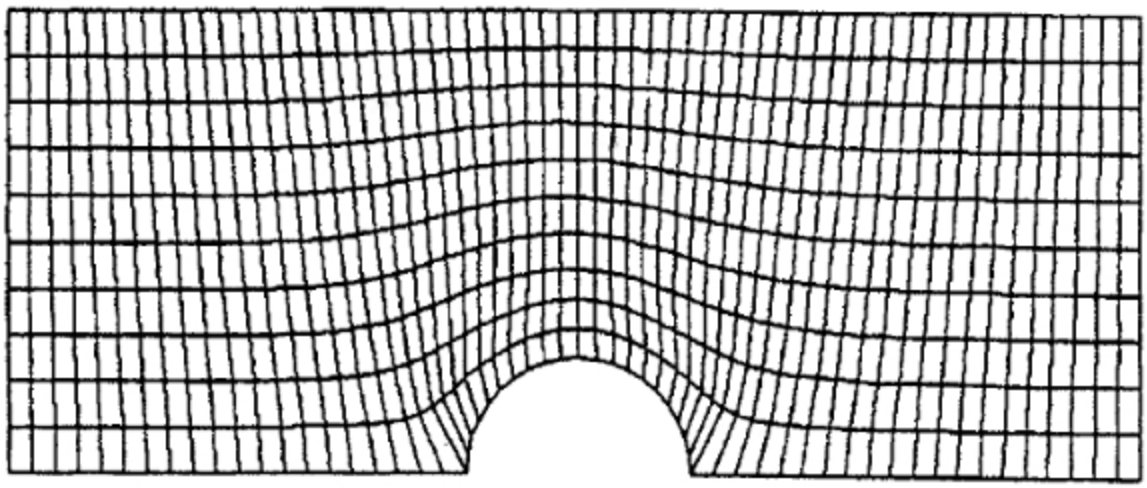
IsoSurfaces

DEF: Scalar Field: $f : \mathbb{R}^3 \to \mathbb{R}$

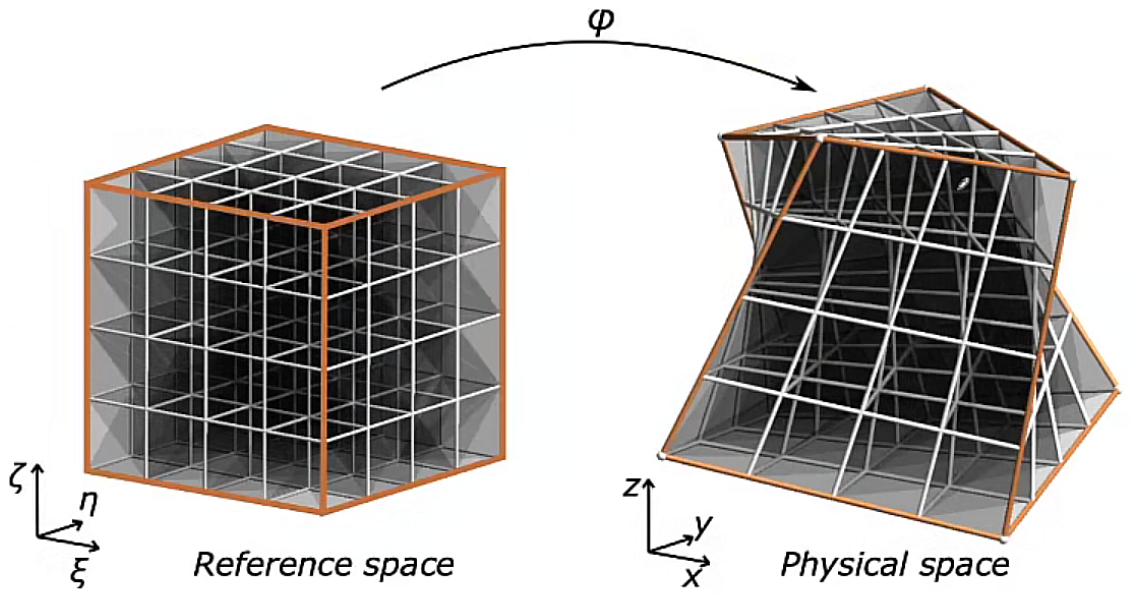
Isosurface: $A_\tau = \{x \in \mathbb{R}^3 : f(x) = \tau\}$, where τ is called the **isovalue**

curvilinear grid: a grid made of not straight lines



we represent the curvilinear grid (3D) as an array $F[i][j][k]$, where each element stores a point in space for the grid $p_{i,j,k} \in \mathbb{R}^3$ and a scalar value $f_{i,j,k} \in \mathbb{R}$ we can

also rappresent the isosurface in the so said **“computational space”** or **reference space**, being a linearization of the curvilinear grid, that maintains the same cells, said **Voxels**, and nodes

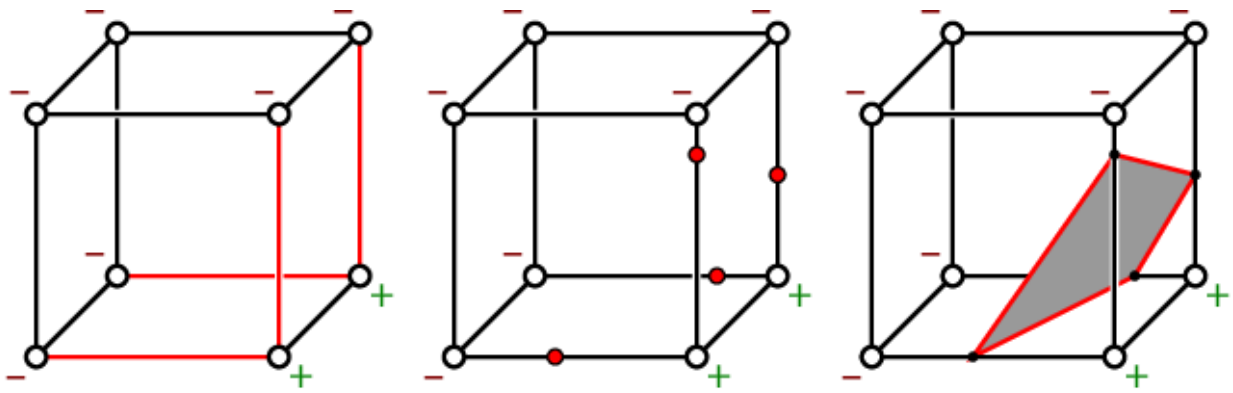


Discrete IsoSurfaces

an isosurface in a discrete space can be defined as a surface

$$A'_\tau : \forall \text{ voxels } v \exists v_i, v_j : f(v_i) \leq \tau, f(v_j) \geq \tau, v = \{v_1, \ldots, v_8\}$$

i.e. if the voxel intecepts the isosurface, there must be a couple of nodes one bigger and one smaller than τ

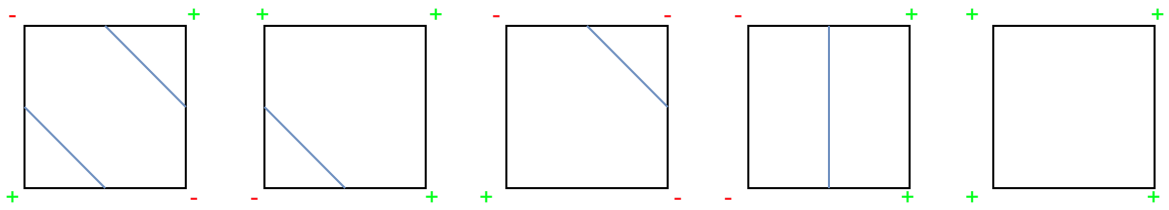


the problem is now: **how do we find the isosurfaces?**

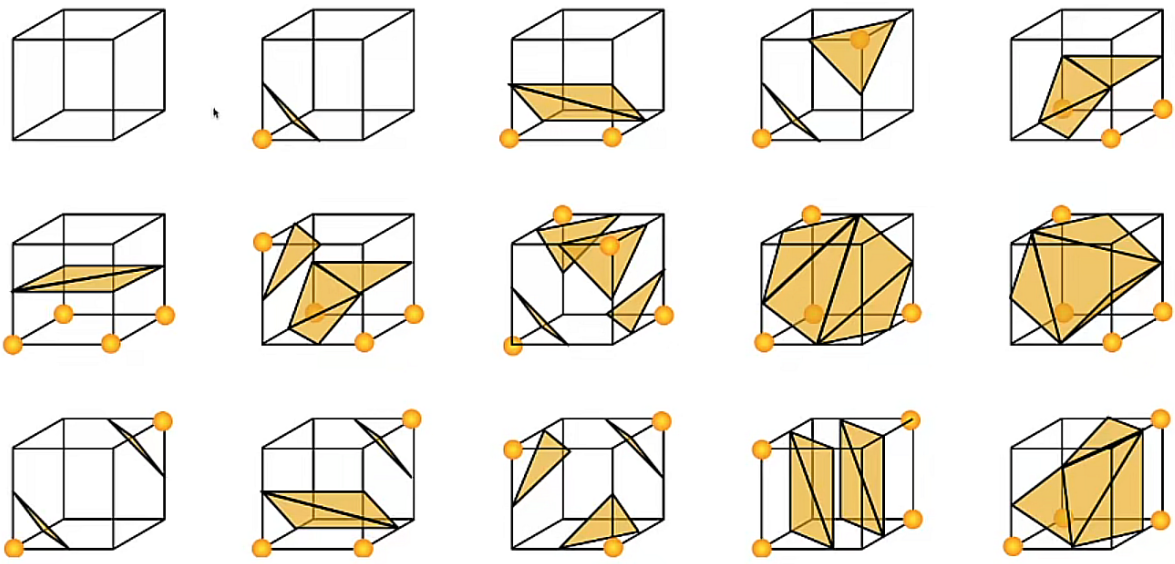
Marching cubes algorithm

```
for all voxels:
    determine the signs of the nodes of v(- if vi < tau, + if vi>tau)
    triagulate according to templates
```

templates in 2d



templates in 3d



Construction with OctTrees

Min-Max OctTree

We construct a complete OctTree over the nodes of the scalar field, where each child points to the lower left verteces of the voxels and each leaf stores the minimum and maximum values of the nodes, for inner nodes the reasoning is propagated with respect to the children.

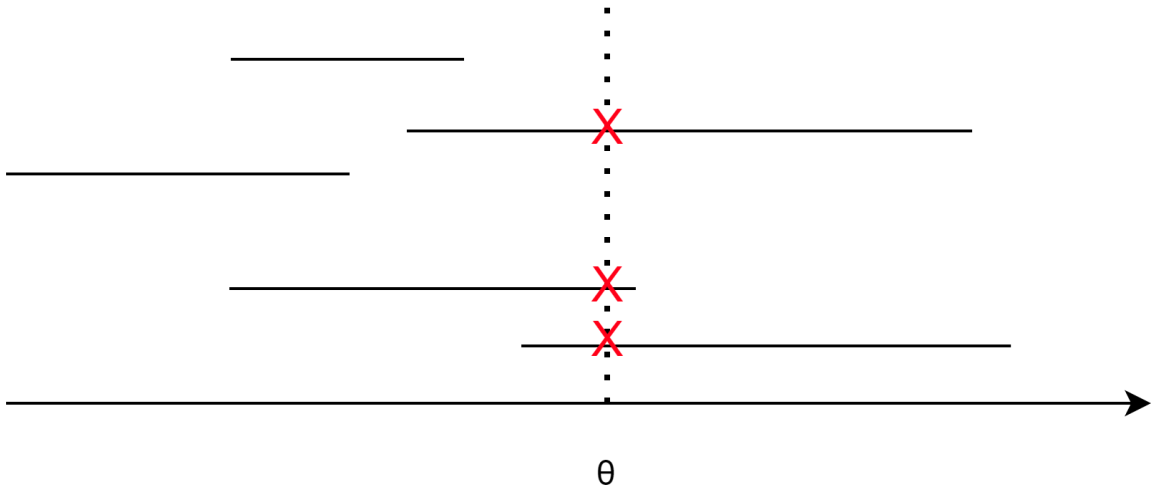
The isosurface now passes throug a region of the octre node if and only if $min(v) \leq \tau \leq max(v)$

at this point the algorithm is a simple recursive visit of the tree, where we seek if the node has children, then find the child that respects the condition

Interlude: Span space

Problem: “1-dimension stabbing query”

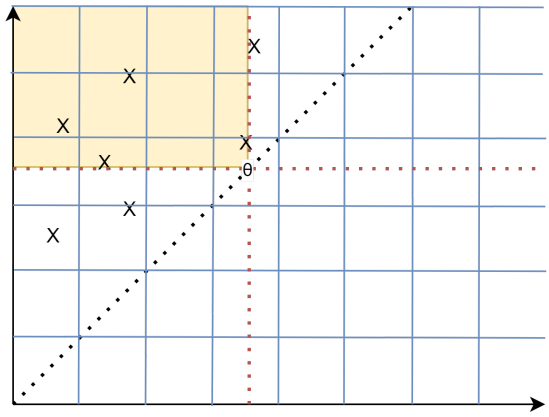
we have N intervals $\{a_i, b_i\} \in \mathbb{R}$ and a point θ called query point. we searh for all the intervals “stabbed” by θ, i.e. all the intervals containing θ:



Solution idea

we consider the intervals as points in $\mathbb{R}^2$, where the x coordinate is the begin and the y is the end.

obviously since $\forall points, y \geq x$ everything will live over the line splitting the first quadrant. at this point theta is a point on the diagonal, we can overlay a lattice on the span space, such that all the lines of the lattice are equidistant and the points are equidistributed over the lattice lines.



we derive that lattice cells are intercepted exactly diagonally or not at all and a and b will be sorted by the lattice.

we can create 2 lists:

- L_i^a = list of all points in row i in ascending order by a from columns 1 to i-1
- L_i^b = list of all points in row i in descending order by b from columns 1 to i-1

for all the remaining points on diagonal we build a new lattice recursively

we can now derive this algorithm:

```
for all cells (l, l) in lattice containing theta:
  for i = l+1 to L:
    traverse lia[j].a>0
  for row l
    traverse Lll up to llb[j].b<theta
  for cell (l,l):
    recurse into the sub lattice
```

running time

each cell contains $\frac{n}{\frac{L^2}{2}} = \frac{2n}{L^2}$ points

we perform a binary searh, followed by a loop, followed by the recursion, finding the points we found(k), the total complexity is:

$$T(n) = O(logL) + O(L) + O(k) + T(\frac{2n}{L^2}) = O(\frac{log^2 n}{log(log(n))})$$

Construction of an isosurface on a time varing field

Given N 3d scalar fields, for $t_i \in \{t_0, t_{N-1}\}$, we can define

$\min_t(v) = \min\{\text{nodes of voxel v at time t}\}$

$\max_t(v) = \max\{\text{nodes of voxel v at time t}\}$

$\min_i^j(v) = \min\{\min_i(v), \ldots, \min_j(v)\}$

$\max_i^j(v) = \max\{\max_i(v), \ldots, \max_j(v)\}$

if we consider a specific voxel and its $\min_t(v)$ and $\max_t(v)$ and its trajectory over time, we want a criterion o consider if the voxel experiences a larve variation over time

we say that the voxel has a **small temporal variation** if and only if $\forall t \in i \ldots, j$ all points $(\min_t(v), \max_t(v))$ are contained in a 2x2 contiguos cells in span space

i.e when t varies the isosurface stays in a 2x2 grid in the lattice

we can construct a **Temporal Index Tree:**

where T_i^J contains all voxels v that have small temporal variation over time $[i, j]$

to construct it, we start with the set of voxels V and T_0^N as a root, then we create the span space of all the voxels in v and time interval from 0 to N. at this point for each voxel $v \in V$ we check the ones with small temporal variance, at this point if they have it we add the voxels to the root, otherwise we create a $T_0^{\frac{N}{2}}$ and $T_{\frac{N}{2}}^N$ nodes and we add to them the voxels in $V/V(T_0^N)$. at this point we build an octtree for each node in $V(T_j^i)$