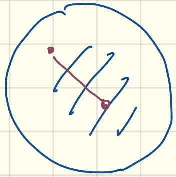
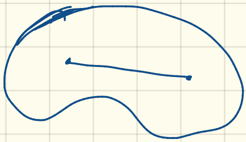
convex set



convex



non-convex

Let $x, y \in R^n$. The line segement between x and y in the set

$$\{(1-\lambda)x + y : \lambda \in [0,1]\}.$$

A set $C \subseteq R^n$ is convex if $\forall x, y \in C$, C contains the line segment between x and.

Halfspace, a set of points satisfies a single linear inequality with the coefficiensies are not all zero.

conditions define the set

Consider the set

$$H = \{\alpha \in R^n : a^T x \geq \beta\}$$

$$\boxed{\text{where } \alpha \in R^n,\ a \neq 0 \quad \beta \in R}$$

Take $x, y \in H$,

consider $z = (1-\lambda)x + \lambda y$ for some $\lambda \in [0,1]$

Then $a^T z = a^T\left((1-\lambda)x + \lambda y\right)$

apply distribute law

You can treat the word polyhedron as a fancy name for the feasible region of a linear programming problem.

$$a^T z = \underbrace{(1-\lambda)}_{\geq 0}\underbrace{a^T x}_{\geq \beta} + \underbrace{\lambda}_{\geq 0}\underbrace{a^T y}_{\geq \beta}$$

$$\geq (1-\lambda)\beta + \lambda\beta = \beta$$

The set of feasible solutions are going to be convex.

$a^T z \geq \beta \Rightarrow z \in H$, so H is convex

Fact.

due to this fact

The intersection of two convex sets is convex.

A well studied problem.

A polyhedron is the intersection of finitely many halfspace. This set is called plyhedron.

e.g. $\left\{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} : x_1 + 2x_2 \geq 5\right\} \cap \left\{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} : -2x_1 + x_2 \geq 0\right\}$

$$= \left\{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} : \begin{array}{l} x_1 + 2x_2 \geq 5 \\ -2x_1 + x_2 \geq 0 \end{array}\right\}$$

Basically. we can see a polyhedron is simply a set of feasible solutions of a linear programming problem.

polyhedral compliation

mathmatical foundation

Integer ploryhedra Presburger relation

isl: integer set library

built around Presburger relations, and many ideas originally introduced in the Omega project. parametric integer programming is an important concept at the core of isl.

Important concepts

1. integer sets:

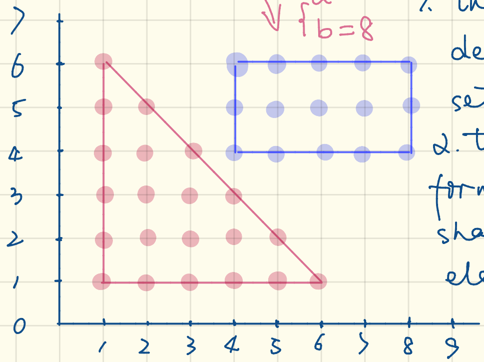
sets of integer tuples from $\mathbb{Z}^d$ described by Presburger formulas.

parameters

$$S = \{(i,j) \mid (a \le i, j \wedge i+j < b) \vee (4 \le i, j \wedge i \le b \wedge j \le 6)\}$$

2-d integer set

$\begin{cases} a=1 \\ b=8 \end{cases}$



1. the colored shapes derived from integer set.
2. the colored shapes form a set of convex shapes. That enclose elements in S.

In general, an integer set has the form:

$$S = \{ \underbrace{\vec{s}}_{\text{integer tuples}} \in \mathbb{Z}^{\overset{\text{dimensionality}}{d}} \mid \underbrace{f}(\vec{s}, \underbrace{\vec{p}}) \}$$

integer tuples

dimensionality

$\vec{p} \in \mathbb{Z}^e$: parameters

$f(\vec{s}, \vec{p})$: presburger formula that evaluates to true iff $\vec{s}$ is element of S for given parameters $\vec{p}$.

1.1. Presburger formula

defined recursively as:

① boolean constant

② boolean operation

- negation: $\neg p$
- conjunction: $p_1 \wedge p_2$ ($\wedge$, 合取词)
- disjunction: $p_1 \vee p_2$ ($\vee$, 析取词)
- implication: $p_1 \Rightarrow p_2$ ($\rightarrow$, 条件词, $\leftrightarrow$ 双条件词)

③ a quantified expression $\forall x : p$, $\exists x : p$

④ comparison between different quasi-affine expressions

$e_1 \oplus e_2$, $\oplus \in \{<, \leq, \geq, >\}$

Side Note:

1. quantifed expression / quantifier This is a concept from mathematical logic.

In natural language, a quantifer turns a sentence about something having some property into a sentence about the number of things having the property.

For example: In English

quantifier: some, all, many, few, most, no

quantified sentence:
- all people are mortal. True
- some people are mortal. True
- no people are mortal False

In mathematical logic, in particular in first-order logic, a quantifier achieves a similar task, operating on a mathematical formula.

2. affine expression / Affine vs Linear

2.1 Linear Transformation

use the definition in Linear Algebra

a linear function is a linear mapping, or a linear transformation.

A transformation f is linear when for any two vectors $\vec{v}$ and $\vec{w}$:

1. $f(\vec{v}+\vec{w}) = f(\vec{v}) + f(\vec{w})$
2. $f(k\vec{v}) = kf(\vec{v})$ for some scalar k

For the kind of vector space we are interested in: finite-dimensional vector spaces with a defined basis any linear mapping can be represented by a matrix that is multiplied by the input vector.

We can represent any vector in terms of standard basis vectors: $\vec{v} = \vec{v}_1\vec{e}_1 + \cdots + \vec{v}_n\vec{e}_n$

since f is linear,

orthonormal basis / standard basis

$$f(\vec{v}) = f\left(\sum_{i=1}^{N} v_i \vec{e_i}\right) = \sum_{i=1}^{N} v_i f(\vec{e_i})$$

think of $f(\vec{e_i})$ as column vector

$$f(\vec{v}) = \begin{pmatrix} | & | & & | \\ f(\vec{e_1}), & f(\vec{e_2}), & \cdots, & f(\vec{e_N}) \\ | & | & & | \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix}$$

→ linear transformation of a vector $\vec{v}$, is precisely the multiplication of a matrix by $\vec{v}$.

can been seen as a change of basis for $\vec{v}$ from the standard basis to a basis defined by f.

For example:

$f(\vec{v}) = \langle 3v_1 - 4v_2, v_2 \rangle$, we can represent the mapping in such a matrix:

$$\begin{pmatrix} 3, & -4 \\ 0, & 1 \end{pmatrix}$$

$$f(\vec{v}) = \begin{pmatrix} 3, & -4 \\ 0, & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$$

This representation gives interesting tools to work with them.

compositions of mapping as asociativity of matrix multiplication.

To visualize a mapping, it useful to examine its effects on some standard vectors.

Let's see an example

Let's consider two mappings: long chains of transformations work in exactly the same way. and the fact that we ~~can represent such chains with a single matrix is very useful.~~

1. $S = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$, "stretches"

equational form: $S(\vec{v}) = \langle 2v_1, 2v_2 \rangle$

2. $R = \begin{pmatrix} \cos\theta, & \sin\theta \\ -\sin\theta, & \cos\theta \end{pmatrix}$ "rotation"

combine these transformation: to stretch and the rotate a vector, we would do:

$$f(\vec{v}) = R(S\vec{v})$$ ← matrix multiplication is associative

$$f(\vec{v}) = (RS)\vec{v}$$ ← we can ~~find a matrix $A = RS$ which represents the combined transformation.~~

2.2. Affine Transformations

For an affine space, every affine transformation is of the form $g(\vec{v}) = A\vec{v} + \vec{b}$

A: a matrix representing a linear transformation

Obviously:

1. Every linear transformation is affine.
2. Not every affine transformation is linear.

An affine transformation combines a linear transformation with a translation.

Let's see an example:

$$f(\vec{v}) = \underbrace{\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}}_{S:\ \text{stretch}} \vec{v} + \underbrace{\begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}}_{T:\ \text{translation}}$$

With some clever agumentation, we can represent affine transformation as a multiplication by a single matrix.

$$f(\vec{v}) = T\vec{v} = \begin{pmatrix} 2 & 0 & \boxed{0.5} \\ 0 & 2 & \boxed{0.5} \\ 0 & 0 & \boxed{1} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ 1 \end{pmatrix}$$

stack translation vector on the right-hand side of transformation matrix.

Affine transformations can be composed using matrix multiplication.

Augmented matrix for stretches:

$$S = \begin{pmatrix} 2 & 0 & 0.5 \\ 0 & 2 & 0.5 \\ 0 & 0 & 1 \end{pmatrix}$$

Augmented vector for rotation:

$$R = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The composed transformation for "scaling + translation + rotation" is:

$$f(\vec{v}) = T(R(\vec{v})) = (TR)\vec{v}, \text{ its matrix is:}$$

$$TR = \begin{pmatrix} 2 & 0 & 0.5 \\ 0 & 2 & 0.5 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2\cos\theta & \sin\theta & 0.5 \\ -2\sin\theta & 2\cos\theta & 0.5 \\ 0 & 0 & 1 \end{pmatrix}$$

2.3 Affine space

A subset $U \subset V$ of a vector space V is an affine space if there exists a $u \in U$ such that $U - u = \{x - u \mid x \in U\}$ is a vector subspace of V.

1. Linear and affine subspaces are related by using a translation vector.
2. An affine space is a generalization of a linear space, in that it doesn't require a specific origin point.

Affine spaces and transformations have some interesting properties, which leads to useful generalizations, making them useful.

For example.

① an affine transformation always map a line to a line

② any two triangles can be converted one to the other using an affine transformation.

2.4. Affine expressions and array addressing

Affine expression

An expression is affine w.r.t variables $v_1, v_2, \cdots, v_n$ if it can be expressed as $c_0 + c_1 v_1 + \cdots + c_n v_n$, where $c_0, c_1, \cdots, c_n$ are constant.

Affine expressions are interesting because they are often used to index arrays in loops.

```
for (int i = 0; i < M; ++i) {
    for (int j = 1; j < N; ++j) {
        arr[i][j-1] = arr[i][j];
    }
}
```

This statement assigns a value to arr [i * N + j - 1] at every iteration.

this is an affine expression w.r.t i and j

keep this fact in mind

When all expressions in a loop are affine, the loop is amenable to some advanced analyses and optimizations.