Arbitrary Point Search and Interpolation on Surface Meshes

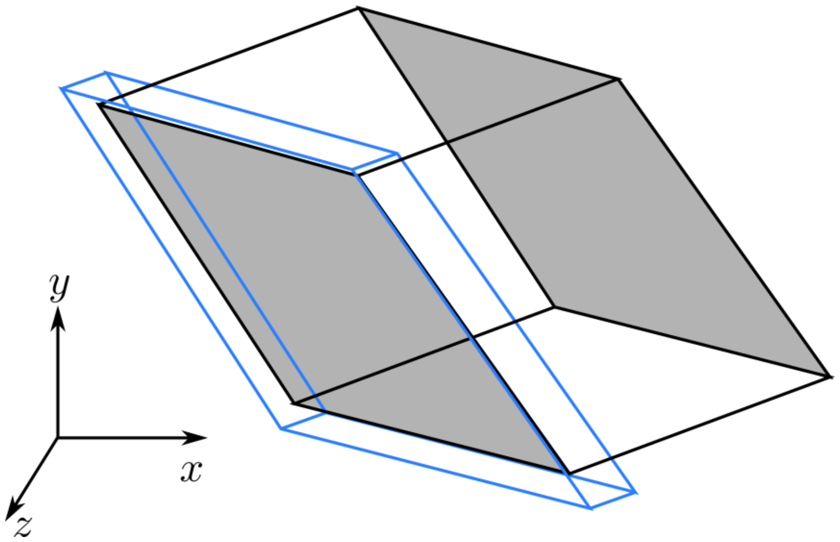
Aditya Parik, Ketan Mittal

PhD candidate, Mechanical Engineering, Mechanical and Aerospace Engineering

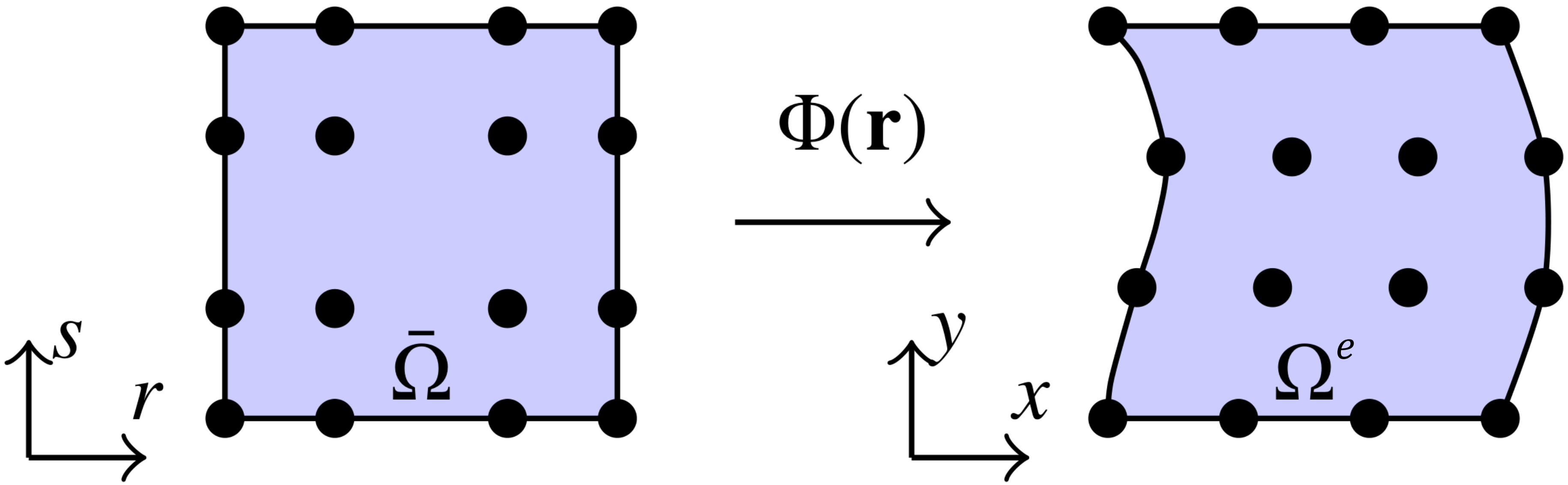
Utah State University, Logan, Utah

Motivation and some surface element peculiarities

- Surface elements map a N dimensional physical space to a (N-1) dimensional reference space element.
- Thus, we encounter a lack of information in the normal direction, which volume elements have.
- This peculiarity results in substantial changes to the evaluation of axis-aligned and oriented bounding boxes.
- A bounding box algorithm for dedicated surface meshes would reduce the physical space required to be searched for the poin, thus reducing the cost of finding process.



What does “Searching” for a point mean?



$$\mathbf{x}^e(\mathbf{r}) = \sum_{l=1}^{N_{1D}^D} \mathbf{x}_l^e \phi_l(\mathbf{r}) = \sum_{i=1}^{N_{1D}} \sum_{j=1}^{N_{1D}} \sum_{k=1}^{N_{1D}} \mathbf{x}_{ijk}^e \phi_i(r)\phi_j(s)\phi_k(t), \quad \mathbf{r} \in \bar{\Omega}$$

- An element (Ω^e) in the physical space is described by a transformation $\Phi(\boldsymbol{r})$ on the reference element $\bar{\Omega}$.
- To search for a point x^*, we identify the element Ω^e it belongs to and the corresponding rank p, and the corresponding reference coordinate $\boldsymbol{r}^*$.

Overview of the Search process

Rapidly identify the rank that contains the point $\boldsymbol{x}^*$

↓

Rapidly Identify the candidate elements in the local mesh that could contain the point $\boldsymbol{x}^*$

Requires the construction (per mesh) of data structures that map:

- The cells of a local map to all contributing local elements, and
- The cells of a global map to all contributing ranks.

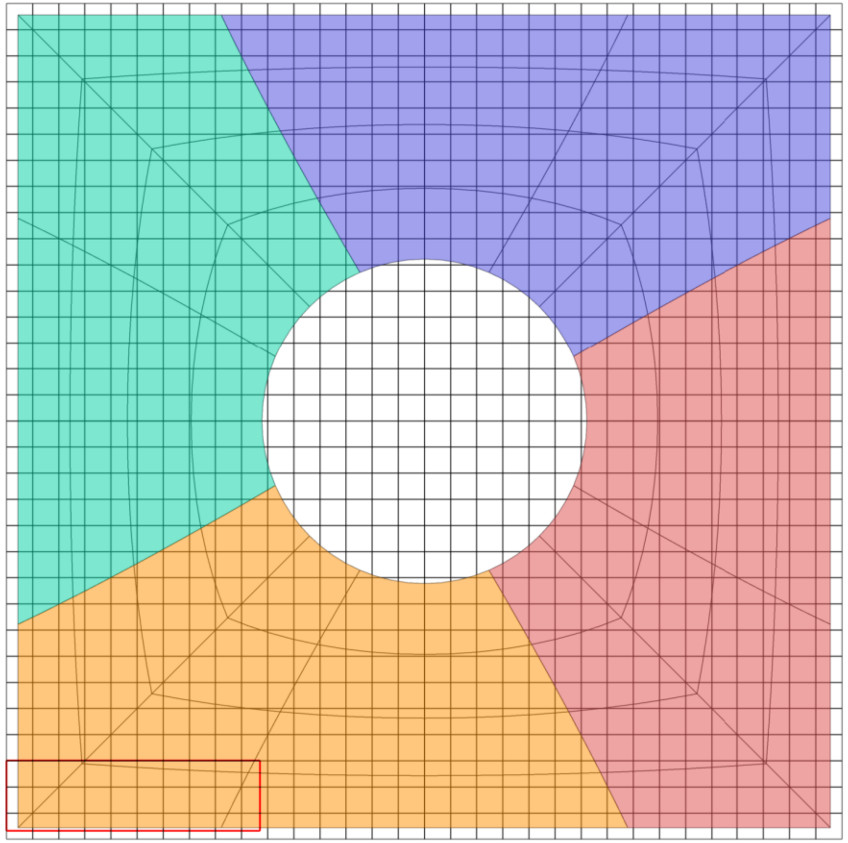
ONE-TIME COST for each mesh!
Called the "Setup" stage of the search process.

↓

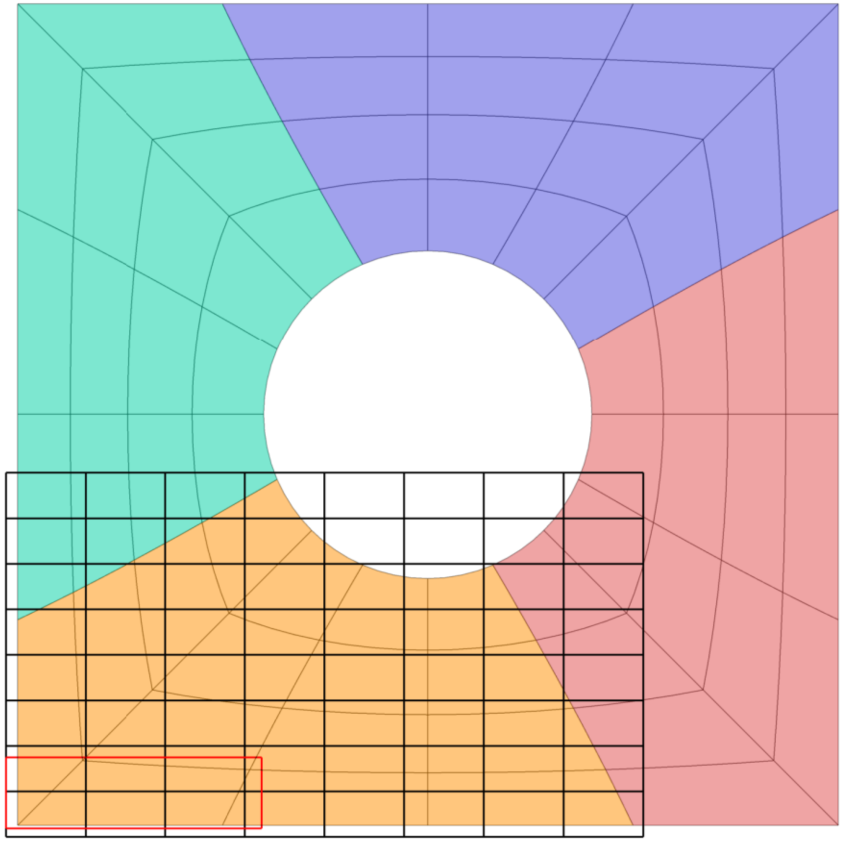
Do an iterative search based on Newton's method on each candidate element until the point is found.

- Only performed on the list of candidate elements that cannot be further narrowed down by explicit fast evaluations.
- Used to determine the reference space coordinates corresponding to the given point and candidate element, **if the point is found**.

The Setup Process: Bounding Boxes



Global Cartesian Mesh $\mathcal{M}_G$ spanning the entire mesh



Local Cartesian Mesh $\mathcal{M}_L$ spanning the partition of a specific Rank

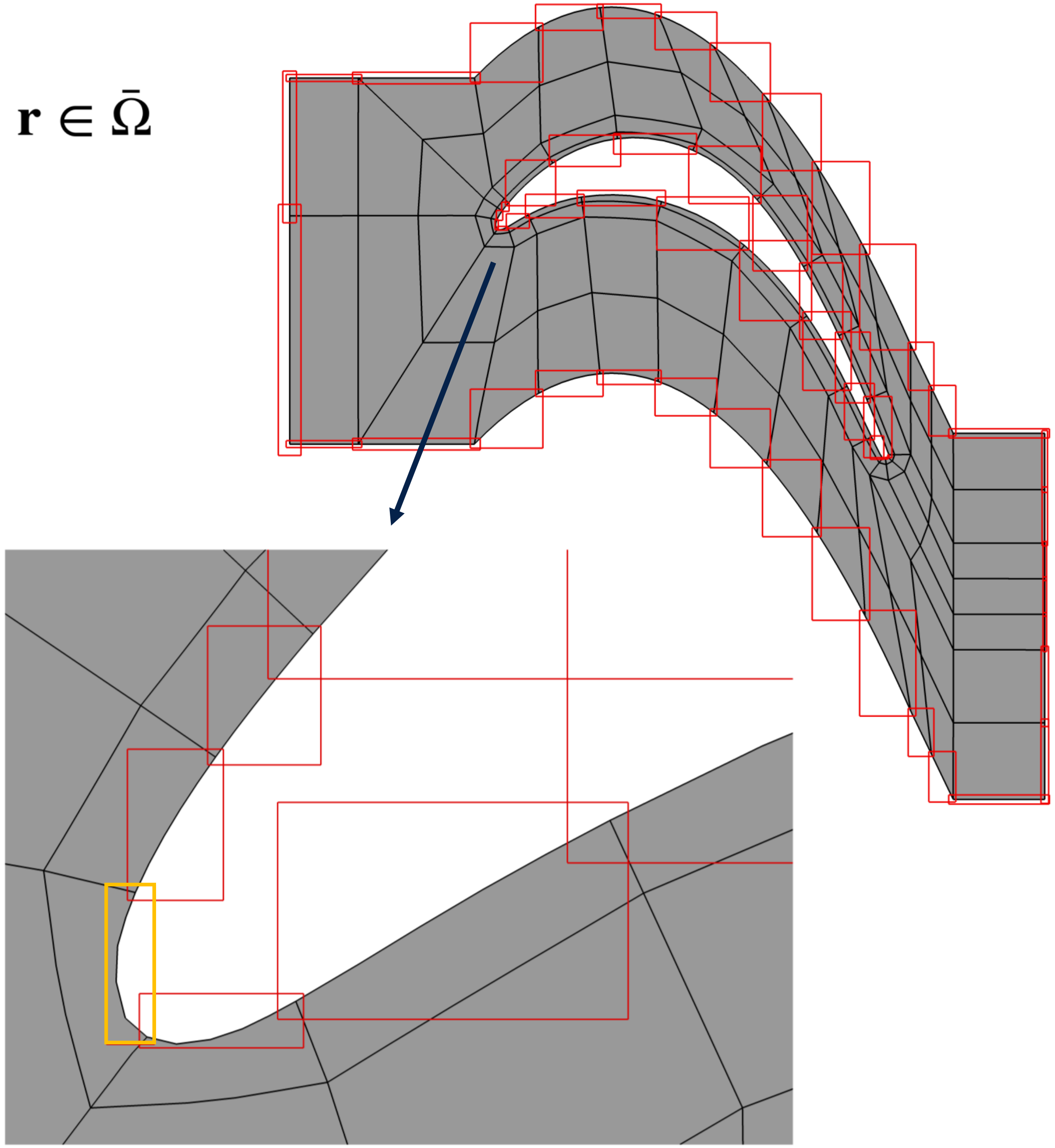
Finding intersections of high order elements with $\mathcal{M}_G$ or $\mathcal{M}_L$ is not trivial. So, how to do this efficiently?

By testing contributions of corresponding Axis-Aligned (linear) Bounding Boxes!

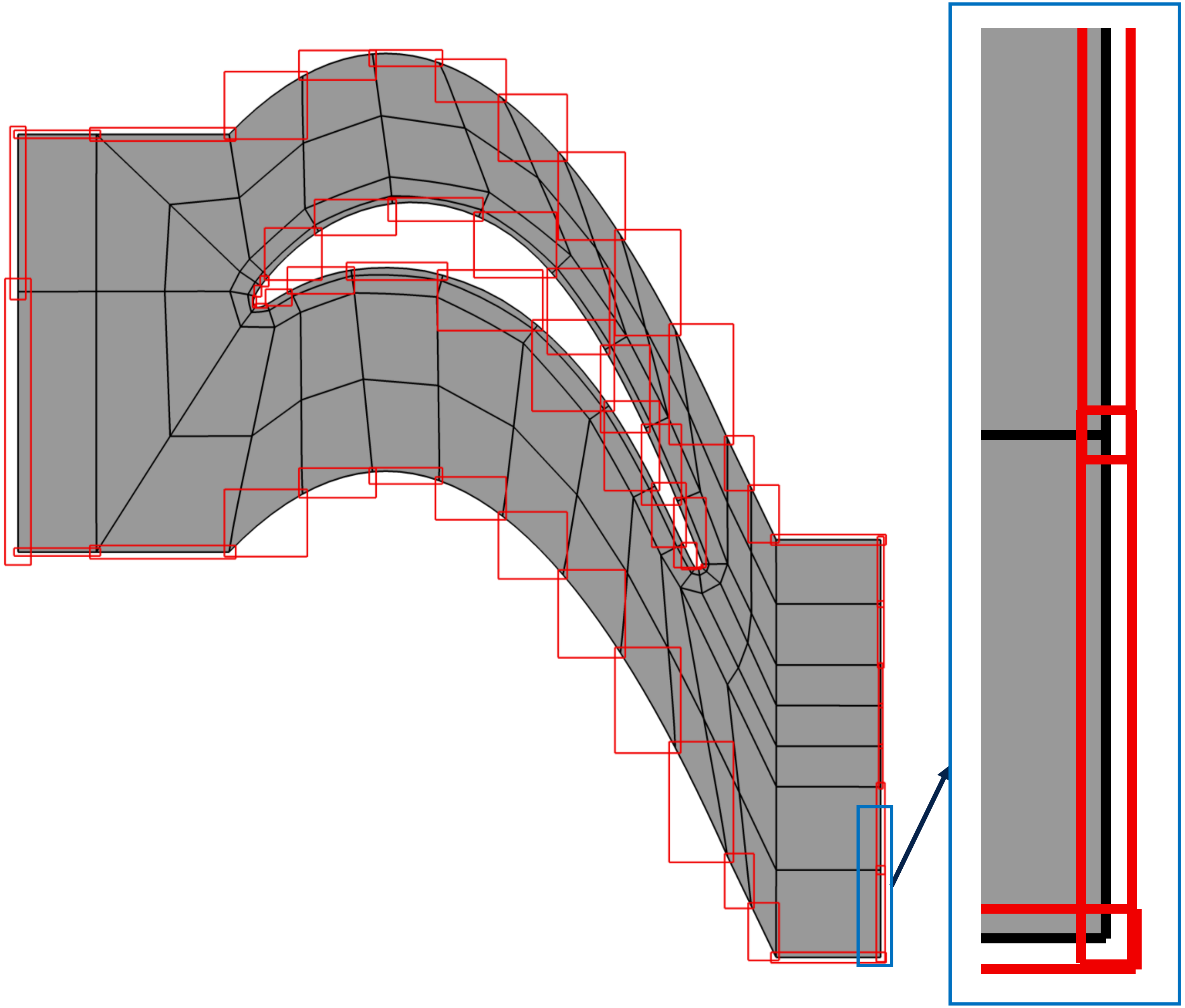
Axis Aligned Bounding Boxes (AABB)

$$\mathbf{x}^e(\mathbf{r}) = \sum_{l=1}^{N_{1D}^D} \mathbf{x}_l^e \phi_l(\mathbf{r}) = \sum_{i=1}^{N_{1D}} \sum_{j=1}^{N_{1D}} \sum_{k=1}^{N_{1D}} \mathbf{x}_{ijk}^e \phi_i(r)\phi_j(s)\phi_k(t), \quad \mathbf{r} \in \bar{\Omega}$$

- For a high order element, generating a AABB is non-trivial since the maxima/minima need not lie on a node.
- We instead bound each Lagrange polynomial through a piecewise-linear function, which can be bound trivially just using its nodal locations.
- $x^e(\boldsymbol{r})$ can hence be bounded using a linear combination of piecewise functions, and hence the corresponding AABB can be obtained.
- The elements of the surface mesh of this blade mesh are bounded using axis-aligned bounding boxes.

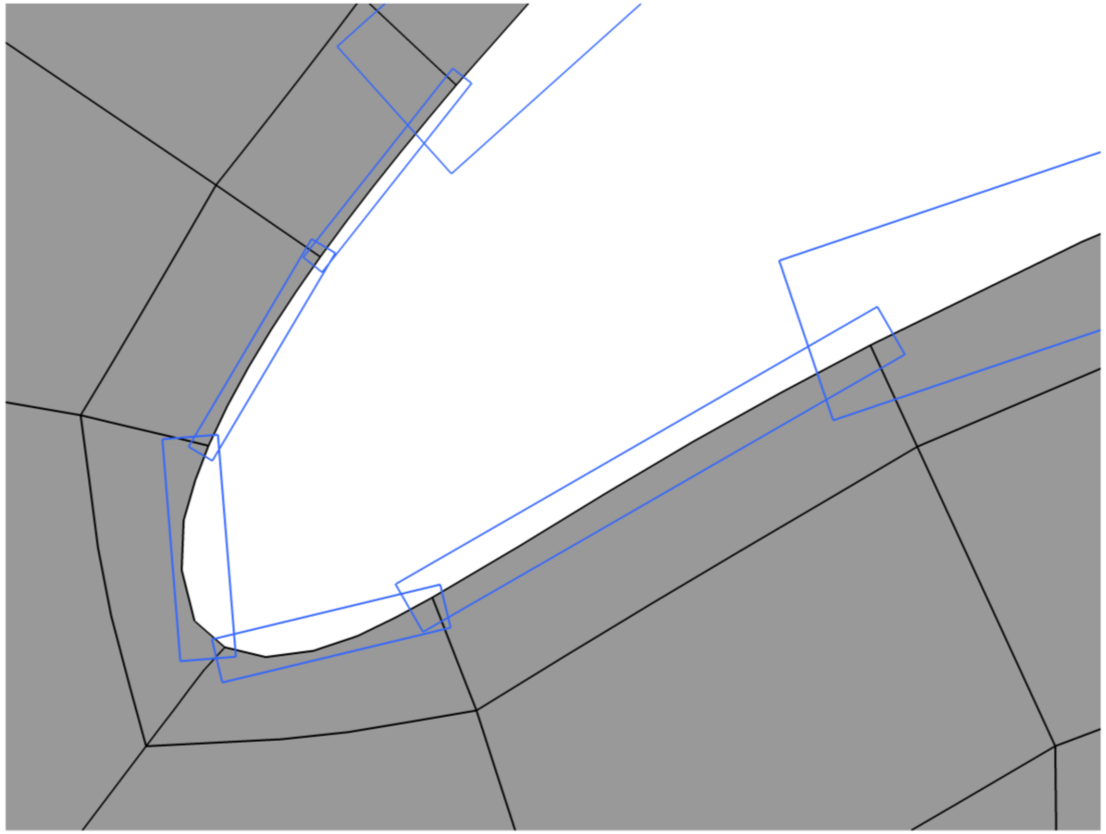


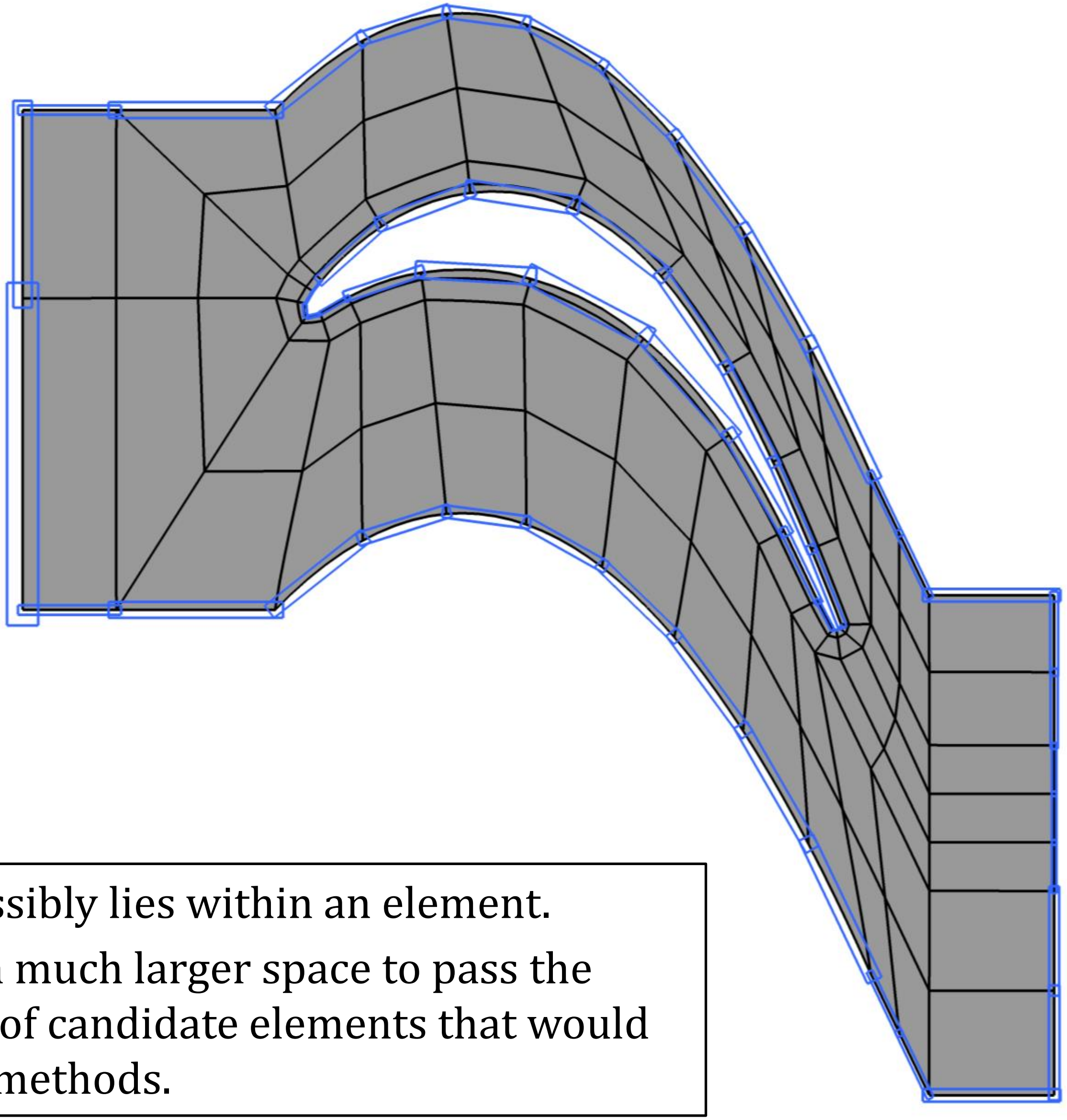
AABBs perfectly aligned with an axis may have zero bounds



- A surface element could be perfectly aligned with one of the physical coordinate axes, resulting in zero-bounding length along the normal direction.
- We provide an additional tolerance in the normal direction for this case.
- This is not an issue for volume elements, since the number of dimensions in the reference and physical space are the same.

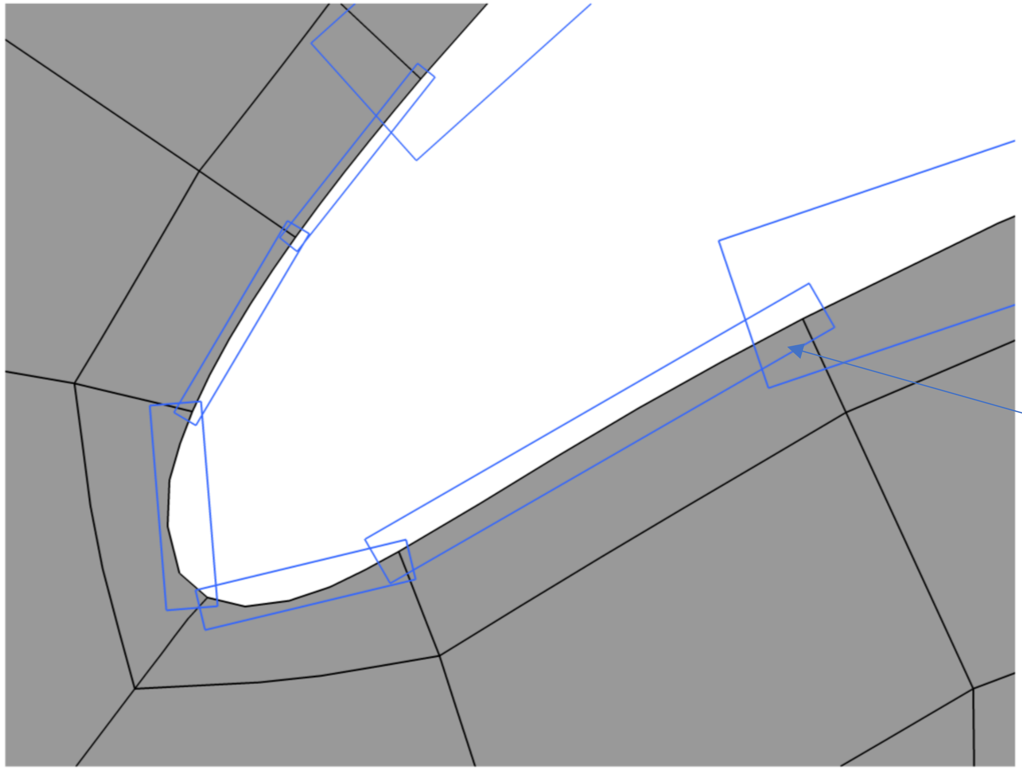
Oriented Bounding Boxes (OBB)



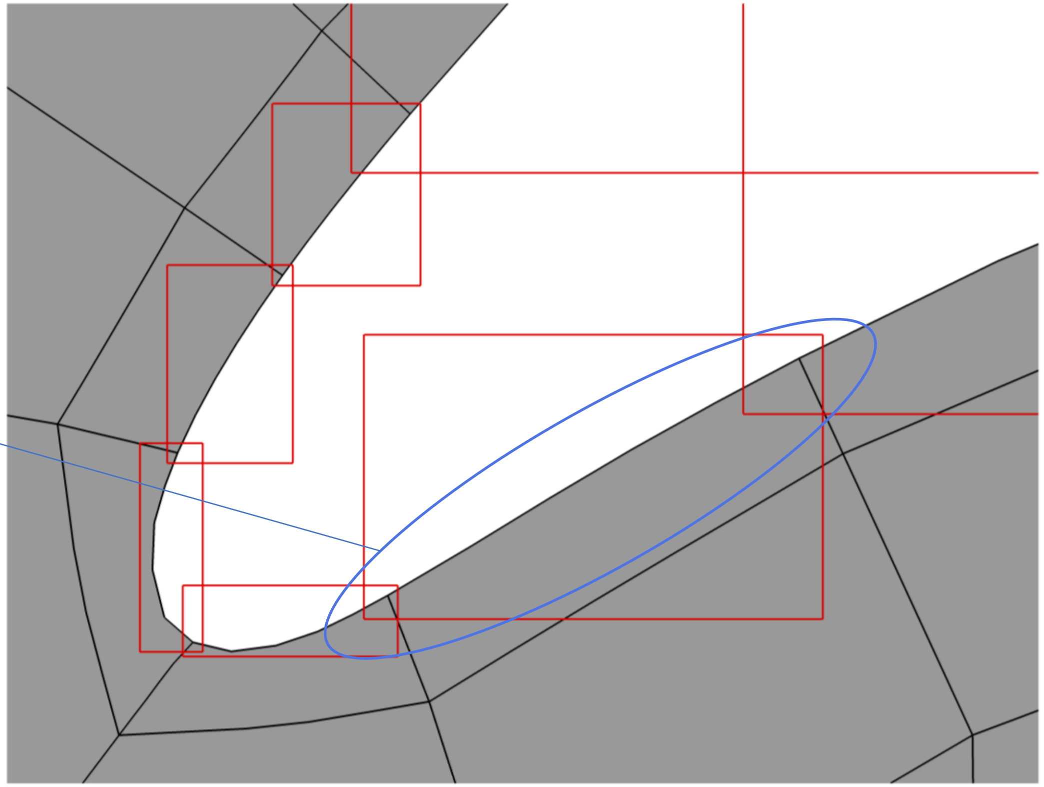


- OBBs provide a stricter yet fast check if a point possibly lies within an element.
- AABBs, compared to OBBs, would allow points in a much larger space to pass the bounding box tests, and thus increase the number of candidate elements that would have to be searched using Newton’s minimization methods.

Procedure for obtaining OBBs in 2D space



OBBs



AABBs

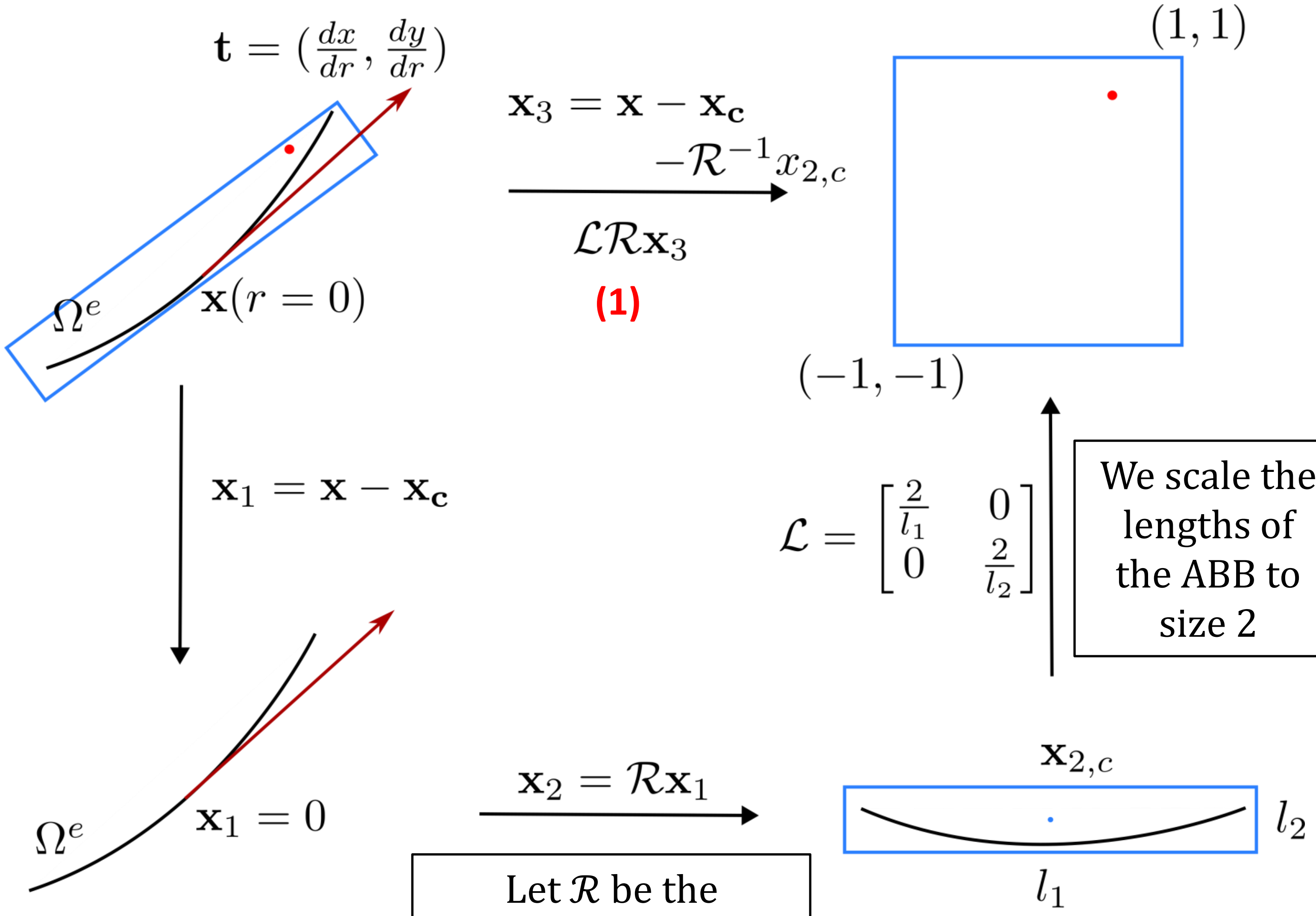
- Depending on the orientation of the element, its AABB may contain more or less volume. When the element is aligned with any of the two axes, the corresponding AABB occupies the smallest volume and is then coincident to an OBB of that same element.

Procedure for obtaining OBBs in 2D space

Obtain the tangential direction (Jacobian) at the element center ($r = 0$).

Translate the element nodes such that the $r = 0$ is at origin.

Given any point inside the OBB of the element Ω^e, its transformation **(2)** lands it inside a square of $[-1,1]^2$.



We scale the lengths of the ABB to size 2

Let $\mathcal{R}$ be the transformation required to align the tangent with x-axis

We calculate the center of this obbox

Procedure for obtaining OBBs in 3D space

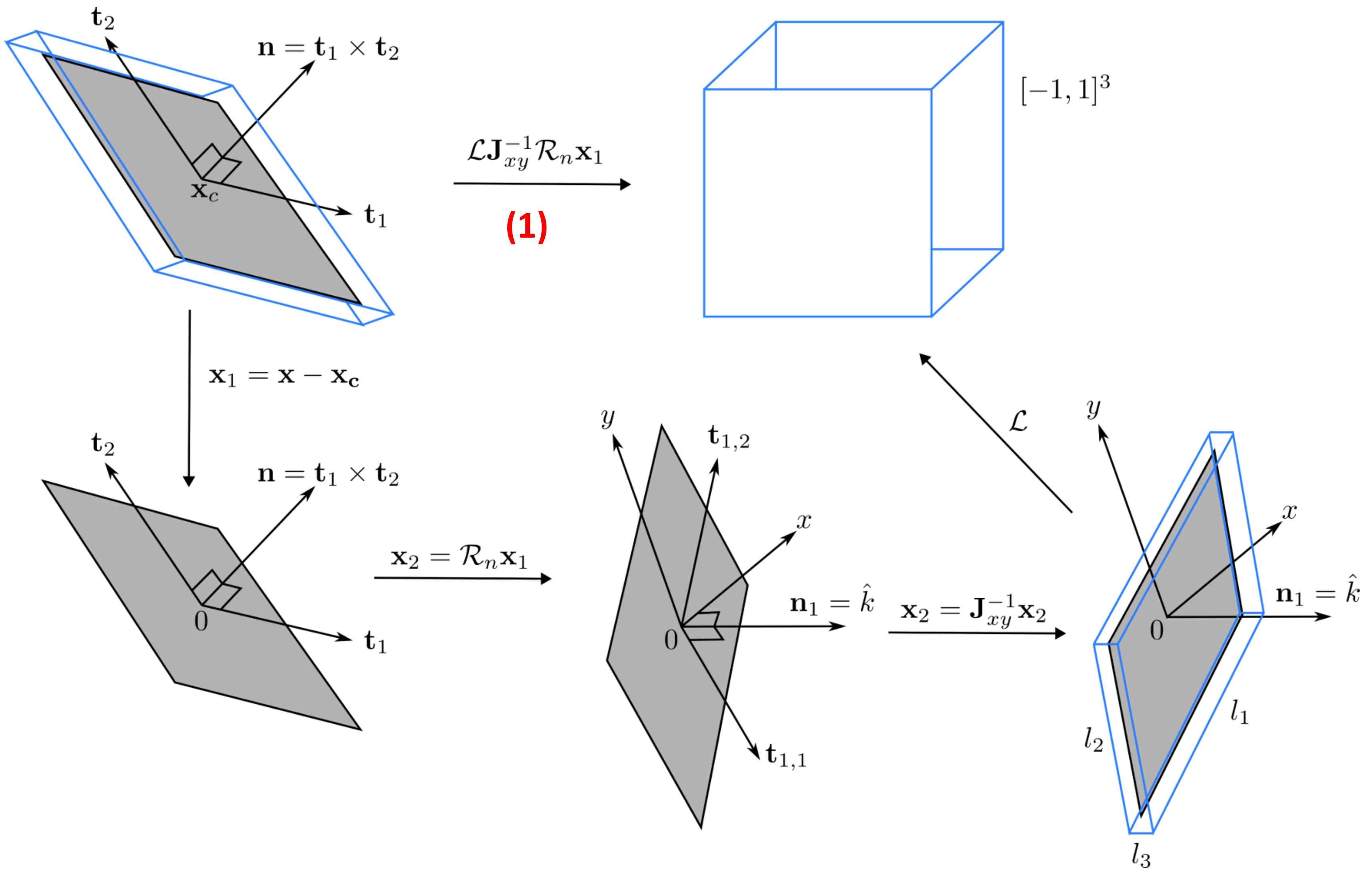
Obtain the tangential direction (Jacobian) at the element center ($r, s = 0{,}0$).

Translate the element nodes such that the x_c is at origin.

Rotate the element such that $\boldsymbol{n}$ is aligned with z.

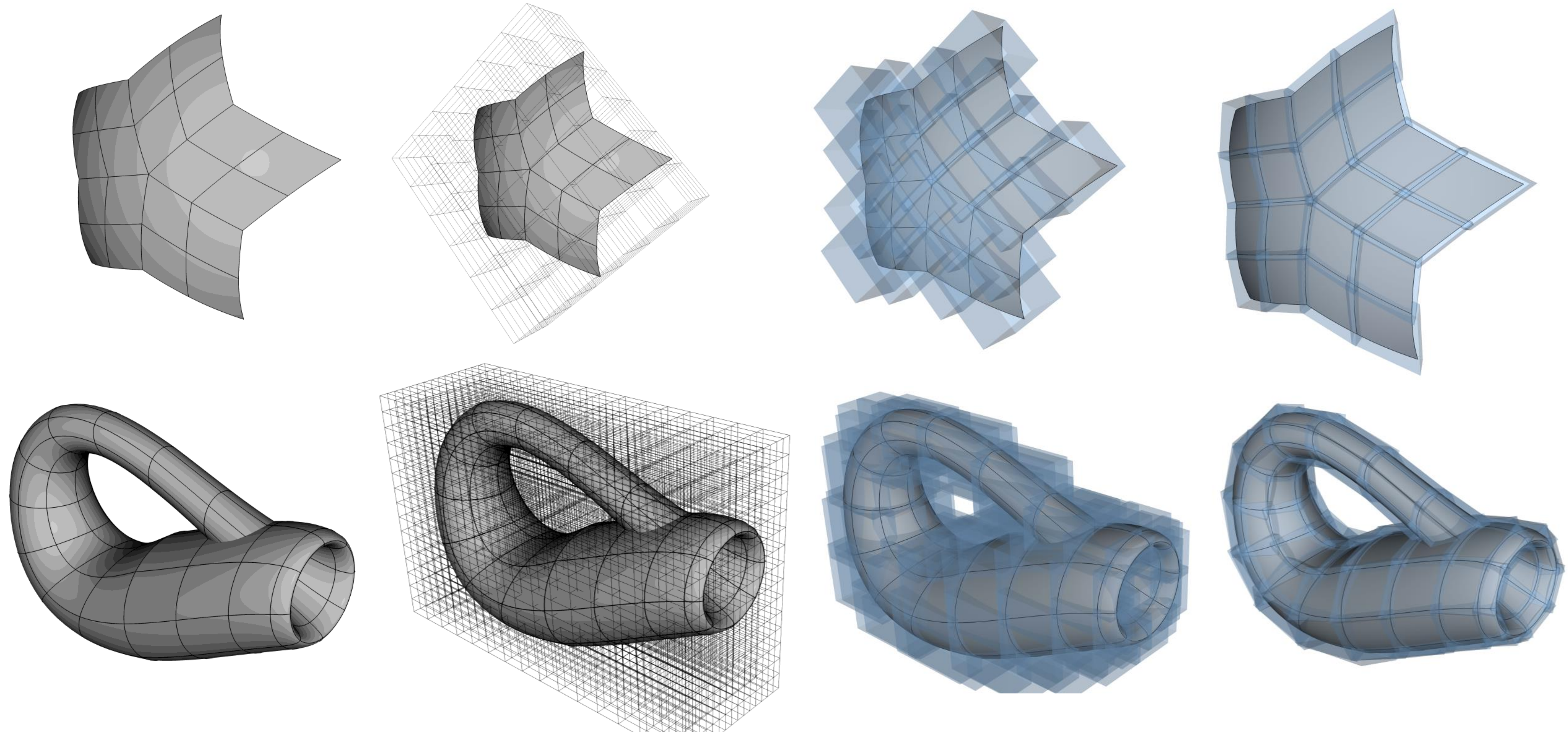
Obtain the Jacobian for the xy-projection of the rotated element and use its inverse to finally orient all 3 axes.

A length scaling transformation is defined.



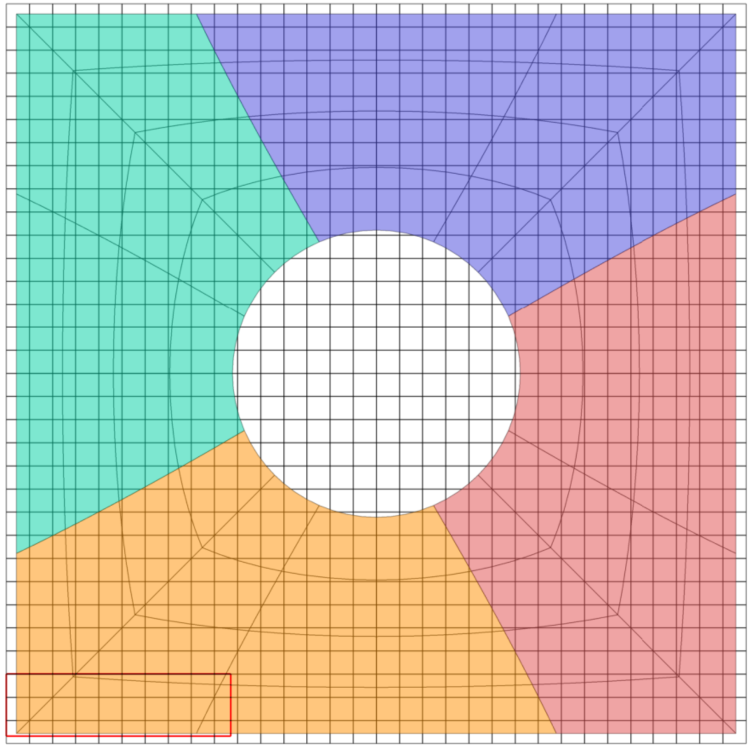
(1) Gives us a single transformation to test if a point lies inside the OBB.

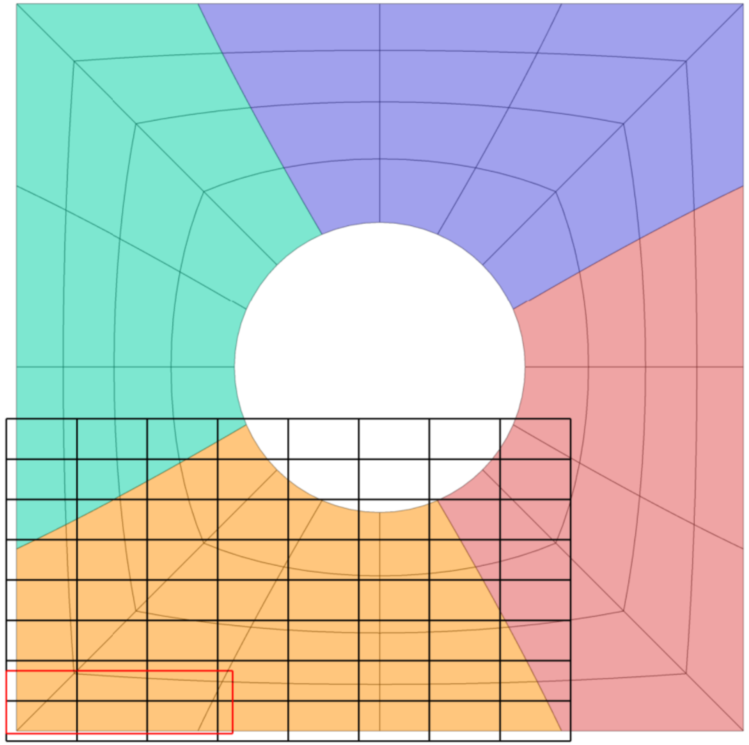
Some OBB tests



The Setup Process: Bounding Boxes

Global Cartesian Mesh $\mathcal{M}_G$





Local Cartesian Mesh $\mathcal{M}_L$

The AABB (in red) contributes to 30 cells of $\mathcal{M}_G$ and the corresponding information to the ranks owning the cells.

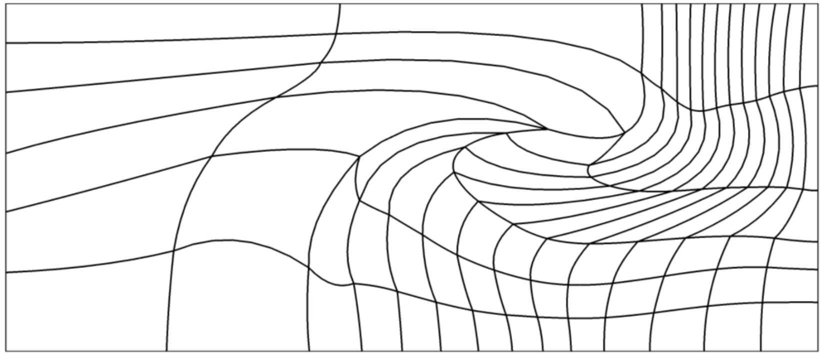
The AABB contributes to 8 cells of $\mathcal{M}_L$ and we store a list of all the elements that contribute to a particular cell.

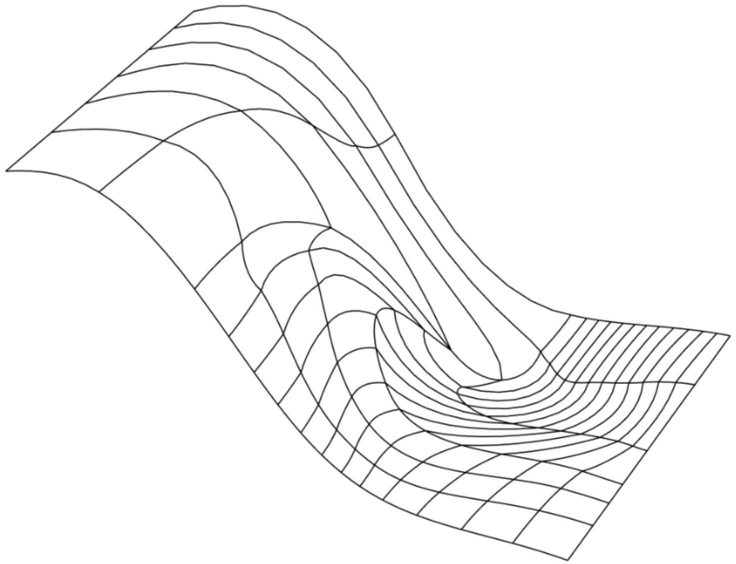
- Given an arbitrary point, we first find the ranks that contribute the the cell of $\mathcal{M}_G$ that contains the point.
- We then query the candidate rank for all the elements (their AABBs) that contribute to the cell of $\mathcal{M}_L$ that contains the point.
- Finally, the OBBs of all candidate elements are tested for the presence of the point, which provides us the final list of elements that must be searched using a Newton's minimization method.

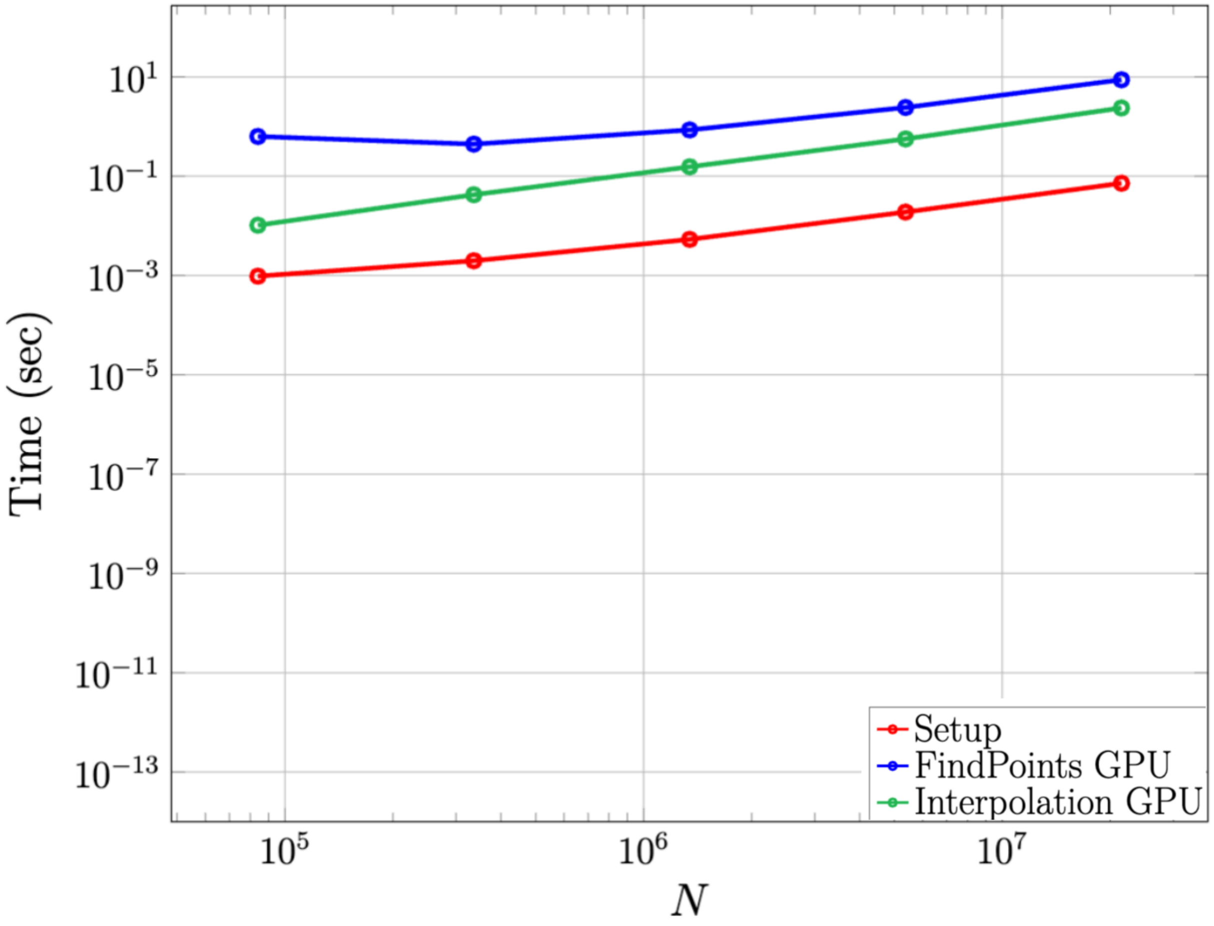
Performance Tests

- 2D Triple Point problem mesh transformed into a $p = 3$ surface mesh.

  - $N = 1000N_E$







Thank You