

Geração de Malhas de Elementos Finitos

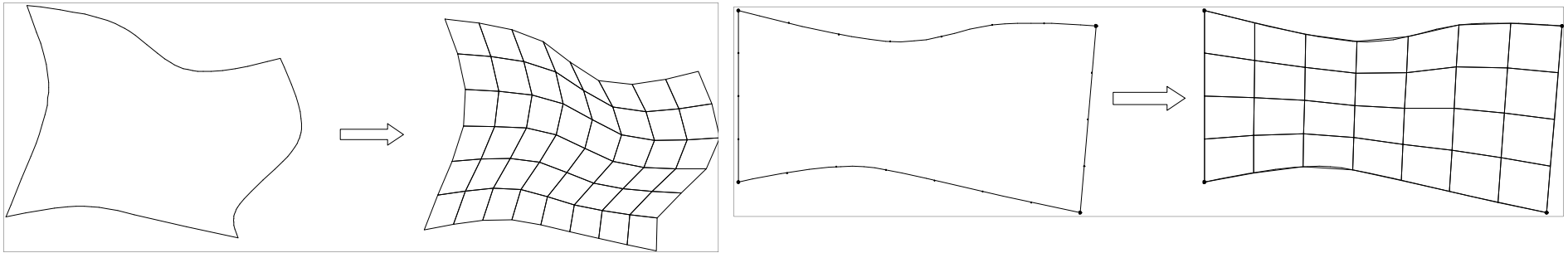
Luiz Fernando Martha
André Pereira

CIV 2802 – Sistemas Gráficos para Engenharia
Departamento de Engenharia Civil e Ambiental – PUC-Rio
2025.1

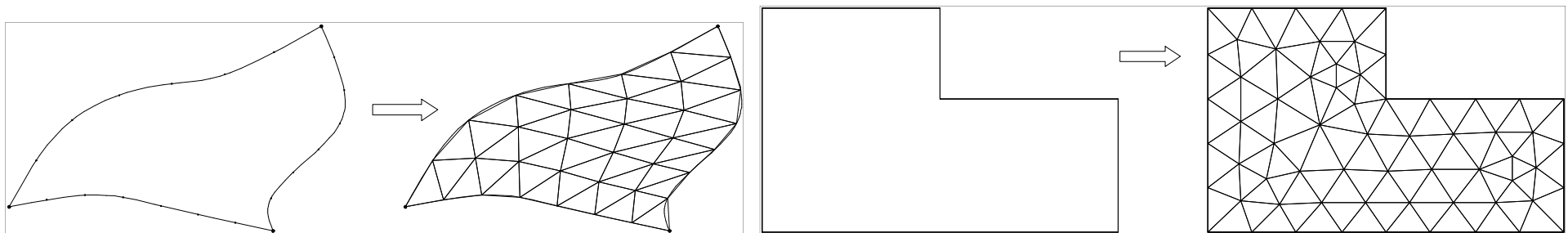


Library of mesh generation algorithms

2D structured meshes

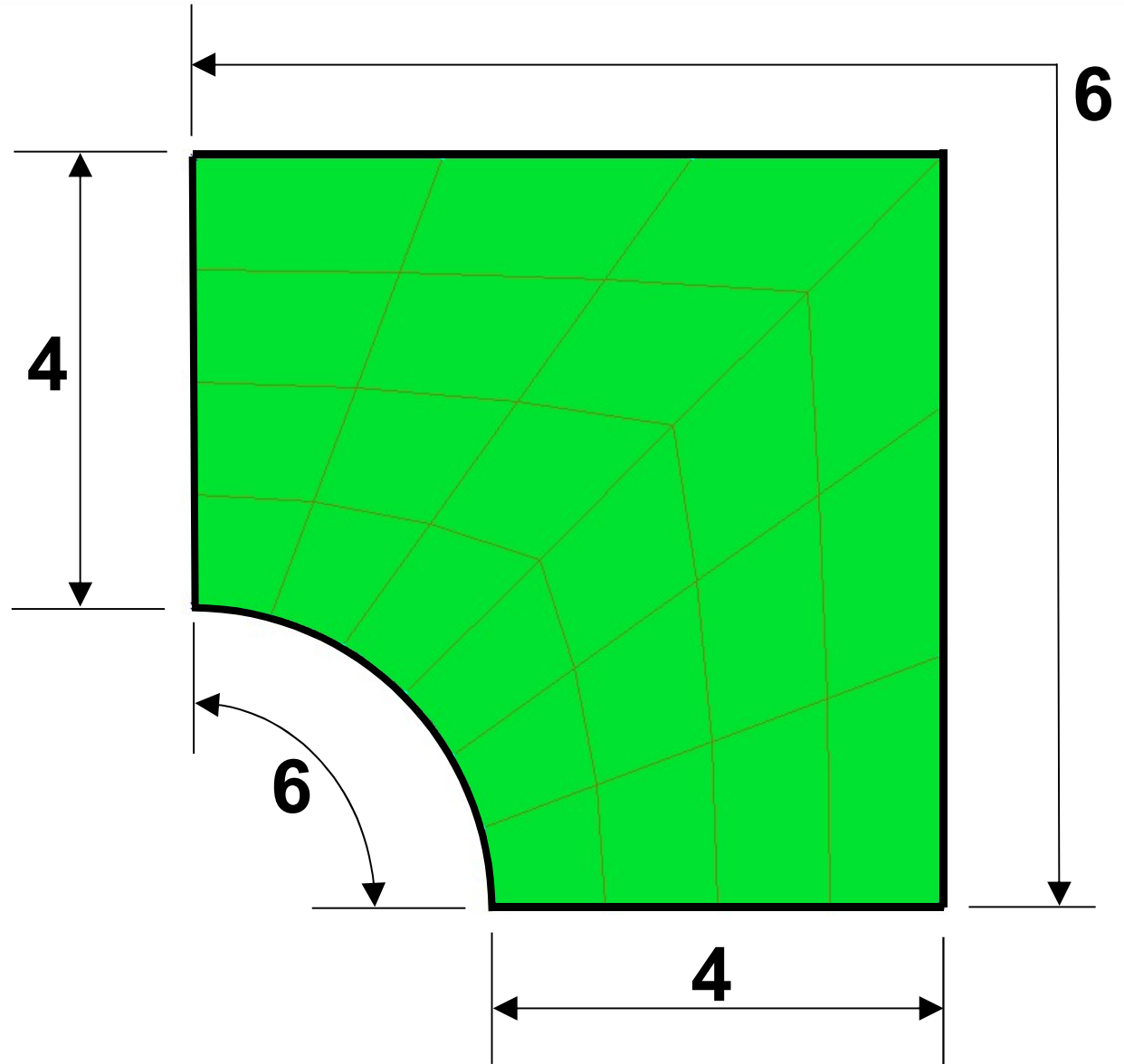


2D structured and non-structured meshes



Structured mesh – 2D Mapping

- **Geometry Requirements**
 - 4 topological sides
 - Opposite sides must have similar discretization





INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 17, 1015–1044 (1981)

A GENERAL TWO-DIMENSIONAL, GRAPHICAL FINITE ELEMENT PREPROCESSOR UTILIZING DISCRETE TRANSFINITE MAPPINGS

ROBERT HABER‡

University of Illinois, Urbana, Illinois, U.S.A.

MARK S. SHEPHARD‡

Rensselaer Polytechnic Institute, Troy, New York, U.S.A.

JOHN F. ABEL§

Cornell University, Ithaca, New York, U.S.A.

RICHARD H. GALLAGHER||

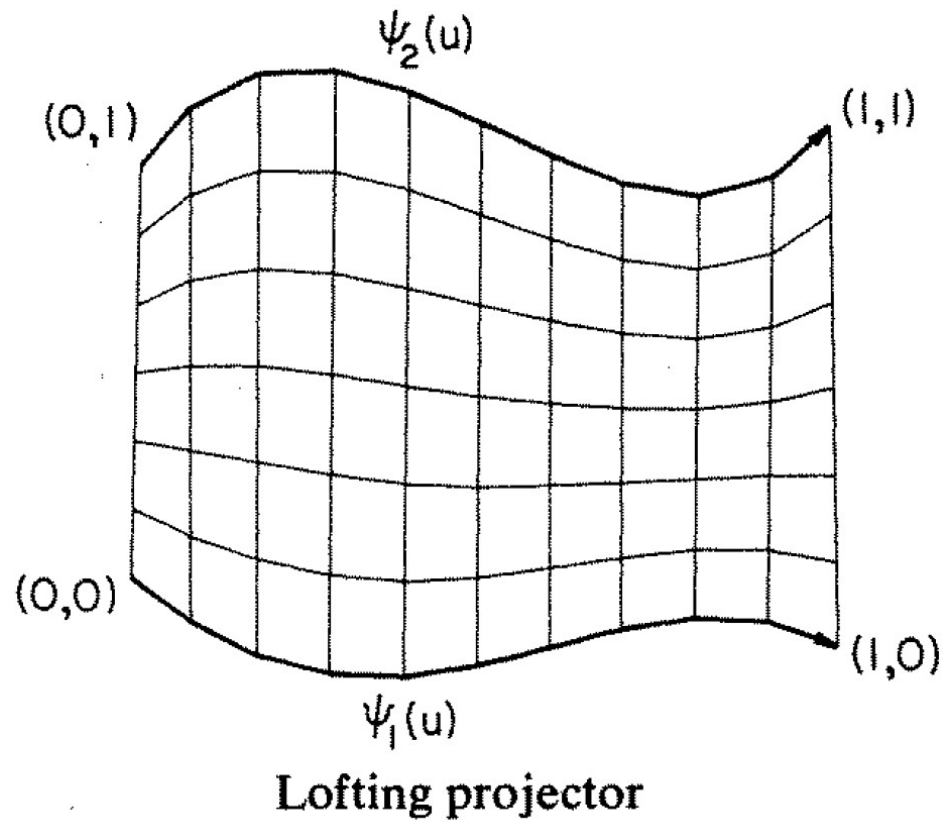
University of Arizona, Tucson, Arizona, U.S.A.

AND

DONALD P. GREENBERG¶

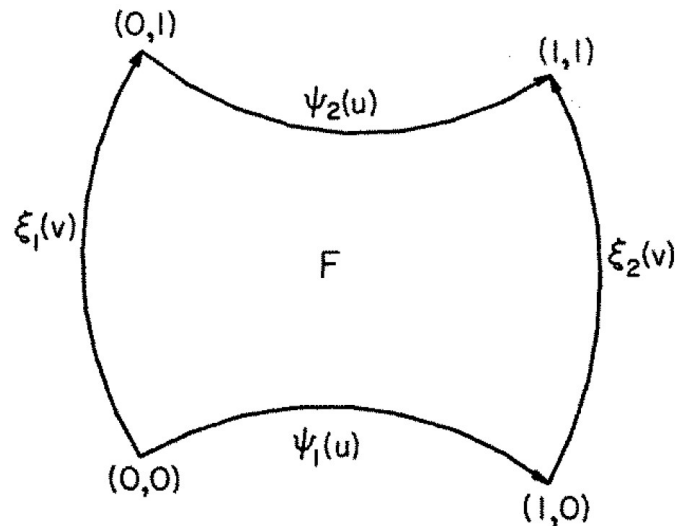
Cornell University, Ithaca, New York, U.S.A.

Structured mesh – 2D Mapping

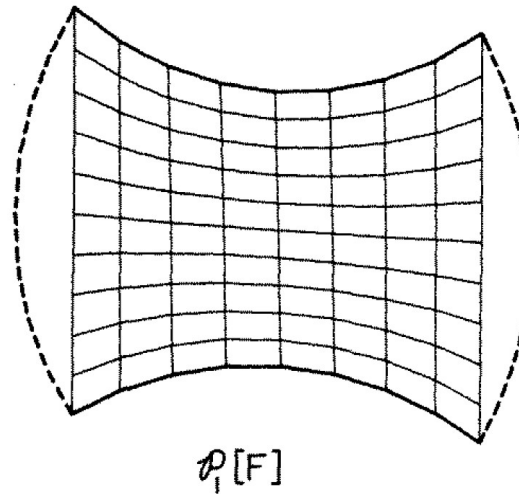


$$\mathcal{P}_1[F] \equiv P_2(u, v) = (1 - v)\psi_1(u) + v\psi_2(u) \quad 0 \leq u \leq 1$$

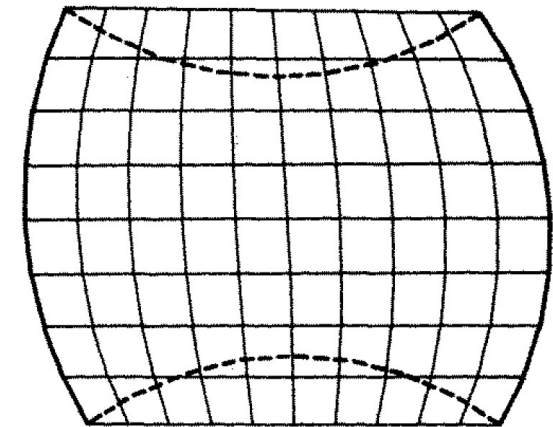
Structured mesh – 2D Mapping



Bilinear projector: co-ordinate system and boundary curves



Bilinear projector: \mathcal{P}_1

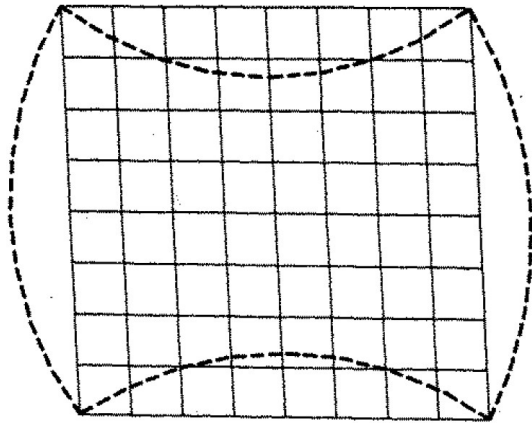


$\mathcal{P}_2[F]$
Bilinear projector: \mathcal{P}_2

$$\mathcal{P}_1[F] \equiv P_2(u, v) = (1-v)\psi_1(u) + v\psi_2(u) \quad 0 \leq u \leq 1$$

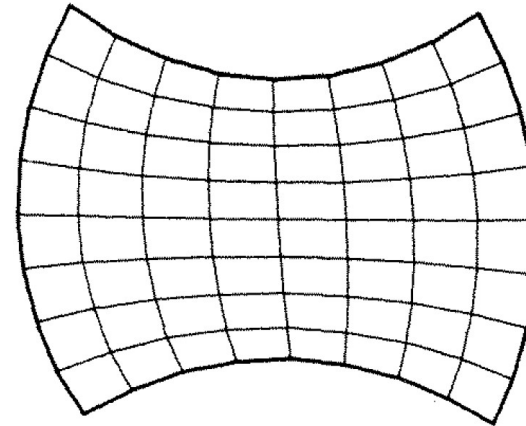
$$\mathcal{P}_2[F] \equiv P_2(u, v) = (1-u)\xi_1(v) + u\xi_2(v) \quad 0 \leq v \leq 1$$

Structured mesh – 2D Mapping



$$\mathcal{P}_1 \mathcal{P}_2 [F]$$

Bilinear projector: $\mathcal{P}_1 \mathcal{P}_2$



$$\mathcal{P}_1 \oplus \mathcal{P}_2$$

Bilinear projector: $\mathcal{P}_1 \oplus \mathcal{P}_2$

$$(\mathcal{P}_1 \oplus \mathcal{P}_2)[F] \equiv \mathcal{P}_1[F] + \mathcal{P}_2[F] - \mathcal{P}_1 \mathcal{P}_2[F]$$

$$= P_B(u, v)$$

$$= (1-v)\psi_1(u) + v\psi_2(u) + (1-u)\xi_1(v) + u\xi_2(v)$$

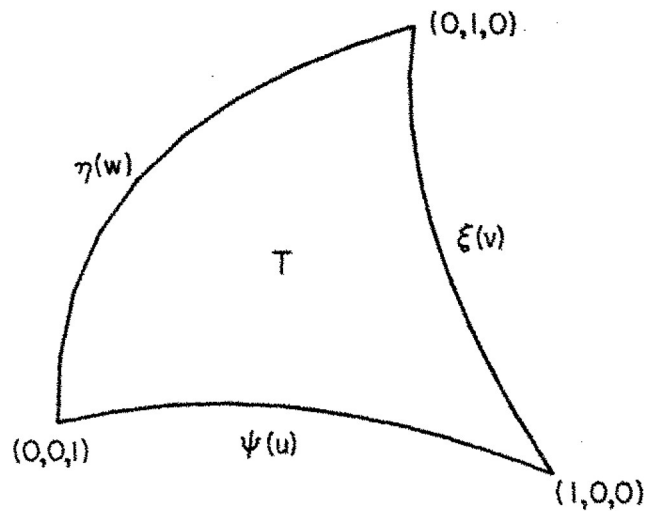
$$- (1-u)(1-v)F(0, 0) - u(1-v)F(0, 1)$$

$$- uvF(1, 1) - (1-u)vF(1, 0) \quad 0 \leq u \leq 1, 0 \leq v \leq 1$$

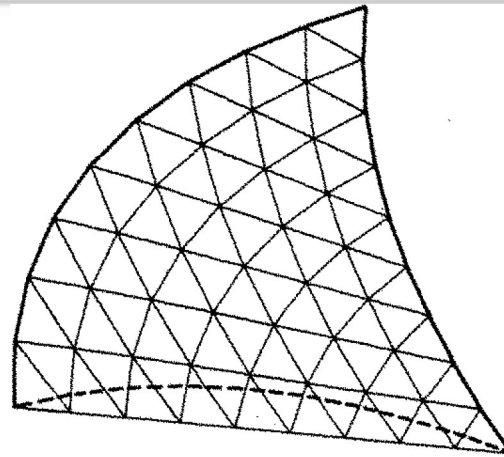
Assumed discrete representation of curves:

$$\{\xi_1(v_i), \xi_2(v_i)\}_{i=1, n}, \quad \{\psi_1(u_j), \psi_2(u_j)\}_{j=1, m}$$

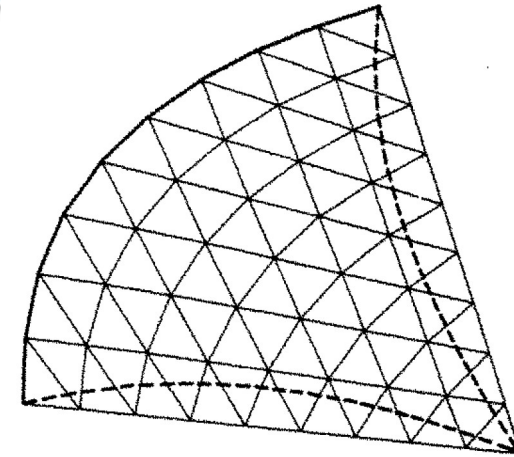
Structured mesh – 2D Mapping



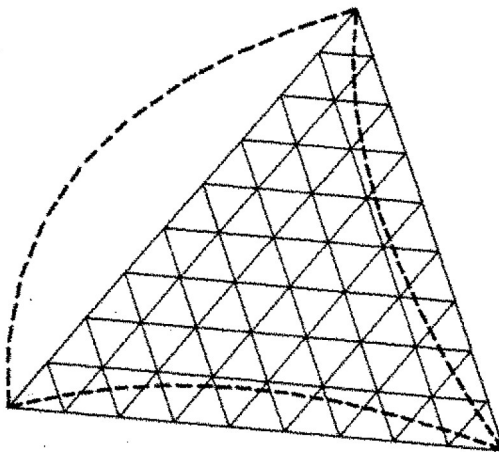
Trilinear projector: co-ordinate system and boundary curves



$\eta_1[T]$
Trilinear projector: \mathcal{N}_1



$\eta_1\eta_2[T]$
Trilinear projector: $\mathcal{N}_1\mathcal{N}_2$



$\eta_1\eta_2\eta_3[T]$
Trilinear projector: $\mathcal{N}_1\mathcal{N}_2\mathcal{N}_3$

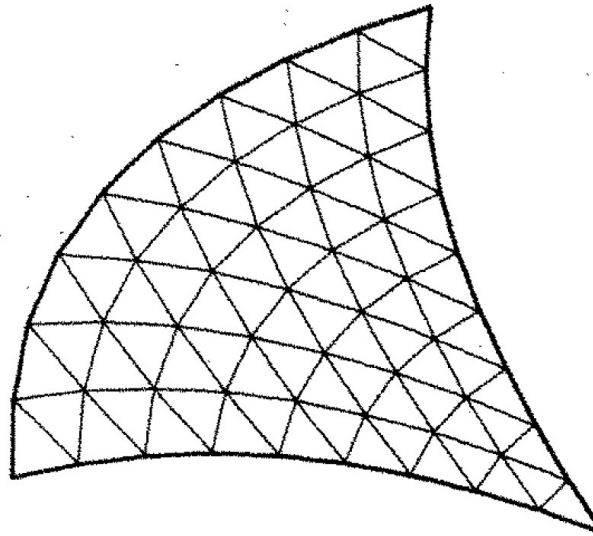
$$\mathcal{N}_1 \equiv N_1(u, v, w) = \left(\frac{u}{1-v} \right) \xi(v) + \left(\frac{w}{1-v} \right) \eta(1-v)$$

$$\mathcal{N}_2 \equiv N_2(u, v, w) = \left(\frac{v}{1-w} \right) \eta(w) + \left(\frac{u}{1-w} \right) \psi(1-w)$$

$$\mathcal{N}_3 \equiv N_3(u, v, w) = \left(\frac{w}{1-u} \right) \psi(u) + \left(\frac{v}{1-u} \right) \xi(1-u)$$

$$0 \leq u \leq 1, \quad 0 \leq v \leq 1, \quad 0 \leq w \leq 1, \quad u + v + w = 1$$

Structured mesh – 2D Mapping



$Q[T]$

Trilinear projector: \mathcal{Q}

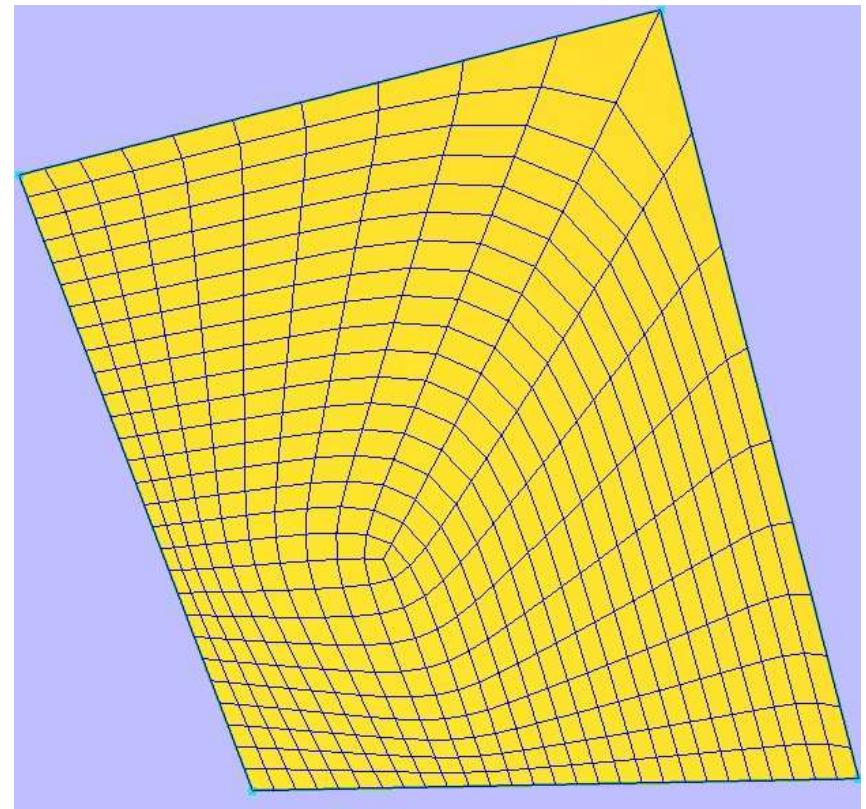
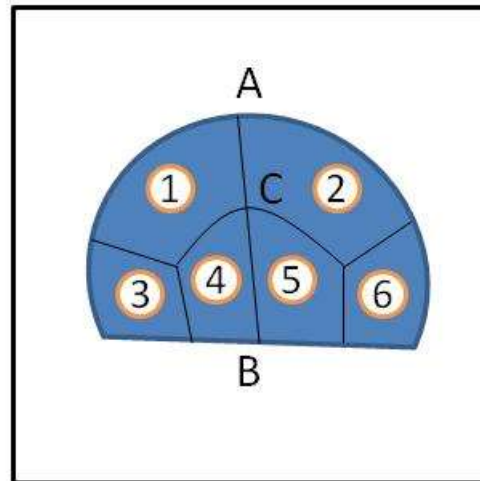
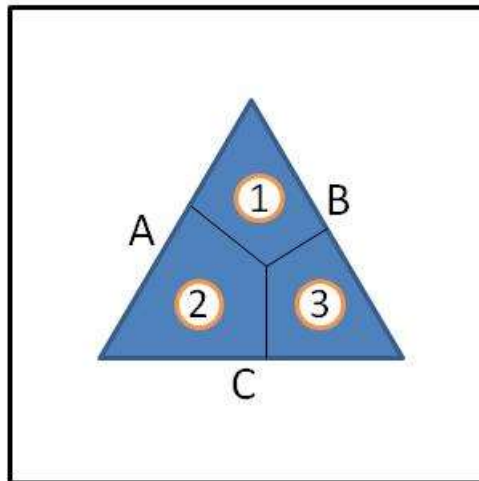
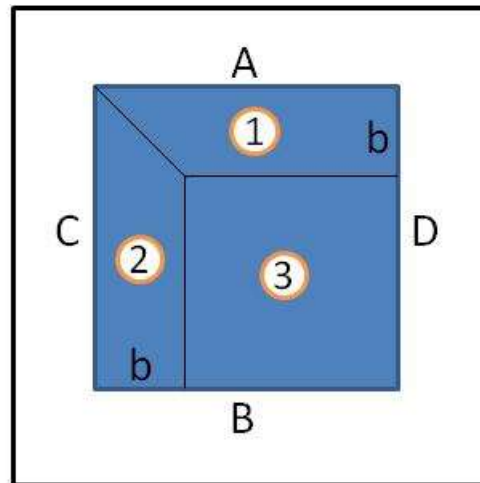
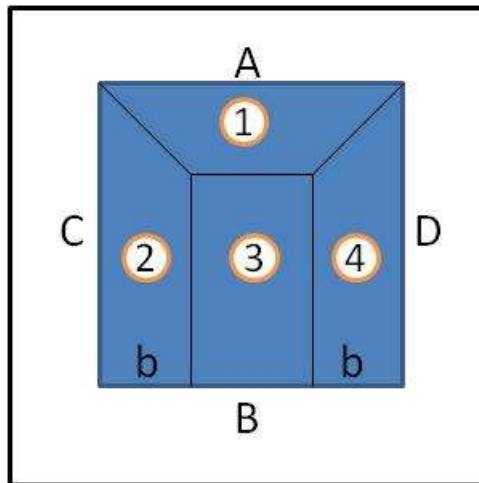
$$\begin{aligned}\mathcal{Q} \equiv Q(u, v, w) = & \frac{1}{2} \left[\left(\frac{u}{1-v} \right) \xi(v) + \left(\frac{w}{1-v} \right) \eta(1-v) + \left(\frac{v}{1-w} \right) \eta(w) + \left(\frac{u}{1-w} \right) \psi(1-w) \right. \\ & \left. + \left(\frac{w}{1-u} \right) \psi(u) + \left(\frac{v}{1-u} \right) \xi(1-u) - w\psi(0) - u\xi(0) - v\eta(0) \right]\end{aligned}$$

Assumed discrete representation of curves:

$$\{\psi(u_i), \xi(v_i), \eta(w_i); i = 1, n\}$$

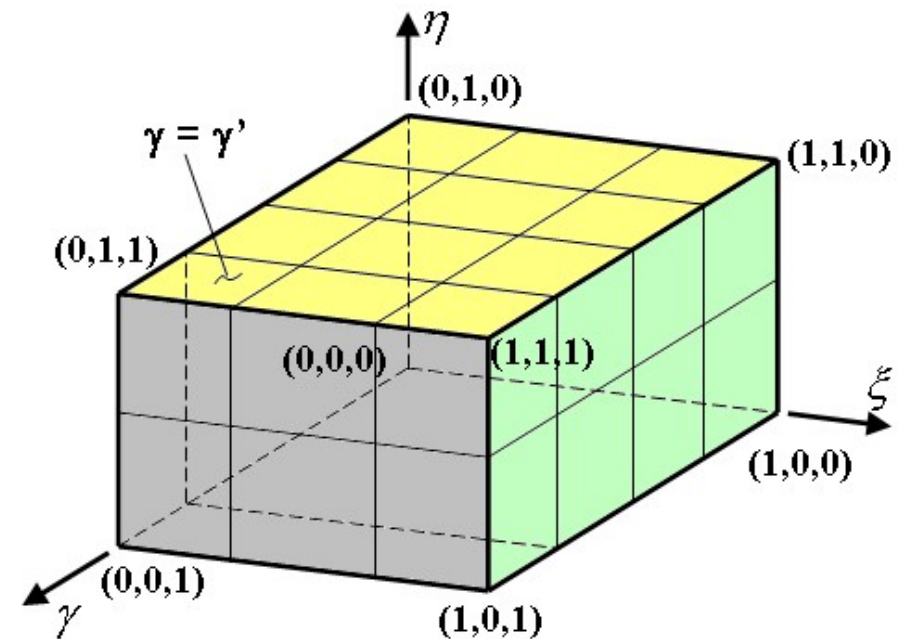
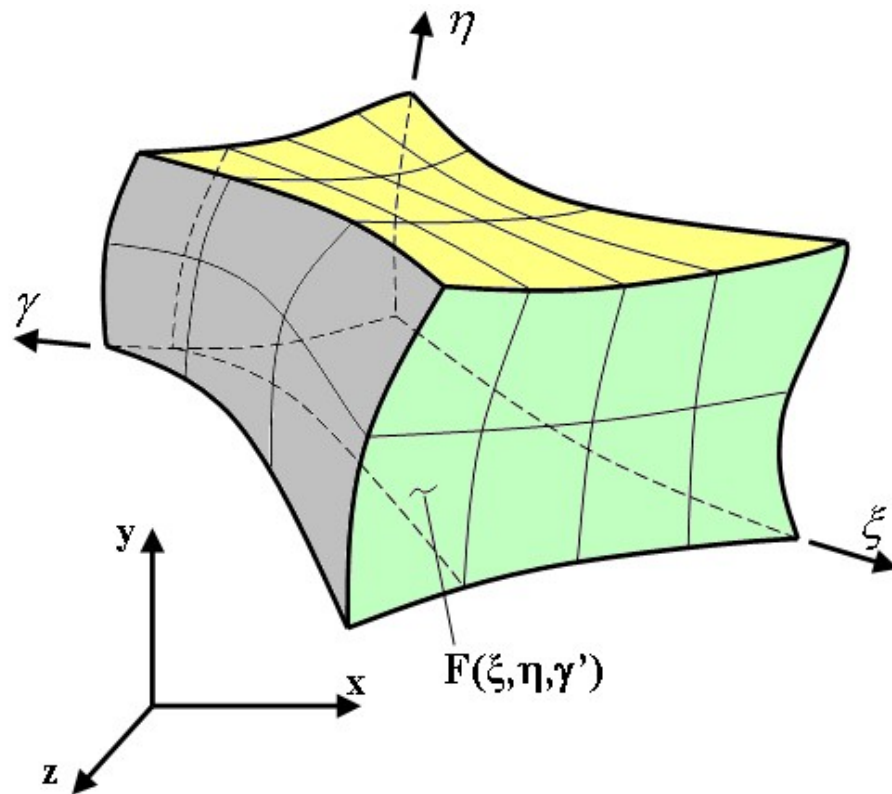
Library of mesh generation algorithms

Quadrilateral template (new)



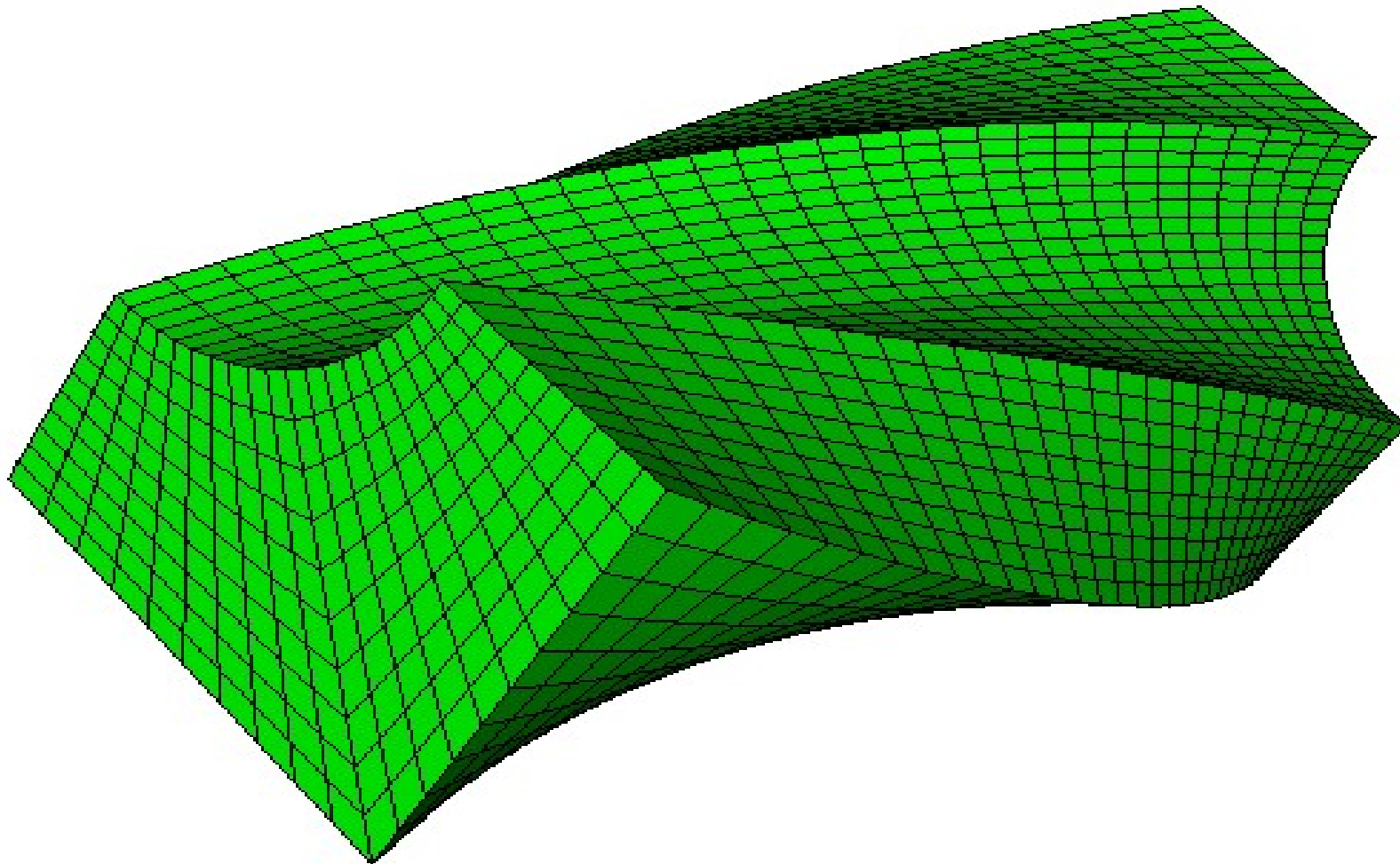
Structured mesh – 3D Mapping

- **Geometry Requirements**
 - 6 topological surfaces
 - Opposite surfaces must have similar mapped meshes



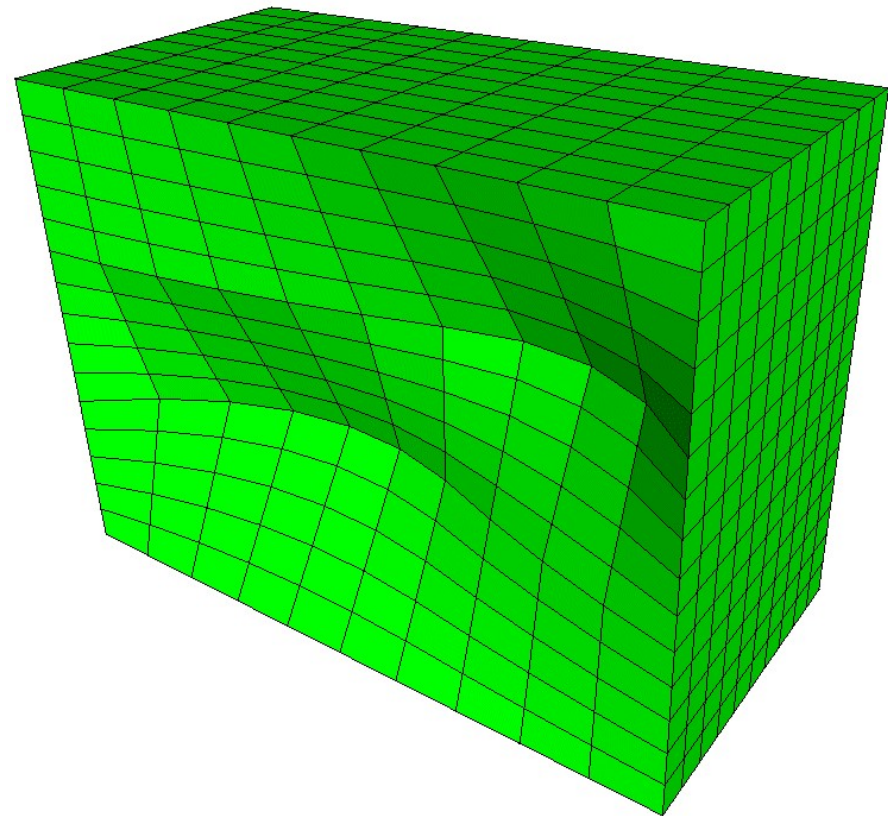
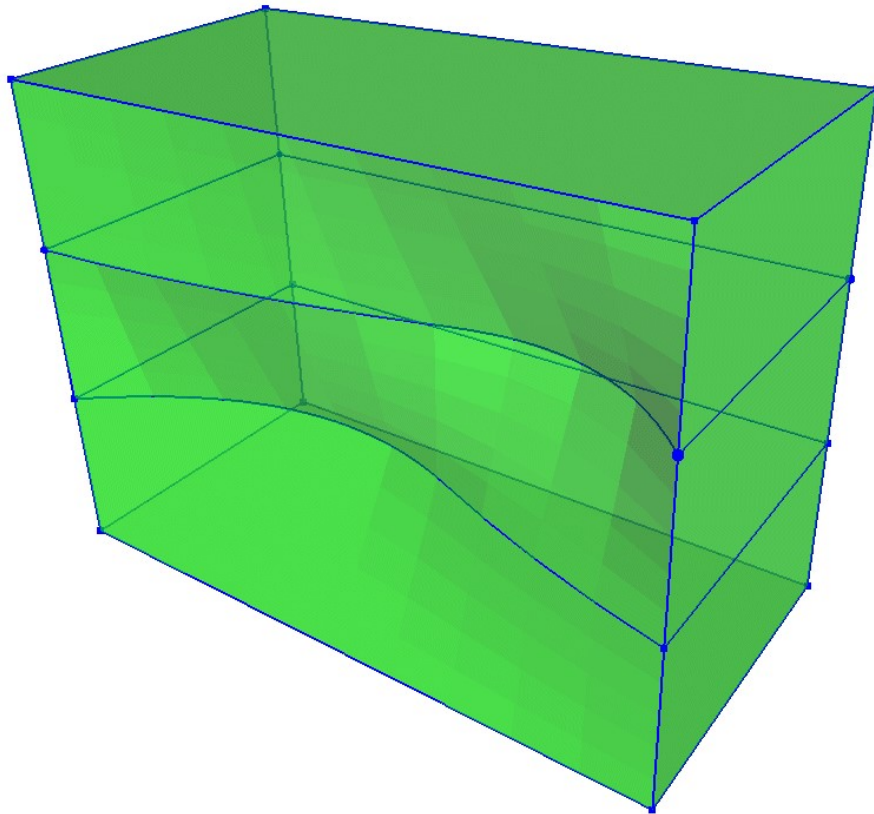
Structured mesh – 3D Mapping

- **Many complex domains can be mapped**



Structured mesh – 3D Mapping

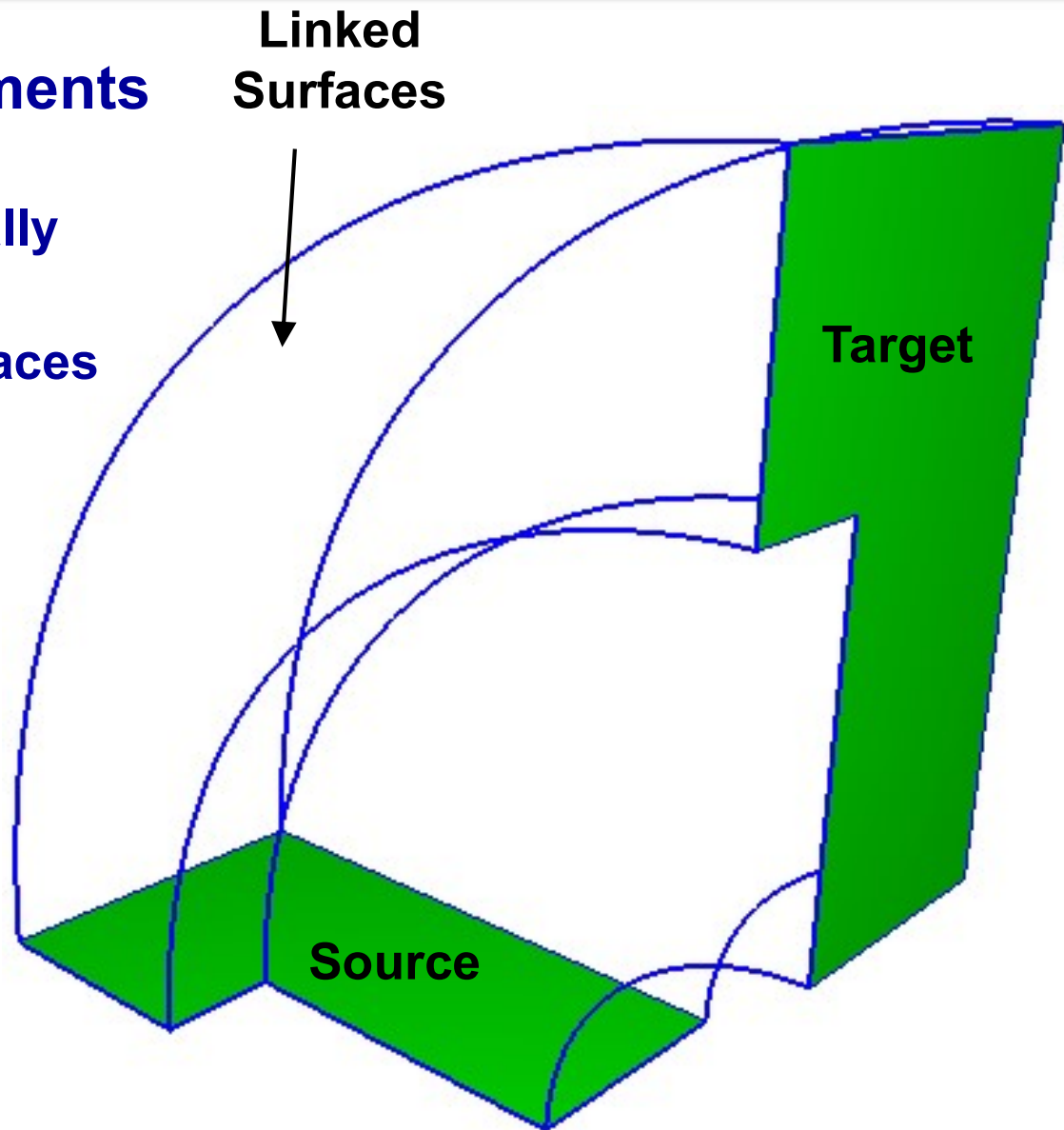
- **Algorithm must deal with:**
 - Multiple surfaces on boundary
 - Concave surfaces



Structured mesh – Sweeping

- **Geometry Requirements**

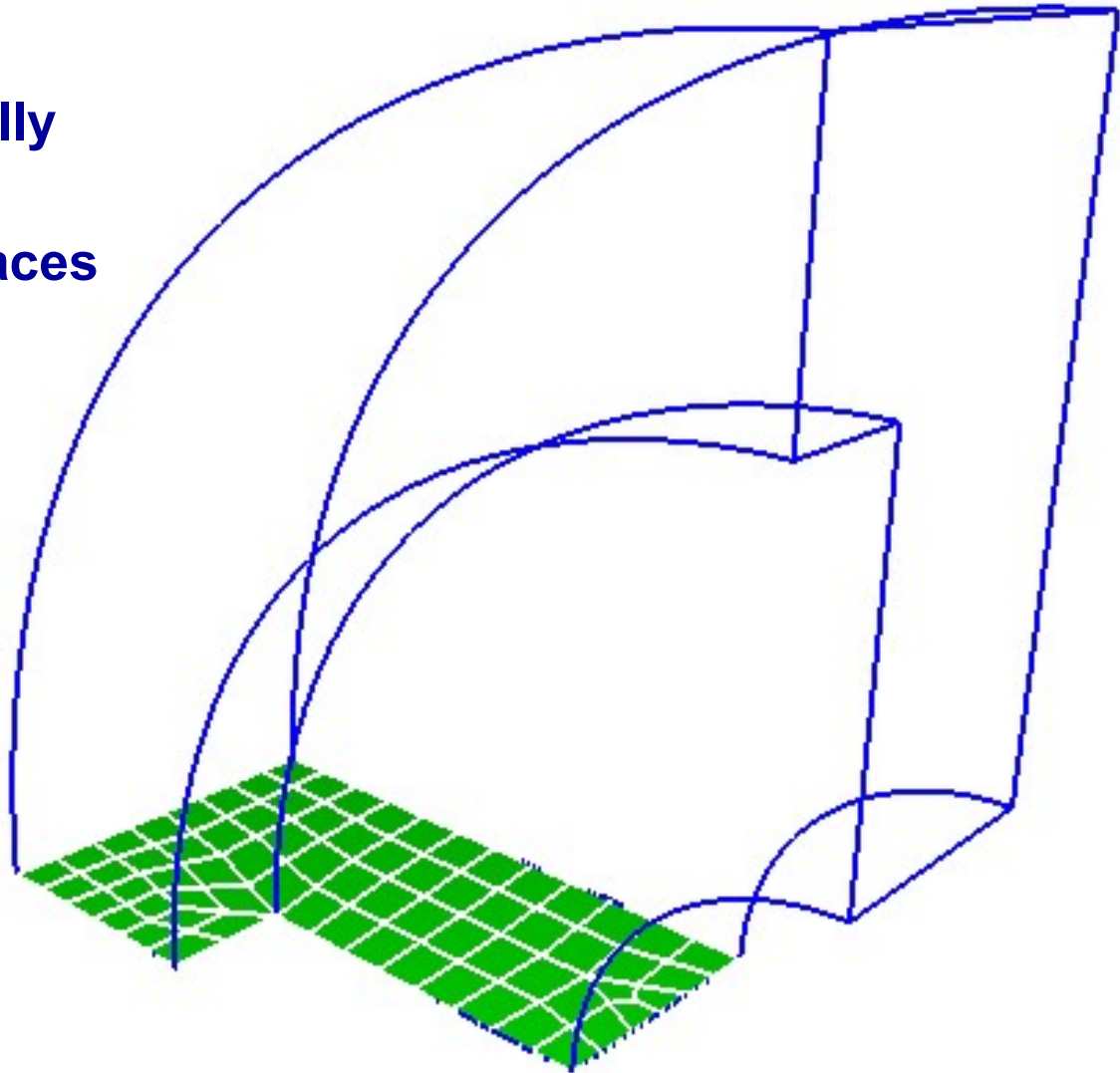
- Source and target surfaces topologically similar
- Mapped linked surfaces



Structured mesh – Sweeping

- **Geometry Requirements**

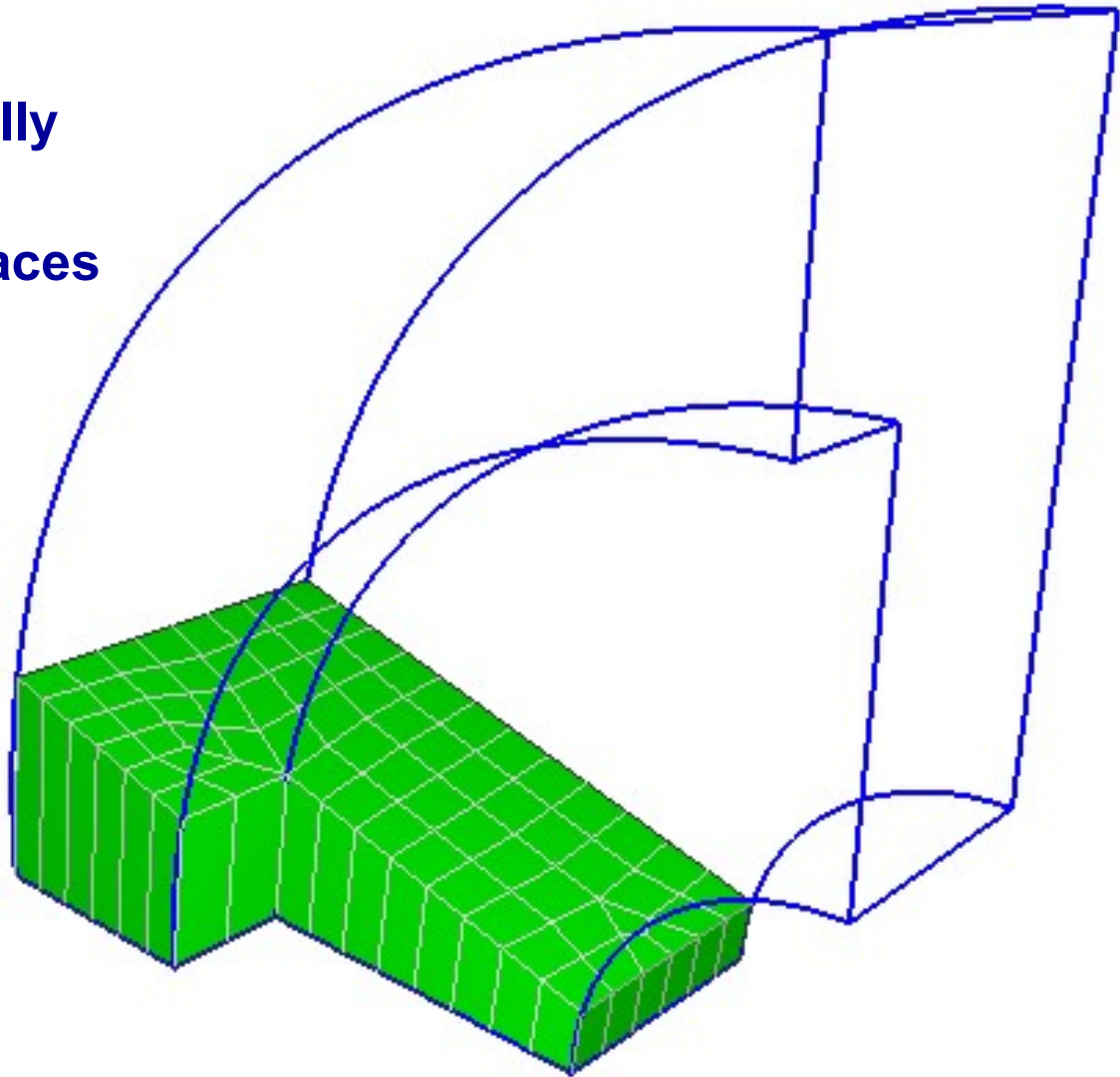
- Source and target surfaces topologically similar
- Mapped linked surfaces



Structured mesh – Sweeping

- **Geometry Requirements**

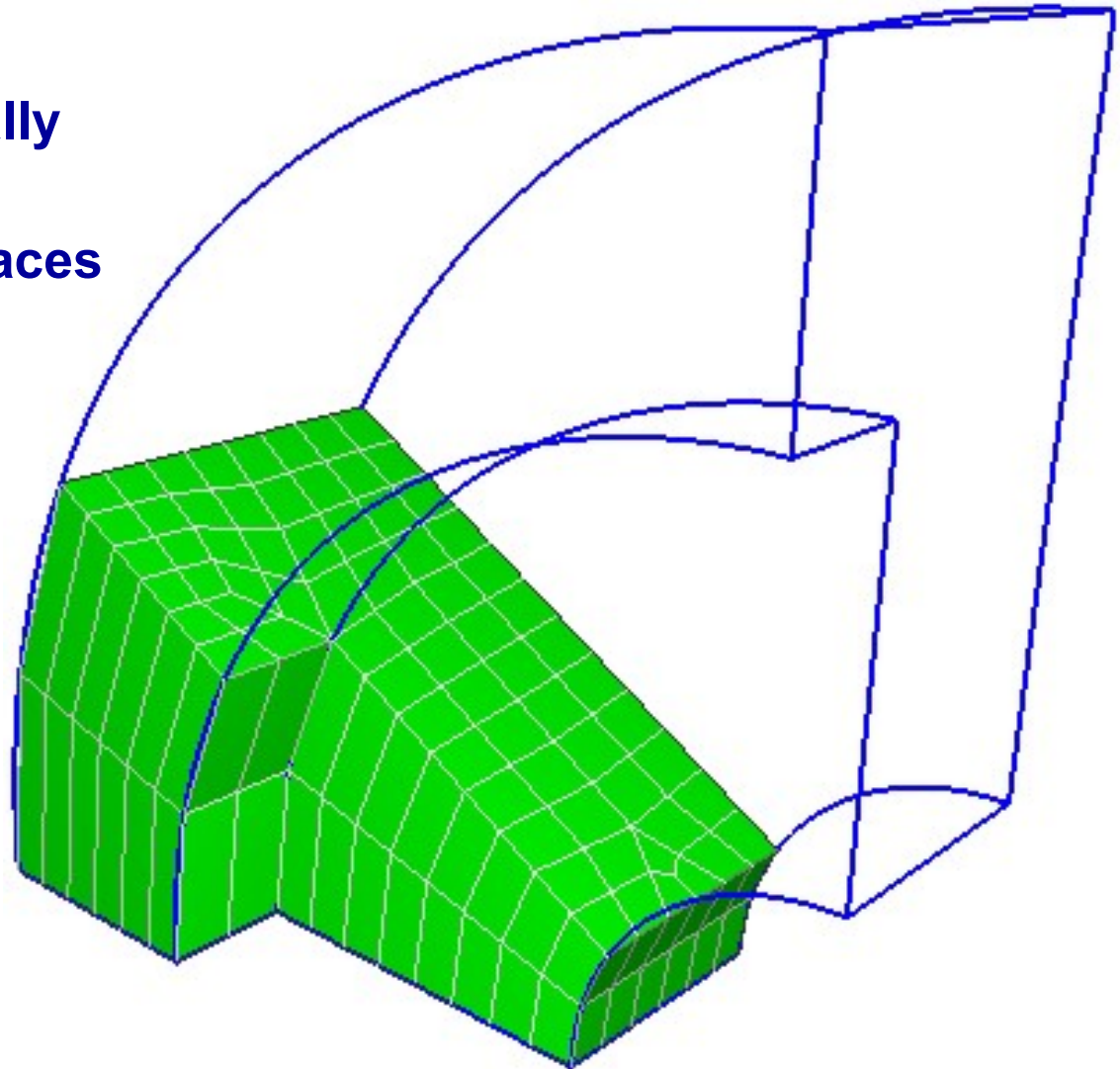
- Source and target surfaces topologically similar
- Mapped linked surfaces



Structured mesh – Sweeping

- **Geometry Requirements**

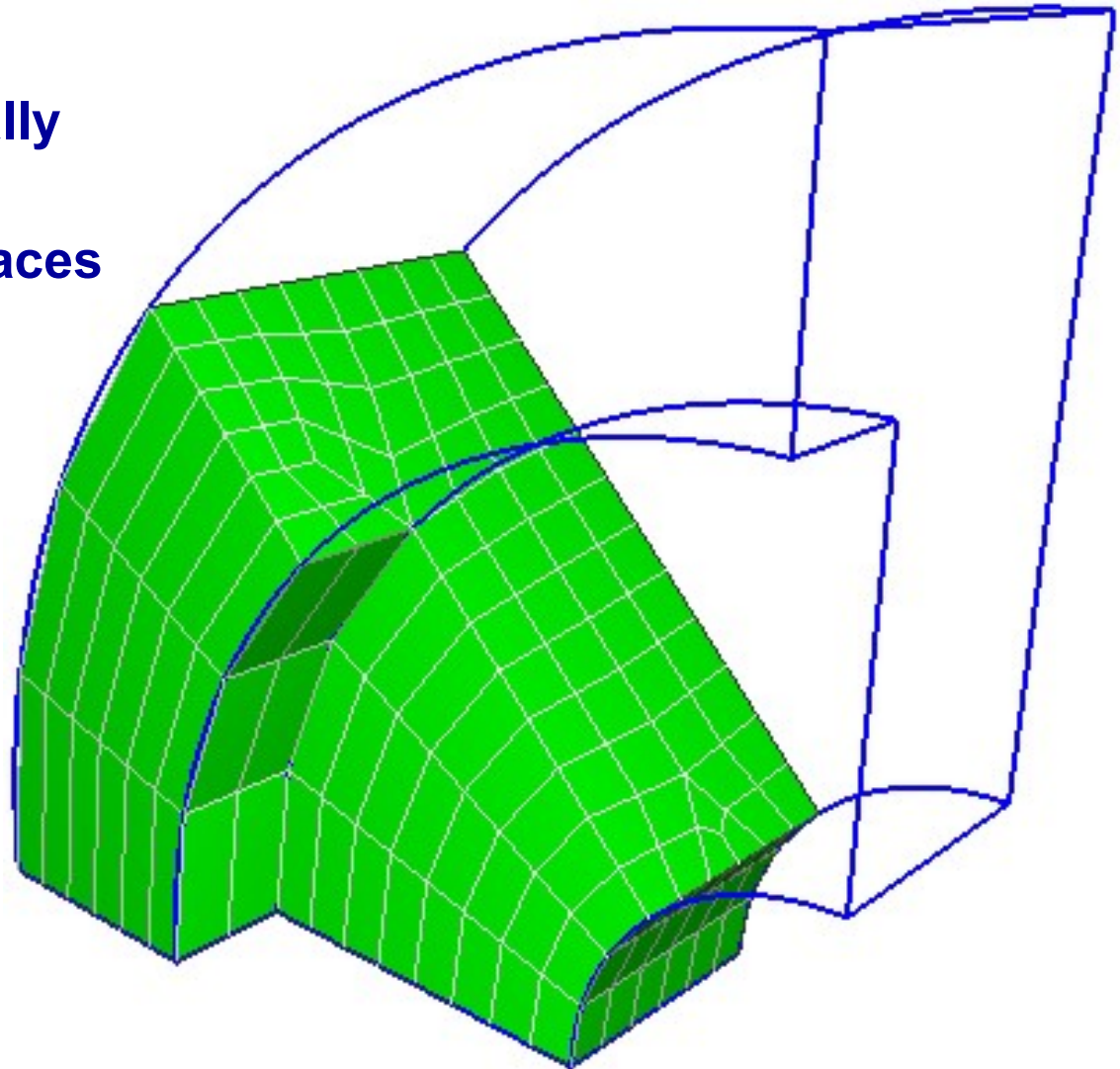
- Source and target surfaces topologically similar
- Mapped linked surfaces



Structured mesh – Sweeping

- **Geometry Requirements**

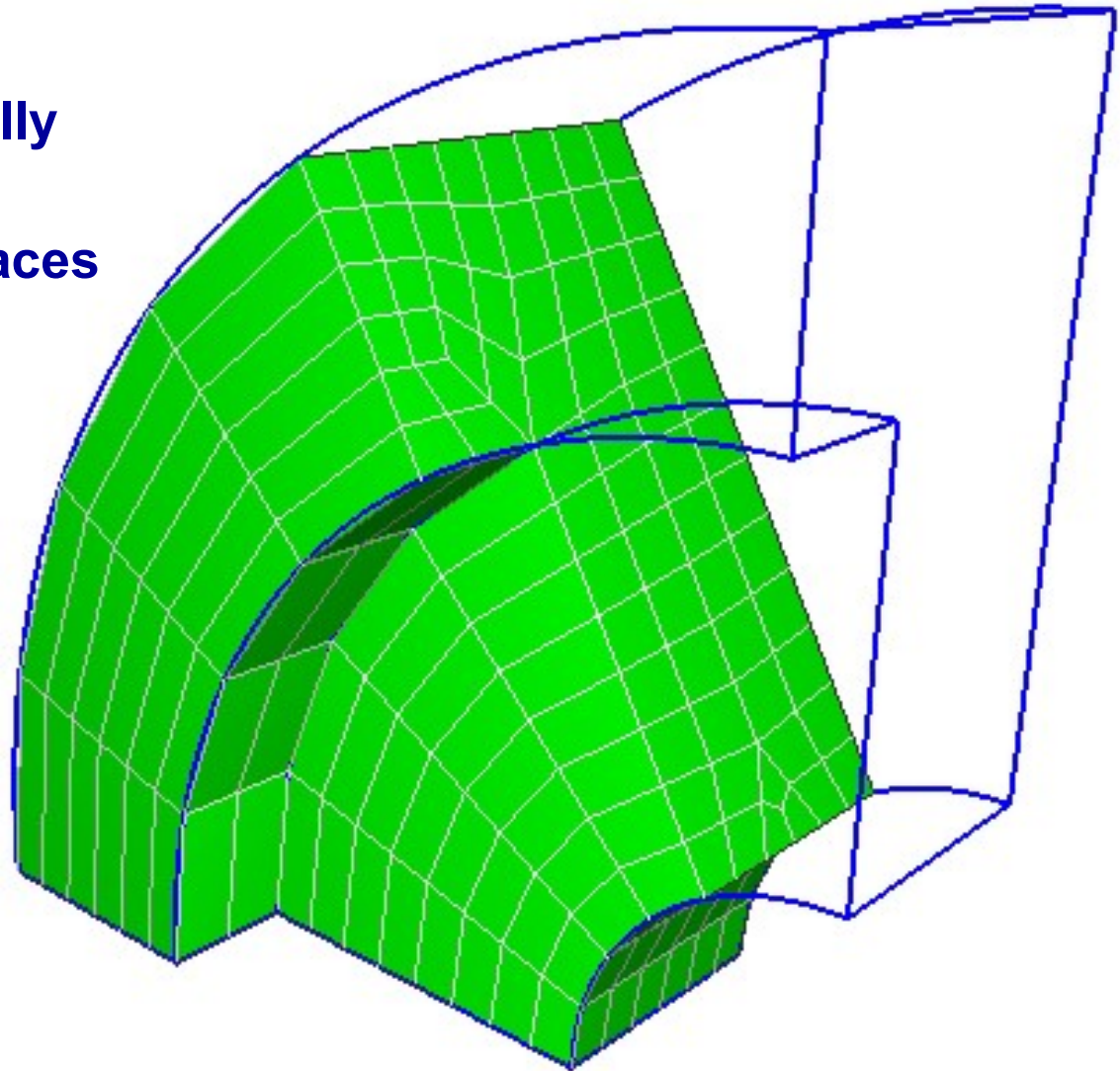
- Source and target surfaces topologically similar
- Mapped linked surfaces



Structured mesh – Sweeping

- **Geometry Requirements**

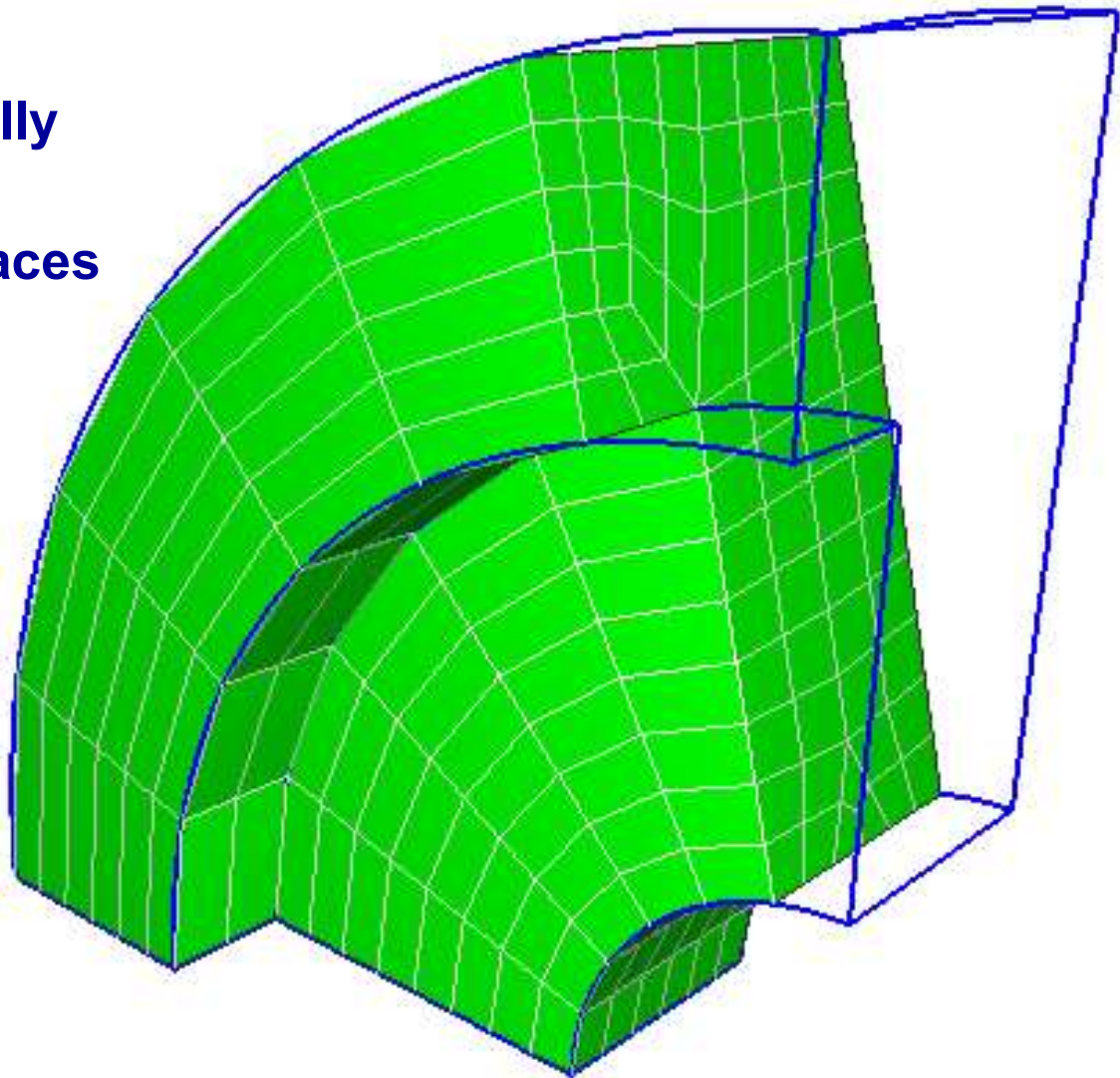
- Source and target surfaces topologically similar
- Mapped linked surfaces



Structured mesh – Sweeping

- **Geometry Requirements**

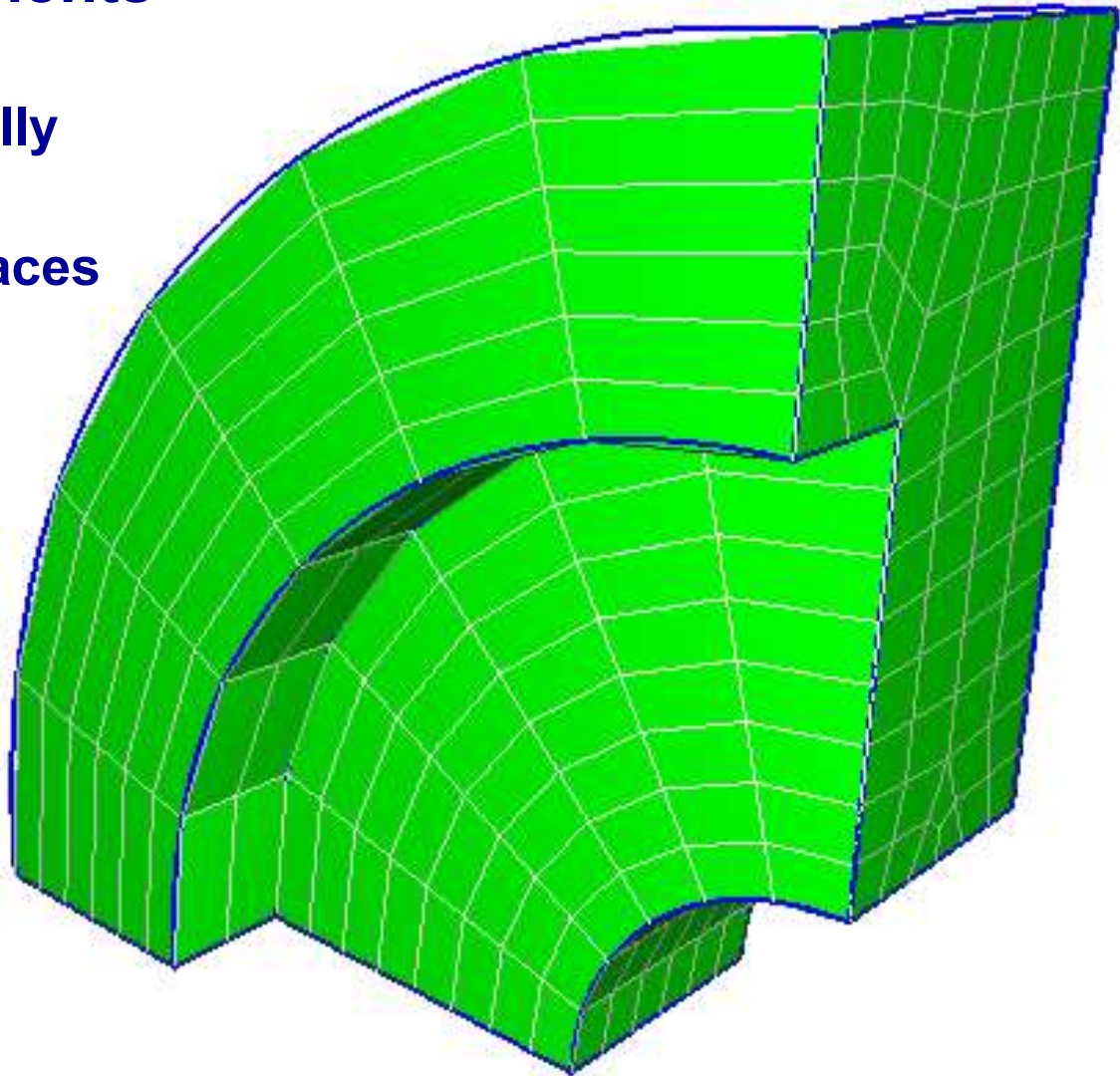
- Source and target surfaces topologically similar
- Mapped linked surfaces



Structured mesh – Sweeping

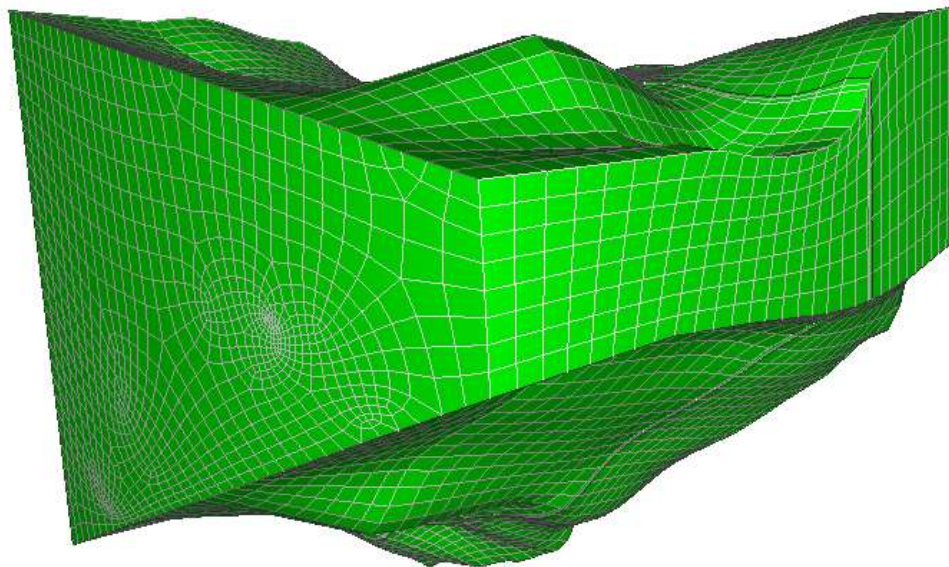
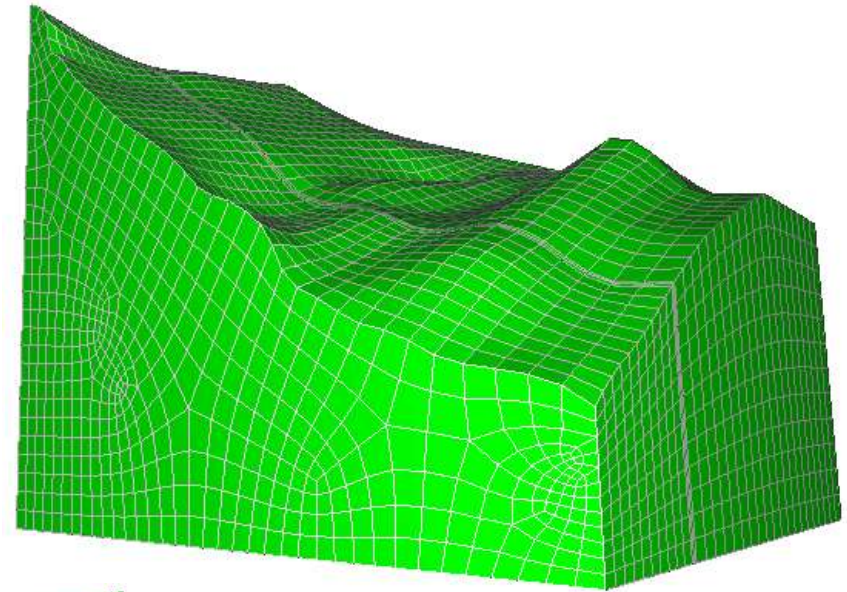
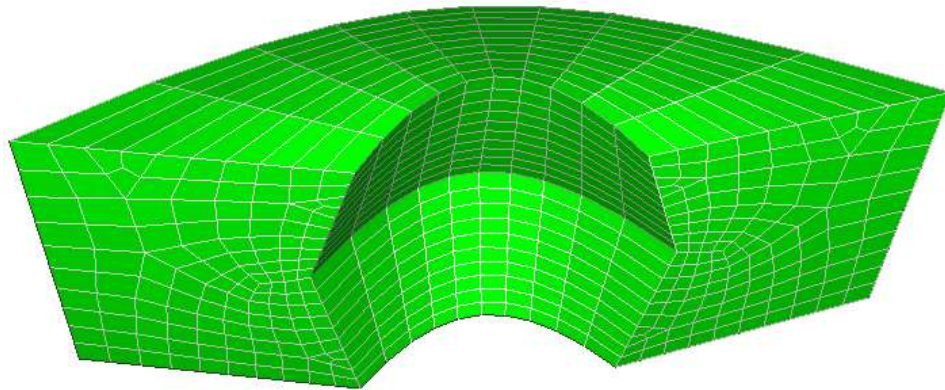
- **Geometry Requirements**

- Source and target surfaces topologically similar
- Mapped linked surfaces



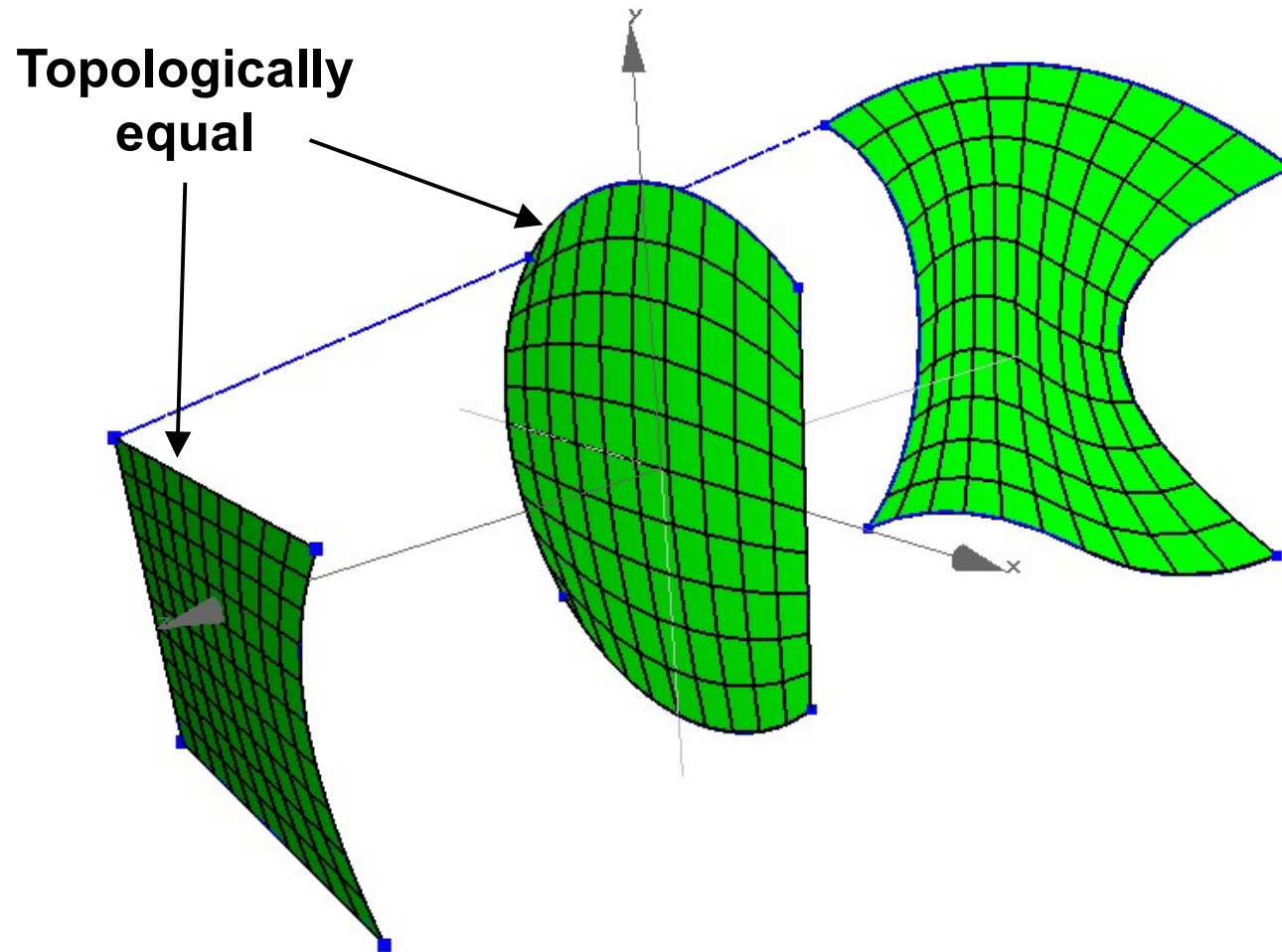
Structured mesh – Sweeping

- **Examples**



Structured mesh – Spline Sweeping

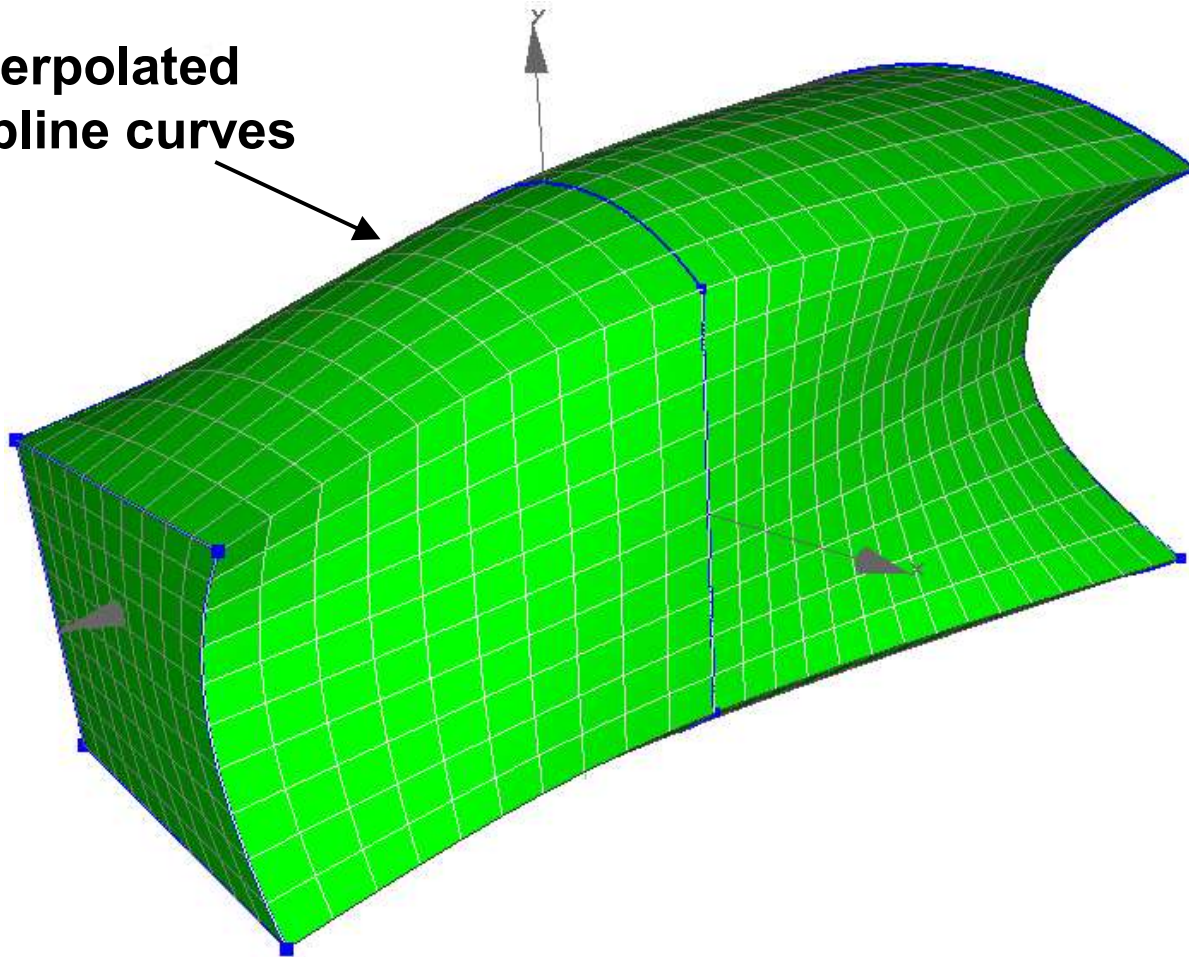
- **Geometry Requirements**
 - Sequence of sections
 - Meshes must be topologically equal



Structured mesh – Spline Sweeping

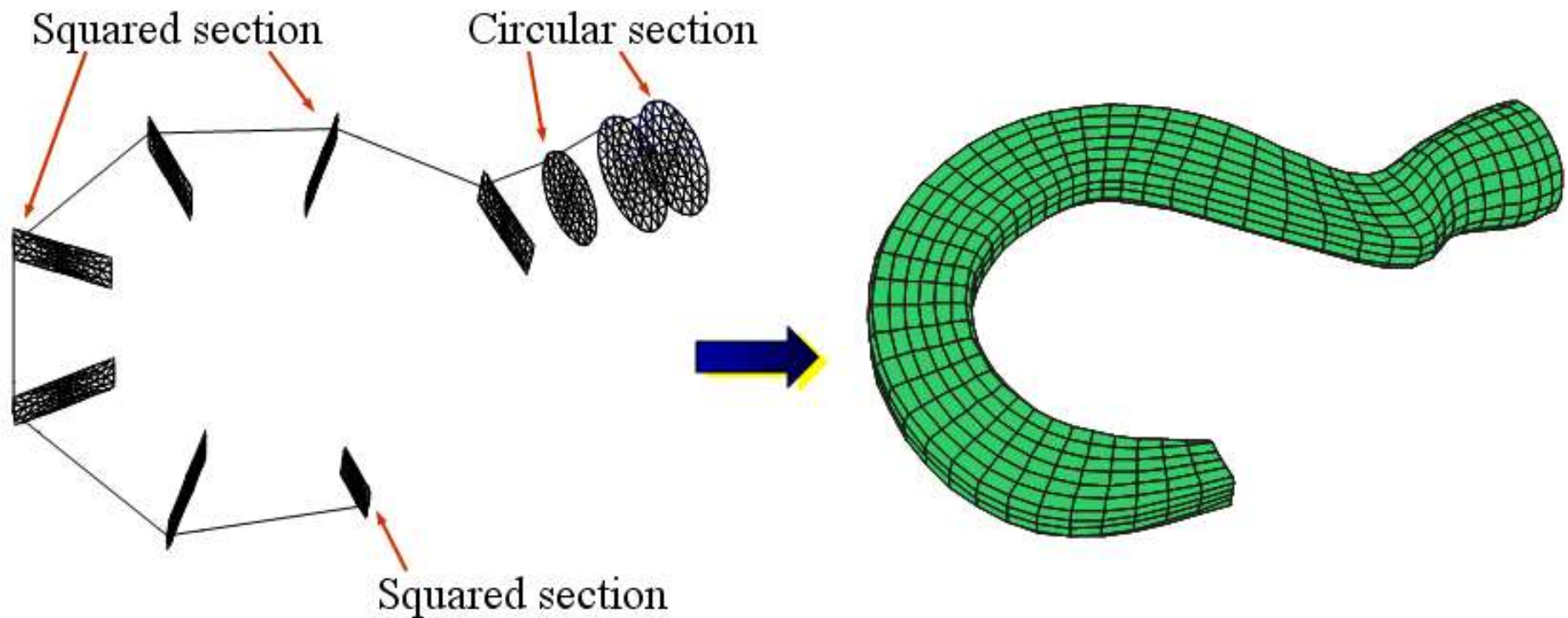
- **Geometry Requirements**
 - Sequence of sections
 - Meshes must be topologically equal

Interpolated
by spline curves



Structured mesh – Spline Sweeping

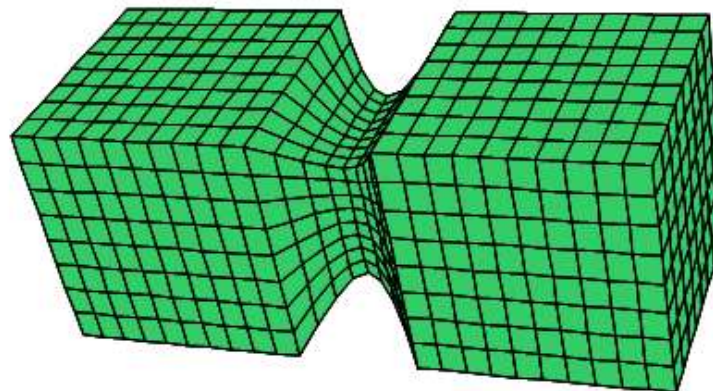
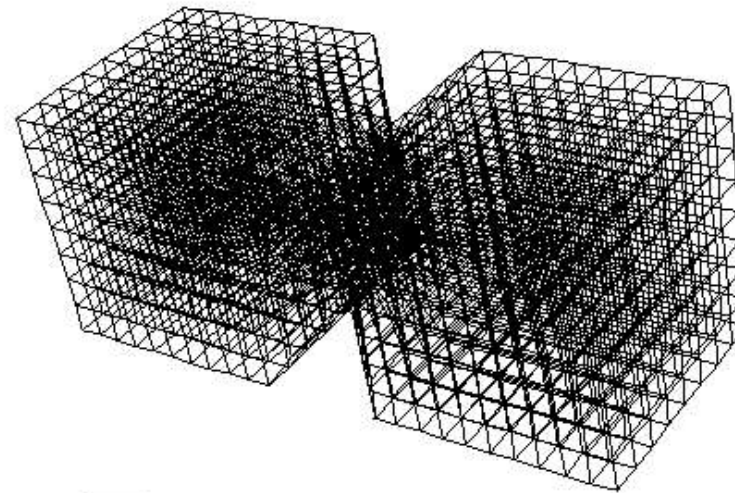
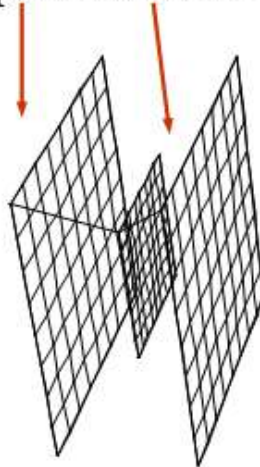
- **Geometry Requirements**
 - Sequence of sections
 - Meshes must be topologically equal



Structured mesh – Spline Sweeping

- **Geometry Requirements**
 - Sequence of sections
 - Meshes must be topologically equal

Squared sections



Unstructured mesh – Requirements



- **Specific algorithm requirements inherited from its ancestor**
J-Mesh (Joaquim Cavalcante-Neto, Wawrzynek, Carvalho, Martha & Ingraffea; 2001):
 - Generation of well-shaped elements
 - Ability to conform to an existing refinement at the boundary of region
 - Ability to transition well between regions with different element sizes
 - Capability for modeling discontinuities (internal restriction and cracks)
- **Additional requirements for surfaces**
 - Locally refine the mesh in regions with curvatures

Unstructured mesh generation outline

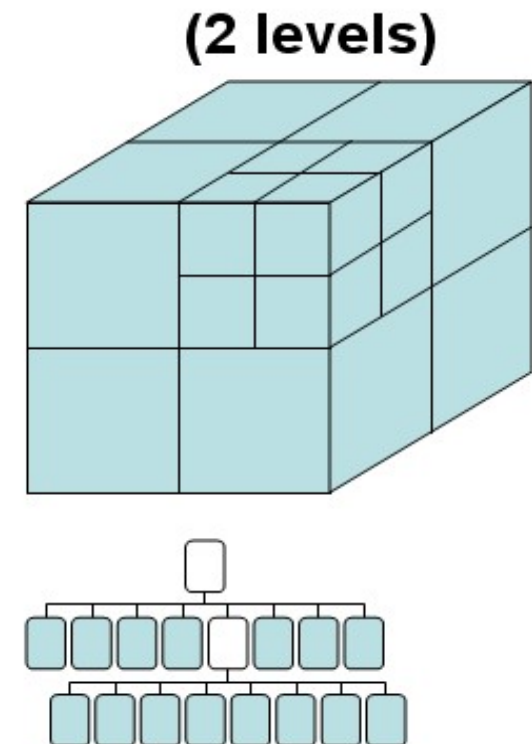
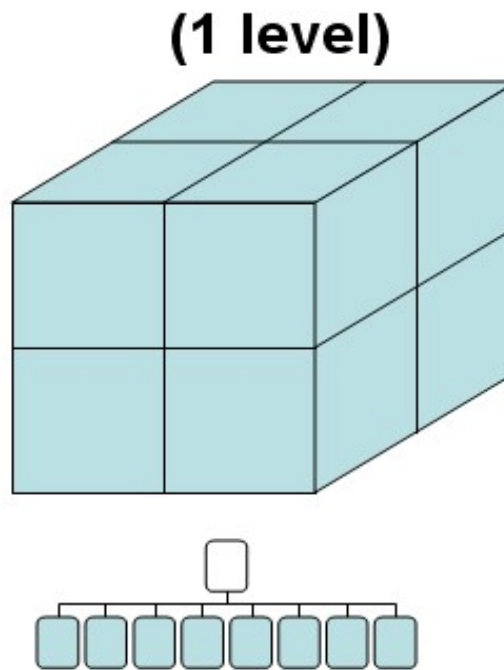
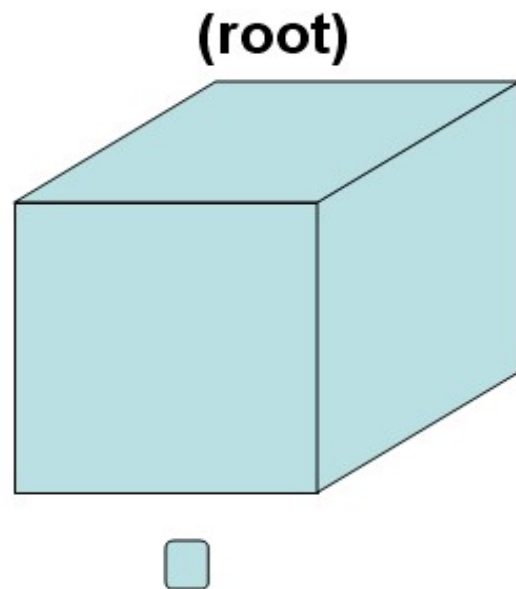


- **Background mesh generation – quadtree/octree**
 - Initialization based on boundary mesh.
 - Refinement to force a maximum cell size.
 - Refinement to provide minimum size disparity for adjacent cells.
- **Advancing-front procedure**
 - Geometry-based element generation
 - Topology-based element generation
 - Element generation based on back-tracking with face deletion.
- **Local mesh improvement**
 - Laplacian smoothing,
 - Local back-tracking with element deletion, or
 - Taubin smoothing (surfaces)

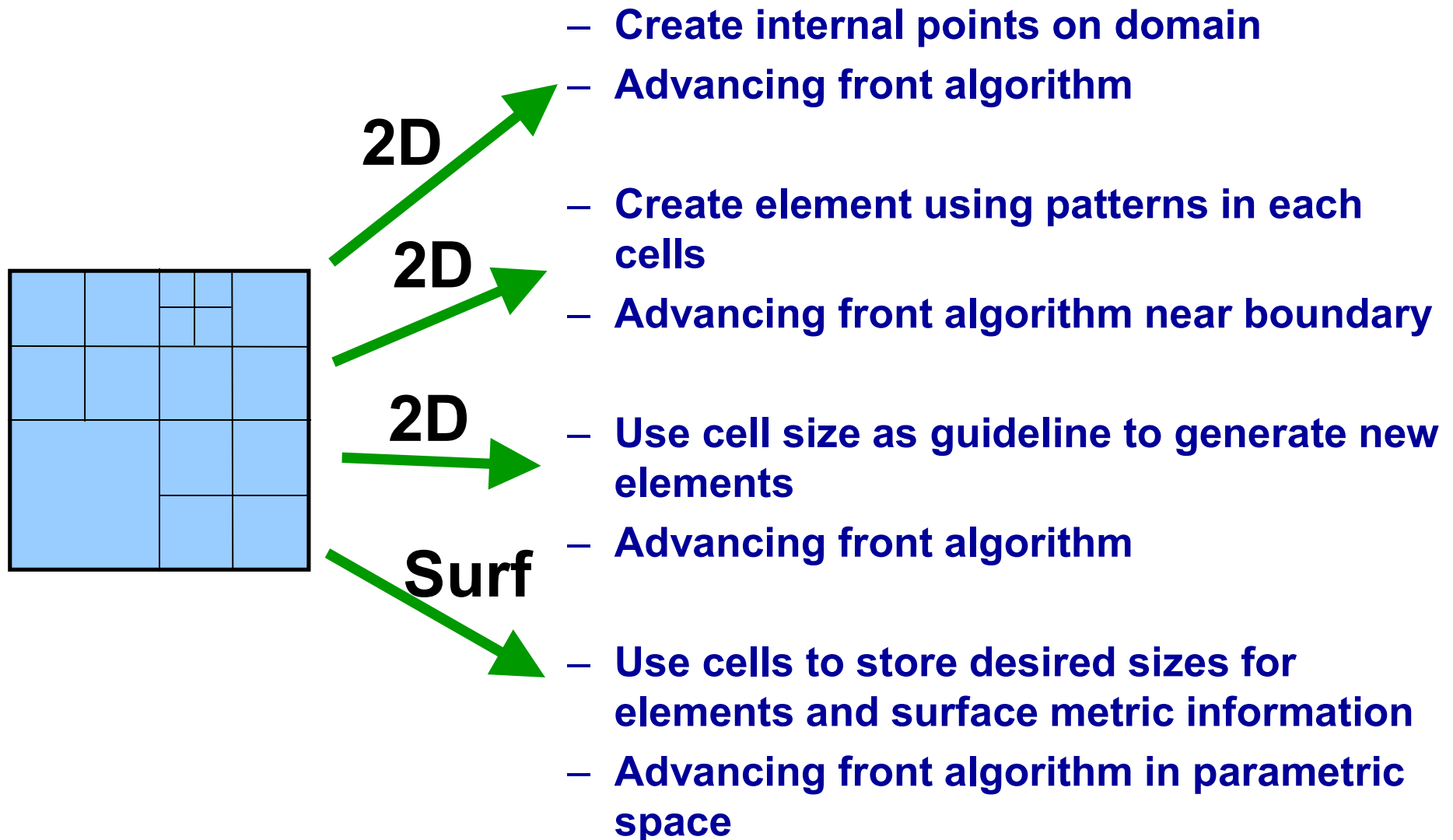
Unstructured mesh – auxiliary background structure

- **Quadtree and Octree**

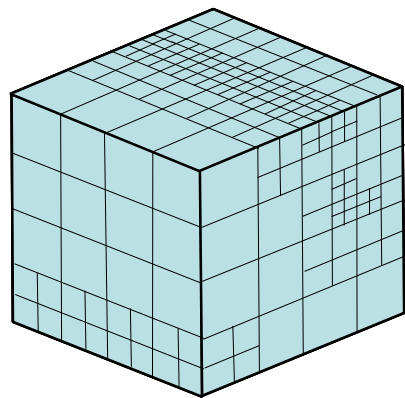
- Fast search procedures to navigate through end leaves
- Represent the desired size of elements with nearly the same size as the end leaves



Unstructured mesh – 2D auxiliary background structure



Unstructured mesh – 3D auxiliary background structure



3D



- Use cells to store desired sizes for elements
- Advancing front algorithm

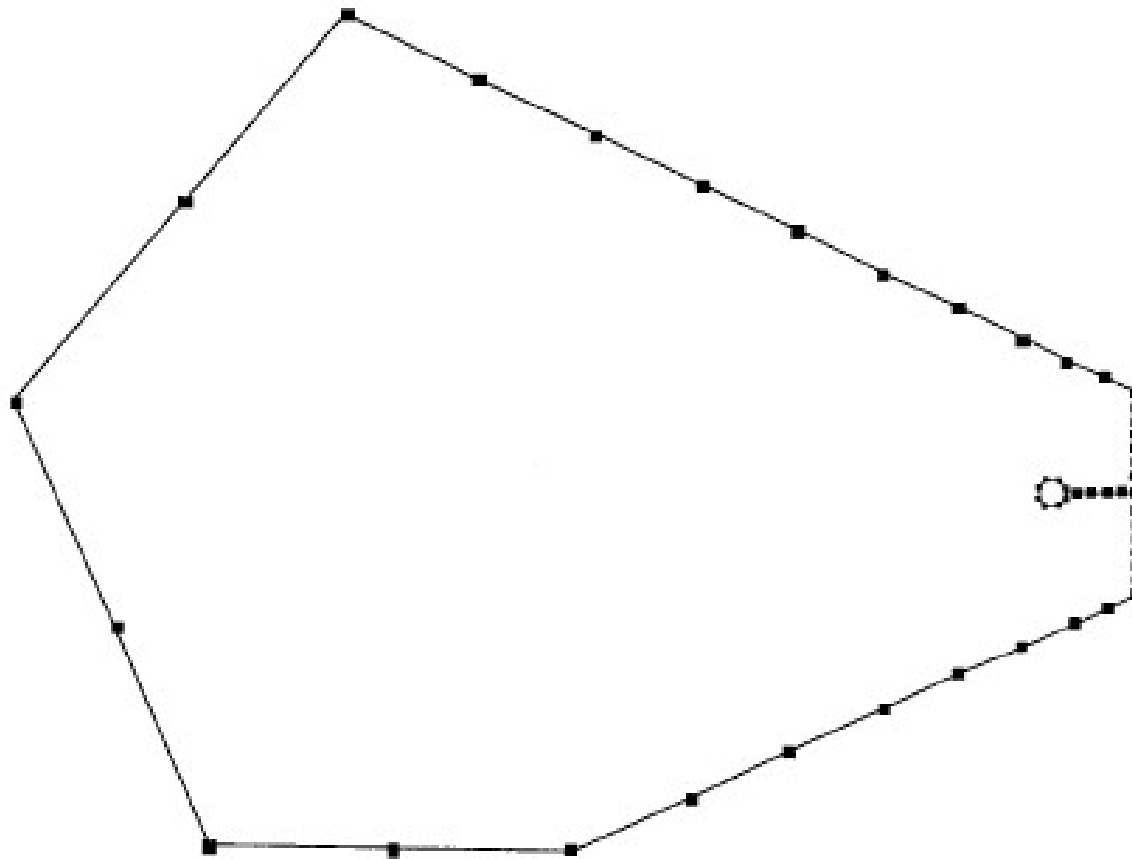
Surf



- Use cells to store desired sizes for elements and surface metric information
- Advancing front algorithm direct in 3D space

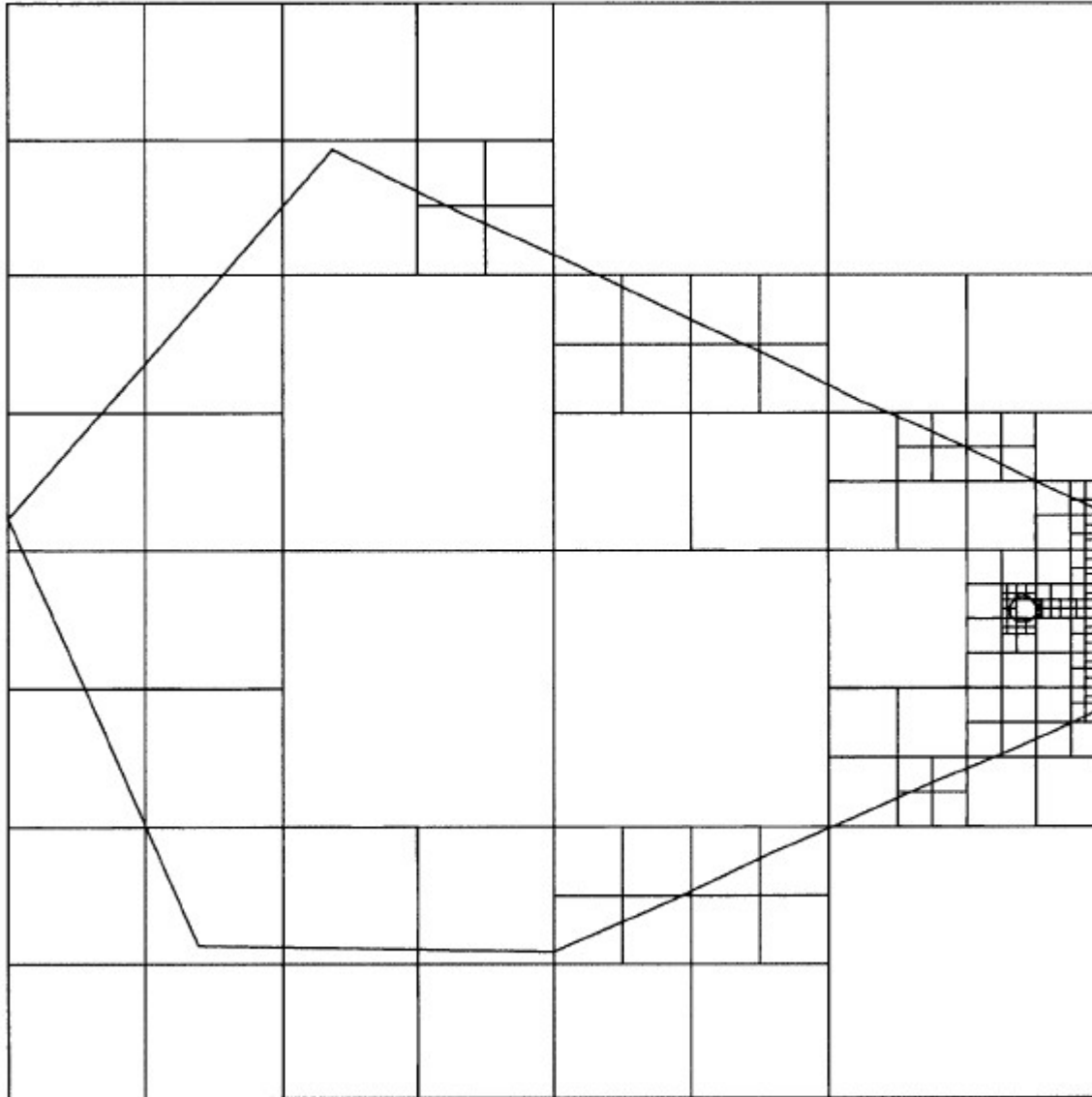
Unstructured mesh – background structure generation

- Hypothetical 2D model and its boundary refinement



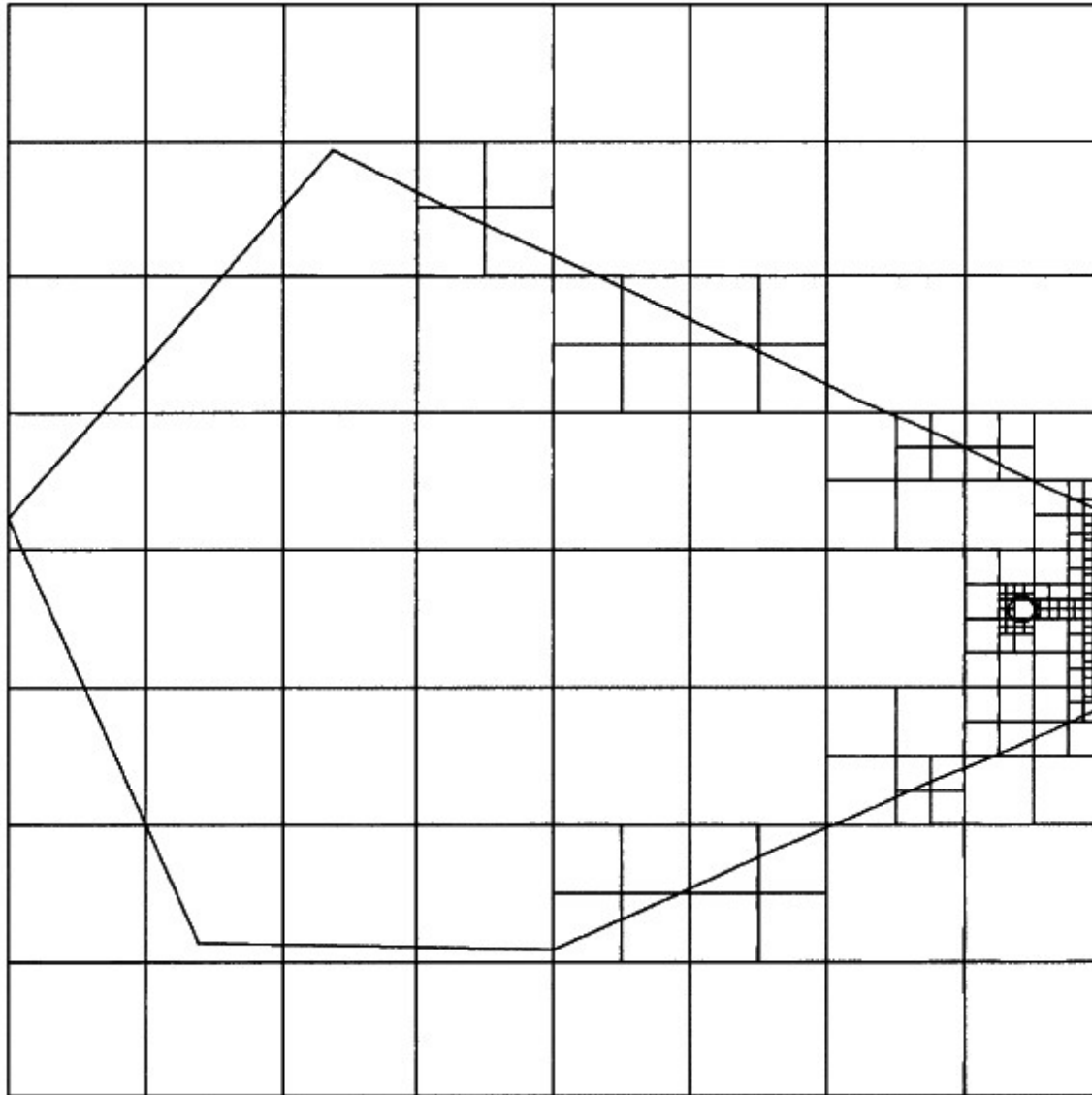
Unstructured mesh – background structure generation

- **Initialization based on boundary mesh**



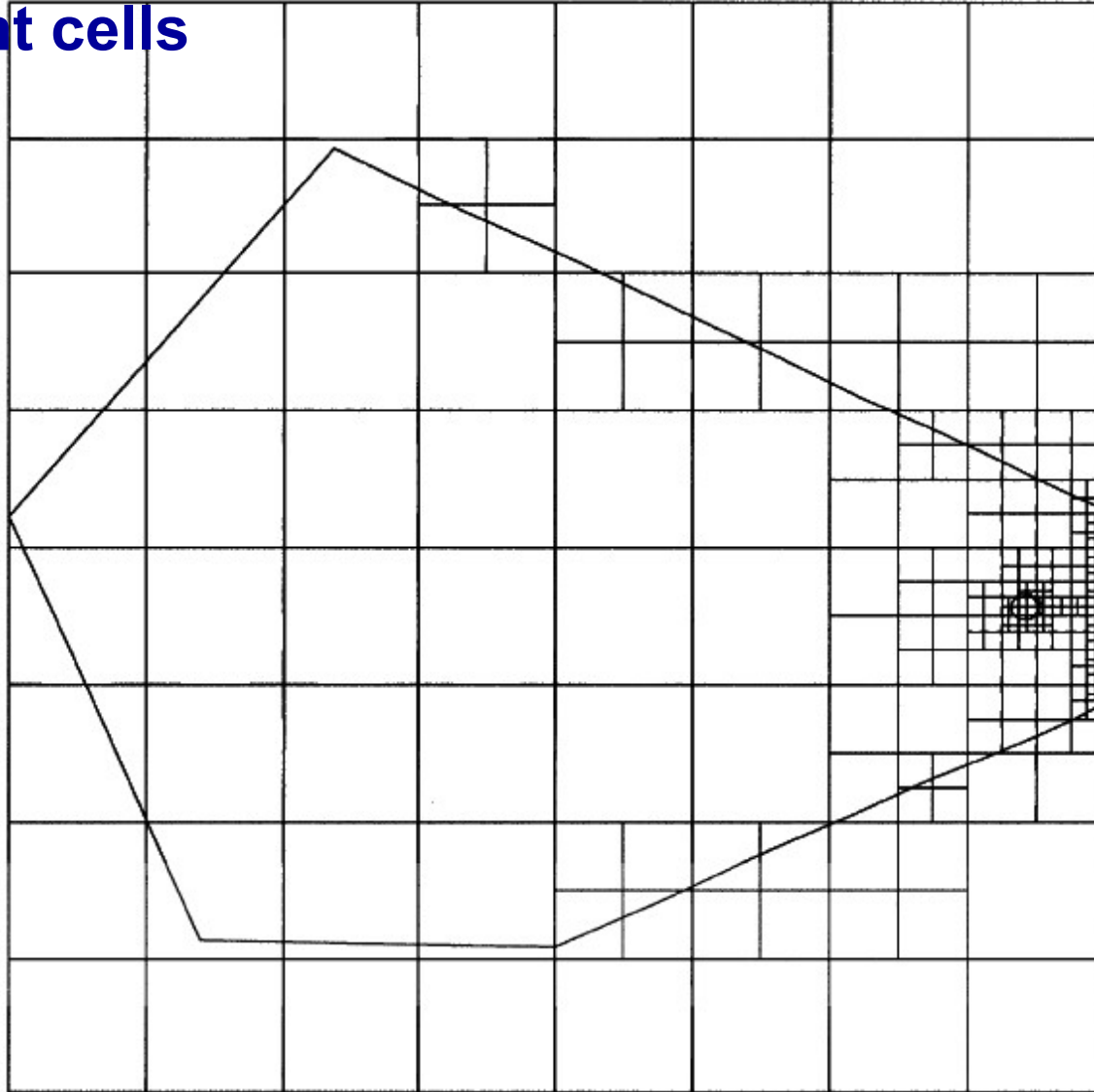
Unstructured mesh – background structure generation

- **Refinement to force a maximum cell size**



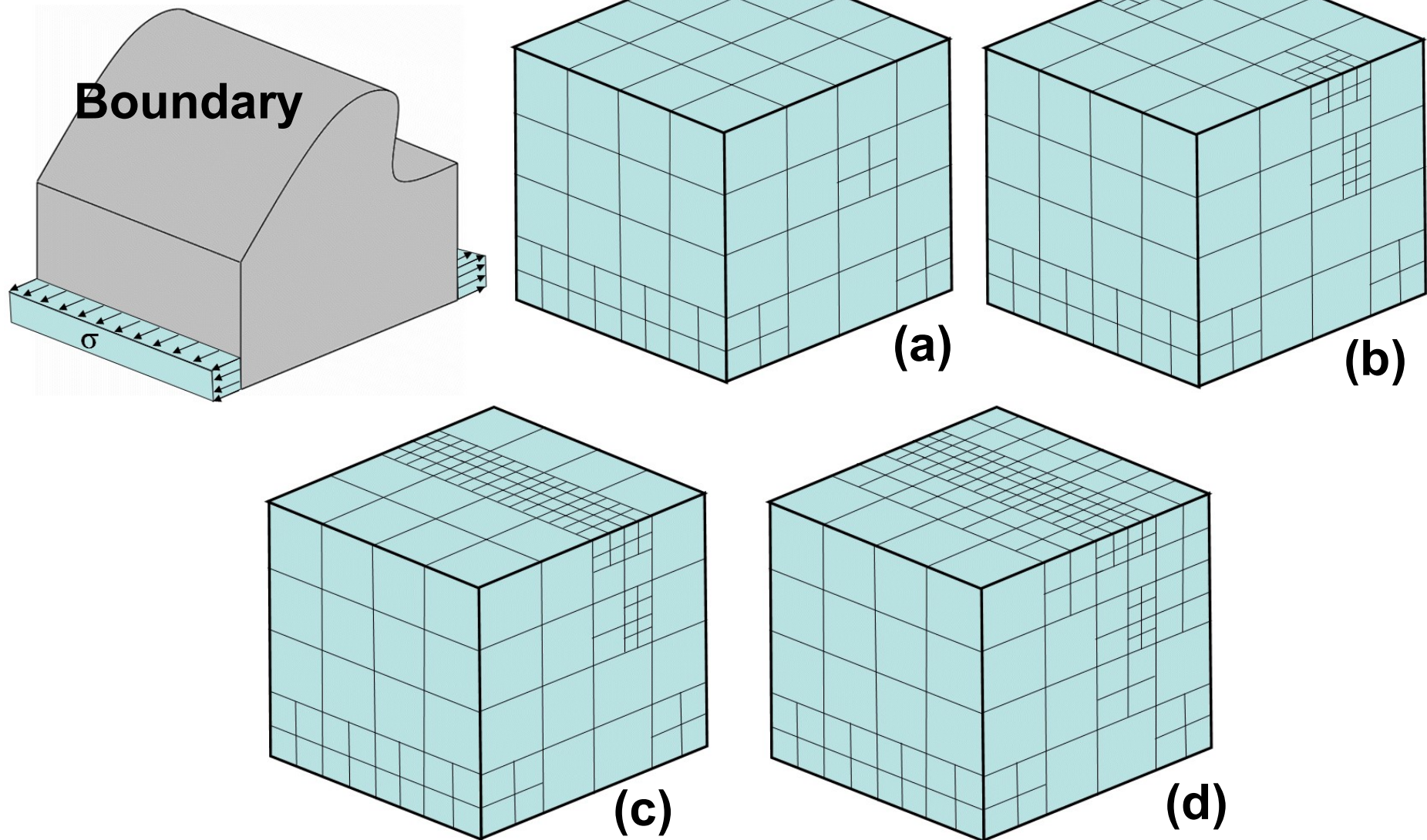
Unstructured mesh – background structure generation

- **Refinement to provide minimum size disparity for adjacent cells**



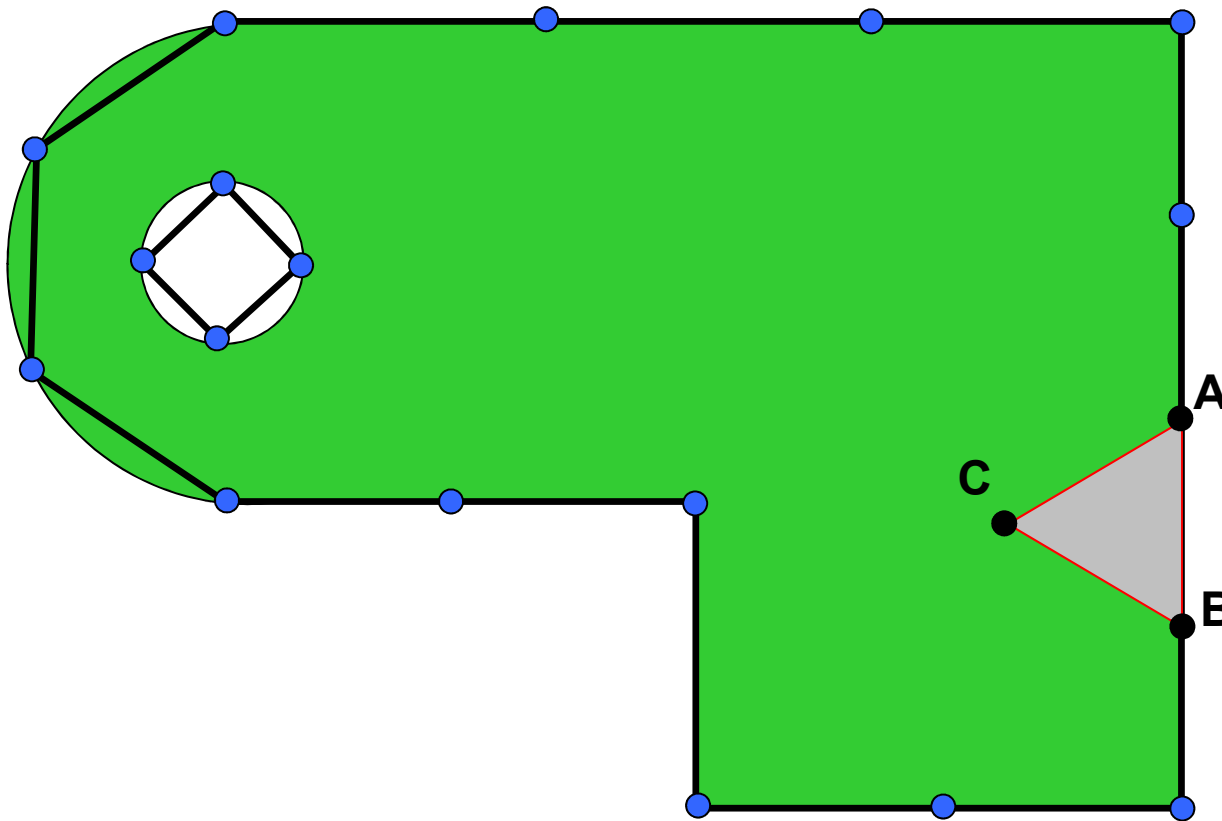
Unstructured mesh – background structure generation

- 3D model: octree refinement



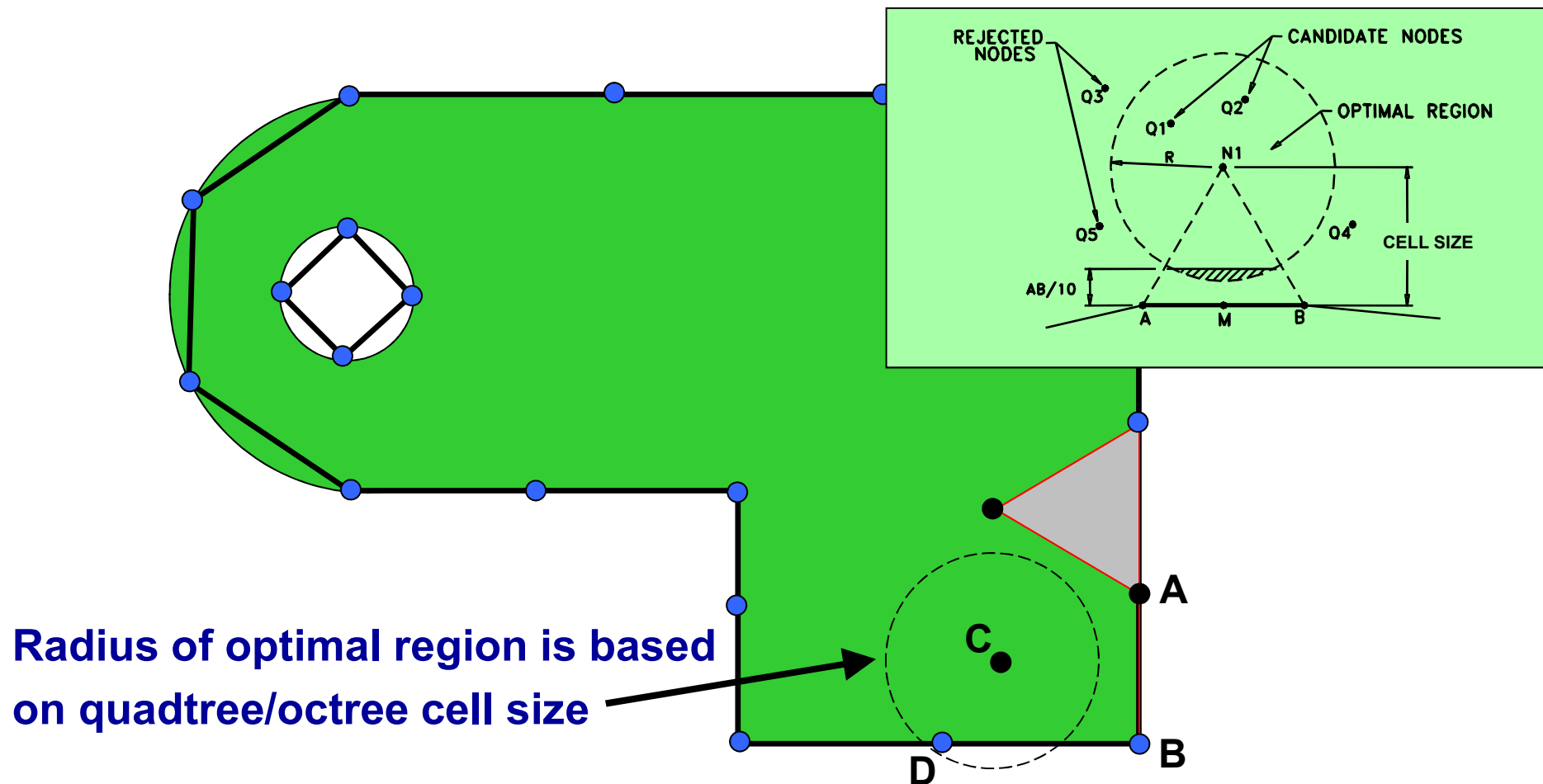
Unstructured mesh – advancing-front technique

- **Advancing front algorithm**
 - Begin with boundary mesh – define as initial *front*
 - For each edge (face) on front, locate initial node C based on front AB



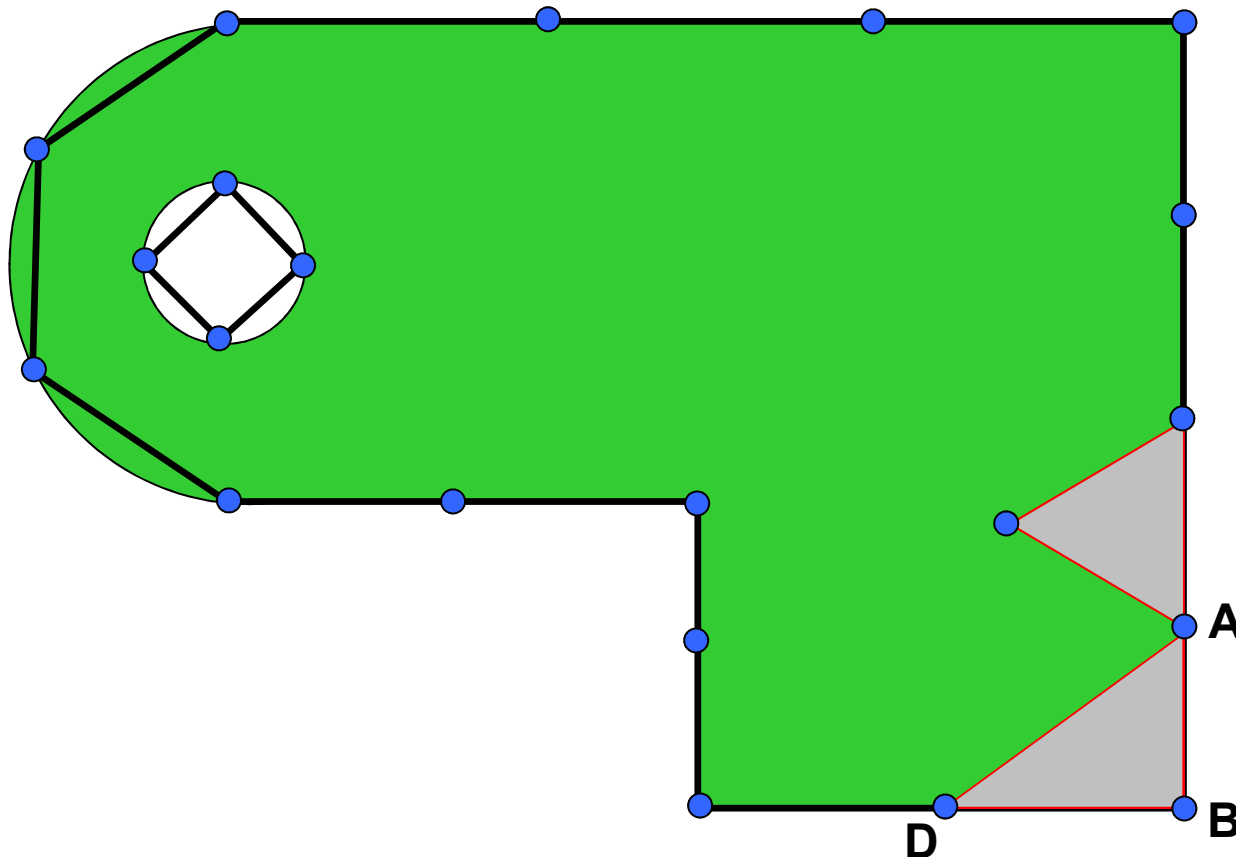
Unstructured mesh – advancing-front technique

- **Advancing front algorithm**
 - Determine if any other node on current front are within search radius r of ideal location C (Choose D instead of C)



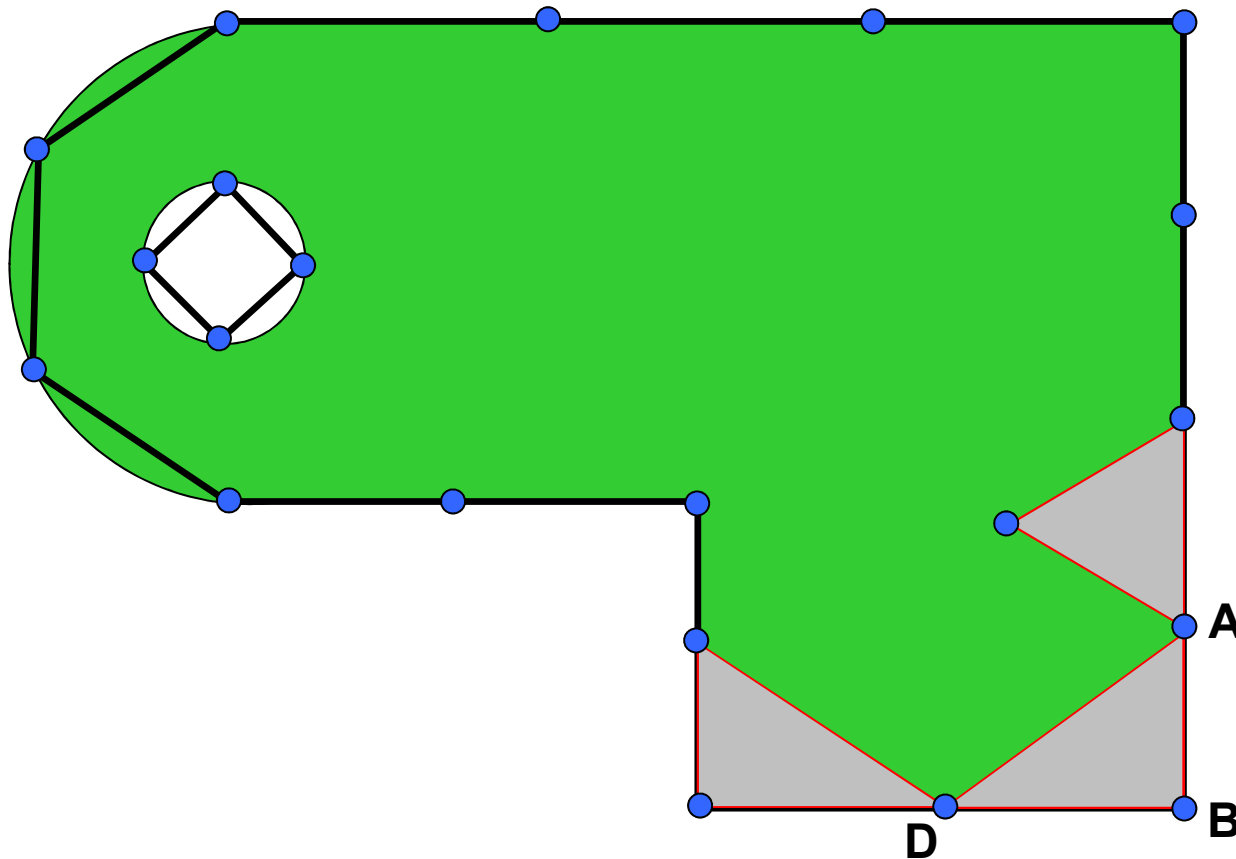
Unstructured mesh – advancing-front technique

- **Advancing front algorithm**
 - New *front edges (faces)* added and deleted from *front* as triangles (tetrahedral) are formed
 - Continue until *front edges (faces)* remain on *front*



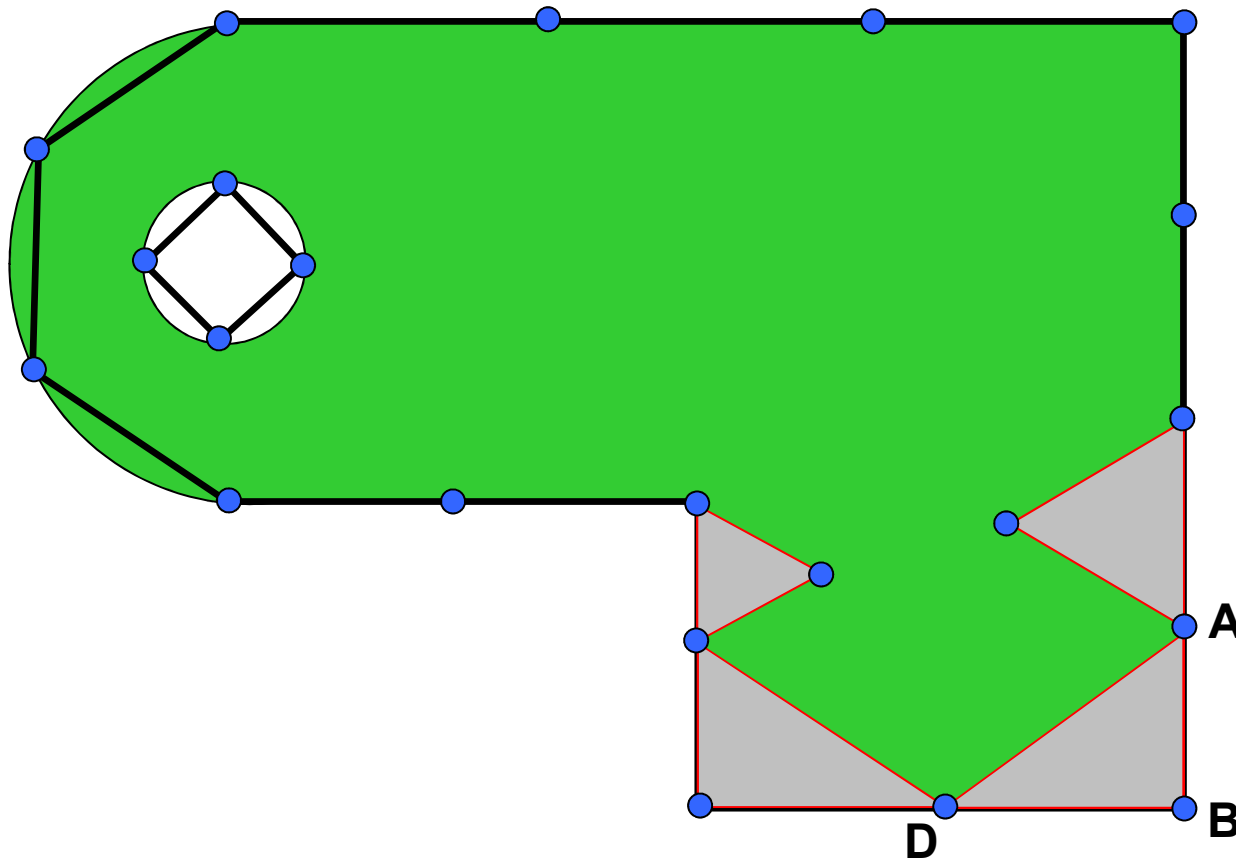
Unstructured mesh – advancing-front technique

- **Advancing front algorithm**
 - New *front edges* added and deleted from *front* as triangles are formed
 - Continue until *front edges* remain on *front*



Unstructured mesh – advancing-front technique

- **Advancing front algorithm**
 - New *front edges* added and deleted from *front* as triangles are formed
 - Continue until *front edges* remain on *front*



Unstructured mesh – advancing-front technique

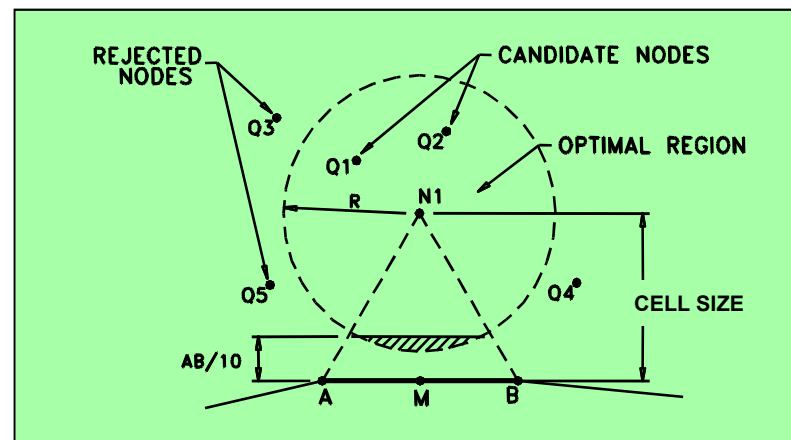
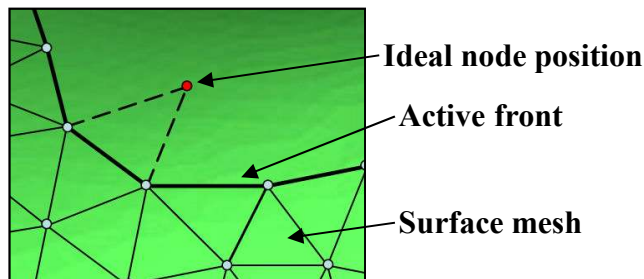
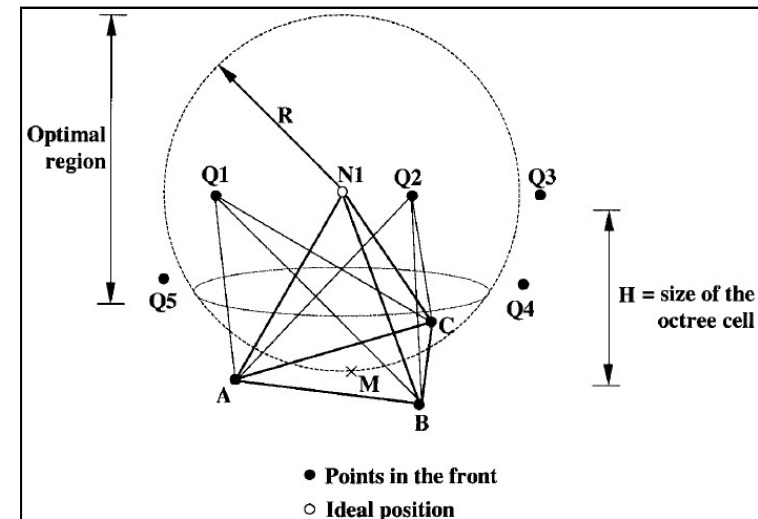
- **Geometry-based element generation**

- **Boundary contraction list**

- List of active edges
 - List of rejected edges

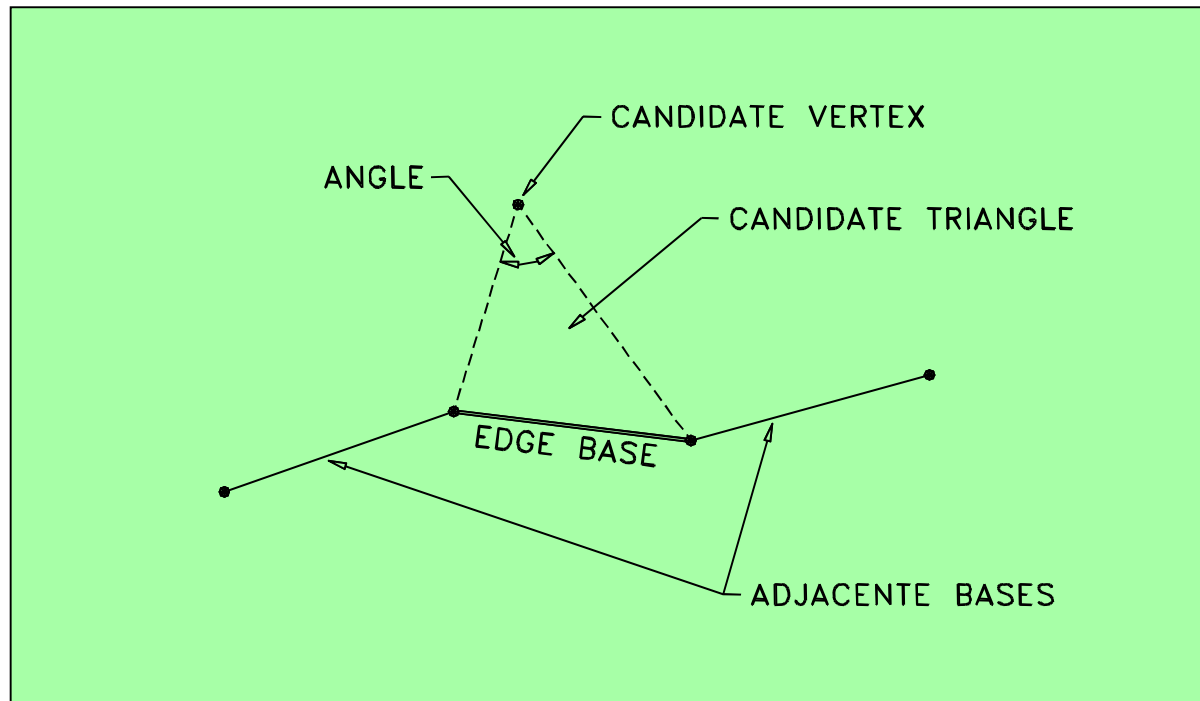
- **Generation of optimal elements**

- Size of element
 - Optimal location N1
 - $\text{Ratio} = 0.85 * \text{size}$
 - Upper bound and lower bound
 - Range Tree Search



Unstructured mesh – advancing-front technique

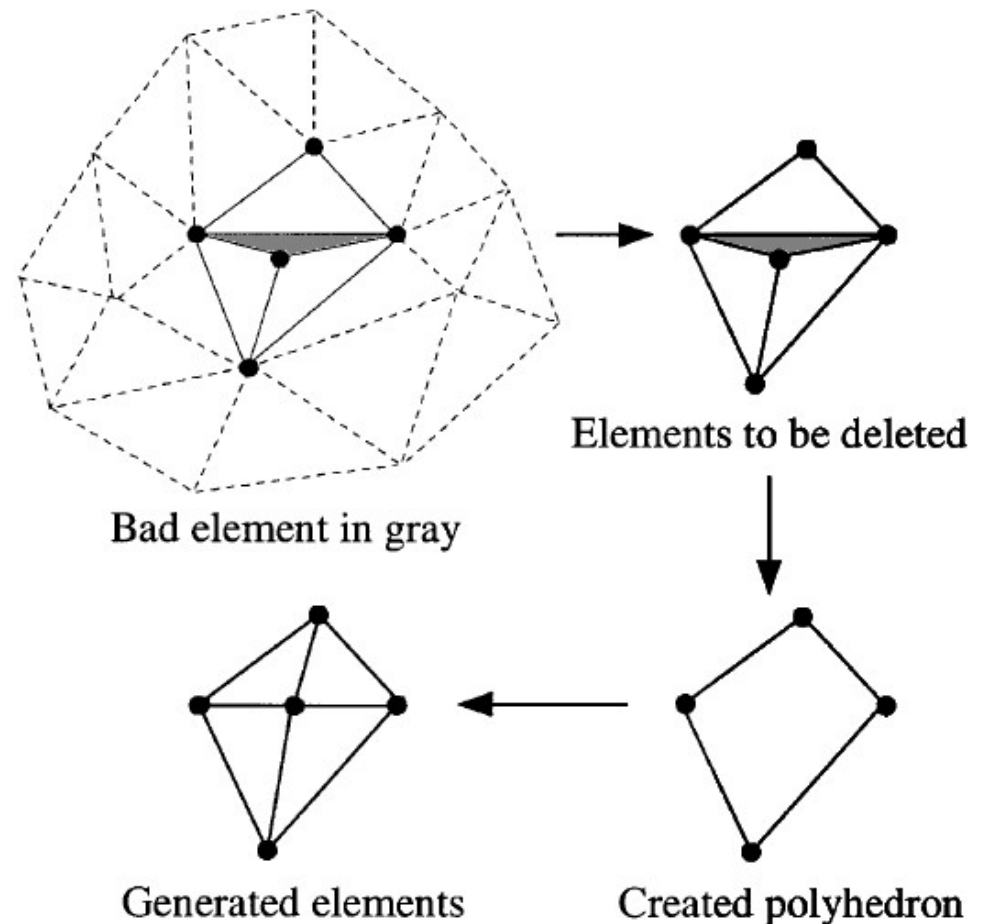
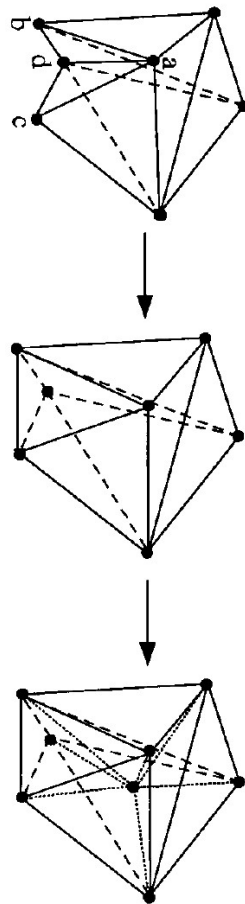
- **Topology-based element generation**
 - List of rejected edges becomes active edges
 - Generation of elements by any node close to the base edge (best angle)
 - Generate a valid mesh, although not optimal



Unstructured mesh – advancing-front technique

- **Back-Tracking**

- Locally modify the advancing front, deleting already generated adjacent tetrahedra until a ‘near’ convex non-meshed polyhedron is formed



Unstructured mesh – local mesh improvement

- **Laplacian smoothing**

- Uses Laplacian equation and the closest point function for surface

$$X_0^{n+1} = X_0^n + \phi \frac{\sum_{i=1}^m w_{i0} (X_i^n - X_0^n)}{\sum_{i=1}^m w_{i0}}$$

- $\phi = 1.0$ and $w_{i0}=1.0$

- **Taubin smoothing (surfaces)**

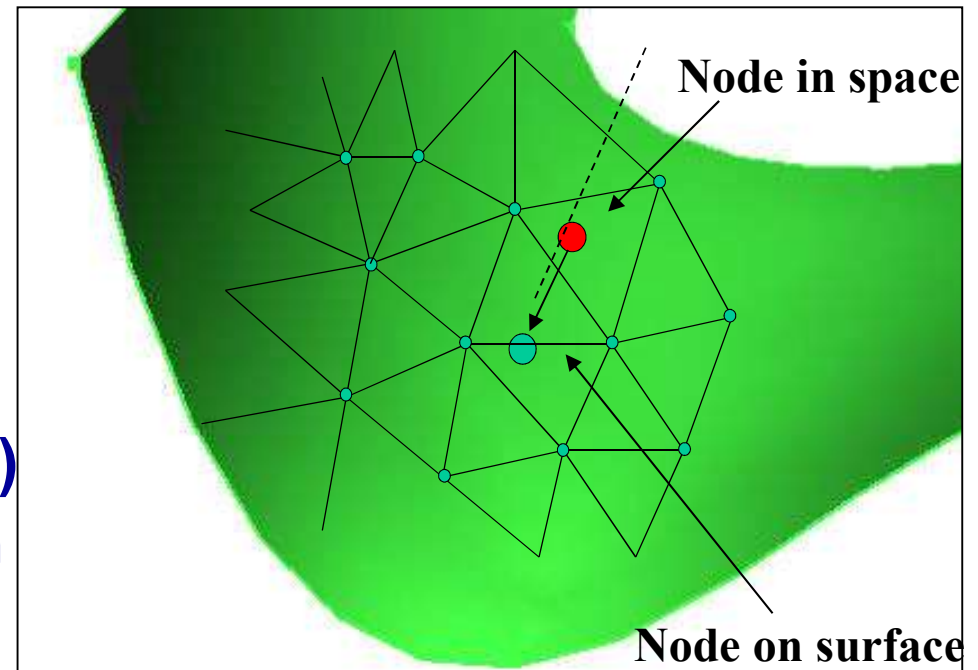
- Uses twice Laplacian equation

- $\phi = 1.0$ and $w_{i0} = 0.63$
- $\phi = 1.0$ and $w_{i0} = -0.67$

- Filters high frequencies

- Preserves the low frequencies

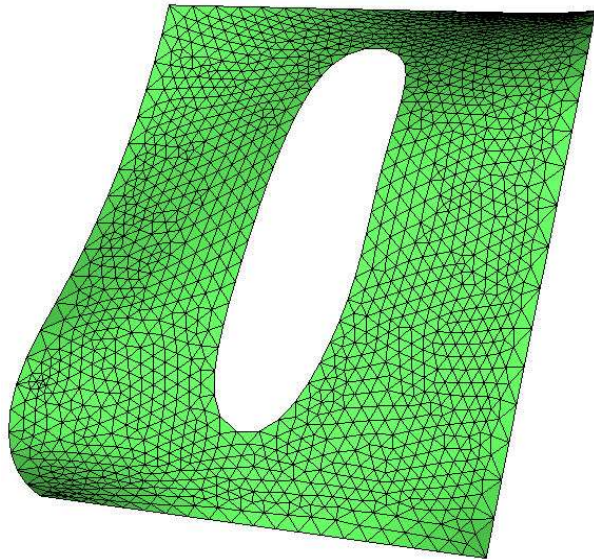
- Good results with geological and microstructure surfaces



Unstructured mesh – Surface Meshing

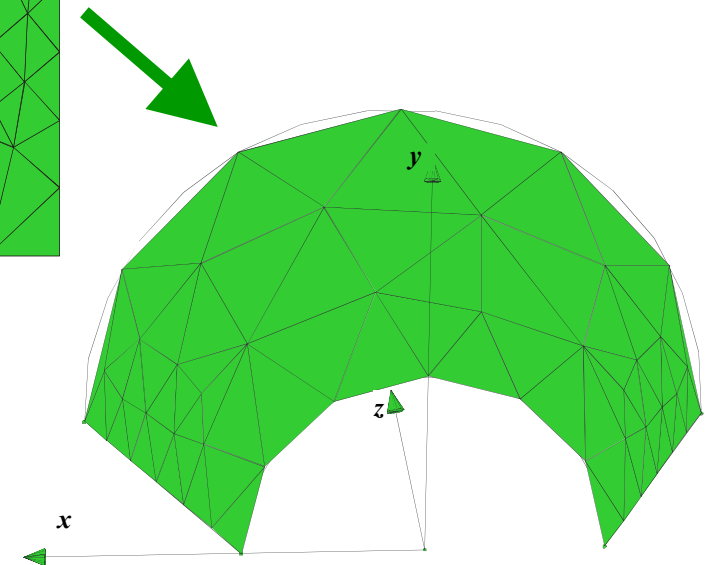
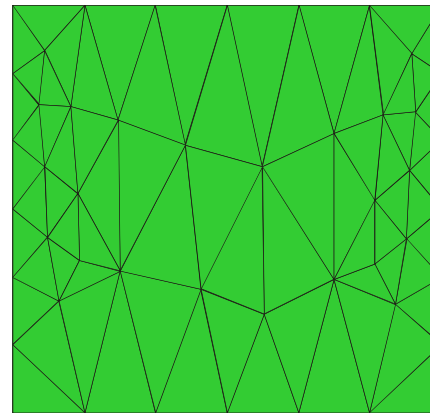
- **Direct 3D Meshing**

- Elements formed in 3D using actual x-y-z representation of surface



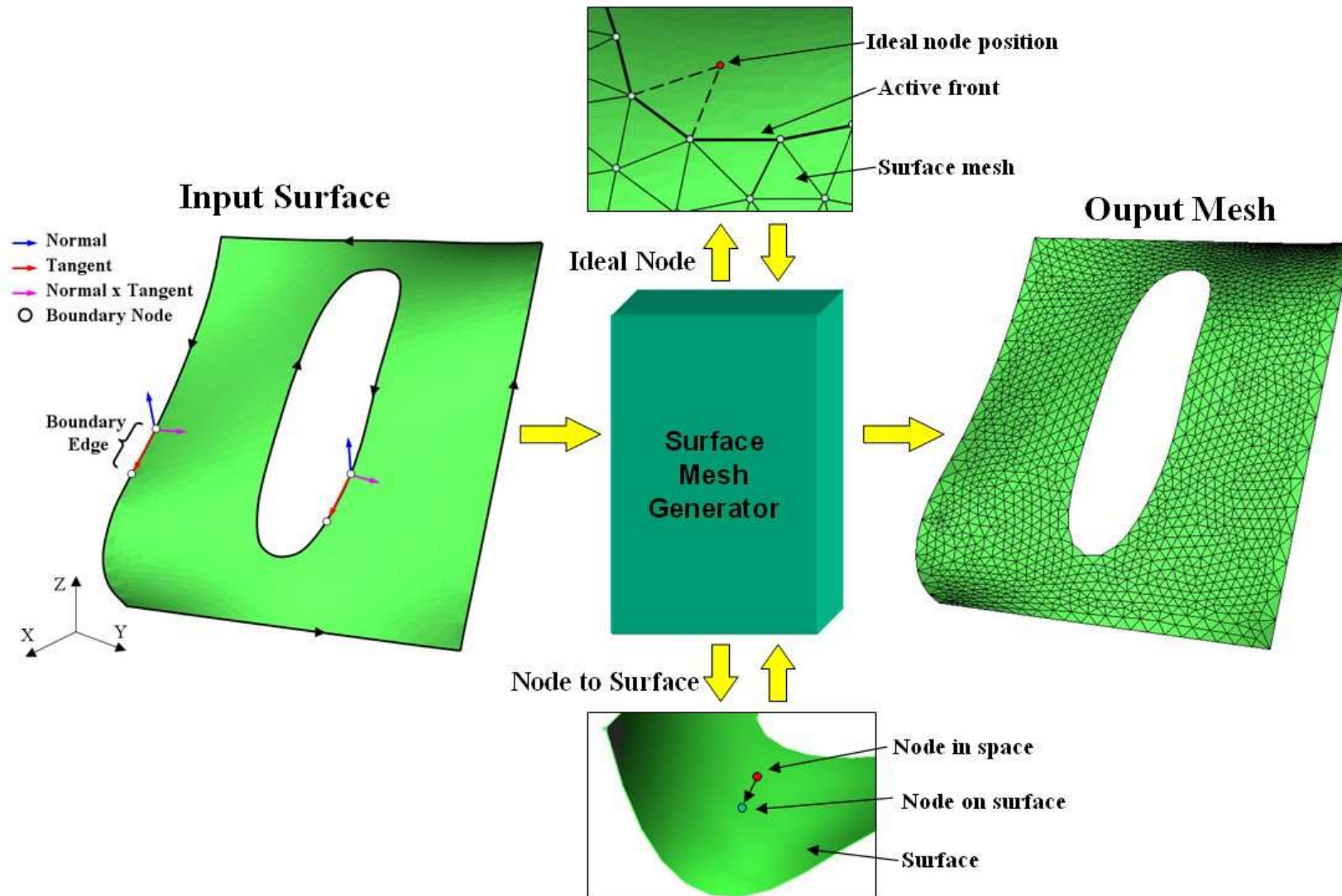
- **Parametric Space Meshing**

- Elements formed in 2D using parametric representation of surface
- Nodes locations later mapped to 3D space



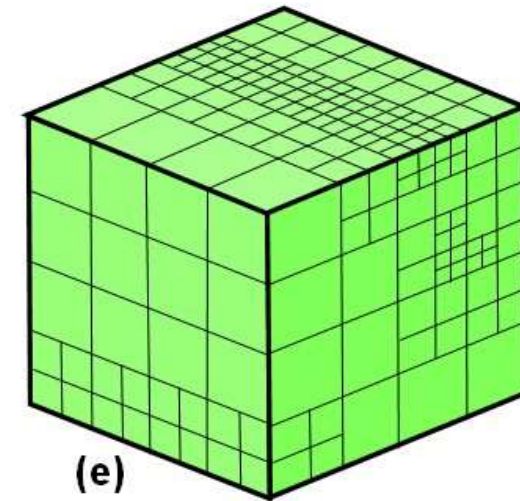
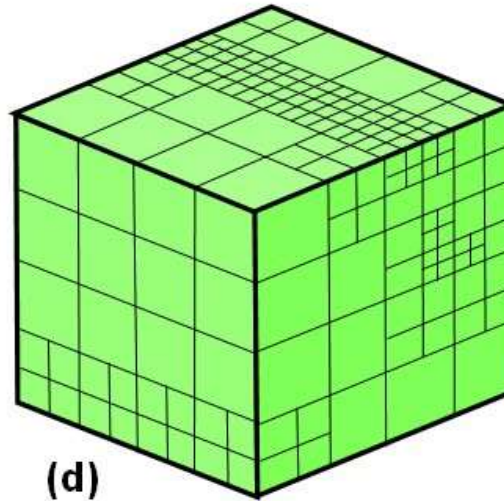
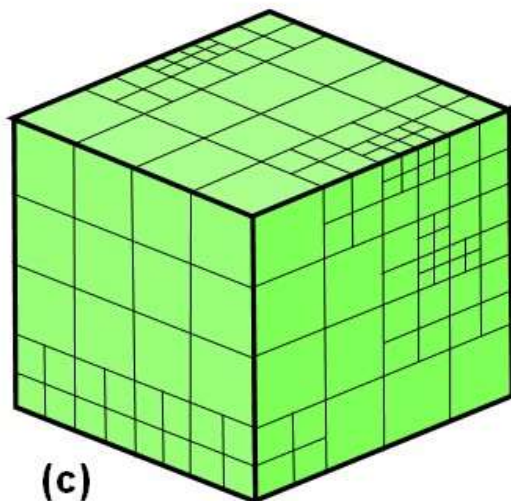
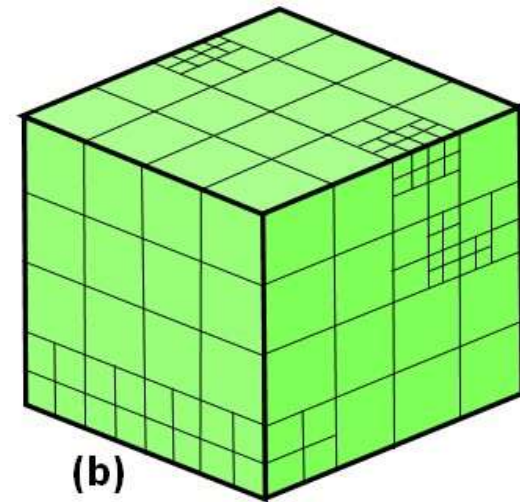
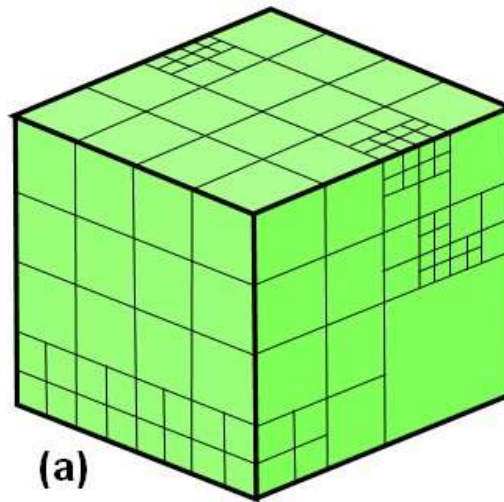
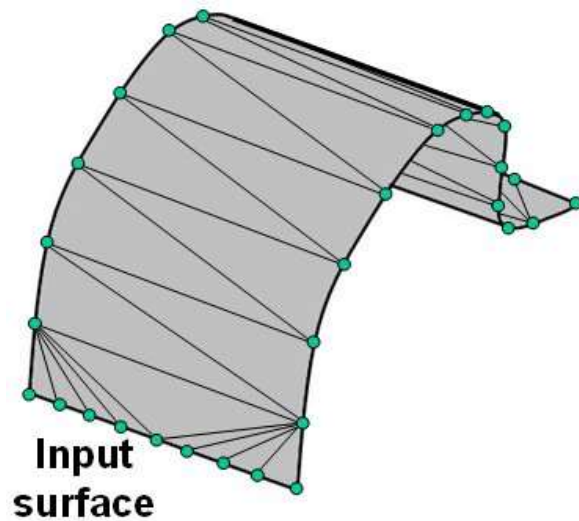
Unstructured mesh – Surface Meshing

- Direct 3D Meshing



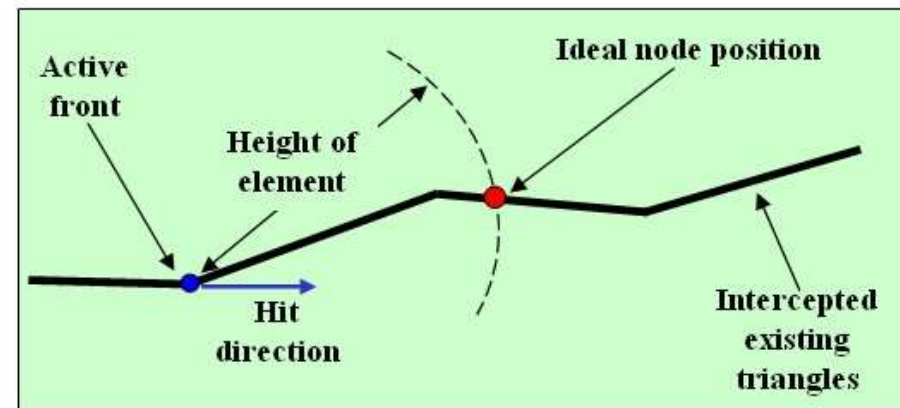
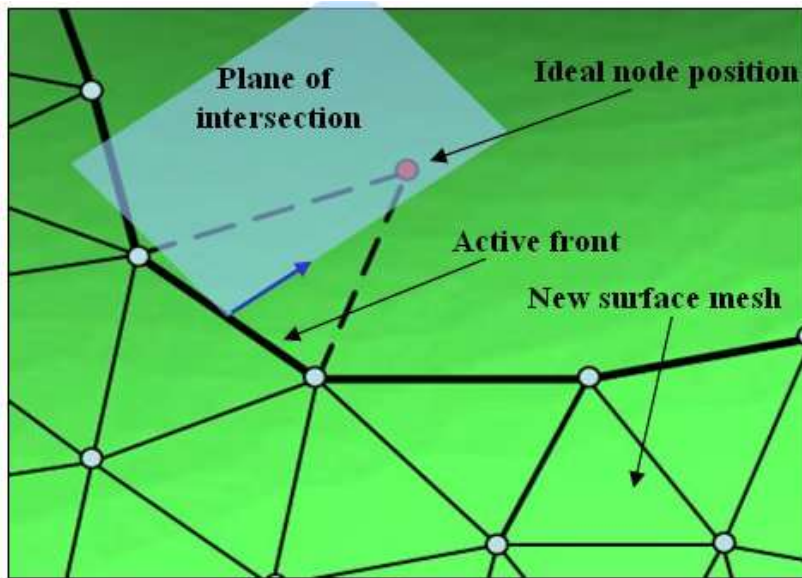
Unstructured mesh – Surface Meshing

- **Direct 3D Meshing – refinement of octree**



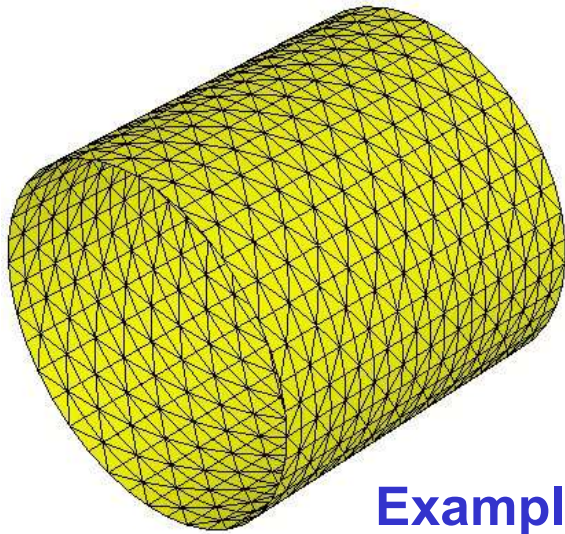
Unstructured mesh – Surface Meshing

- **Direct 3D Meshing – node location**

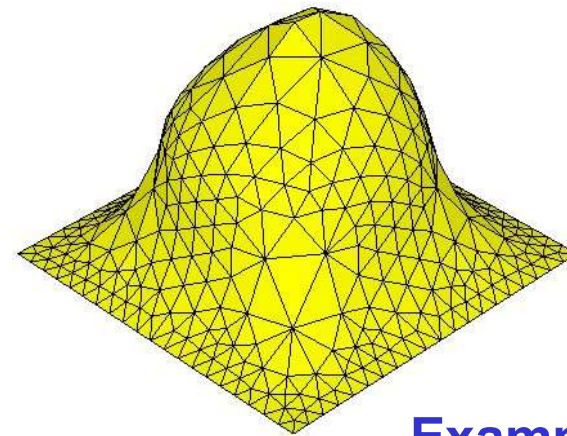


Unstructured mesh – Surface Meshing

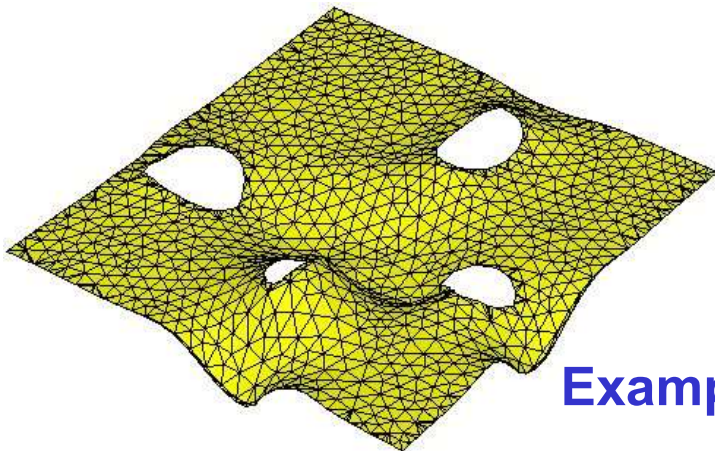
- **Direct 3D Meshing – Examples**



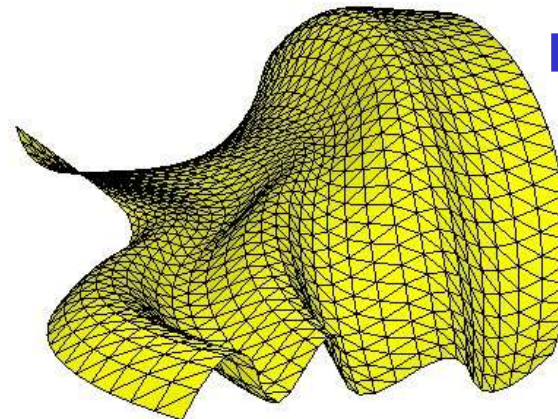
Example 1



Example 2



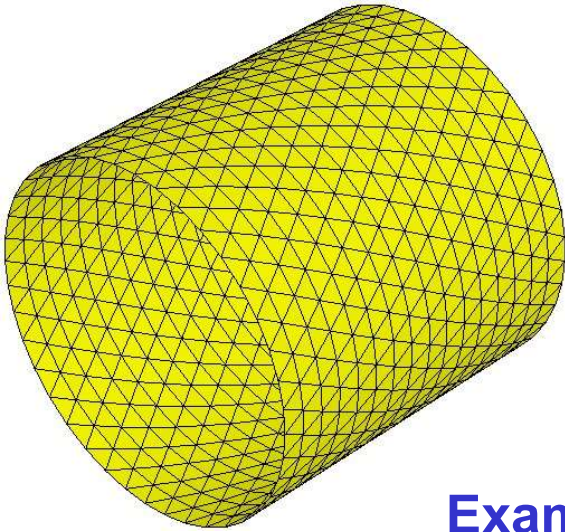
Example 3



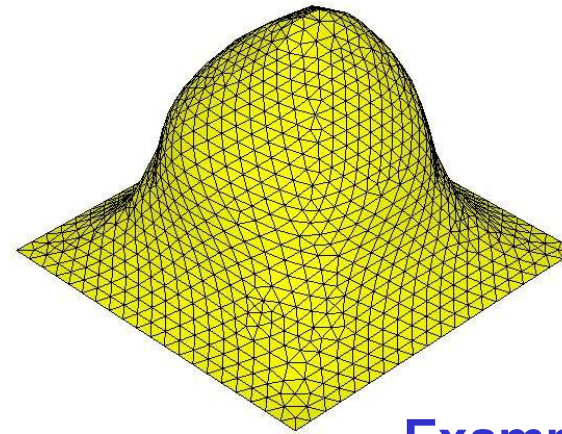
Example 4

Unstructured mesh – Surface Meshing

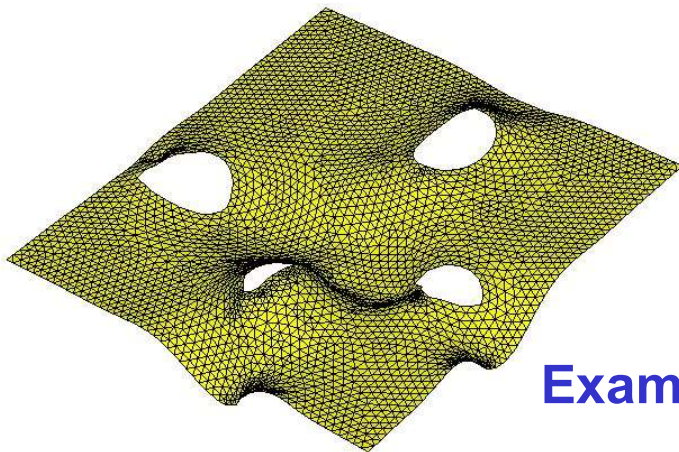
- **Direct 3D Meshing – Examples**



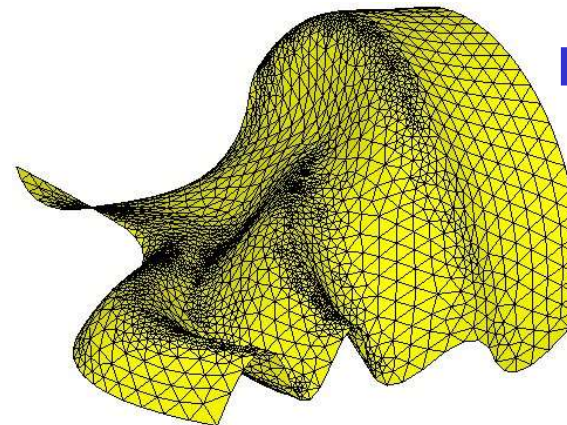
Example 1



Example 2

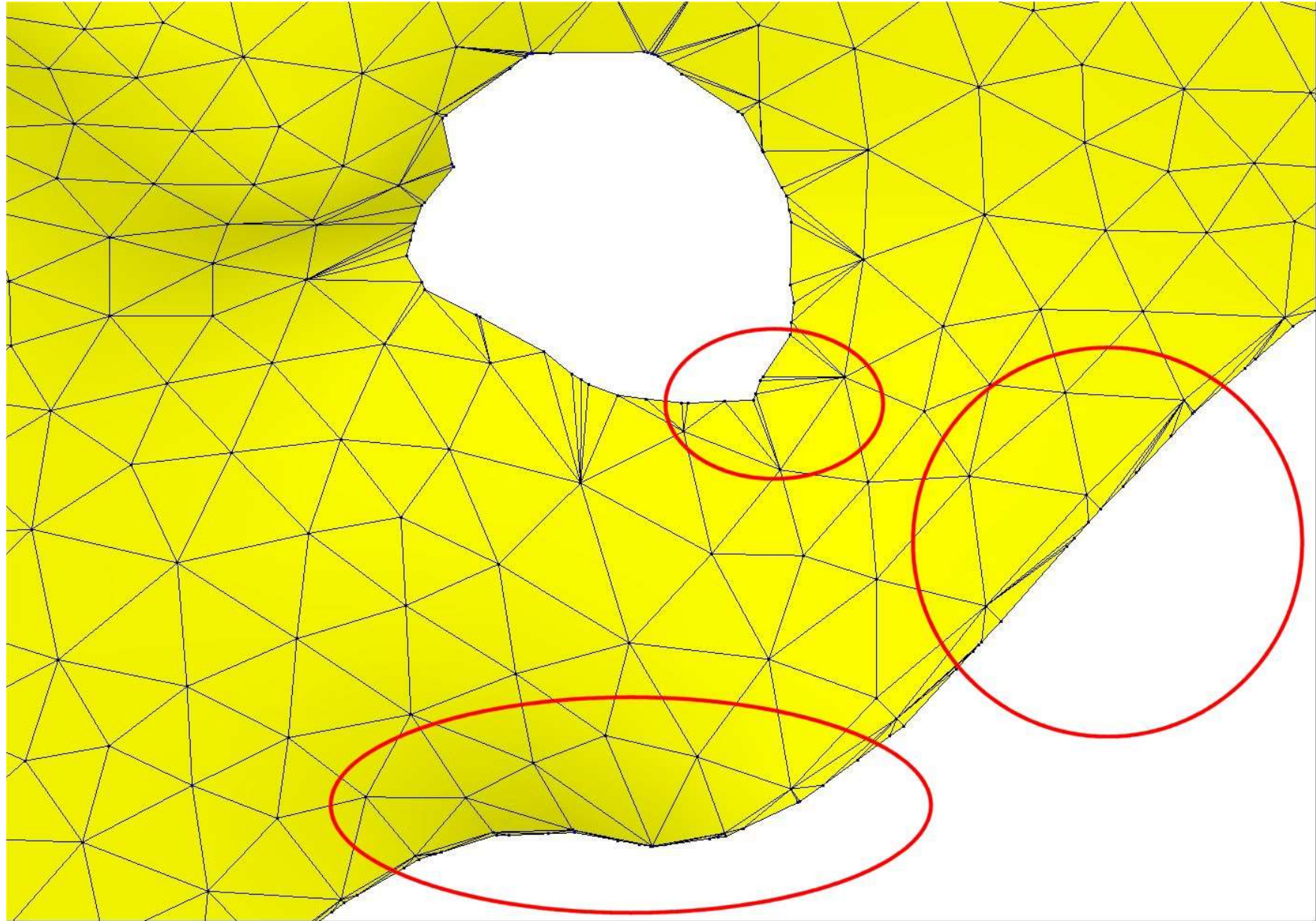


Example 3

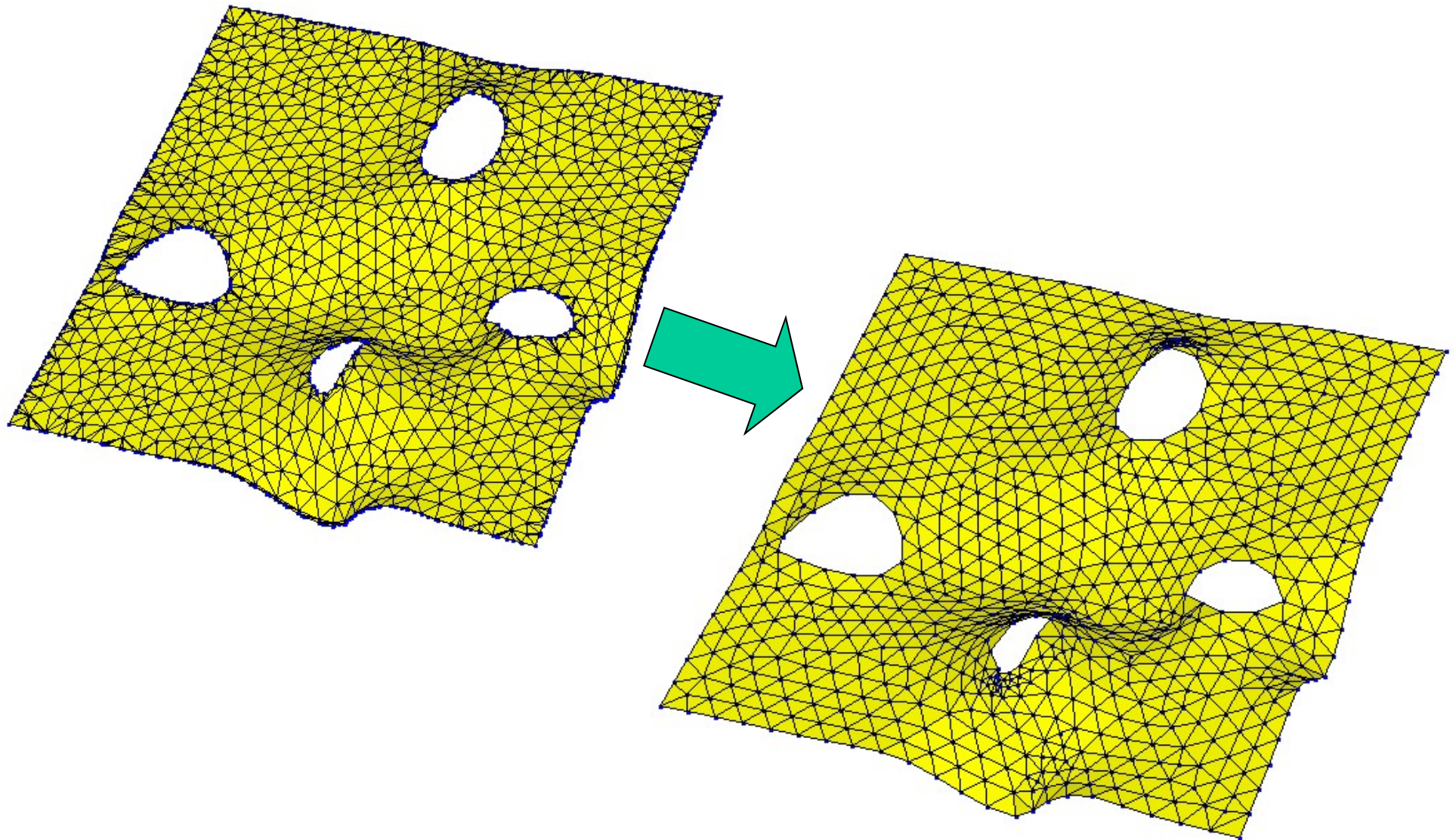


Example 4

Imported triangulation with poorly-shaped elements

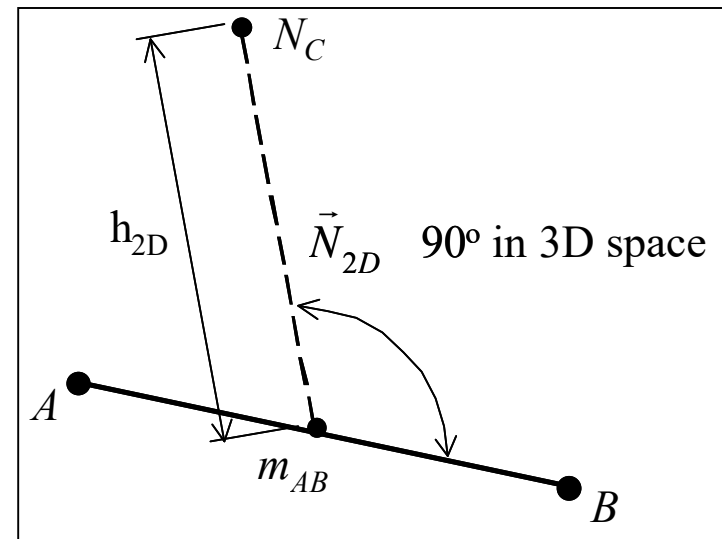
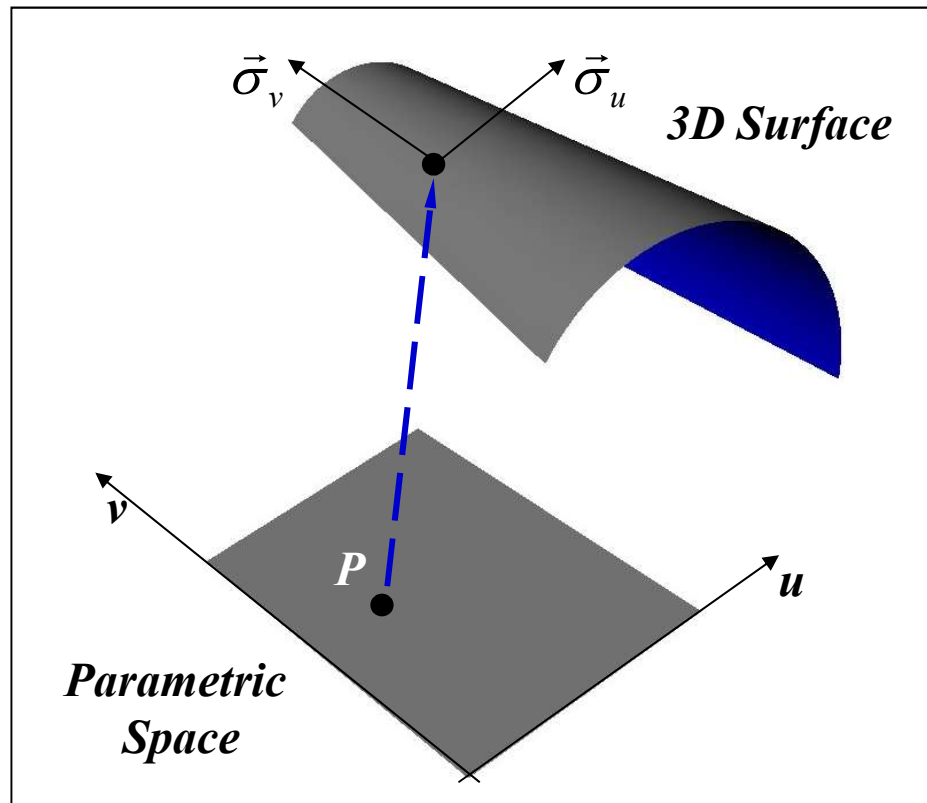


Example of surface re-triangulation



Unstructured mesh – Surface Meshing

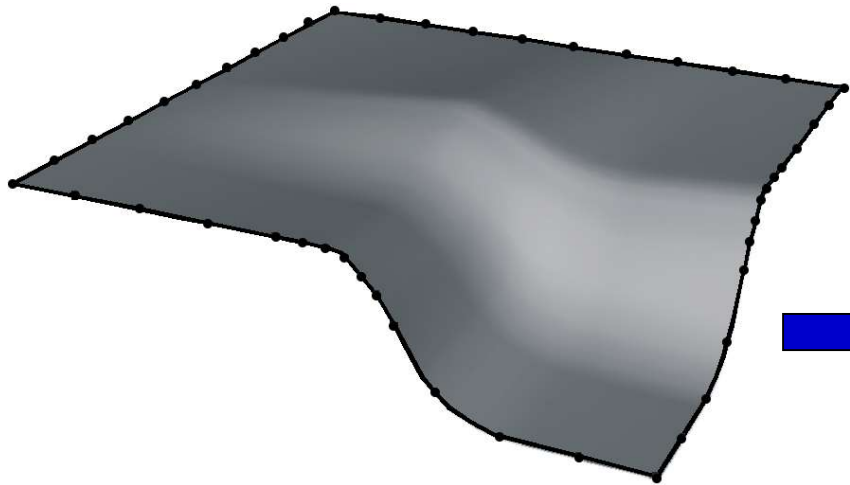
- **Parametric Space Meshing**
 - Elements formed in 2D using parametric representation of surface
 - Distance and angles are distorted in parametric space
 - Nodes locations later mapped to 3D space



Unstructured mesh – Surface Meshing

- **Parametric Space Meshing**

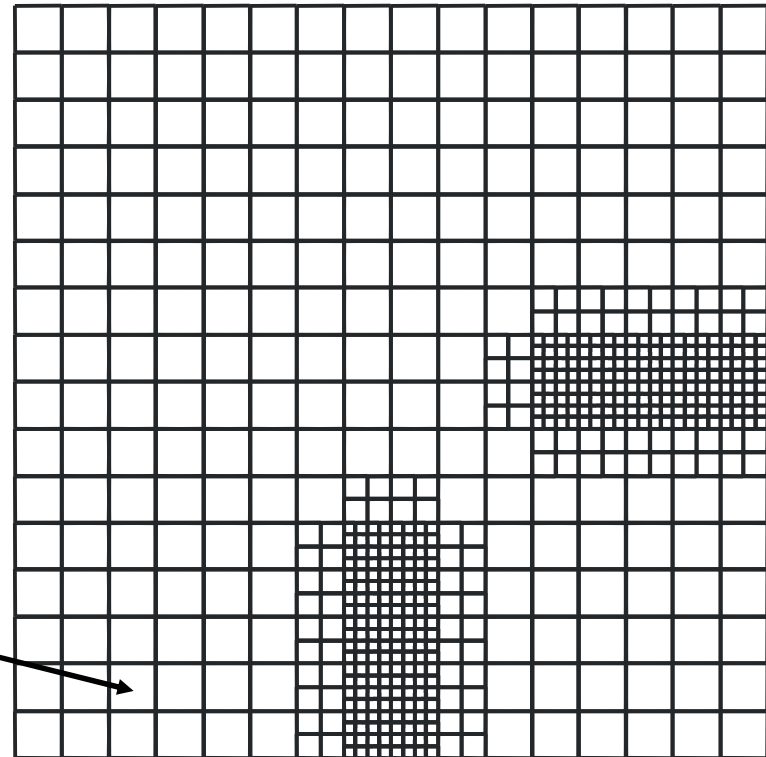
- Given an analytical surface description and boundary segments



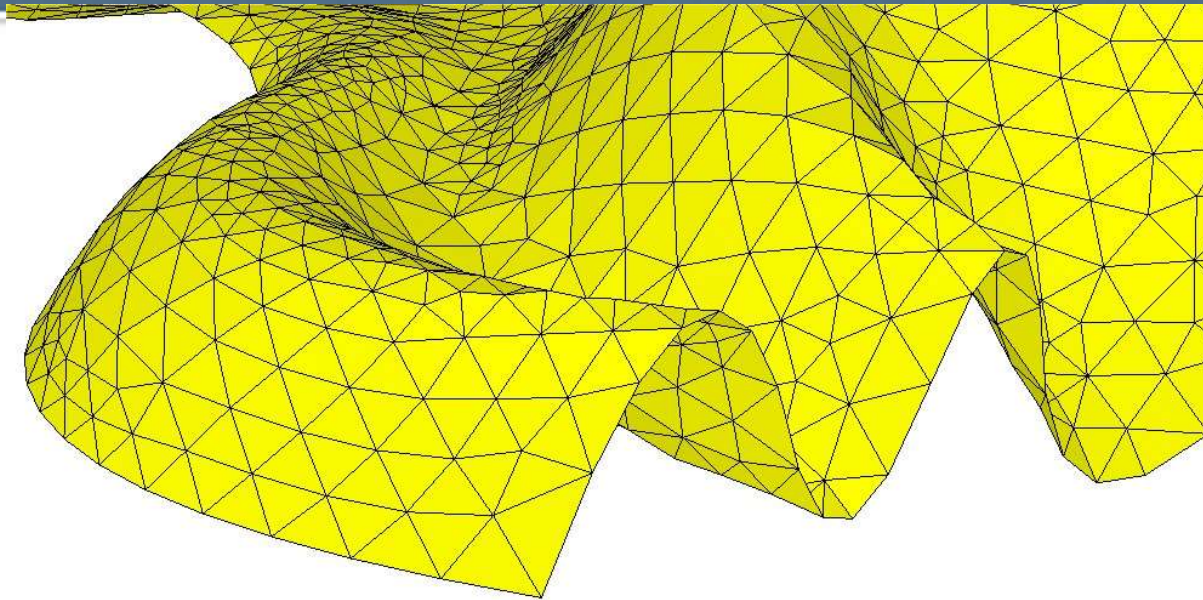
**Metric
Information**



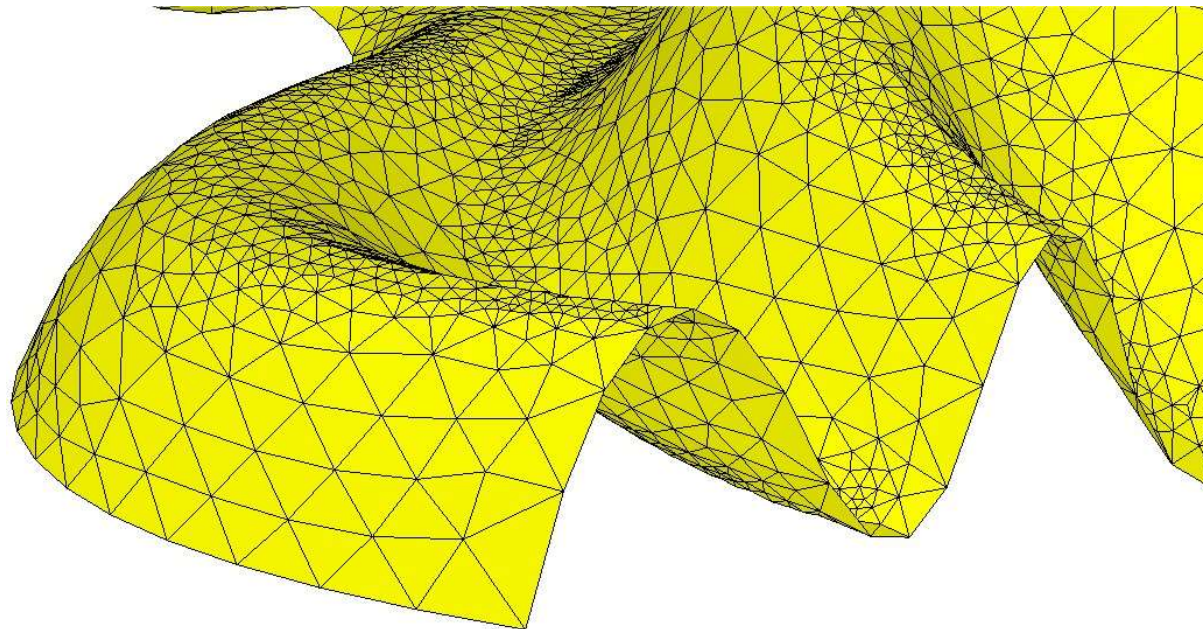
- **Background quadtree**



Importance of considering the curvature

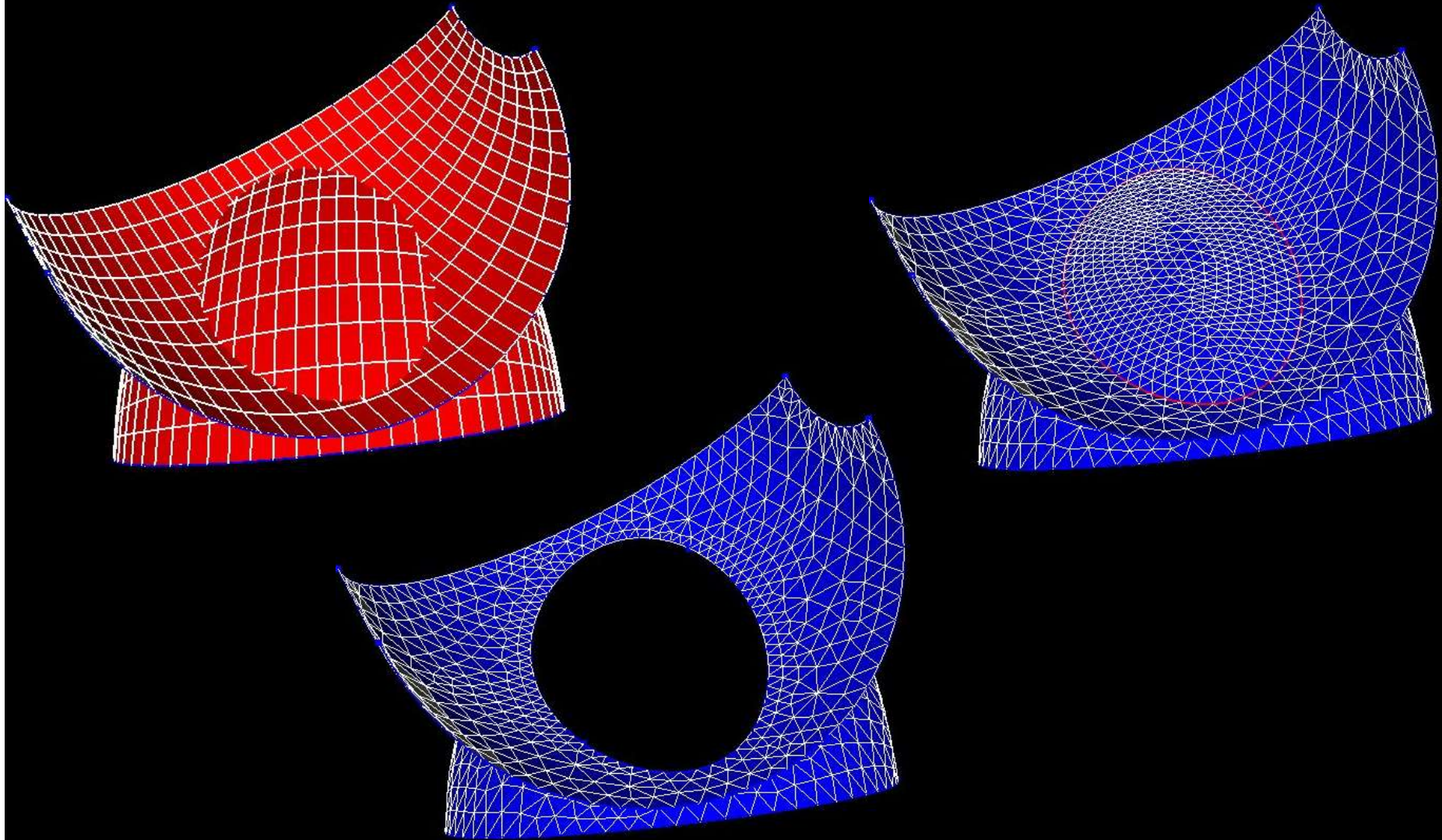


No consideration of curvature

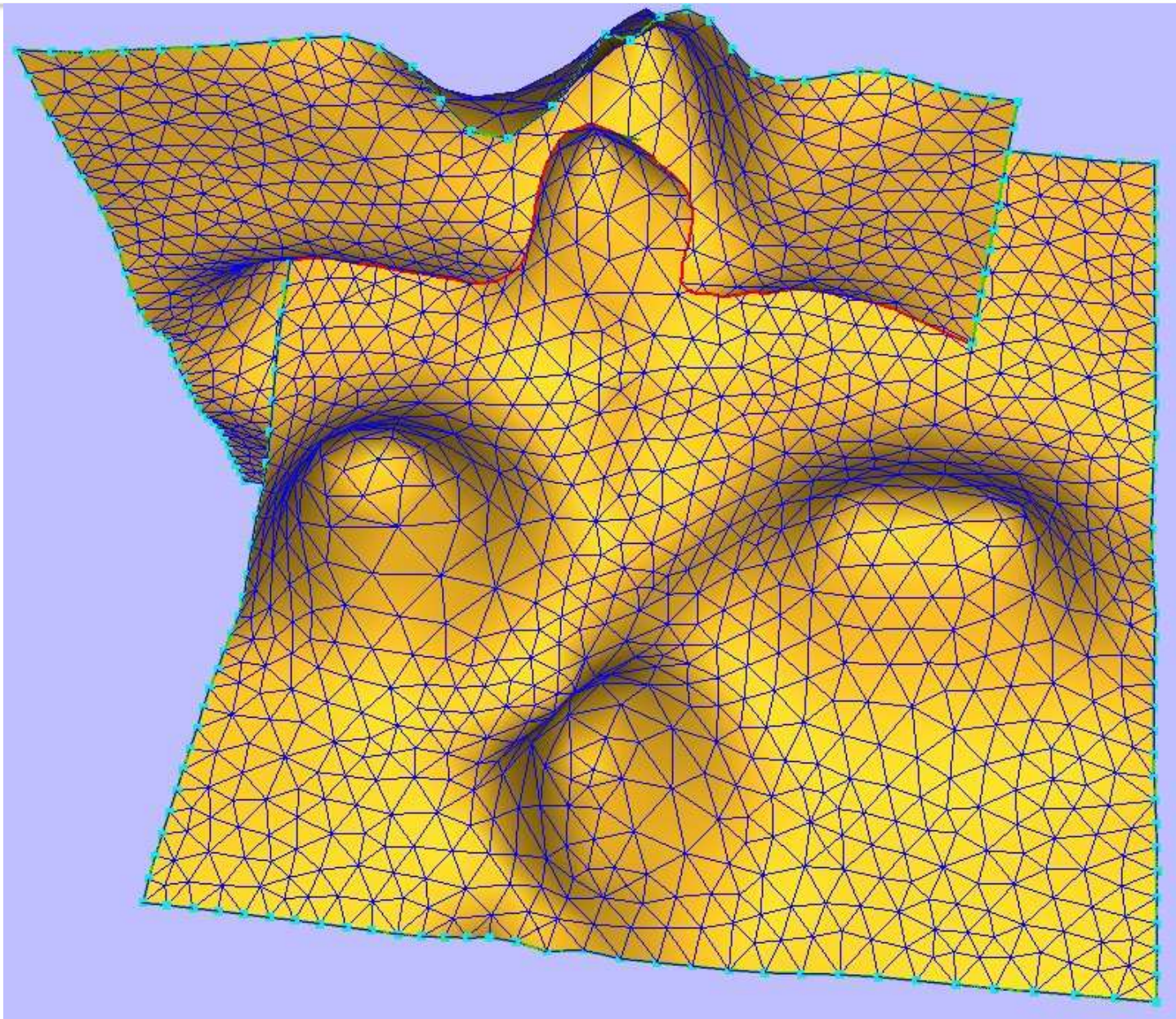


Consideration of curvature

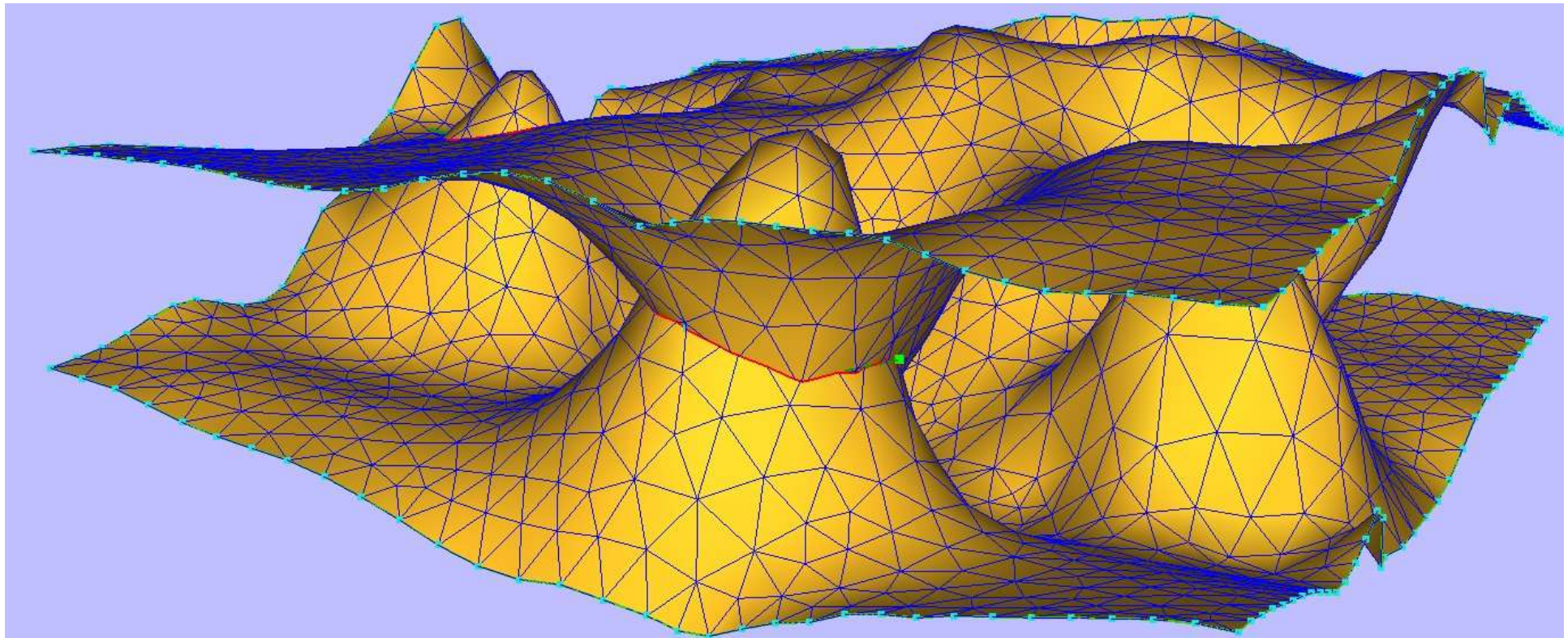
Surface mesh intersection



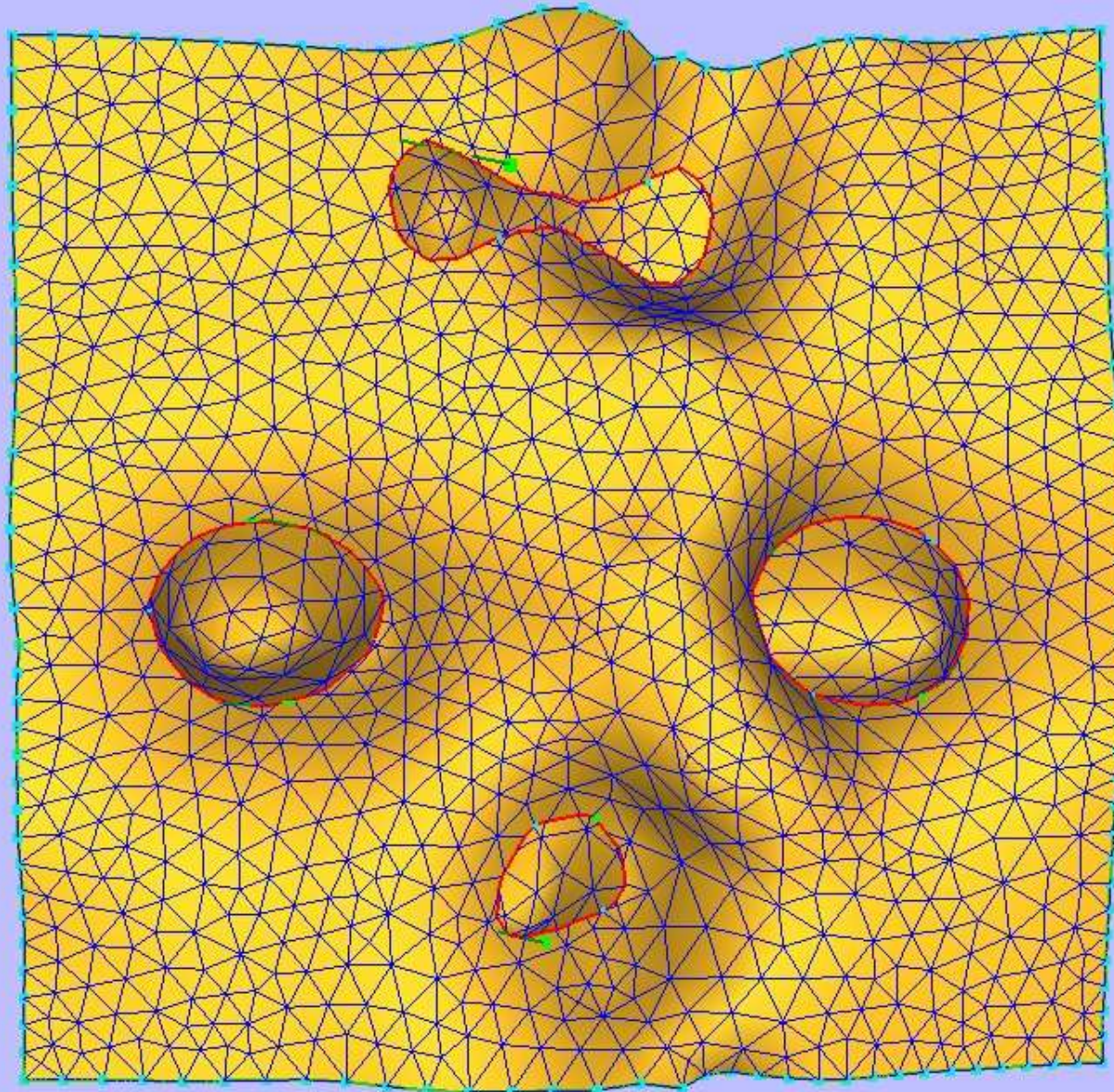
Surface intersection with exact arithmetic



Surface intersection with exact arithmetic



Surface intersection with exact arithmetic



Surface intersection with exact arithmetic

