :)$.

(ii) Then, we drop entirely the n rows `nbs` (storing those rows perhaps in some other place against the possibility that we need to compute x from $y_{\mathtt{nbs}}$ at some later date), and also drop the first n columns.

In our example, this leaves us with the following, smaller, array B:

$$B = \begin{bmatrix} 1 & 0 & 1/2 & 1/2 & 0 & 5 \\ 0 & 1 & -1 & 2 & 0 & 8 \\ 0 & 0 & 2 & -1 & 1 & 8 \\ 0 & 0 & 3/2 & -1/2 & 0 & 3 \end{bmatrix}, \quad \mathtt{bs} = [1, 2, 5], \quad \mathtt{nbs} = [3, 4].$$

This change of independent variables, from x to $y_{\mathtt{nbs}}$, turns the n hyperplanes $h(A_{k:}, b(k))$, $k \in \mathtt{nbs}$, into coordinate planes; see (14.5)Figure. In particular, the choice $y_{\mathtt{nbs}} = 0$ places us at the (unique) point of intersection of these n hyperplanes. In our example, that point is $x = (0, 3)$, and it is marked in (14.4)Figure, and functions as the origin in (14.5)Figure(a).



(14.5) Figure. The feasible set for five linear constraints in the plane, as filled out by some level lines of the cost function, viewed in terms of the (nonbasic) variables (a) y_3, y_4; (b) y_5, y_4. From the latter, the minimizing vertex will be reached in one step of the Simplex Method.

In terms of this B as just constructed, and with

$$m' := m - n = \#\mathtt{bs},$$

our minimization problem now reads: *Minimize the cost function*

$$y_{\texttt{nbs}} \mapsto B(\texttt{end},\texttt{end}) - B(\texttt{end},\texttt{nbs})^{\mathrm{t}} y_{\texttt{nbs}} \tag{14.6}$$

over all $y_{\texttt{nbs}} \in \mathbb{R}_+^n$ *for which*

$$B(\texttt{bs},\texttt{nbs})\, y_{\texttt{nbs}} \le B(\texttt{bs},\texttt{end}),$$

i.e., for which

$$y_{\texttt{bs}} := B(\texttt{bs},\texttt{end}) - B(\texttt{bs},\texttt{nbs}) y_{\texttt{nbs}} \in \mathbb{R}_+^{m'}.$$

This is the form in which linear programming problems are usually stated, and from which most textbooks start their discussion of such problems.

Note how easily accessible various relevant information now is.

(i) $B(\texttt{end},\texttt{end})$ tells us the value of the cost function at the current point, $y_{\texttt{nbs}} = 0$.

(ii) For any $k \in \texttt{nbs}$, the entry $B(\texttt{end}, k)$ tells us how the cost function would change if we were to change the value of the nonbasic variable y_k in the only way permitted, i.e., from 0 to something positive. Such a move would lower the cost function if and only if $B(\texttt{end}, k) > 0$.

(iii) Our current point, $y_{\texttt{nbs}} = 0$, is feasible if and only if $B(1{:}m', \texttt{end}) \ge 0$.

(iv) If we were to change the nonbasic variable y_k from zero to something positive, then the basic variable $y_{\texttt{bs}(i)}$ would change, from $B(i,\texttt{end})$ to $B(i,\texttt{end}) - B(i,k)y_k$. Hence, assuming $B(i,\texttt{end}) > 0$ and $B(i,k) > 0$, we could change y_k only to $B(i,\texttt{end})/B(i,k)$ before the $\texttt{bs}(i)$th constraint would be violated.

In our example (have a look at (14.5)Figure(a)), we already observed that our current point, $y_{\texttt{nbs}} = 0$, is, indeed, feasible. But we notice that, while $B(\texttt{end}, 4) < 0$, hence any feasible change of y_4 would only increase the cost function (14.6), $B(\texttt{end}, 3)$ is positive, hence know that we can further decrease the cost function (14.6) by increasing y_3. Such a change is limited by concerns for the positivity of y_1 and y_5. As for y_1, we would reach $y_1 = 0$ when $y_3 = 5/(1/2) = 10$, while, for y_5, we would reach $y_5 = 0$ when $y_3 = 8/2 = 4$. We take the smaller change and thereby end up at a new vector y, with $y_4 = 0 = y_5$, i.e., are now at the intersection of the constraints 4 and 5, with the cost further reduced by $(3/2)4 = 6$, to -3.

In other words, after this change, y_4 and y_5 are now the nonbasic variables. In order to have our B tell us about this new situation, and since $5 = \texttt{bs}(3)$, we merely divide its 3rd row by $B(3,3)$, then use the row to eliminate y_3 from all other rows of B. This leads to

$$B = \begin{bmatrix} 1 & 0 & 0 & 3/4 & -1/4 & 3 \\ 0 & 1 & 0 & 3/2 & 1/2 & 12 \\ 0 & 0 & 1 & -1/2 & 1/2 & 4 \\ 0 & 0 & 0 & 1/2 & -3/4 & -3 \end{bmatrix}, \quad \texttt{bs} = [1,2,3], \quad \texttt{nbs} = [5,4].$$

In particular, we readily see that the cost at $y_4 = 0 = y_5$ is, indeed, -3. We also see (see also (14.5)Figure(b)) that it is possible to reduce the cost further by changing y_4, from 0 to something positive. Such a change would reduce $y_1 = 3$ by $(3/4)y_4$ and would reduce $y_2 = 12$ by $(3/2)y_4$. Hence, this change is limited to the smaller of $3/(3/4) = 4$ and $12/(3/2) = 8$, i.e., to the change $y_4 = 4$ that makes $y_1 = 0$.

We carry out this exchange, of y_4 into $\texttt{bs}$ and y_1 into $\texttt{nbs}$, by dividing $B(1,:)$ by $B(1,4)$ and then using that row to eliminate y_4 from all other rows, to get the following B:

$$B = \begin{bmatrix} 4/3 & 0 & 0 & 1 & -1/3 & 4 \\ -2 & 1 & 0 & 0 & 1 & 6 \\ 2/3 & 0 & 1 & 0 & 1/3 & 6 \\ -1/3 & 0 & 0 & 0 & -2/3 & -4 \end{bmatrix}, \quad \texttt{bs} = [4,2,3], \quad \texttt{nbs} = [5,1].$$

In particular, now $B(\texttt{end},\texttt{nbs}) \le 0$, showing that no further improvement is possible, hence -4 is the minimum of the cost function on the feasible set. At this point, $y_{3:4} = (6,4)$, hence, from the rows used as pivot rows to eliminate x, we find that, in terms of x, our optimal point is $x = (0,3) - (1/2)\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}(6,4) = -(1,2)$, and, indeed, $c^{\mathrm{t}}x = (2,1)^{\mathrm{t}}(-1,-2) = -4$.

The steps just carried out for our example are the standard steps of the **Simplex Method**. In this method (as proposed by Dantzig), one examines the value of the cost function only at a **vertex**, i.e., at the unique intersection of n of the constraint hyperplanes, i.e., at a point corresponding to $y_{\texttt{nbs}} = 0$ for some choice of `nbs`. Assuming that vertex feasible, one checks whether $B(\texttt{end}, \texttt{nbs}) \le 0$. If it is, then one knows that one is at the minimum. Otherwise, one moves to a neighboring vertex at which the cost is less by exchanging some y_k for which $B(\texttt{end}, k) > 0$ with some $y_{\texttt{bs}(i)}$ with i chosen as the minimizer for $B(i, \texttt{end})/B(i, k)$ over all i with $B(i, k) > 0$. This exchange is carried out by just one full elimination step applied to B, by dividing $B(i, :)$ by $B(i, k)$ and then using this row to eliminate y_k from all other rows, and then updating the sequences `bs` and `nbs`.

This update step is one full elimination step (sometimes called a **(Gauss-)Jordan** step in order to distinguish it from the **Gauss** step, in which the unknown is eliminated only from the rows not yet used as pivot rows).

Since all the information contained in the columns $B(:, \texttt{bs})$ is readily derivable from `bs` and `nbs`, one usually doesn't bother to carry these columns along. This makes the updating of the matrix $B(:, \texttt{nbs})$ a bit more mysterious.

Finally, there are the following points to consider:

unbounded feasible set If, for some $k \in \texttt{nbs}$, $B(\texttt{end}, k)$ is the only positive entry in its column, then increasing y_k will strictly decrease the cost *and* increase all basic variables. Hence, if $y_{\texttt{nbs}} = 0$ is a feasible point, then we can make the cost function on the feasible set as close to $-\infty$ as we wish. In our example, this would be the case if we dropped constraints 1 and 5. Without these constraints, in our very first Simplex step, we could have increased y_3 without bound and so driven the cost to $-\infty$.

finding a feasible point In our example, we were fortunate in that the very first vertex we focused on was feasible. However, if it is not, then one can use the very Simplex Method to obtain a feasible point, simply by introducing an additional variable, y_0, which is added to each infeasible row, and then using the Simplex Method to minimize the cost function

$$y \mapsto y_0.$$

In this, the variable y_0 starts off nonbasic, i.e., $y_0 = 0$, and, then, as an extraordinary first step, we would exchange y_0 for the most negative basic variable, and then proceed until the minimum of this auxiliary cost function is reached. If it is positive, then we now know that *the feasible set is empty*. Otherwise, the minimum is zero (i.e., y_0 is again a nonbasic variable) and can simply be dropped now since the point corresponding to the remaining nonbasic variables being zero is feasible.

Note that, in this way, the Simplex Method can be used to solve any finite set of linear inequalities in the sense of either providing a point satisfying them all or else proving that none exists.

convergence in finitely many steps If we can guarantee that, at each step, we strictly decrease the cost, then we must reach the vertex with minimal cost in finitely many steps since, after all, there are only finitely many vertices. A complete argument has to deal with the fact that the cost may not always strictly decrease because the current point may lie on more than just n of the constraint hyperplanes.

Approximation by broken lines

tbc

Flats: points, vectors, barycentric coordinates, differentiation

In CAGD and Computer Graphics, Linear Algebra is mainly used to change one's point of view, that is, to change coordinate systems. In this, even the familiar 3-space, $\mathbb{R}^3$, is often treated as an 'affine space' or 'flat' rather than a vector space, in order to deal simply with useful maps other than linear maps, namely the affine maps.

For example, the **translation**

$$\mathbb{R}^3 \to \mathbb{R}^3 : p \mapsto p + v$$

of $\mathbb{R}^3$ by the vector v is not a linear map. Nevertheless, it can be represented by a matrix, using the following trick. Embed $\mathbb{R}^3$ into $\mathbb{R}^4$ by the 1-1 map

$$\mathbb{R}^3 \to \mathbb{R}^4 : x \mapsto (x, 1).$$

The image of $\mathbb{R}^3$ under this map is the 'flat'

$$F := \mathbb{R}^3 \times 1 = \{(x, 1) : x \in \mathbb{R}^3\}.$$

Consider the linear map on $\mathbb{R}^4$ given by

$$T_v := \begin{bmatrix} \mathrm{id}_3 & v \\ 0 & 1 \end{bmatrix}.$$

Then, for any $x \in \mathbb{R}^3$,

$$T_v(x, 1) = (\,\mathrm{id}_3 x \;+ v, 0^{\mathrm{t}}x + 1) = (x + v, 1).$$

In other words, the linear map T_v carries F into itself in such a way that the point $p = (x, 1)$ is carried to its 'translate' $(p + v, 1) = p + (v, 0)$.

Let, now, $\widehat{A} \in \mathbb{R}^{4\times 4}$ be an arbitrary linear map on $\mathbb{R}^4$ subject only to the condition that it map F into itself. Breaking up $\widehat{A}$ in the same way as we did T_v, i.e.,

$$\widehat{A} =: \begin{bmatrix} A_0 & v \\ [u]^{\mathrm{t}} & t \end{bmatrix},$$

we get

$$\widehat{A}\,(x, 1) = (A_0 x \;+ v, u^{\mathrm{t}}x + t),$$

hence want $u^{\mathrm{t}}x + t = 1$ for all $x \in \mathbb{R}^3$, and this holds if and only if $u = 0$ and $t = 1$, i.e.,

$$\widehat{A} =: \begin{bmatrix} A_0 & v \\ 0 & 1 \end{bmatrix}$$

is the most general such map.

Look at its action as concerns the general point $p = (x, 1)$ in F: After subtracting $p_0 := (0, 1)$ from p, we obtain a vector in the linear subspace

$$F{-}F = \{p - q : p, q \in F\} = \mathrm{ran}[e_1, e_2, e_3].$$

Since $\widehat{A}$ maps F into itself, it also maps this subspace into itself, hence $\widehat{A}(p - p_0) = (A_0\tau_{p-p_0}, 0)$ is again in this subspace and, after adding to it the element $\widehat{A}p_0 = (v, 1) \in F$, we finally obtain $\widehat{A}p$ as $\widehat{A}(p - p_0) + \widehat{A}p_0$.

Three-dimensional plots in `MATLAB` show, in fact, the orthogonal projection onto the (x,y)-plane *after* an affine transformation of $\mathbb{R}^3$ that makes the center of the plotting volume the origin and a rotation that moves a line, specified by azimuth and elevation, to the z-axis. This affine map is recorded in a matrix of order 4, obtainable by the command `view`, and also changeable by that command, but, fortunately, in down-to-earth terms like *azimuth* and *elevation*, or *viewing angle*.

□

Consider now the question which weighted sums

$$\sum_{j=0}^{r} p_j \alpha_j$$

of $p_j \in F$ are again in F. Apparently, all that is required is that

$$\sum_{j=0}^{r} \alpha_j = 1.$$

Such a weighted sum is called an **affine combination**. Thus, as far as the set F is concerned, these are the only linear combinations allowed. Note that such an affine sum can always be rewritten as

$$p_0 + \sum_{j=1}^{r} (p_j - p_0)\alpha_j,$$

where now the weights α_j, $j = 1{:}r$, are *arbitrary*. In other words, an affine sum on F is obtained by adding to some point in F an arbitrary weighted sum of elements in the vector space $F{-}F$.

An **affine map** on F is any map from F to F that preserves affine combinations, i.e., for which

$$A(p_0 + \sum_j (p_j - p_0)\alpha_j) = Ap_0 + \sum_j (Ap_j - Ap_0)\alpha_j$$

for all $p_j \in F$, $\alpha_j \in \mathbb{R}$. It follows that the map on $F{-}F$ defined by

$$A_0 : F{-}F \to F{-}F : p - q \mapsto Ap - Aq$$

must be well-defined and linear, hence A is necessarily the restriction to F of some linear map $\widehat{A}$ on $\mathbb{R}^4$ that carries F into itself and therefore also carries the linear subspace $F{-}F$ into itself.

The main pay-off, in CAGD and in Computer Graphics, of these considerations is the fact that one can represent the composition of affine maps by the product of the corresponding matrices.

This concrete example has led to the following abstract definition of a flat, whose notational conventions strongly reflect the concrete example. It should be easy for you to verify that the standard example is, indeed, a flat in the sense of this abstract definition.

(14.7) Definition: A **flat** or **affine space** or **linear manifold** is a nonempty set F of **point**s, a vector space T of **translation**s, and a map

(14.8) $$\varphi : F \times \mathrm{T} \to F : (p, \tau) \mapsto \tau(p) =: p + \tau$$

satisfying the following:

(a) $\forall\{(p,\tau) \in F \times \mathrm{T}\}\ \ p + \tau = p \iff \tau = 0.$

(b) $\forall\{\tau, \sigma \in \mathrm{T}\}\ \ (\cdot + \tau) + \sigma = \cdot + (\tau + \sigma).$

(c) $\exists\{p_0 \in F\}\ \ \varphi(p_0, \cdot)$ is onto.

Translations are also called **vector**s since (like 'vehicles' or 'conveyors', words that have the same Latin root as 'vector') they carry points to points.

Condition (a) ensures the uniqueness of the solution of the equation $p + ? = q$ whose existence (see the proof of (3) below) is guaranteed by (c).

Condition (b) by itself is already satisfied, for arbitrary F and T, by, e.g., $\varphi : (p, \tau) \mapsto p$.

Condition (c) is needed to be certain that T is rich enough. (a) + (b) is already satisfied, e.g., by $\mathrm{T} = \{0\}$, $\varphi(\cdot, 0) = \mathrm{id}$. As we will see in a moment, (a)+(b)+(c) together imply that $\varphi(p, \cdot)$ is onto for every $p \in F$. In other words, there is nothing special about the p_0 that appears in (c). In fact, the notion of a flat was developed explicitly as a set that, in contrast to a vector space which has an origin, does not have a distinguished point.

Consequences

(1) $\varphi(\cdot, 0) = \mathrm{id}$ (by (a)).

(2) For any $\tau \in \mathrm{T}$, $\varphi(\cdot, \tau)$ is invertible; its inverse is $\varphi(\cdot, -\tau)$ (by (1) and (b)). The corresponding abbreviation

$$p - \tau := p + (-\tau)$$

is helpful and standard.

(3) $\forall\{p, q \in F\}\ \exists!\{\tau \in \mathrm{T}\}\ \ p + \tau = q$. This unique τ is usually denoted

$$q - p,$$

for obvious reasons.

Proof: If $p+\tau = q = p+\sigma$, then, by (2) and (b), $p = q+(-\sigma) = (p+\tau)+(-\sigma) = p+(\tau-\sigma)$, therefore, by (1), $\tau - \sigma = 0$, showing the *uniqueness* of the solution to $p+? = q$, regardless of p and q. The *existence* of a solution is, offhand, only guaranteed, by (c), for $p = p_0$. However, with the invertibility of $\varphi(p_0, \cdot) : \mathrm{T} \to F$ thus established, hence with $p - p_0$ and $q - p_0$ well-defined, we have $q = p_0 + (q - p_0)$ and $p = p_0 + (p - p_0)$, hence $p_0 = p - (p - p_0)$, therefore

$$q = p - (p - p_0) + (q - p_0),$$

showing that the equation $p+? = q$ has a solution (namely the vector $(q - p_0) - (p - p_0)$). □

(4) Note that (3) provides a 1-1 correspondence (in many different ways) between F and T. Specifically, for any particular $o \in F$,

$$F \to \mathrm{T} : p \mapsto p - o$$

is an invertible map, as is its inverse,

$$\mathrm{T} \to F : \tau \mapsto o + \tau.$$

However, the wish to avoid such an arbitrary choice of an ‘origin’ o in F provided the impetus to define the concept of flat in the first place. The **dimension of a flat** is, by definition, the dimension of the associated vector space of translations. Also, since the primary focus is usually the flat, F, it is very convenient to write its vector space of translations as

$$F{-}F.$$

(5) The discussion so far has only made use of the additive structure of T. Multiplication by scalars provides additional structure. Thus, for arbitrary $Q \subset F$, the **affine hull** of Q is, by definition,

$$\flat(Q) := q + \mathrm{span}(Q - q),$$

with the right side certainly independent of the choice of $q \in Q$, by (4). The affine hull of Q is, itself, a flat, with $\mathrm{span}(Q - q)$ the vector space of its translations.

(6) In particular, the affine hull of a finite subset Q of F is

$$\flat(Q) = q_0 + \mathrm{ran}[q - q_0 : q \in Q\backslash q_0], \quad q_0 \in Q.$$

Let
$$q_0 + \sum_{q \neq q_0} (q - q_0)\alpha_q$$
be one of its elements. In order to avoid singling out $q_0 \in Q$, it is customary to write instead
$$\sum_q q\alpha_q, \quad \text{with} \ \ \alpha_{q_0} := 1 - \sum_{q \neq q_0} \alpha_q.$$

This makes $\flat(Q)$ the set of all **affine combination**s
$$\sum_{q \in Q} q\alpha_q, \quad \sum_q \alpha_q = 1,$$

of the elements of Q. The affine hull $\flat(q_0, \ldots, q_r)$ of a sequence $q_0, \ldots, q_r$ in F is defined analogously. But I prefer to work here with the set Q in order to stress the point of view that, in a flat, all points are of equal importance.

A special case is the straight line through $p \neq q$, i.e.,
$$\flat(p, q) = p + \mathbb{R}(q - p) = q + \mathbb{R}(p - q) = \{(1 - \alpha)p + \alpha q : \alpha \in \mathbb{R}\}.$$

(7) The finite set $Q \subset F$ is called **affinely independent** in case, for some (hence for every) $o \in Q$, $[q - o : q \in Q\backslash o]$ is 1-1. In that case, each $p \in \flat(Q)$ can be written in exactly one way as an affine combination
$$p =: \sum_q q\ell_q(p), \quad \sum_q \ell_q(p) = 1,$$

of the $q \in Q$. Indeed, in that case, for any particular $o \in Q$, $V_o := [q - o : q \in Q\backslash o]$ is a basis for the vector space of translations on $\flat(Q)$, hence, for all $p \in \flat(Q)$,
$$p = o + (p - o) = o + V_o V_o^{-1}(p - o) = \sum_{q \in Q} q\ell_q(p),$$

with
$$(\ell_q(p) : q \in Q\backslash o) := V_o^{-1}(p - o), \quad \ell_o(p) := 1 - \sum_{q \neq o} \ell_q(p).$$

The 'affine' vector $\ell(p) = (\ell_q(p) : q \in Q) \in \mathbb{R}^Q$ constitutes the **barycentric coordinates of** p **with respect to** Q.

It follows that, for arbitrary $p_i \in \flat(Q)$ and arbitrary $\alpha_i \in \mathbb{R}$ with $\sum_i \alpha_i = 1$, we have
$$\sum_i \alpha_i p_i = \sum_i \alpha_i \sum_q \lambda_q(p_i)q = \sum_q (\sum_i \alpha_i \lambda_q(p_i))q,$$

with
$$\sum_i \alpha_i (\sum_q \lambda_q(p_i)) = \sum_i \alpha_i = 1.$$

Hence, by the uniqueness of the barycentric coordinates, the map
$$\lambda : \flat(Q) \to \mathbb{R}^Q : p \mapsto (\lambda_q(p) : q \in Q)$$

is **affine**, meaning that
$$\lambda(\sum_i \alpha_i p_i) = \sum_i \alpha_i \lambda(p_i).$$

It is also 1-1, of course, and so is, for our flat $\flat(Q)$, what a coordinate map is for a vector space, namely a convenient structure-preserving numerical representation of the flat.

It follows that, with $f_0 : Q \to G$ an arbitrary map on Q into some flat G, the map

$$f : \flat(Q) \to G : \sum_{q\in Q} \lambda_q(p)q \mapsto \sum_{q\in Q} \lambda_q(p) f_0(q)$$

is affine. Hence, if $A : f \to G$ is an affine map that agrees with f_0 on Q, then it must equal f.

(8) Let the $r+1$-subset Q of the r-dimensional flat F be affinely independent. Then, for any $o \in Q$, $[q - o : q \in Q\backslash o]$ is a basis for $F{-}F$, and the scalar-valued map

$$\ell_o : F \to \mathrm{I\!R} : p \mapsto \ell_o(p)$$

is a **linear polynomial** on F. Some people prefer to call it an **affine polynomial** since, after all, it is not a *linear* map. However, the adjective 'linear' is used here in the sense of 'degree ≤ 1', in distinction to quadratic, cubic, and higher-degree polynomials. A description for the latter can be obtained directly from the ℓ_q, $q \in Q$, as follows. The column map

$$[\ell_\alpha := \prod_{q\in Q} (\ell_q)^{\alpha(q)} : \alpha \in \mathbb{Z}_+^Q, |\alpha| = k]$$

into $\mathrm{I\!R}^F$ is a basis for the (scalar-valued) polynomials of degree $\le k$ on F.

(9) An affine combination with nonnegative weights is called a **convex combination**. The weights being affine, hence summing to 1, they must also be no bigger than 1. The set

$$[q \,..\, q] := \{(1-\alpha)p + \alpha q : \alpha \in [0\,..\,1]\}$$

of all convex combinations of the two points p, q is called the **interval with endpoints** p**,** q. The set

$$\sigma_Q := \{\sum_{q\in Q} q\alpha_q : \alpha \in [0\,..\,1]^Q, \sum_q \alpha_q = 1\}$$

of all convex combinations of points in the finite set Q is called the **simplex with vertex set** Q in case Q is affinely independent.

(10) Flats are the proper setting for *differentiation.* Assume that the flat F is finite-dimensional. Then there are many ways to introduce a vector norm on the corresponding vector space $F{-}F$ of translations, hence a notion of convergence, but which vector sequences converge and which don't is independent of the choice of that norm. This leads in a natural way to convergence on F: *The point sequence* $(p_n : n \in \mathrm{I\!N})$ *in* F **converges** *to* $p \in F$ *exactly when* $\lim_{n\to\infty} \|p_n - p\| = 0$. Again, this characterization of convergence does not depend on the particular vector norm on $F{-}F$ chosen.

With this, the function $f : F \to G$, on the finite-dimensional flat F to the finite-dimensional flat G, is **differentiable at** $p \in F$ in case the limit

$$D_\tau f(p) := \lim_{h\searrow 0} (f(p + h\tau) - f(p))/h$$

exists for every $\tau \in F{-}F$. In that case, $D_\tau f(p)$ is called the **derivative of** f **at** p **in the direction** τ.

Notice that $D_\tau f(p)$ is a *vector*, in $G{-}G$. It tells us the direction into which $f(p)$ gets translated as we translate p to $p + \tau$. Further, its magnitude gives an indication of the size of the change as a function of the size of the change in p. Exactly,

$$f(p + h\tau) = f(p) + hD_\tau f(p) + o(\|\tau\|h), \quad h \ge 0.$$

In particular, if f is differentiable at p, then

$$Df(p) : F{-}F \to G{-}G : \tau \mapsto D_\tau f(p)$$

is a well-defined map, from $F{-}F$ to $G{-}G$. This map is obviously positively homogeneous, i.e.,

$$D_{h\tau} f(p) = h D_\tau f(p), \quad h \ge 0.$$

If this map $Df(p)$ is linear, it is called the **derivative of f at p**. Note that then

$$f(p + \tau) = f(p) + Df(p)\tau + o(\|\tau\|), \quad \tau \in F{-}F.$$

If V is any particular basis for $F{-}F$ and W is any particular basis for $G{-}G$, then the matrix

$$Jf(p) := W^{-1}\, Df(p)V$$

is the **Jacobian** of f at p. Its (i, j) entry tells us how much $f(p + \tau)$ moves in the direction of w_i because of a unit change in τ in the direction of v_j. More precisely, if $\tau = V\alpha$, then $Df(p)\tau = W\, Jf(p)\alpha$.

A practical high-point of these considerations is the **chain rule**, i.e., the observation that if $g : G \to H$ is a 'uniformly' differentiable map, then their composition gf, is differentiable, and

$$D(gf)(p) = Dg(f(p))Df(p).$$

grad, div, and curl

15. Optimization and quadratic forms

Minimization

We are interested in *minimizing* a given function

$$f : \operatorname{dom} f \subset \mathbb{R}^n \to \mathbb{R},$$

i.e., we are looking for $x \in \operatorname{dom} f$ so that

$$\forall y \in \operatorname{dom} f \quad f(x) \le f(y).$$

Any such x is called a **minimizer for** f; in symbols:

$$x \in \operatorname{argmin} f.$$

The discussion applies, of course, also to finding some $x \in \operatorname{argmax} f$, i.e., finding a **maximizer** for f, since $x \in \operatorname{argmax} f$ iff $x \in \operatorname{argmin}(-f)$.

Finding minimizers is, in general, an impossible problem since one cannot tell whether or not $x \in \operatorname{argmin} f$ except by checking *every* $y \in \operatorname{dom} f$ to make certain that, indeed, $f(x) \le f(y)$. However, if f is a 'smooth' function, then one can in principle check whether, at least, x is a **local minimizer**, i.e., whether $f(x) \le f(y)$ for all 'nearby' y, by checking whether the **gradient**

$$Df(x) = (D_i f(x) : i = 1{:}n)$$

of f at x is zero. Here, $D_i f = \partial f / \partial x_i$ is the derivative of f with respect to its ith argument.

To be sure, the vanishing of the gradient of f at x is only a *necessary* condition for x to be a minimizer for f, since the gradient of a (smooth) function must also vanish at any local *maximum*, and may vanish at points that are neither local minima nor local maxima but are, perhaps, only saddle points. By definition, any point x for which $Df(x) = 0$ is a **critical point** for f.

At a critical point, f is locally flat. This means that, in the Taylor expansion

$$f(x+h) = f(x) + (Df(x))^{\mathrm{t}} h + h^{\mathrm{t}} (D^2 f(x)/2) h + \text{h.o.t.}(h)$$

for f at x, the linear term, $(Df(x))^{\mathrm{t}} h$, is zero. Thus, if the matrix

$$H := D^2 f(x) = (D_i D_j f(x) : i, j = 1{:}n)$$

of second derivatives of f is 1-1, then x is a local minimizer (maximizer) for f if and only if 0 is a minimizer (maximizer) for the **quadratic form**

$$\mathbb{R}^n \to \mathbb{R} : h \mapsto h^{\mathrm{t}} H h$$

associated with the **Hessian** $H = D^2 f(x)$ for f at x.

If all second derivatives of f are continuous, then also $D_i D_j f = D_j D_i f$, hence the Hessian is real symmetric, therefore

$$H^{\mathrm{t}} = H.$$

However, in the contrary case, one simply defines H to be

$$H := (D^2 f(x) + (D^2 f(x))^{\mathrm{t}})/2,$$

thus making it real symmetric while, still,

$$h^{\mathrm{t}} H h = h^{\mathrm{t}} D^2 f(x) h, \quad \forall h \in \mathbb{R}^n.$$

In any case, it follows that *quadratic forms model the behavior of a smooth function 'near' a critical point.* The importance of minimization of real-valued functions is the prime motivation for the study of quadratic forms, to which we now turn.