Recall determinants (6.1, 6.2)

- for a square matrix A,
  $\det(A)$ is a scalar.

- defined using axioms (as a map from matrices to F ← scalars.

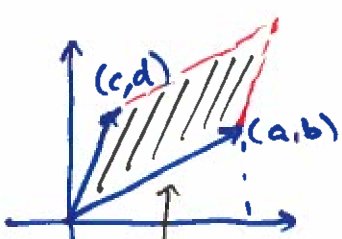
there exists exactly one map satisfying these properties →
$$\left\{ \begin{array}{l} \text{it is linear in each row,} \\ \det(A) = 0 \iff rk(A) < n \\ \quad \text{for an } n\times n\text{-matrix } A \\ \det(Id) = 1 \end{array} \right\}$$

Today: want to get comfortable with computing det.

So far, we have:

1) 1×1-matrix $[a] \xrightarrow{\det} a$.
2) 2×2-matrix: $\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc$.

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} \leftarrow \text{means, det.}$$

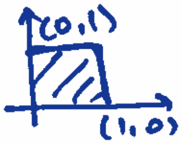


Area $= \begin{vmatrix} a & b \\ c & d \end{vmatrix}$

Proof of the formula for area

- if you scale any side, area should multiply by the (abs. value of) the scala

- Area $= 0 \iff$ vectors are linearly dependent
  $\iff rk \begin{bmatrix} a & b \\ c & d \end{bmatrix} \leq 1$

•  Area(unit square) = 1.

• Then: Area of  satisfies all the properties of det! Then it equals det(A)!

3) for 3x3-matrices:

we have 3 ways of computing det:

1) expansion by a row or column. ← works for any size matrix

$$\begin{vmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{vmatrix} = +a_1 \begin{vmatrix} b_2 & b_3 \\ c_2 & c_3 \end{vmatrix} - a_2 \begin{vmatrix} b_1 & b_3 \\ c_1 & c_3 \end{vmatrix} + a_3 \begin{vmatrix} b_1 & b_2 \\ c_1 & c_2 \end{vmatrix}$$

$$\boxed{\begin{matrix} + & - & + \\ - & + & - \\ + & - & + \end{matrix}}$$ ← signs

2) Elementary row operations: ← works for any size matrix

• interchange 2 rows ↔ det gets multiplied by −1.

• scale a row by a scalar → det gets multiplied by that scalar

• $R_i + cR_j$ (leave the j^{th} row unchanged, replace the i^{th} row with this sum)

these do not change det!

• Once the matrix is upper-triangular, $\begin{pmatrix} * & & * \\ & \ddots & \\ 0 & & * \end{pmatrix}$ det is easy: it is the product of the diagonal entries.

Example

(assume a,b,c are distinct) $a,b,c \in F$

$$\begin{vmatrix} 1 & a & a^2 \\ 1 & b & b^2 \\ 1 & c & c^2 \end{vmatrix} \xrightarrow[R_3-R_1]{R_2-R_1} = \begin{vmatrix} 1 & a & a^2 \\ 0 & b-a & b^2-a^2 \\ 0 & c-a & c^2-a^2 \end{vmatrix}$$

kill this next

$$\xrightarrow[R_3-\frac{c-a}{b-a}R_2]{} = \begin{vmatrix} 1 & a & a^2 \\ 0 & b-a & b^2-a^2 \\ 0 & 0 & c^2-a^2-\frac{b^2-a^2}{b-a}\cdot(c-a) \end{vmatrix}$$

upper-triangular

assuming $b-a \neq 0$

$$= 1\cdot(b-a)\cdot\underbrace{\Big(\underset{\underset{(c-a)(c+a)}{\shortparallel}}{c^2-a^2} - \frac{\overset{b+a}{\cancel{b^2-a^2}}}{\cancel{b-a}}\cdot(c-a)\Big)}_{\underset{(c-a)(c+\cancel{a}-(b+\cancel{a}))}{\shortparallel}}$$

$$= (c-a)(c-b)$$

$$= (b-a)(c-a)(c-b)$$

3) Just for 3×3:

$$\begin{vmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{vmatrix} = a_1b_2c_3 + a_2b_3c_1 + b_1c_2a_3 - a_3b_2c_1 - b_1a_2c_3 - c_2b_3a_1$$

can be generalized to any rank:

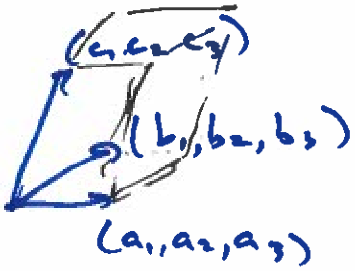
combination of products one element from each row and each column with "tricky" $\pm$ signs. ← Leibniz formula.

(handy for 3×3)

4) Geometric way:

$$\begin{vmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{vmatrix} = \text{vol}\Big(\,$$  $$\Big)$$

build a parallelepiped spanned by the rows of the matrix



(same proof as for area!)

Aside

This tells us how to define volume in $\mathbb{R}^n$.

Vol ( in $\mathbb{R}^n$) = | matrix of components of the vectors spanning it |

(and this is what gives rise to change of variable in integrals formula in multi variable calculus)

Transpose matrix

given A (an $m \times n$-matrix), we can "transpose" it:

interchange rows $\leftrightarrow$ columns

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c_3 & c_4 \end{bmatrix} \longrightarrow A^t = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \\ a_4 & b_4 & c_4 \end{bmatrix}$$

3×4 "A transpose" 4×3-matrix

- Transposition does not change det!
  (for square matrices)

- So everything I said about row ops applies to column ops as well, and when thinking of volumes, could instead make a box of column vectors.