Mesh Improvement Strategies

Two related ingredients

- Mesh correction indication to decide how the mesh is to be improved
- Mesh "enrichment method"

Most common procedure: employ the element contributions to the error with a single enrichment strategy

More optimal strategy would be use combination of information sources and enrichment methods

- Theory indicates higher rates of convergence can be obtained
- Correction indication for combined procedures is complex, but possible, for example, some heuristic methods have been developed for hp-adaptive
- Accounting for directional nature of solution

Mesh Enrichment Strategies

- Using mesh of elements of same order
  - Relocating nodes within a given mesh topology (r-refinement)
  - Nested refinement templates (h-refinement)
    - Non-conforming
    - Conforming
  - Remeshing
  - General local mesh modifications
- Altering the order on a fixed mesh
  - p-version finite elements
  - Spectral elements
- Addition of special functions
  - Elements with required jumps
  - Elements with proper order singular field
- Combinations of procedures
  - h- and r-refinement
  - hp-refinement
  - Etc.

r-Refinement

Correction indication

- Include positions of vertices in functional - too expensive
- Add nodal velocities as unknowns with penalty term to maintain mesh validity

Strategy

- Move mesh vertices to reduce error while ensuring mesh remains valid

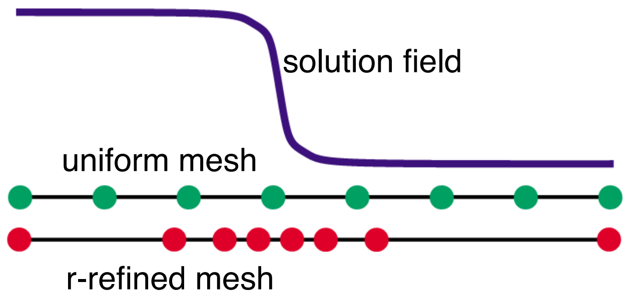
Advantages

- In some cases get large improvements for little effort
- No need to deal with mesh topology changes

Disadvantages

- Fixed limit on level of improvement possible
- Difficult to control on 2-D and 3-D meshes

Good option in combination with other methods



Nested Refinement

Correction indication

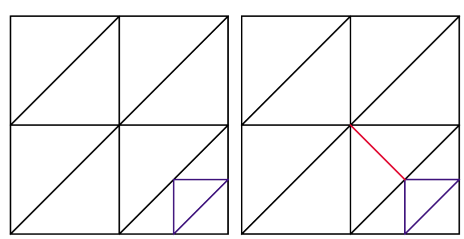
- Elements with large error subdivided with goal of equidistributing error

Strategies

- Bisection of elements yielding non-conforming meshes
- Bisection of elements with temporary splits of neighbors
- Application of strategy to maintain control of shapes such as split longest edge, of alternating edges split

Advantages

- Straight forward for non-conforming case
- *a-priori* control of element shapes
- Allows effective solution transfer processes
- Can obtain level of accuracy desired



bisection   bisection with temporary split

Disadvantages

- Dealing with constraint equation in non-conforming case
- Cannot coarsen past the initial mesh sizes
- Cannot account properly for curved domains

Remeshing

Correction Indication

- Need to use *a posteriori* information (error estimates, error indicators, or correction indicator) to construct new mesh size field
- Mesh size field typically defined discretely over previous mesh or some background grid

Strategy

- Employ automatic mesh generator that can function from a general mesh size field

Advantages

- Supports general changes in mesh size including construction of anisotropic meshes
- Can deal with any level of geometric domain complexity
- Can obtain level of accuracy desired

Disadvantages

- Cost of complete mesh generation
- Solution field transfer expensive and can be inaccurate

General Local Mesh Modification

Goal is the flexibility of remeshing while reducing some of the disadvantages

Correction Indication

- *a posteriori* information (error estimates, error indicators, or correction indicator) to mark elements or construct new mesh size field

Strategy

- Employ a "complete set" of mesh modification operations to alter the given mesh into the desired

Advantages

- Supports general changes in mesh size including construction of anisotropic meshes
- Can deal with any level of geometric domain complexity
- Can obtain level of accuracy desired
- Solution transfer can be applied incrementally - may have more control

Disadvantages

- Nearly as complex as complete mesh generation

Altering the Order on a Fixed Mesh

Correction Indication

- *a posteriori* information (error estimates, error indicators, or correction indicator) determine how to alter element basis functions

Strategy

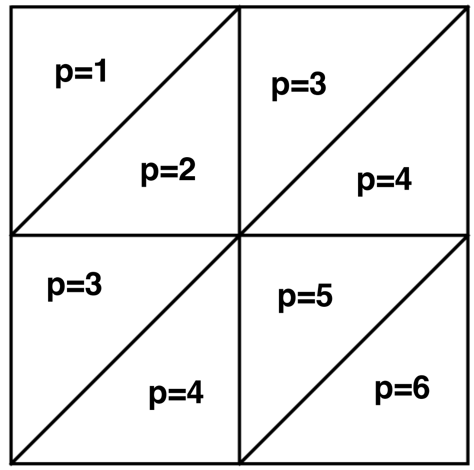
- Employ hierarchical spectral or p-version finite element basis to easily change order

Advantages

- For specific classes of problems method has improved orders of convergence
- Can support limited anisotropy
- Can obtain level of accuracy desired

Disadvantages

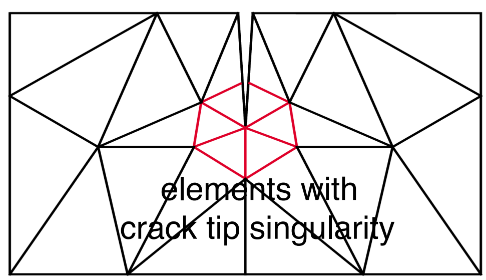
- Need analysis code that effectively supports variable order elements
- Dealing with curved element meshes



Addition of Special Functions

Correction Indication

- Indicators to detect and isolate features like jumps and singularities
- Procedures to detect order of singularity can be required



Strategy

- Add appropriate analytic functions for the jumps and singularities
- Tool like partition of unity functions and level sets being used to effectively add the desired functions

Advantages

- Can be quite effective when such features are present

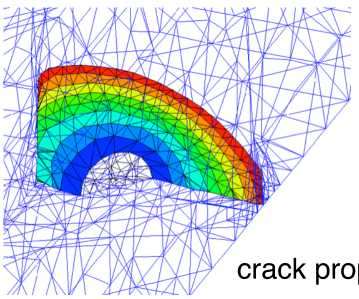
Disadvantages

- Need specialized methods not commonly supported by available codes
- Cannot ensure given level of accuracy

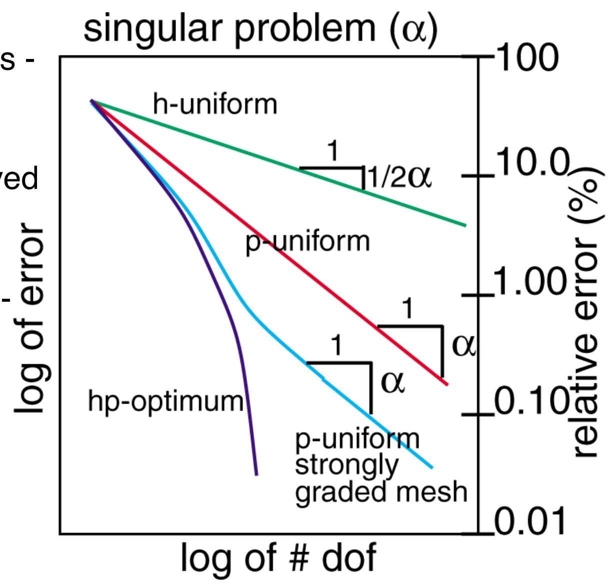
Good option in combination with others when appropriate features are present

Combination of Methods

- hr-for can get desired level of accuracy and advantages of r-refinement
- h-refinement and special elements - isolate singularity and refine to control the remaining error
- hp-adaptive method - gain improved rates of convergence possible
- Local mesh modification with p-refinement or special elements - can deal with evolving domains



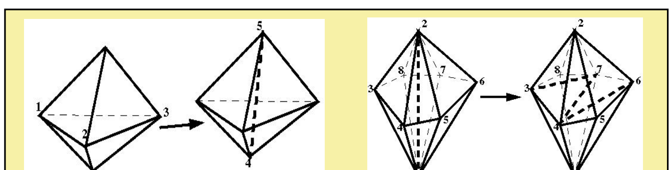
crack propagation



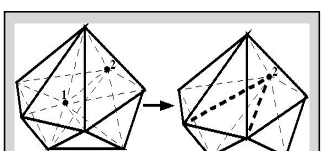
Simplex Element Mesh Modification Operators

Single step operators

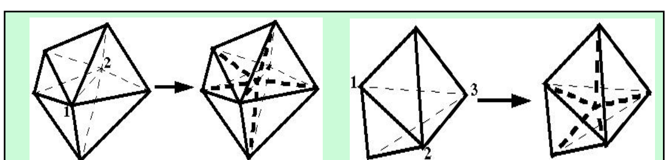
- swap
- collapse
- split
- Vertex motion



Face swap Edge swap



Edge collapse

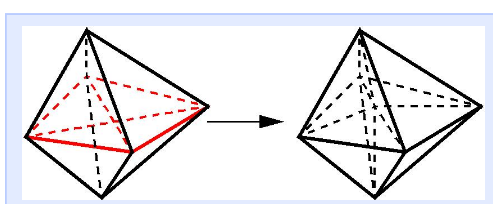


Edge split face split

Compound operators chain single step operators.

Examples of compound operators

- Double split collapse operator
- Swap(s) followed by collapse operator
- Split, then move the created vertex
- Etc.



Double split collapse to remove the red sliver

Implementation of Mesh Modification Procedure

Given the "mesh size field":

- Drive the mesh modification loop at the element level
  - Look at element edge lengths and shape (in transformed space)
  - If both satisfactory continue to the next element
  - If not satisfied select "best" modification
  - Elements with edges that are too long must have edges split or swapped out
  - Short edges eliminated
- Continue until size and shape is satisfied or no more improvement possible

Determination of "best" mesh modification

- Selection of mesh modifications based on satisfaction of the element requirements
- Appropriate considerations of neighboring elements (not fully resolved in the anisotropic case)
- Choosing the "best" mesh modification

Placing Vertices on Curved Boundaries

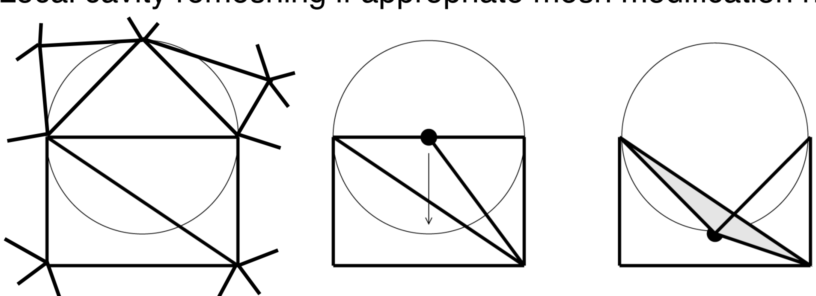
Must improve mesh geometric approximation as mesh is refined

New mesh vertices classified on boundary must be placed on the boundary

- In the case of curved domains the moving of vertices to the boundary can produce invalid elements
- Must locally correct the mesh in such cases

Three options for moving vertices to curved boundaries

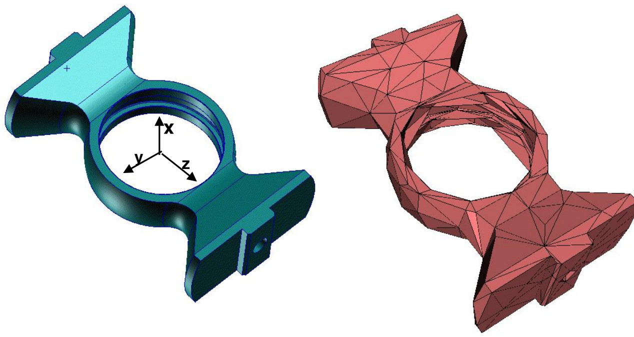
- Simple mesh motion when all connected elements remain valid and of acceptable shape
- Local mesh modification to eliminate the invalid elements created when moving the vertex
- Local cavity remeshing if appropriate mesh modification not satisfactory

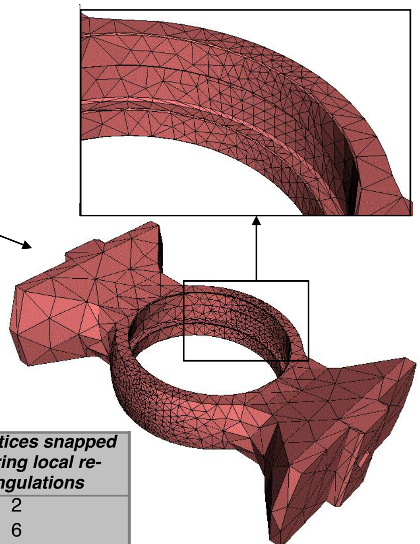


a) b) c)

Accounting for Curved Domains During Refinement

- Moving refinement vertices to boundary required mesh modification (see IJNME paper, vol58 pp247-276, 2003)
- Coarse initial mesh and the mesh after multiple refinement/coarsening





- Operations to move refinement vertices

Iterations of adaptations	# of vertices to be snapped	# of vertices snapped by a reposition	# of vertices snapped by local modifications	# of vertices snapped requiring local re-triangulations
1	342	204	136	2
2	485	369	110	6
3	340	286	52	2
4	74	34	40	-
5	26	3	23	-

Mesh adaptation to an Anisotropic Mesh Size Field

Based on applying mesh modifications following mesh metric

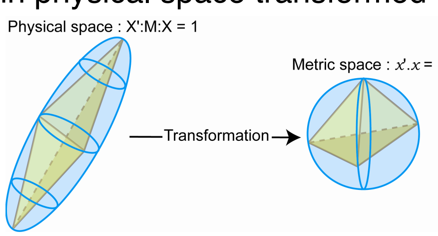
- Transformation matrix field T(x,y,z)

$$T(x,y,z) = \underbrace{\begin{bmatrix} 1/h_1 & 0 & 0 \\ 0 & 1/h_2 & 0 \\ 0 & 0 & 1/h_3 \end{bmatrix}}_{\text{Distortion}} \cdot \underbrace{\begin{bmatrix} \vec{e}_1 \\ \vec{e}_2 \\ \vec{e}_3 \end{bmatrix}}_{\text{Rotation}}$$

$\vec{e}_1, \vec{e}_2, \vec{e}_3$: **Unit vectors associated with three principle directions**

h_1, h_2, h_3 : **Desired mesh edge lengths in these directions**

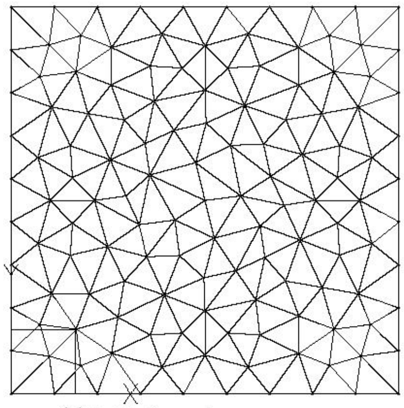
- Ellipsoidal in physical space transformed to normalized sphere



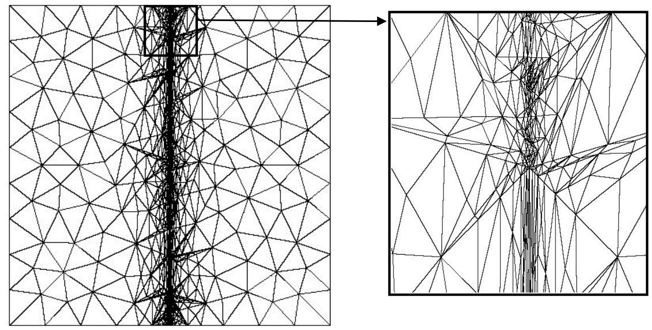
- Volume relation between physical space and the transformed space:

$$V_{transformed} = |T(x,y,z)| \cdot V_{physical}$$

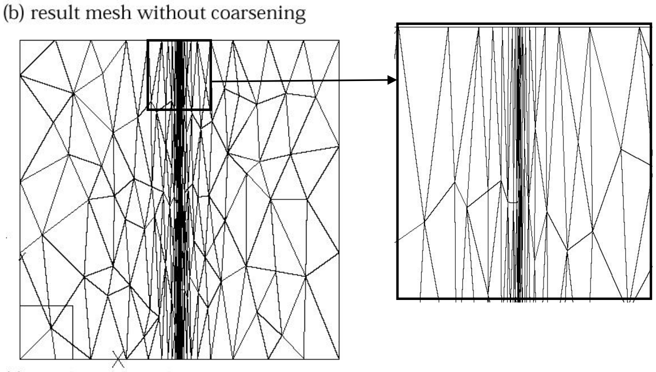
Example of Coarsening of Newly Created Edges



(a) initial mesh

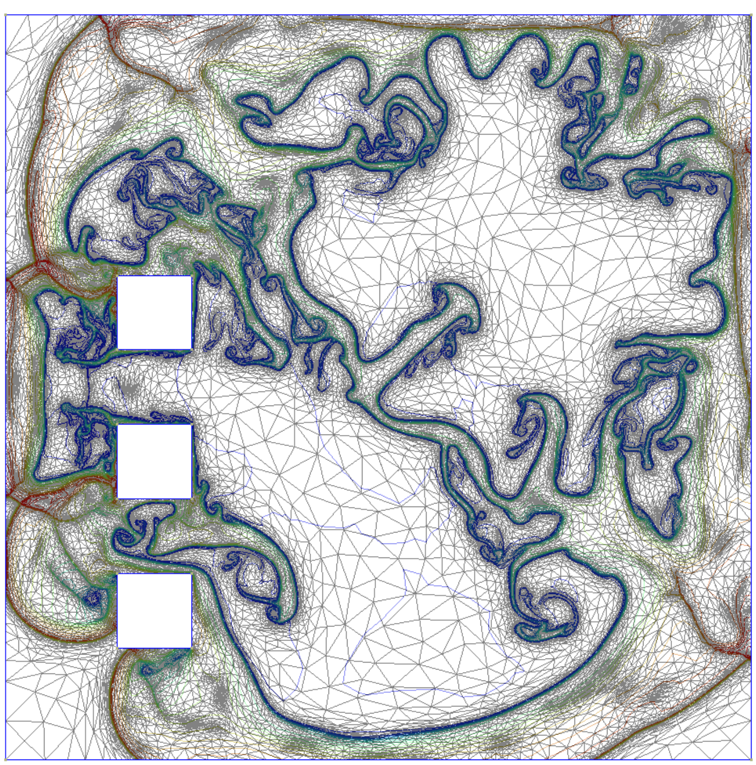


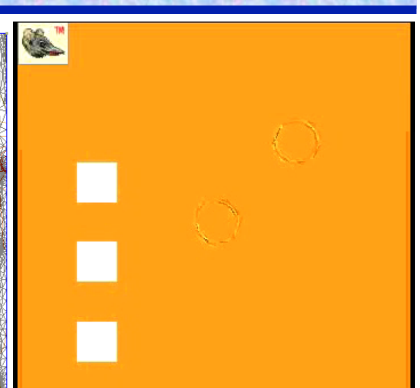
(b) result mesh without coarsening



(c) result mesh with coarsening

Colliding Explosions - 250 refinement steps

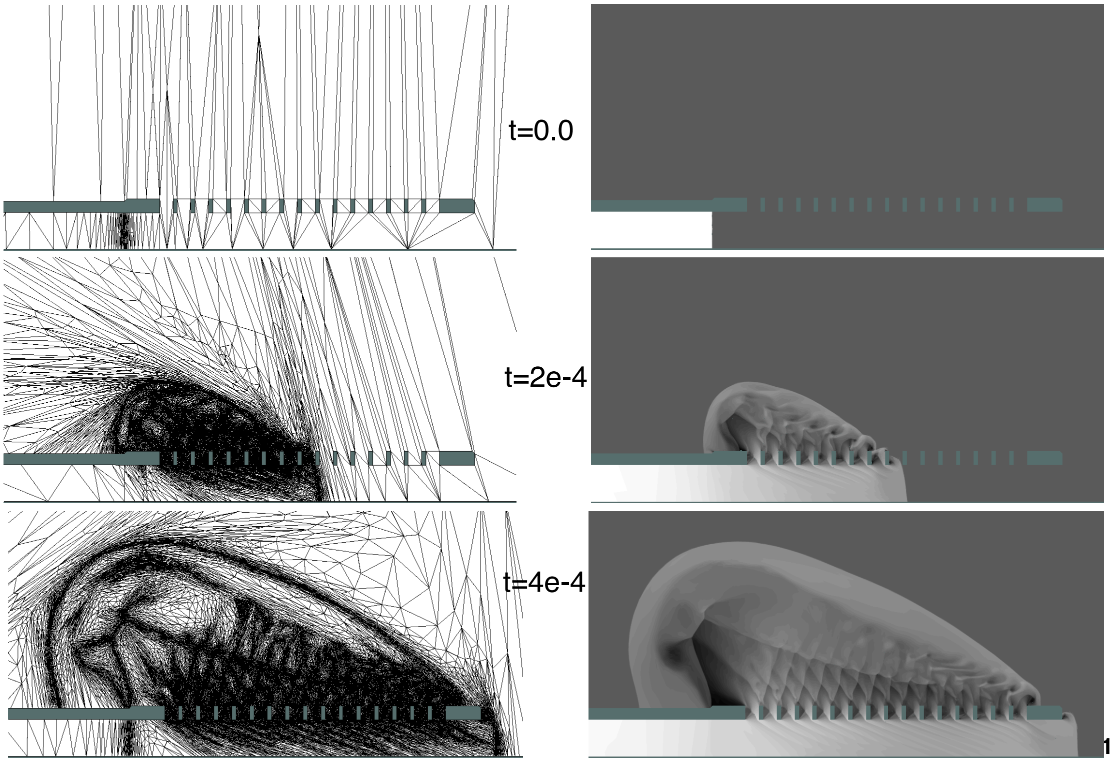




Evolution of density contours

Problem dominated by shock/contact interactions - A very complex problem

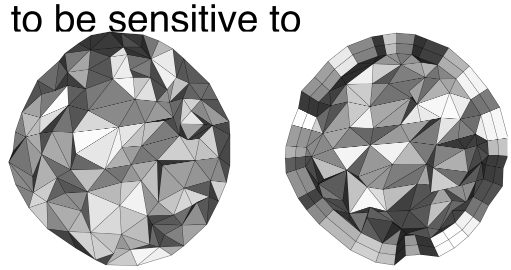
Muzzle Blast



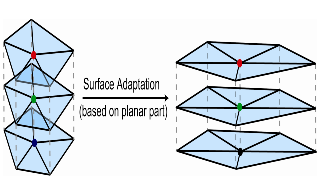
Maintaining “Structure” for Derivative Recovery

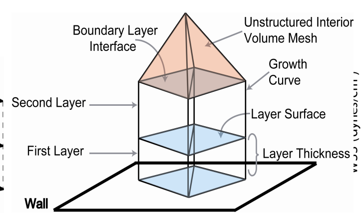
Post-processing procedure for recovering conservative wall shear stress has been observed to be sensitive to near wall mesh “structure”.

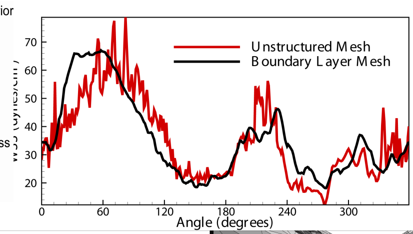
- Coarse example of arterial cross section

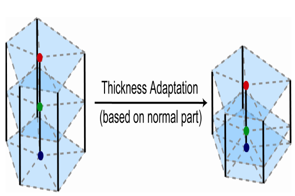


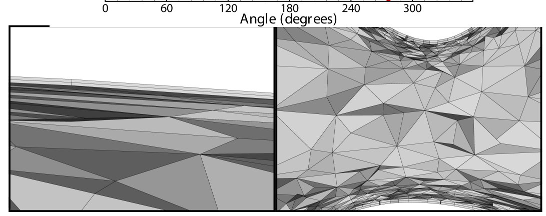
Semi-structured meshes mesh adaptation added





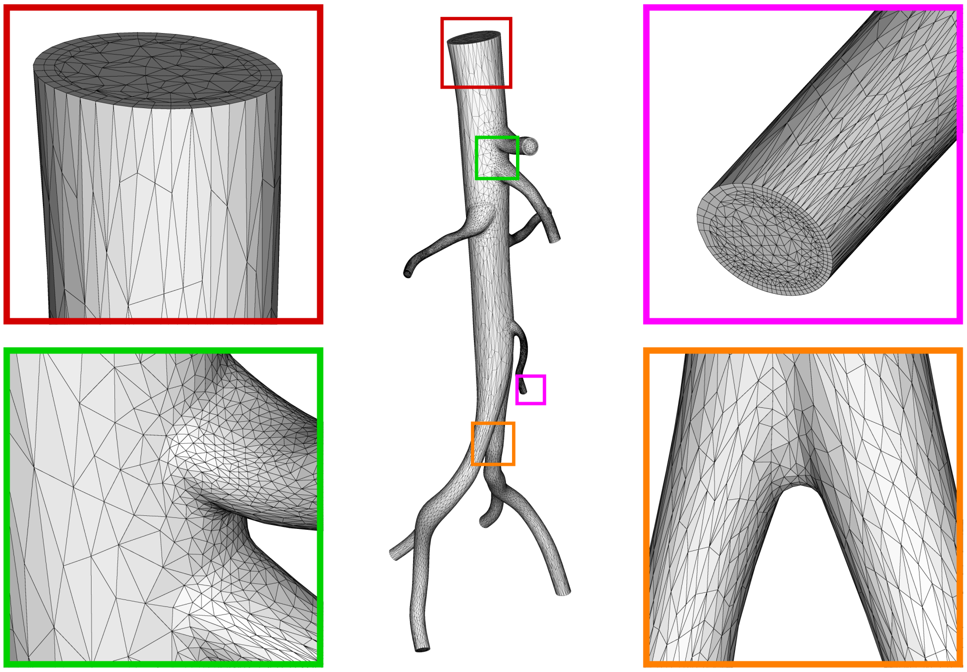






Example

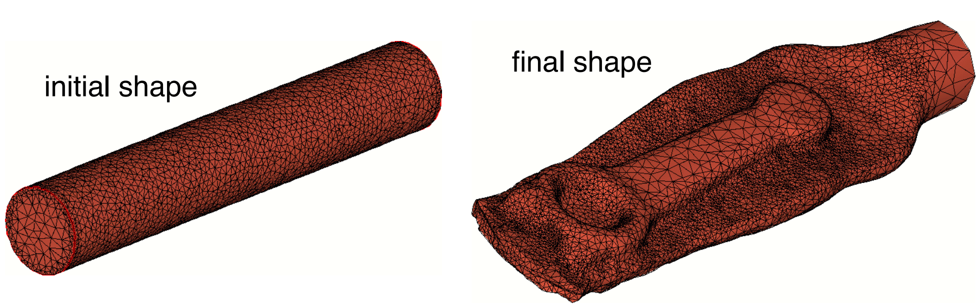
Surface of **adapted** mesh for human abdominal aorta



Adapting with Evolving Geometry - Metal Forming

Components of automated adaptive simulation

- Commercial analysis engine
- Monitoring of mesh discretization errors, and element shapes
- Model topology update
- Construct mesh size field based on discretization errors
- General mesh modification to obtain the desired mesh size field
- Adjust mesh size and shape to control geometric approximations
- Local solution transfer



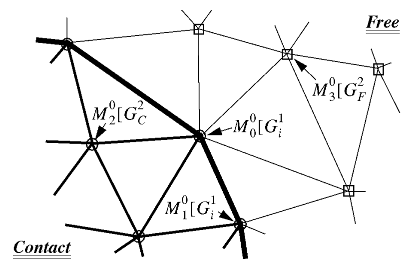
Model Topology Update

Geometric components in forming simulations

- Workpiece
- Dies
- Die motions

Model topology needs to be updated

- Contact conditions change as simulation proceeds
- Mesh updates require complete model topology
- Simulation engine tracts only nodal contact
  - Must update model topology based on this information
- Model update procedure
  - Maintain non-manifold model representation
  - Process uses initial classification to build up topology and then corrects ambiguities
  - Mesh classified against updated model topology
    - Mesh modifications controlled
    - Attributes properly associated to mesh



Free Surface Smoothing & Volume Control

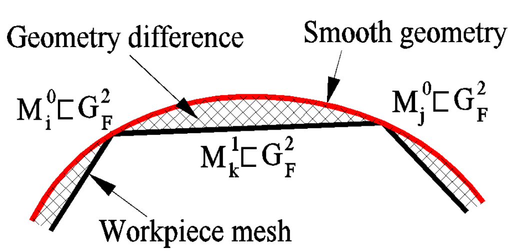
Accuracy degrades due to

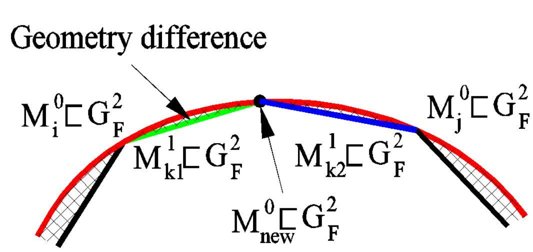
- Poor approximations to the smooth workpiece free surfaces
- Volume change of the workpiece during analysis and mesh updating

- Needs
  - Control the volume over the free surfaces in critical areas by
    - Volume compensation during free surface refinement
    - Constrain on the free surface coarsening based on curvature information

Workpiece free surface smoothing and volume control

- Interpolating subdivision surface procedures
  - For refinement of free boundary mesh edges,
    - Calculate target positions of the subdivision using interpolation template
    - Place new refinement mesh vertices to the these points





Geometric Approximation in Pre-contact areas

Contact prediction strongly influence by geometric approx.

- Time and location of the contact occurrence evolves
- Need for good geometric approximation in pre-contact areas

Mesh entities in pre-contact areas determined and geometric approximations improved as needed

- Candidate contact mesh entities
  - Mesh entities classified on the free model face, $M_i^{d_i} \sqsubset G_{Free}^2$, where $0 \le d_i \le 2$
  - Expected to come in contact with die surfaces in a few analysis increments
- Construction of smooth approximation on candidate contact areas
  - Steps
    - Prediction of the candidate contact mesh entities via an octree
    - Evaluation of geometric approximation on the candidate contact areas
    - Improvement of geometric approximation in the required areas

History Variable Solution Transfer

Variables to be transferred from original mesh to updated mesh

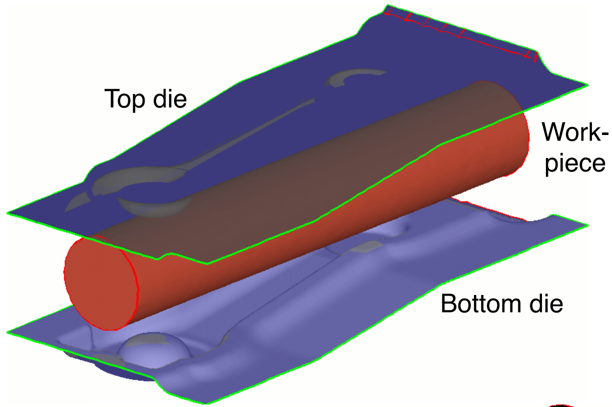
- Nodal velocities
- Nodal temperature
- Elemental effective strains
- Elemental stress tensor (for elasto-plastic materials)

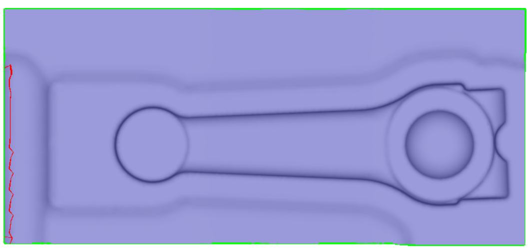
Two approaches for solution transfer

- Global solution transfer
  - Computationally expensive
  - Procedures tend to diffuse information - accuracy loss
- Local solution transfer
  - Performed as local mesh modification performed
  - Limited number of elements involved - efficient
  - No accuracy loss with some operations, others easier to control due to local nature

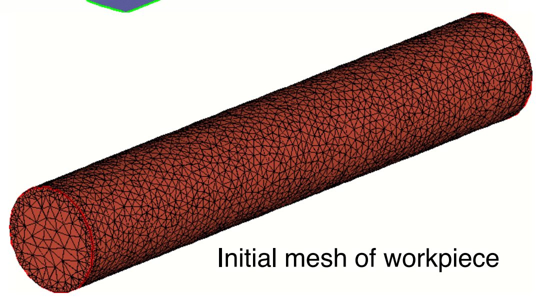
Example Simulation: Steering Link Problem

- Problem definition
  - Material type = Plastic
  - Total stroke = 41.7 mm
  - Stroke/step = 0.15 mm
  - Total steps = 279
  - Allowed GI δ_p = 1.00 mm
  - Target volume = 24011mm^3
  - Starting workpiece mesh:
    - 6765 mesh vertices
    - 28885 mesh regions.



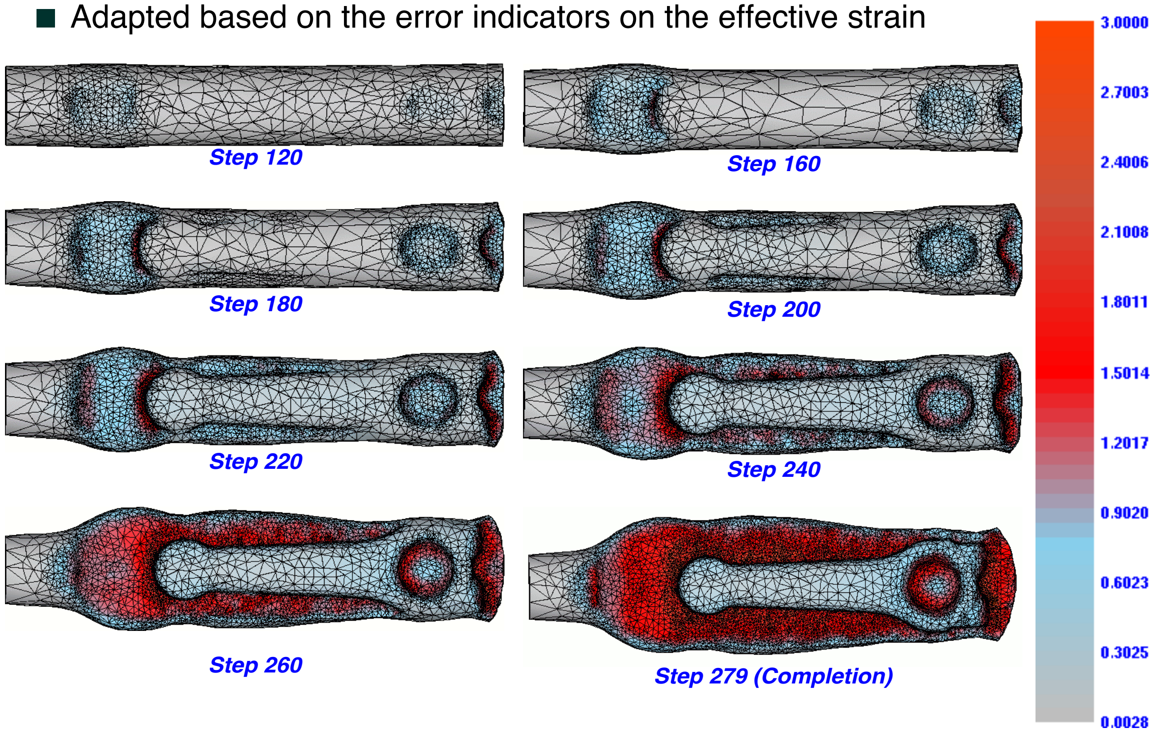


Bottom Die (Top view)



Steering Link Problem

- Adapted based on the error indicators on the effective strain

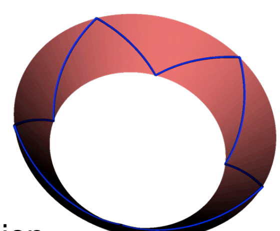


Meshes for p-Version FEM on Curved Domains

Requirements

- Coarse, strongly graded meshes
- Must ensure the validity of curved elements
- Element geometric order and level of geometric approximation need to be related to geometric shape order
- Approach being taken
  - Construct algorithms using key existing straight edged meshing tools
  - Add new components for curved elements
- Steps in the procedure
  - Automatic identification and isolate singular features
  - Create graded linear feature isolation mesh around features
  - Generate coarse linear surface mesh accounting for the boundary layers
  - Curve coarse surface mesh to boundary
  - Curve graded linear feature isolation mesh
  - Generate coarse linear interior mesh
  - Modify interior linear mesh to ensure validity with respect to the curved surface and graded linear feature isolation mesh

Application of Bezier Polynomial in Curved Meshing



- Advantageous properties of Bezier polynomials
  - Can be as high a degree as desired
  - Convex hull provides smoother and more controllable approximation
  - Derivatives and products of Beziers are also Beziers
  - Efficient algorithms for degree elevation and subdivision
  - Better properties to allow more efficient intersection checks

- Mesh region validity determination
  - Traditional validation methods test the Jacobian at integration points
  - A general validation for Bezier Regions is provided: Relate Jacobian to the region control points and determine its minimum bound
    - A region is valid globally if the minimum control point of the Jacobian determinant function is > 0

Mesh Modification

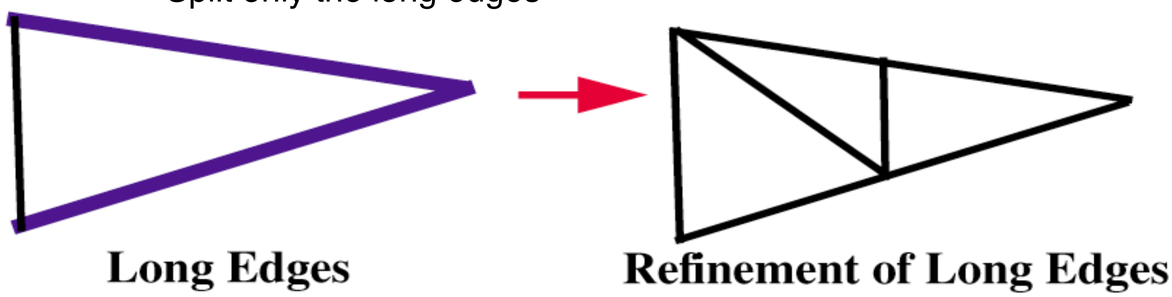
"Standard approach"

- Independent application of refinement/coarsening and shape control
- Adaptive refinement based on templates
- Some procedures limit coarsening to undoing refinements
- Shape control often limited to vertex repositioning

Current approach focused on examination of local situation

- Mesh modification selected from full set of operations with the goal of satisfaction of general mesh size field

Example: Refinement of element with poor shape caused by a short edge - Split only the long edges



Element Shape

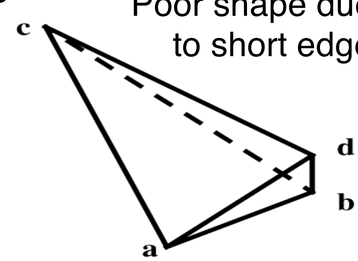
For isotropic meshes several efficient measures possible (Operations for anisotropic meshes done in transformed space)

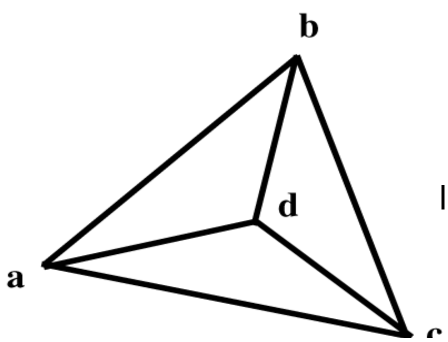
- Can normalize to 1 for equilateral tetrahedron and 0 for zero volume
- Use such a measure to identify elements of unsatisfactory shape

Consideration of operations to improve shape

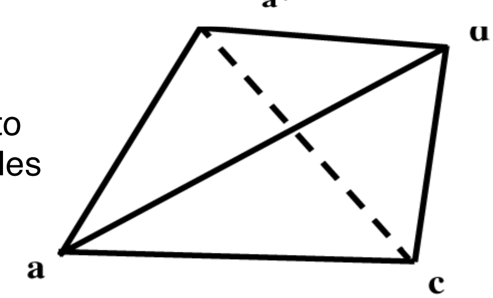
- Requires more careful analysis
  - Dihedral angles
  - Edge lengths
- Best operators to improve based on situation

Poor shape due to short edge





Poor shape due to large dihedral angles



Implementation of Mesh Modification Procedure

Given the “mesh size field”:

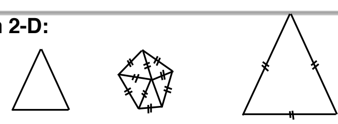
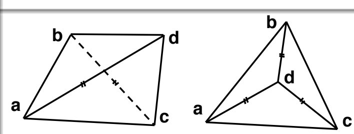
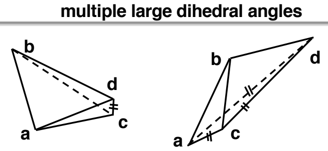
- Drive the mesh modification loop at the element level
  - Look at element edge lengths and shape (in transformed space)
  - If both satisfactory continue to the next element
  - If not satisfied select “best” modification
  - Elements with edges that are too long must have edges split or swapped out
  - Short edges eliminated so long and other criteria remain satisfied
- Continue until size and shape is satisfied or no more improvement possible

Determination of “best” mesh modification

- Selection of mesh modifications based on satisfaction of the element requirements
- Appropriate considerations of neighboring elements (not fully resolved in the anisotropic case)
- Ranking mesh modifications by utility values and choosing the “best” mesh modification

Choice of Desired Mesh Modifications

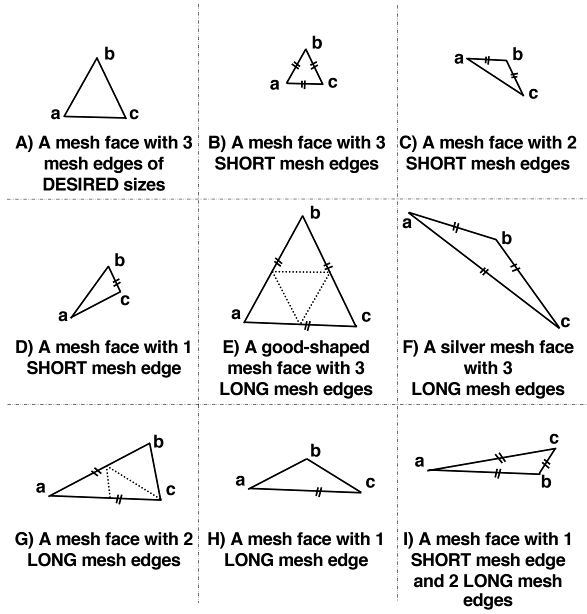
Desired mesh modifications based on satisfaction analysis of element requirements

Satisfaction on Requirements	Examples	Analysis of Unsatisfaction	Desired Mesh Modifications
Good shape + Undesired sizes	In 2-D: 	Consistently too short or long edges ==> too small or large element	Local coarsening or refinement
Bad shape + Desired sizes	 “Flat” elements with desired sizes contain multiple large dihedral angles	2 opposite or 3 neighbor ing large dihedral angles ==> flat element	Collapse it if flat enough and topologicall allowed Remove all large dihedral angles by swapping or collapsing any mesh edge bounding a large dihedral angle
Bad shape + Undesired sizes	 Poor Element with a short edge Poor Element with 1 short edge and 2 long edges contains large dihedral angles	Large dihedral angles or relatively “short” edges ==> poor element	Collapse it if flat enough and topologically allowed. Handle short edges first and long edges later based on analysis of short/long edges at the face level by collapsing short edges/small faces, or swapping/refining long edges

Face-Based Considerations of Undesired Edges

Choice of most desired mesh modifications based on face-level configuration of undesired mesh edges

Undesired Edges In A Face	Figure	Choice of Most Desired Mesh Modification
3 short edges	B	Collapse the small mesh face, *a-b-c*
2 short edges	C	Collapse one of the short edges from *b* à *a* or *b* à *c* or collapse the small mesh face, *a-b-c*
1 short edge	D	Collapse the short mesh edge, *b-c*
3 long edges in a good-shaped face	E	Refine the large mesh face, *a-b-c*, properly
3 long edges in a silver face	F	Swap the long mesh edges appropriately
2 long edges	G	Swap or refine the long mesh edges, *a-b* and *a-c*
1 long edge	H	Swap the long mesh edge, *a-c*
Short + long edges	I	Handle short edges first (Collapse short edge *b-c*)



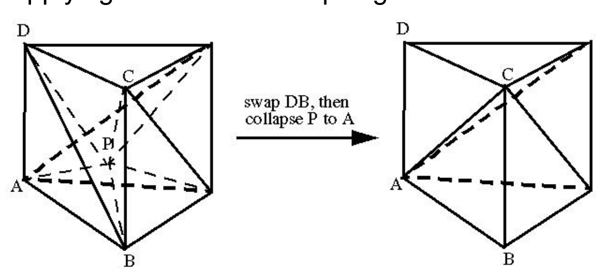
SCOREC

Eliminate Short Edges

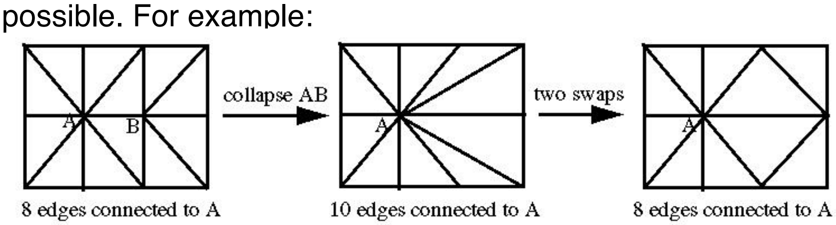
The sequence to determinate a desired modification

- First collapse the short edge by removing either its bounding vertex if possible
- Use compound operator: “swap(s)/collapse(s) + collapsing the short edge”
  - The pre-swap(s) or collapse(s) is determined by analyzing the tetrahedra that become invalid after applying the desired collapsing

    For example:



- After a collapsing, always check the number of edges connected to retained vertex, swap(s) from the longest edge to reduce that number to previous level if possible. For example:

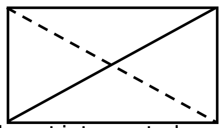


SCOREC

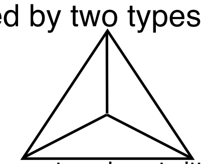
Eliminate Small Volumes

Analysis of possible situations:

- Without short edges, small volume is created by two types of poor connectivity

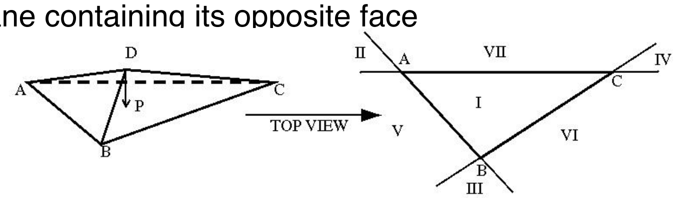


(1) two almost intersected opposite edges



(2) a vertex almost sitting in its opposite face

- the two types can be identified by projecting a vertex of the sliver tetrahedron onto the plane containing its opposite face



- The key problem mesh entities can be determined by the projection location

	A face/vertex pair				Two opposite edges		
Projection location	I	II	III	IV	V	VI	VII
Key Problem mesh entities	Face ABC and Vertex D	Face BCD and vertex A	Face ACD and vertex B	Face ABD and vertex C	Edge AB and edge CD	Edge AD and Edge BC	Edge AC and Edge BC

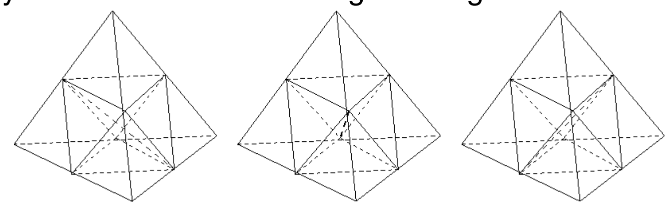
Eliminate Small Volumes

Sequence to determinate a desired modification

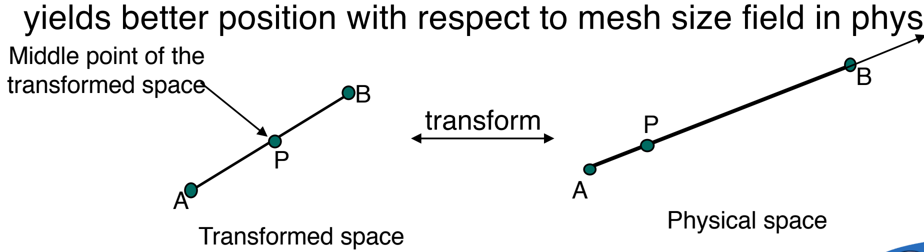
- For small volume due to two opposite edges
  - In case on boundary, simple remove it if possible
  - Swap either of the two opposite edges if possible
  - Split the two opposite edges, followed by collapsing the new diagonal edge if possible
- For small volume due to a vertex/face pair
  - In case on the boundary, simple remove it if possible
  - Split the face and collapse the new interior edge if possible
  - Swap one of the three bounding edges of the face if possible
  - Collapse the vertex to any of the other three vertices

Refinement

- Use edge split templates
  - Forty-two possible configurations to triangulate a tetrahedron
  - Always collapse interior vertices that have to be inserted in four of the forty-two configuration through swap + collapse operations
  - Always create the shortest diagonal edge in case ambiguous

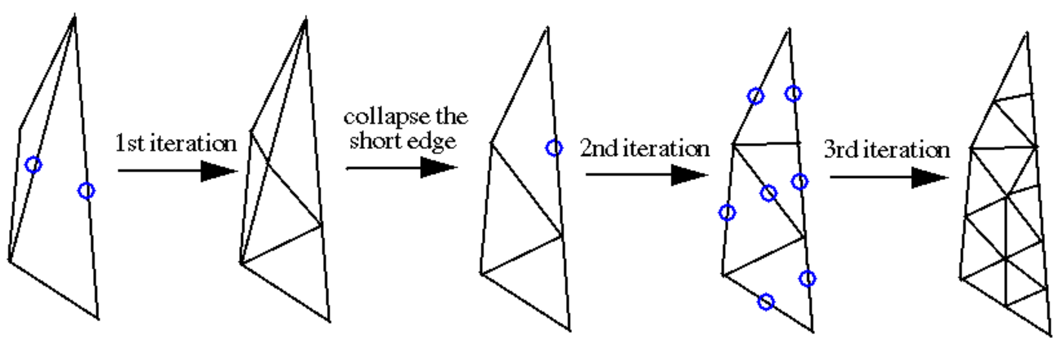


- Adding new vertices at the middle point of edges in transformed space yields better position with respect to mesh size field in physical space



Refinement

- Refinement of edges
  - Allow several steps to reduce element size to desired level
  - At each iteration, only split the longest edge and those edges close to the longest with respect to the mesh size field
- Collapse the short edges refinement produced before starting the next iteration



Blue circles indicate refinement edges in each iteration

Placing Vertices on Curved Boundaries

Input: list of refinement mesh vertices classified on curved b'drys

Two parts of the procedure:

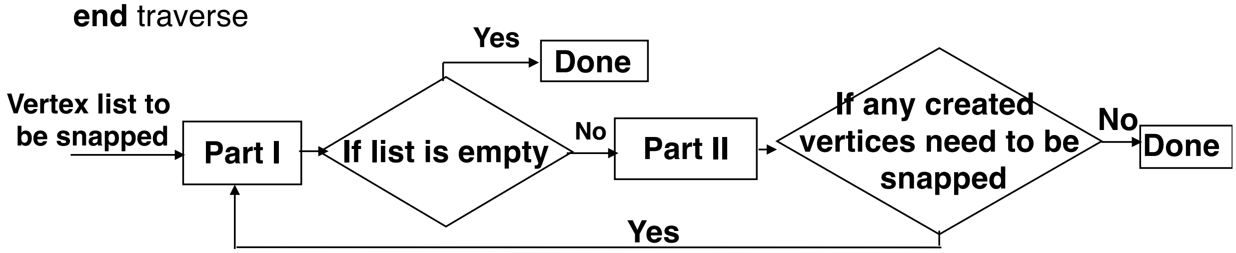
- Part I: Process all that move directly or require local modification only

While any mesh vertex in the list still can be moved ahead
    determine target move for next vertex in list
    if the current vertex can move to target location without a problem **then**
        move to that location and remove vertex from list
    else
        determine the first problem preventing motion
        define a move that ensures the first problem is "passed"
        "analyze" current situation to determine "best" local modification
        if there is an acceptable local modification **then**
            perform selected local modification and remove from list if at target
        end if
    end if
end while

Placing Vertices on Curved Boundaries

- Part II: Ensure moving remaining vertices in the list using cavity re-triangulation

Traverse remaining vertices in the list once
    perform local cavity construction
    apply cavity mesher, remove form list & add any new ones created into list
end traverse

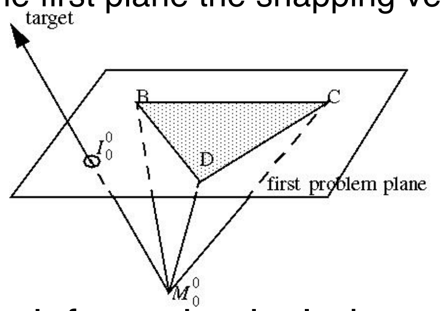


New boundary mesh vertices may be created during cavity re-triangulation. But, this is not common, and they are typically "closer" to boundary

Analyze Current Situation

Definition of first problem plane

- The plane containing a mesh face opposite to current snapping vertex
- The first plane the snapping vertex runs into while moving toward target

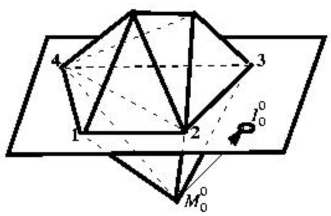


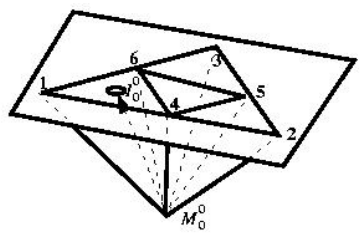
M_0^0: the mesh vertex classified on the model boundary not snapped

B-C-D: a mesh face opposite to M_0^0

Other information includes:

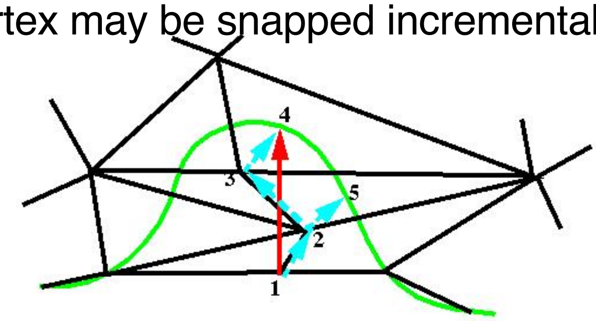
- Key mesh entities to cause mesh invalid
- Problem mesh faces on the first problem plane





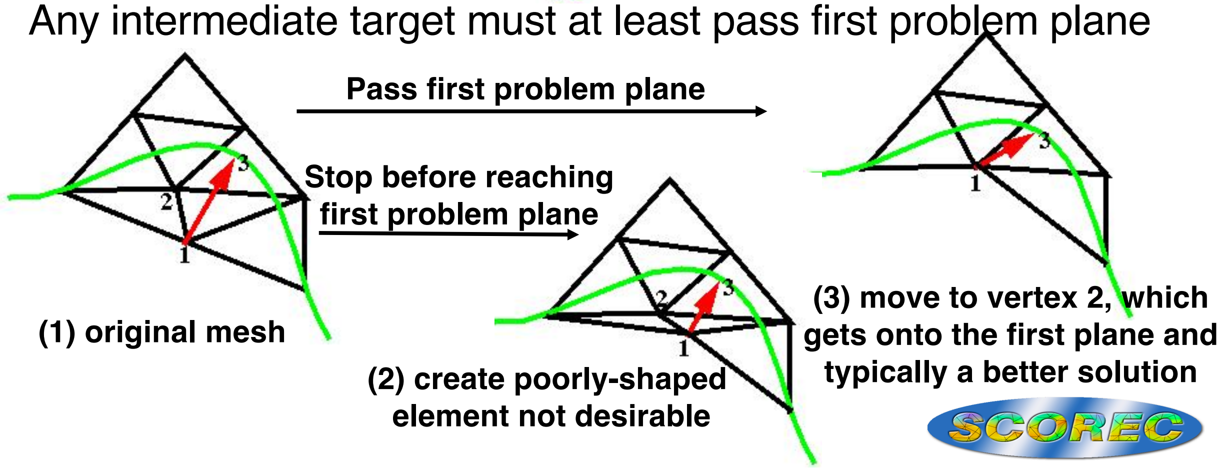
Need to Move Past First Problem Plane

Vertex may be snapped incrementally along different path

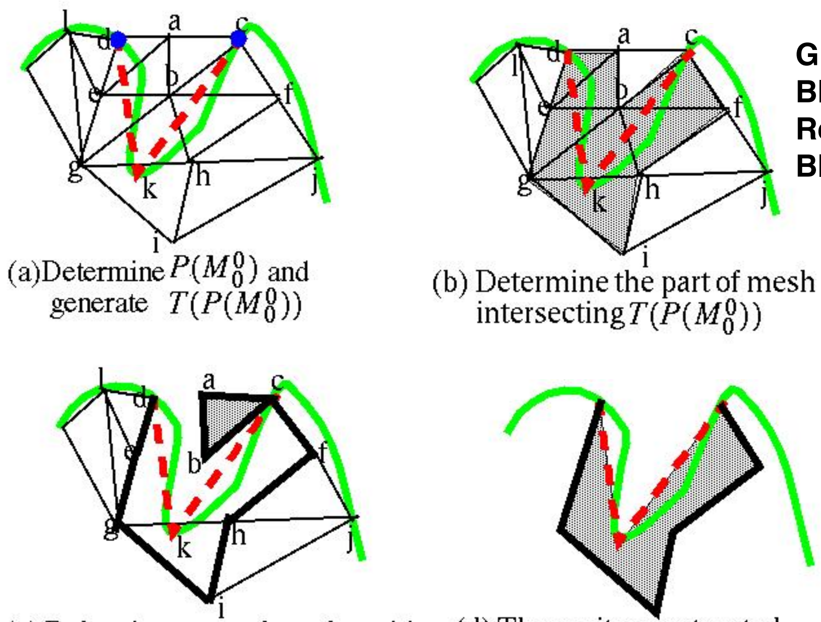


Green line: model
Black line: mesh
Red arrow: snap direction
Cyan arrow: possible snap path

Any intermediate target must at least pass first problem plane



Cavity Creation and Meshing to Place Vertices on B'dry



(a)Determine $P(M_0^0)$ and generate $T(P(M_0^0))$

(b) Determine the part of mesh intersecting $T(P(M_0^0))$

(c) Delete intersected mesh entities and determine these interacted

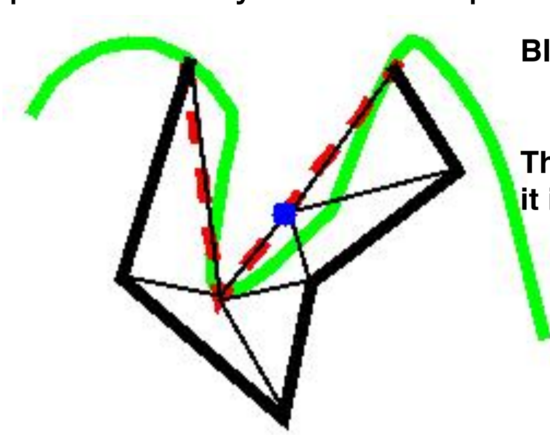
(d) The cavity constructed

Green line: model edge
Black line: mesh edge
Red dash line: $T(P(M_0^0))$
Blue dot: $P(M_0^0)$

Handling of New Boundary Vertices Created

This procedure may create new vertices that need to be snapped – not common and typically "close" to its classified model boundary

These new vertices will be added into the vertex list to be processed by the overall procedure

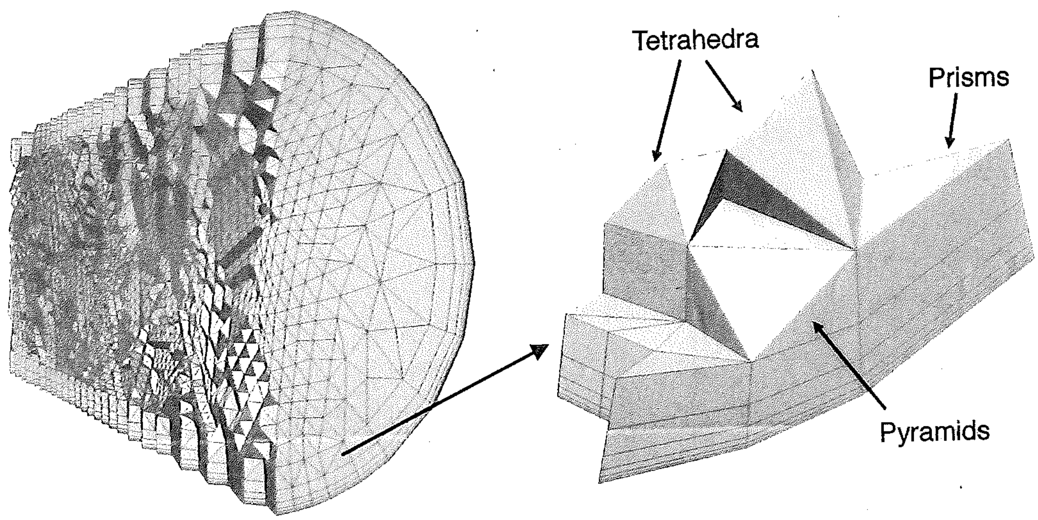


Blue dot: A new boundary vertex introduced during re-triangulation

The new vertex is easier to be snapped since it is "close" to classified model boundary

Meshes with Boundary Layers

- Applied in fluid problems such as viscous flows
  - Experience highly anisotropic boundary layers near no-slip walls
- Anisotropic mesh adaptation with boundary layers vs isotropic one
  - Yields meshes with same level of accuracy
  - Results in one to two orders of magnitude fewer elements



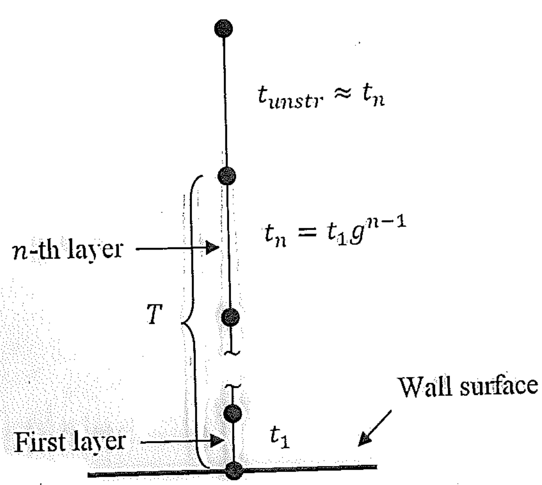
Meshes with Boundary Layers

Normal to the surface spacings define stack quantities

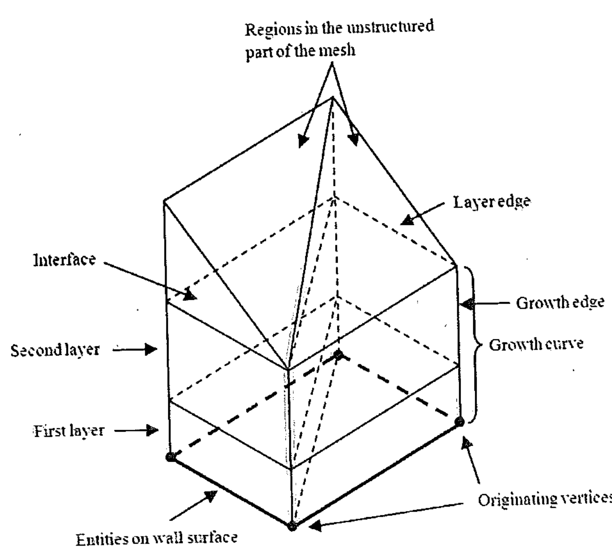
- t_1 – initial spacing off the wall
- g – gradation factor
- n – number of layers
- T – total thickness

$$t_i = t_1 g^{i-1} \qquad (1)$$

$$T = t_1 \sum_{i=1}^{n} g^{i-1} \qquad (2)$$



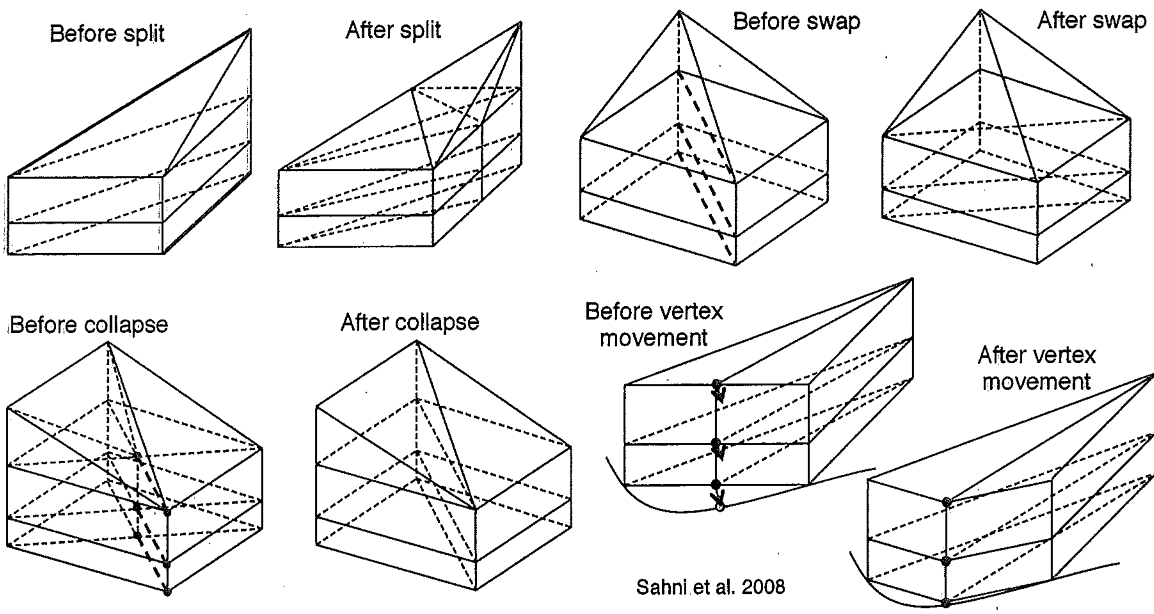
Spacing of layers on the vertex stack



Decomposition of a boundary layer stack

Boundary Layer Mesh Modification Operations

A modification operation on any layer is propagated through the stack and interface elements to preserve attributes of layered elements



Sahni et al. 2008

Implementation of Mesh Adaptation Procedure

Given the "mesh size field":

- Drive the mesh modification loop at the region and stack of regions level
  - Look at region edge lengths and shape (in transformed space)
  - If both satisfactory continue to the next region
  - If not satisfied select best modification
  - Entities with edges that are too long must have edges split or swapped
  - Short edges are eliminated with coarsening
- Continue until size and shape is satisfied or no more improvement possible

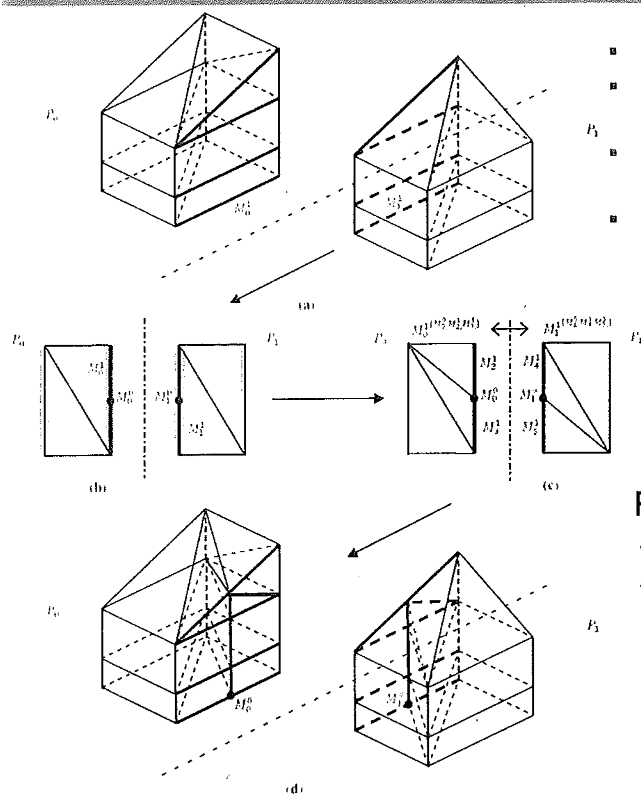
Li et al. 2005: 3D anisotropic mesh adaptation by mesh modifications

Accounting for boundary layer stacks

- Boundary layer part of the mesh has priority in applying mesh modification operations
- Modification of not-in-stack mesh entities has more flexibility
- Easier to satisfy the desired mesh resolution

Sahni et al. 2008: Adaptive boundary layer meshing for viscous flow simulations

Parallel Boundary Layer Mesh Adaptation

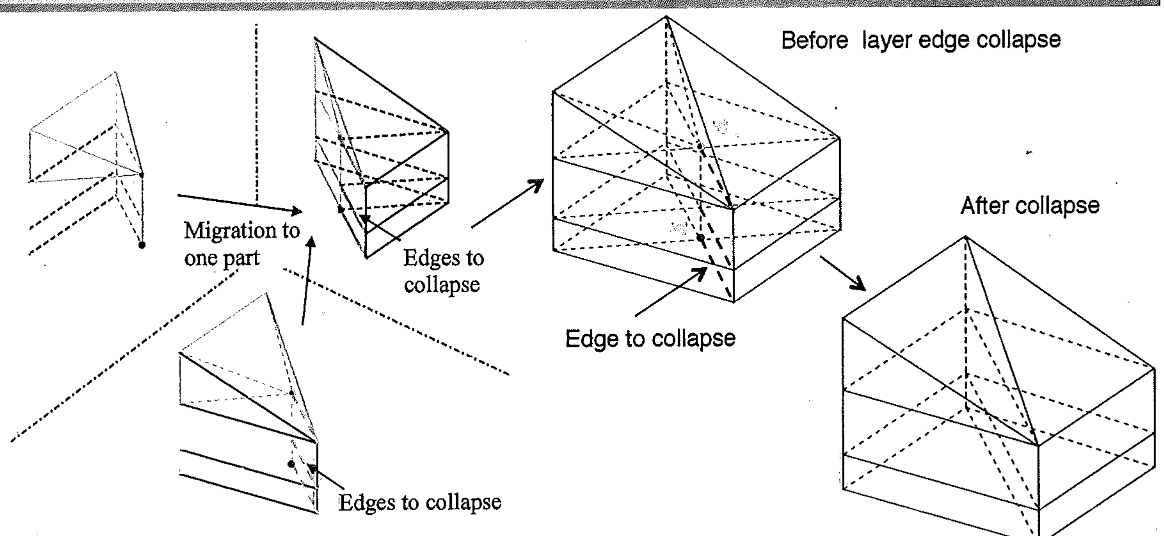


- Entity sets are used to support BL stacks
- A group of entities which stick together on part and perform only collective operations
- A mesh entity only associates with a single entity set, and is defined once in the entity set
- Entity set information is maintained consistently during mesh migration

Refinement

- Similar to the unstructured one
- Requires proper linkage of newly created entities on the part boundary for stack entities and the interface

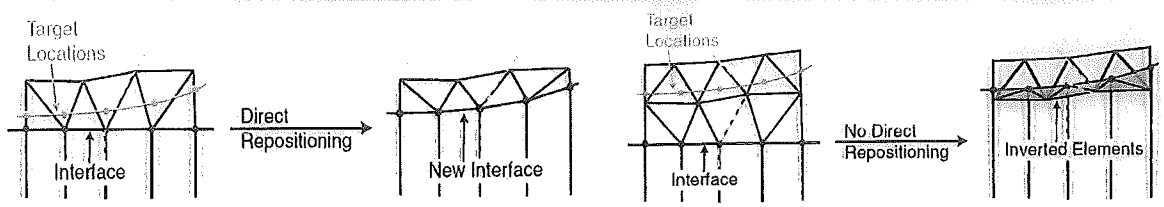
Parallel Boundary Layer Mesh Adaptation



Coarsening, shape correction and vertex repositioning

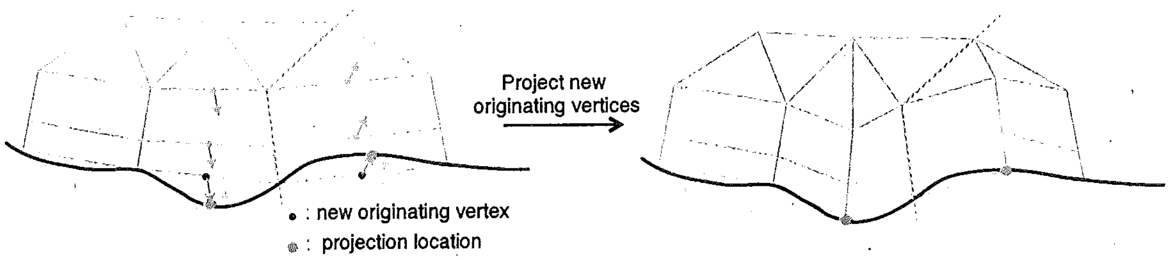
- Boundary layer edge collapse, swap and vertex movement are performed locally
- If cavity is not local, mesh migration of boundary layer and unstructured part of mesh connected to the interface is needed to localize the cavity on one part
- The migration request originates by providing growth curve vertices on part boundary, where the unstructured regions are adjacent to top-most vertex

Parallel Boundary Layer Mesh Adaptation



Vertex repositioning in the interface

- Used to move boundary layer vertices to the curved boundary
- Might not always be possible with direct repositioning
- If inverted elements are introduced, local mesh modification operations are applied

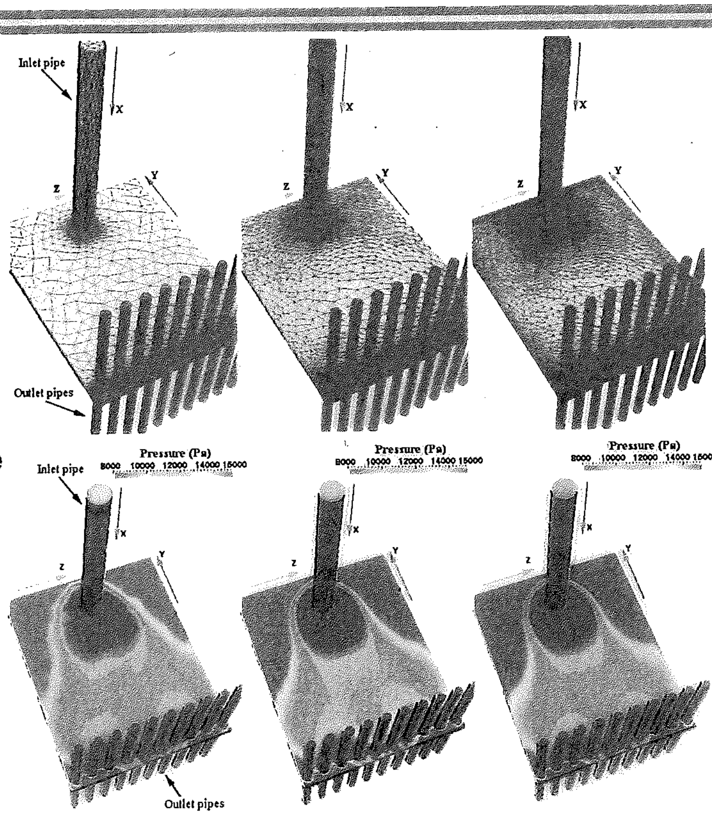


Sahni et al. 2008: Adaptive boundary layer meshing for viscous flow simulations

Application Results. Heat Transfer Manifold

Initial mesh is 3M regions

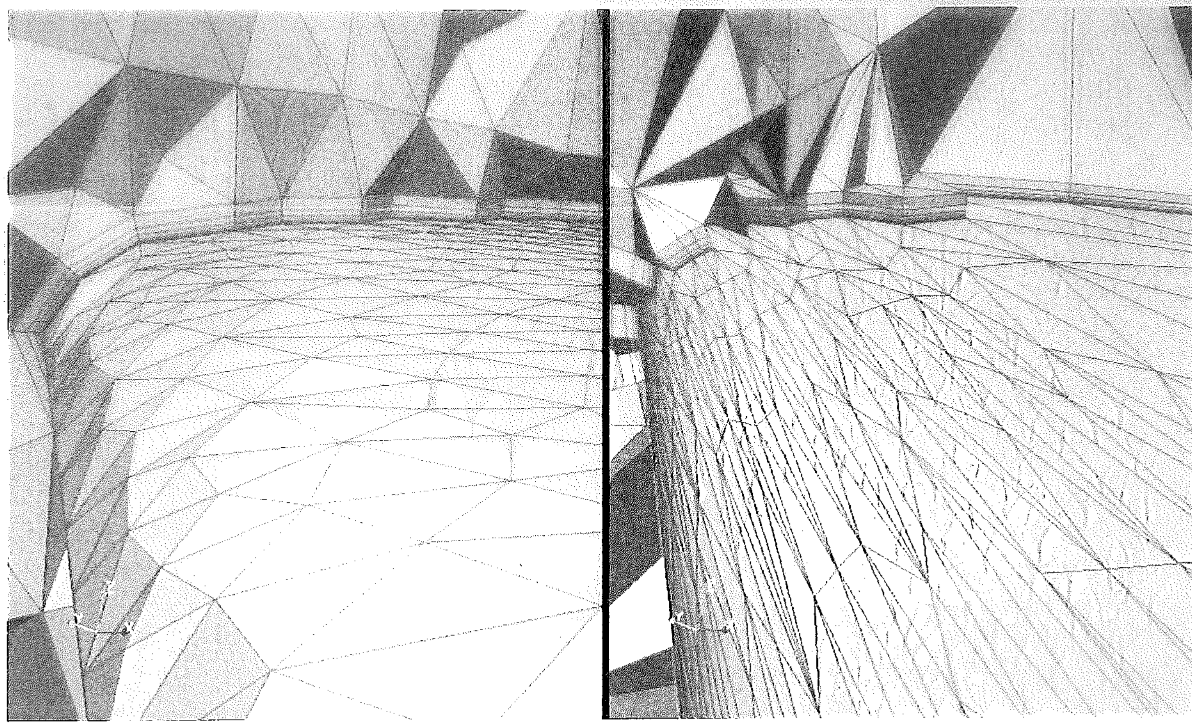
Final mesh is 81M regions



Two adaptive cycles

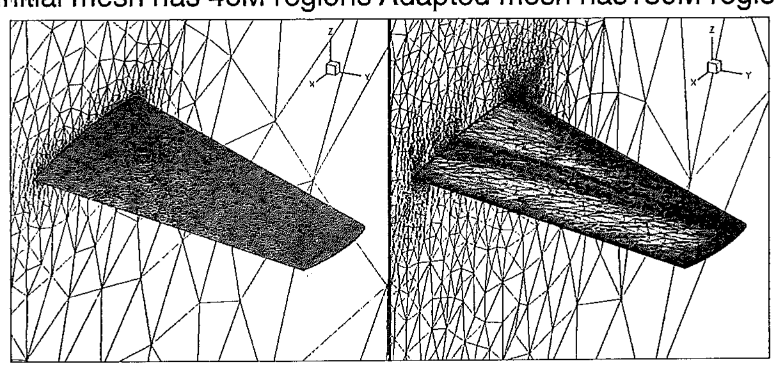
Up to 4096 parts

Example of Anisotropic Adaptation



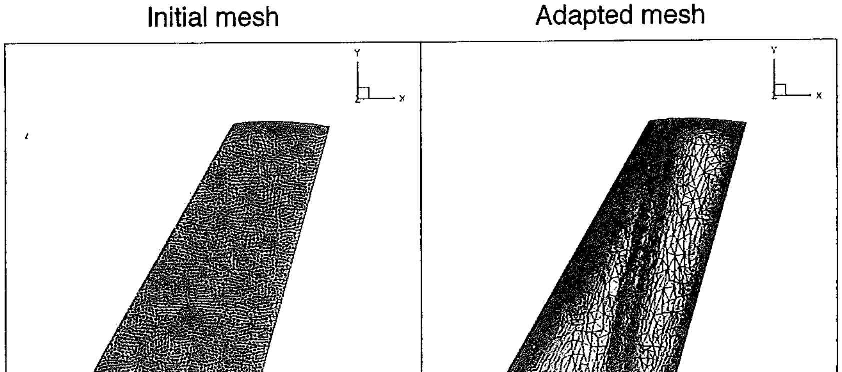
Application Results. OM6 Wing

Initial mesh has 46M regions Adapted mesh has180M regions



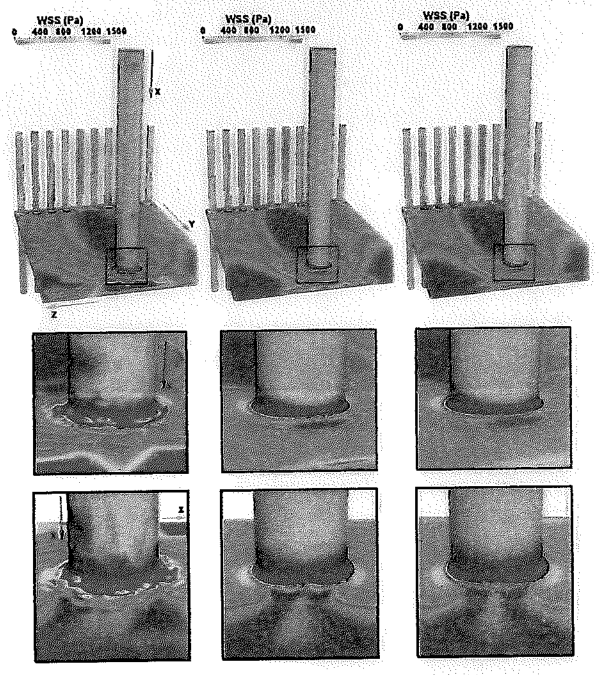
Initial mesh Adapted mesh

Up to 8192 parts



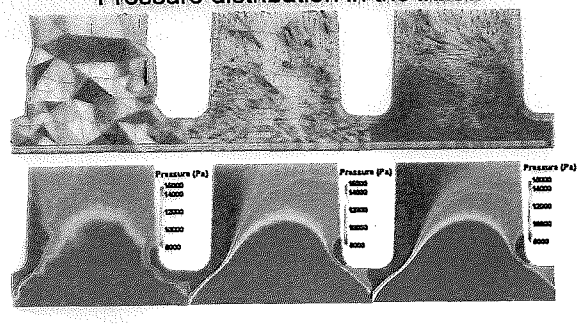
Application Results. Heat Transfer Manifold

Wall shear stress

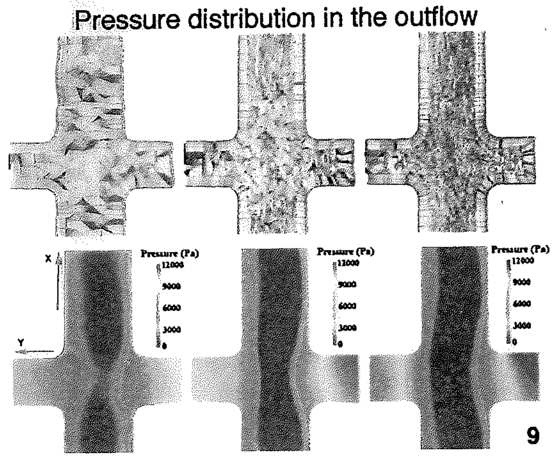


Initial mesh First adapted Second adapted

Pressure distribution in the inflow



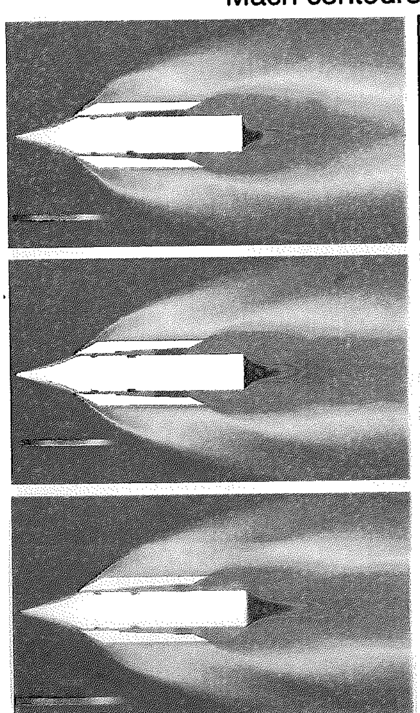
Pressure distribution in the outflow

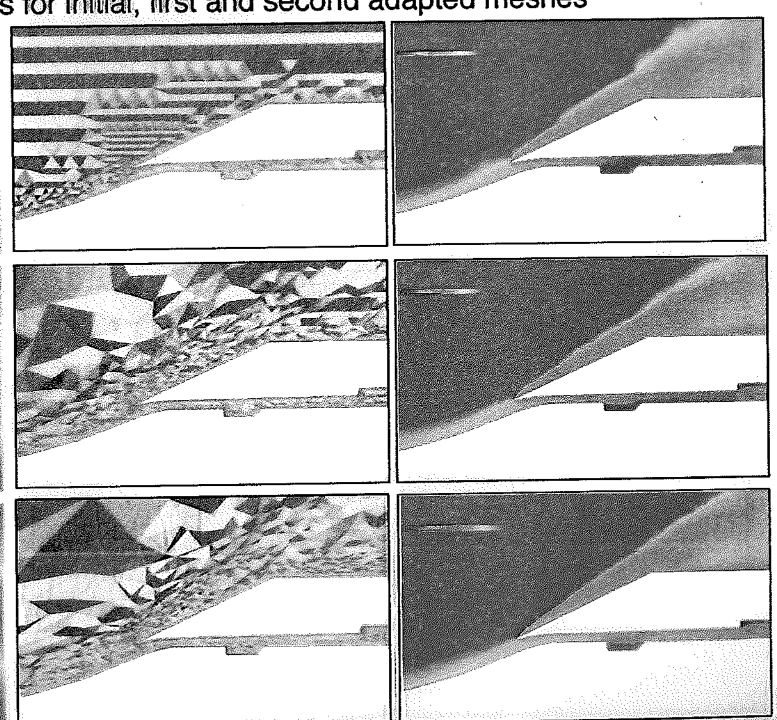


9

Application Results. Scramjet Engine

Mach contours for initial, first and second adapted meshes





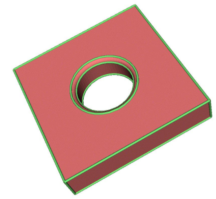
Clip-plane view

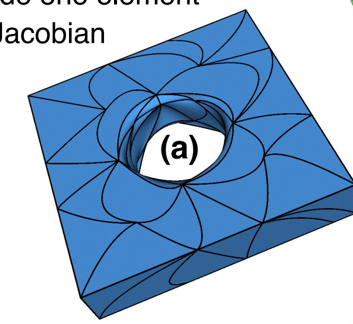
Cowl lip and entry to the combustor engine

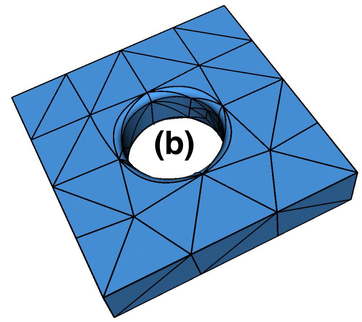
10

Quality of Curved Mesh for p-Version Method

- Quality of curved mesh is still an open issue, some possible components
  - Minimum determinant of Jacobian
  - Determinant of Jacobian variation inside one element
  - Normalized Minimum determinant of Jacobian
  - Geometric approximation error
- Two main difficulties
  - Lack of mathematical proof
  - Hard to relate quality to finite element solution accuracy
- Example
  - Compare two valid curved meshes based on the same geometric mode
  - Quadratic mesh geometry
  - Case **(a)** focused on maximizing the min. Jacobian - leads to strong interior mesh entity curving
  - Clear there are open questions



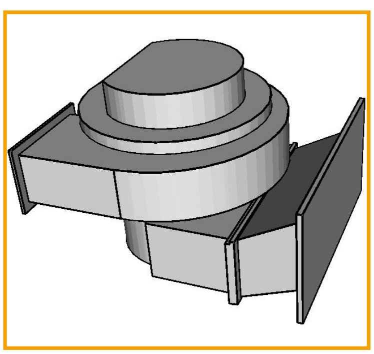




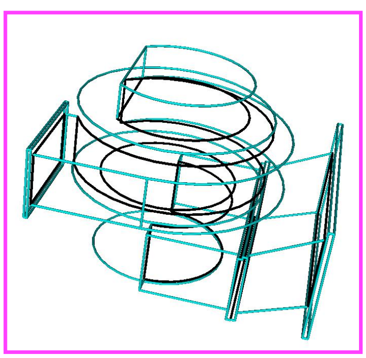
Automatic Isolation of Geometric Singular Features

- Use golden section search algorithm to determine the singular portions of a model edge having variable dihedral angle
- A model vertex is singular once one of its connect model edge is singular

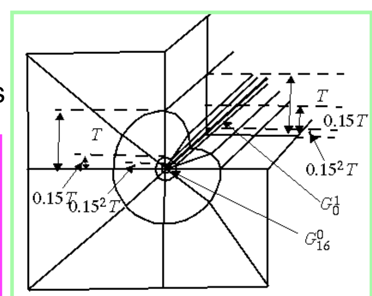
- Required p-version mesh
  - Cylindrical layer elements for singular model edge
  - Spherical layer element for singular model vertices



Geometric model



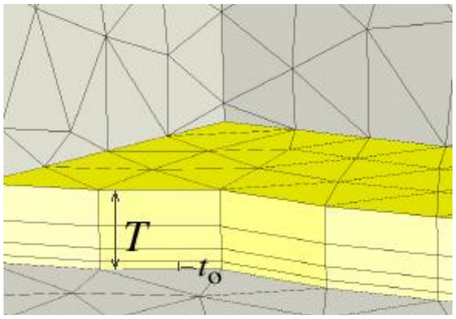
Singular model entities marked as dark

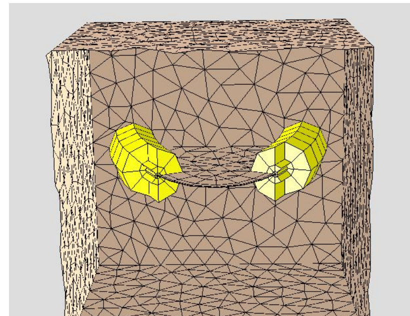


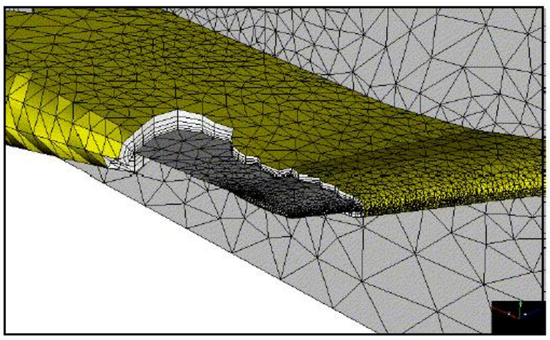
Cylindrical and Spherical layer mesh

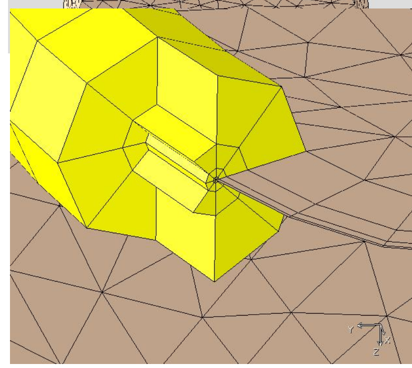
Generation of Linear Feature Isolation Mesh

- Isolated features can be model faces, edges and/or vertices
  - Create advancing-front type meshes from the surface mesh



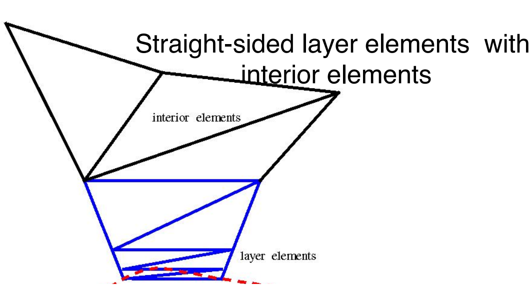


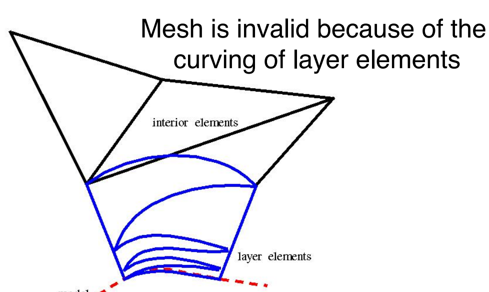


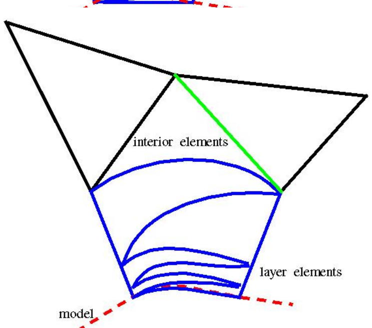


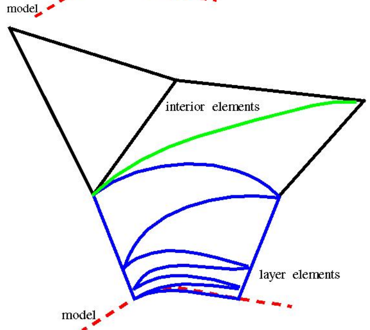
Curving Volume Mesh Around Isolated Features

- Linear element removal to fill in the remainder of the volume
- Mesh Modifications to correct volume mesh



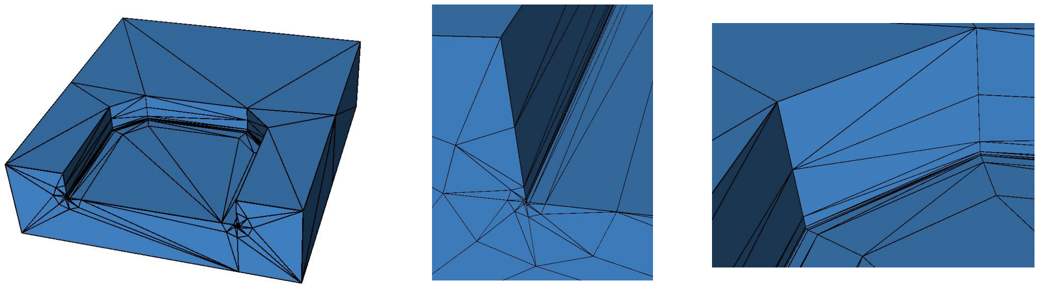




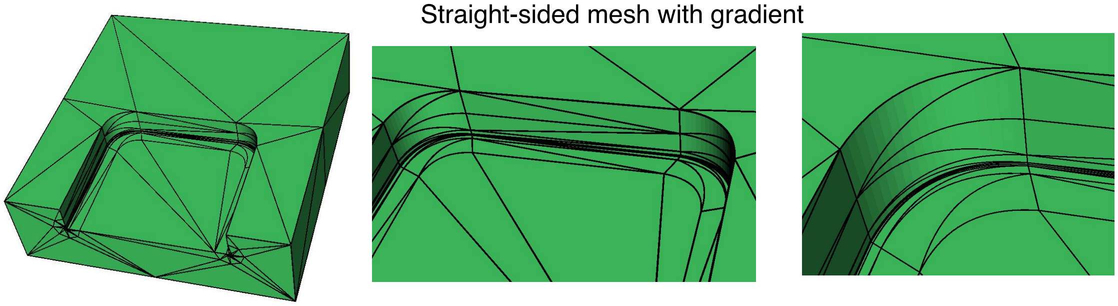


Example p-Version Mesh

- Isolation on model edges



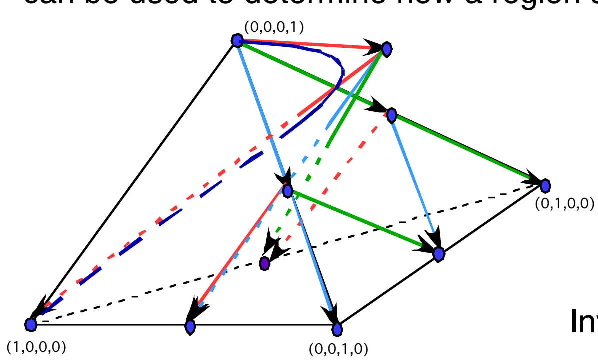
Straight-sided mesh with gradient

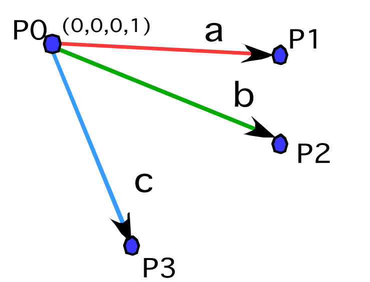


Curved mesh with gradient

Curved Operations - Recurving

- The box product terms that compose the Jacobian determinant function can be used to determine how a region should be corrected





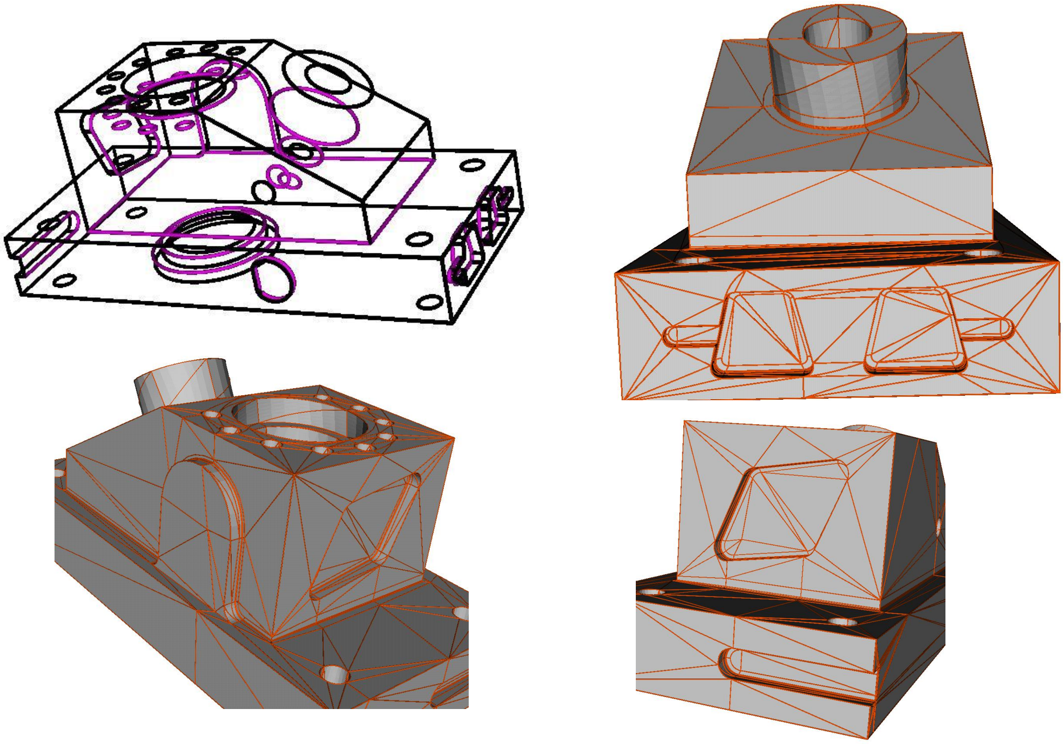
Invalid Tet caused by moving P1 - note that **a** • **(b x c)** < 0

- Jacobian Control point, J_l for a tet region of degree d is equal to:

$$J_l = \frac{1}{\binom{3(d-1)}{l}} \sum_{i+j+k=l} a_{ijk} P_i^{\zeta_1} \bullet (P_j^{\zeta_2} \times P_k^{\zeta_3})$$

$$a_{ijk} = \binom{d-1}{i}\binom{d-1}{j}\binom{d-1}{k}$$

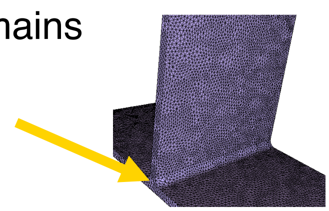
Example p-Version Mesh



p-Version Meshes in Thin Sections

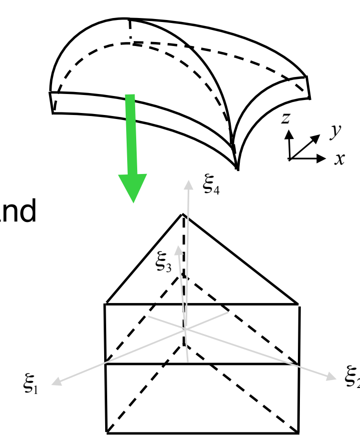
Shell elements for thin sections in 3D domains

- Historic approach is plate or shell models
- Can not deal with fillets, bosses, stiffeners etc.
- Need special finite elements for the 2D to 3D transition



Apply 3D p-version method for thin sections

- One to two layers of prismatic elements through thickness direction most effective
- Choose polynomial order differently for in-plane and thickness directions



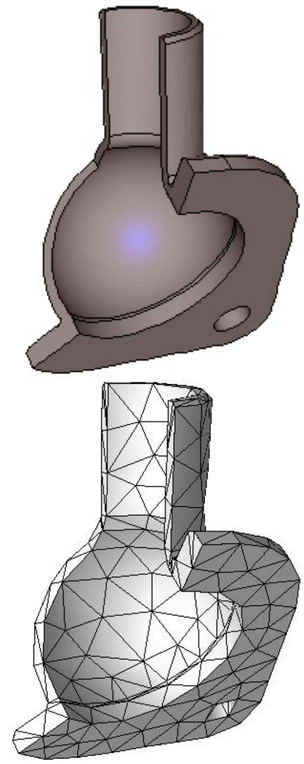
Identification of Thin Section

Geometry characteristics

- Dimension through thickness is small compared to the “in-plane” dimension
- Closely related to size and order of element in mesh

Key steps to identify thin section

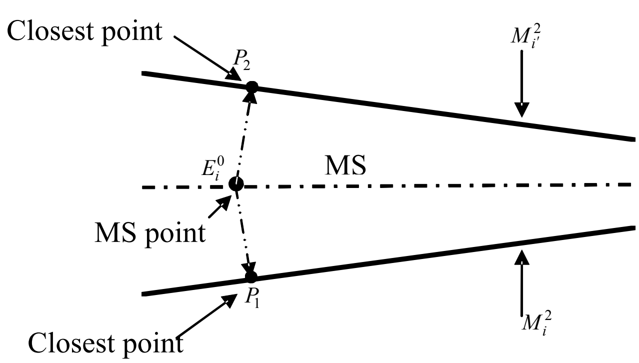
- Start from classified surface triangle
  - Same surface triangulation for volume mesh
- Find pairs of triangles on “opposite model faces” that are close to each other relative to their size
- Use classification information - combine triangles pairs to form thin section surface patches
- Properly adjust the boundary of those surface patches to define thin section



Define Thin Section Triangle Pairs

A pair of triangles M_i^2 and $M_{i'}^2$ is thin section triangle pair if a pair of closest boundary points P_1 and P_2 from a medial surface point E_i^0 such that:

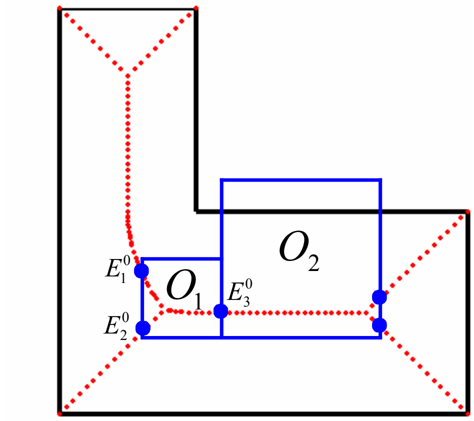
- $P_1 \subset \overline{M}_i^2$, $P_2 \subset \overline{M}_{i'}^2$
- The ratio of thickness to average size of M_i^2 and $M_{i'}^2$ is less than a default value
- The angle formed by the outward normals to M_i^2 and $M_{i'}^2$ is close to π



Compute the Medial Surface Points

Overall

- Classification information used to ignore the medial surface branches of the triangulated model due to facets
- Octree tracing algorithm is used
- Calculate intersection of octant edges and medial surface



Octree-based tracing algorithm

- Determine an octant with an edge that intersects the medial surface – can require octree refinement
- Resolve all intersections of that octant edge by proper traversal
- Continue the traversal on the other edges until all intersections resolved
- Move to neighboring octants to process their octant edge/medial surface intersections

Define Thin Section

Collect thin section triangle pair sets

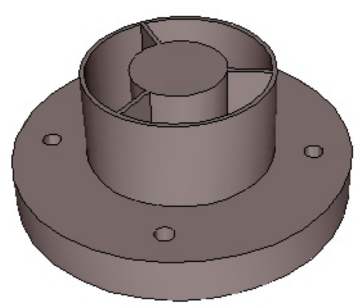
- Unorganized and incomplete thin section triangle pairs by medial points

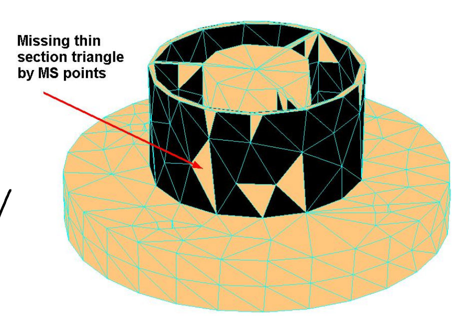
$$\left|E_i^0\right|_* = \begin{cases} 1 & thin \\ 0 & not\ thin \end{cases} \qquad [E_i^0]_* = \{M_k^2, M_{k'}^2\}$$

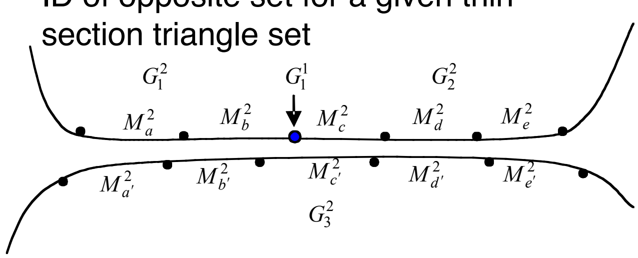
- Organize thin pairs into sets by classification

$$\hat{G}_j^2 = \left\{ M_i^2 \middle| M_i^2 \left[G_j^2 \text{ and } M_i^2 \in [E_l^0]_* \text{ and } \left|E_l^0\right|_* = 1 \right. \right\}$$

  - Each set is uniquely associated with a model face
  - Classification information also provides ID of opposite set for a given thin section triangle set



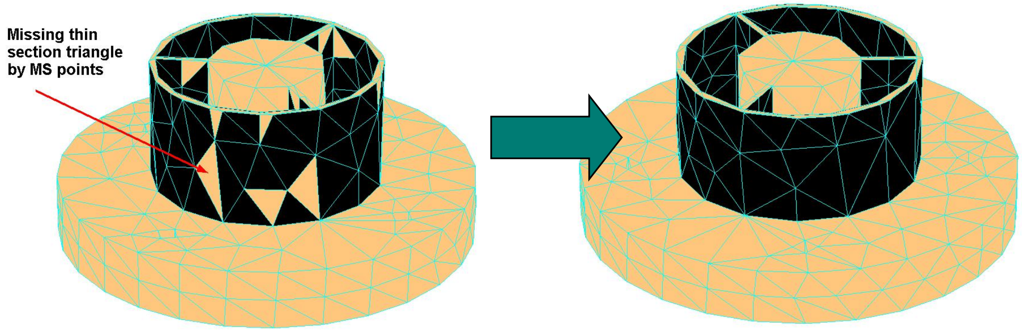




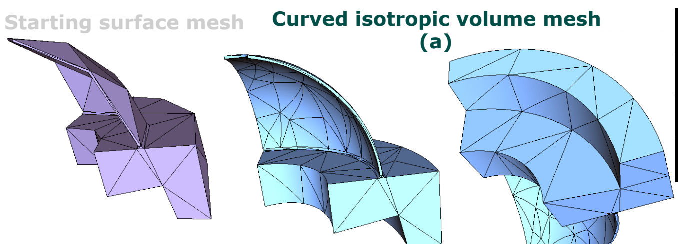
Define Thin Section

Determining the missing thin section triangles

- Use adjacencies, classifications and local thickness
- A triangle classified on a model face adjacent to a known thin section triangle has its thickness defined as the distance to the closest triangle on the opposite face
- If the thickness is small – the triangle is placed in the thin section set



p-Version Mesh Results



Mesh	Regions	Gradation
a	305	0
b	706	0.15
c	350	0.15