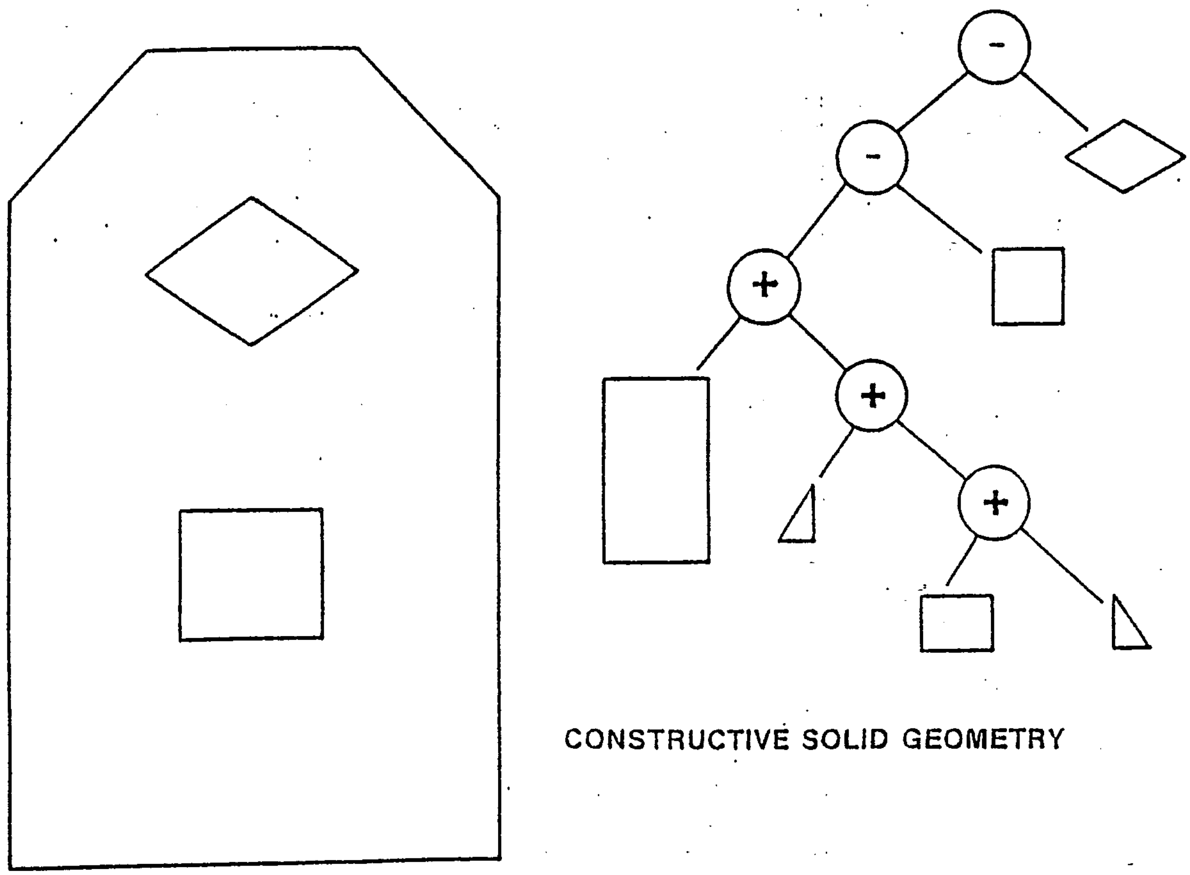
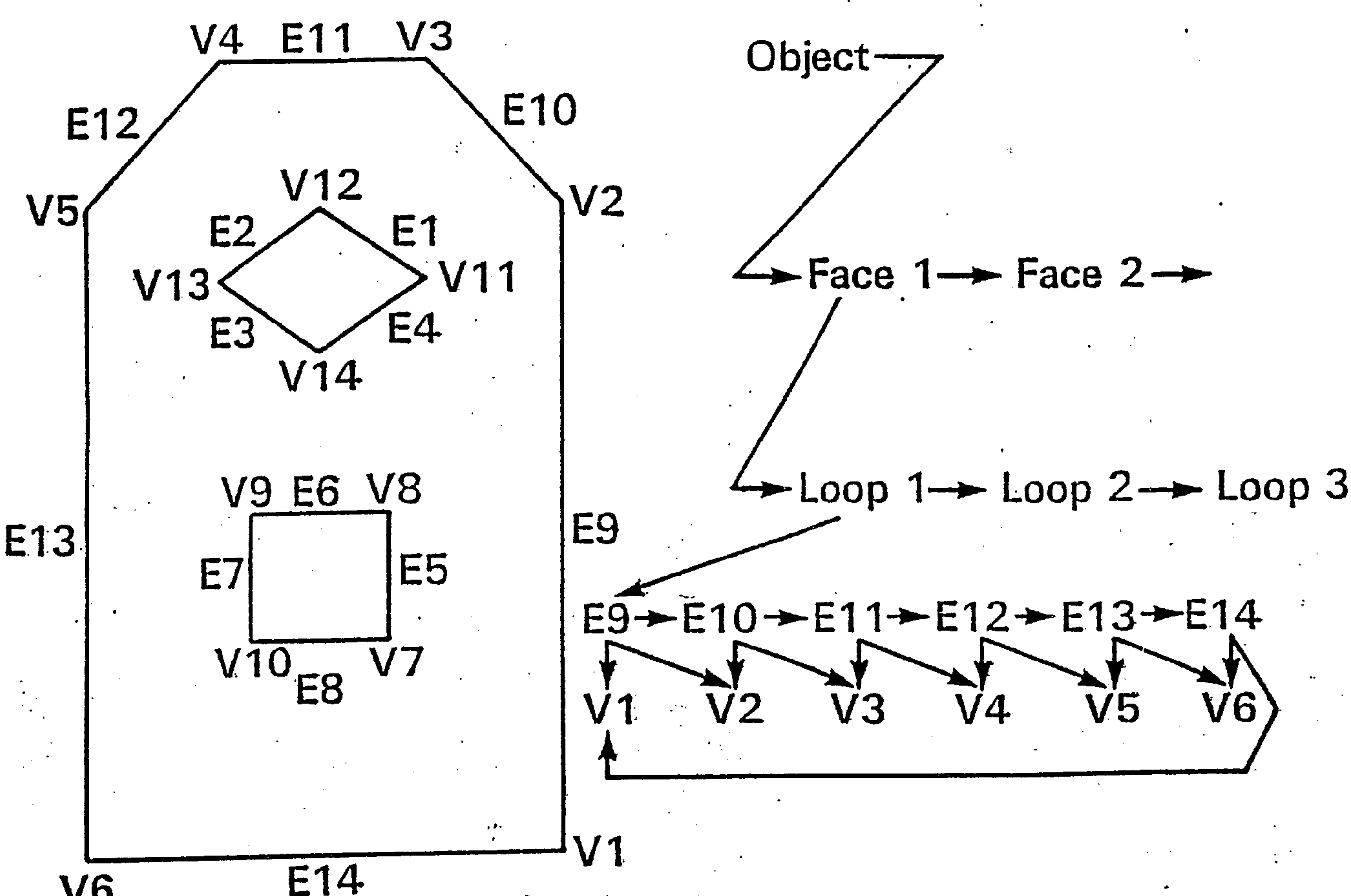
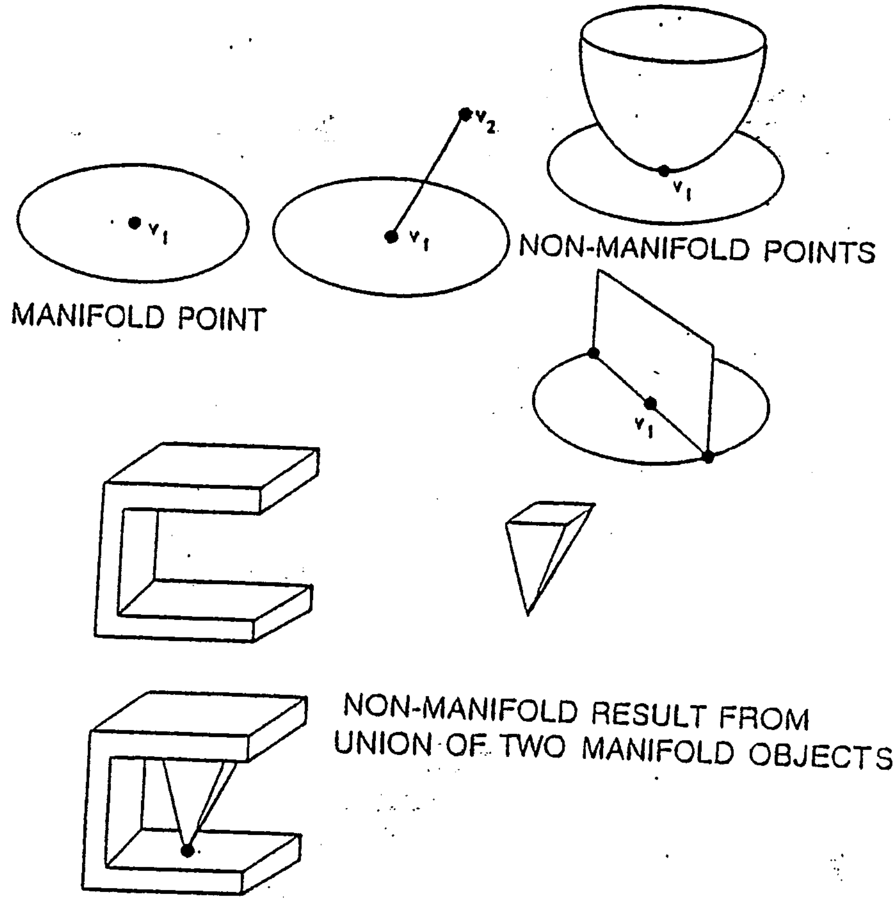


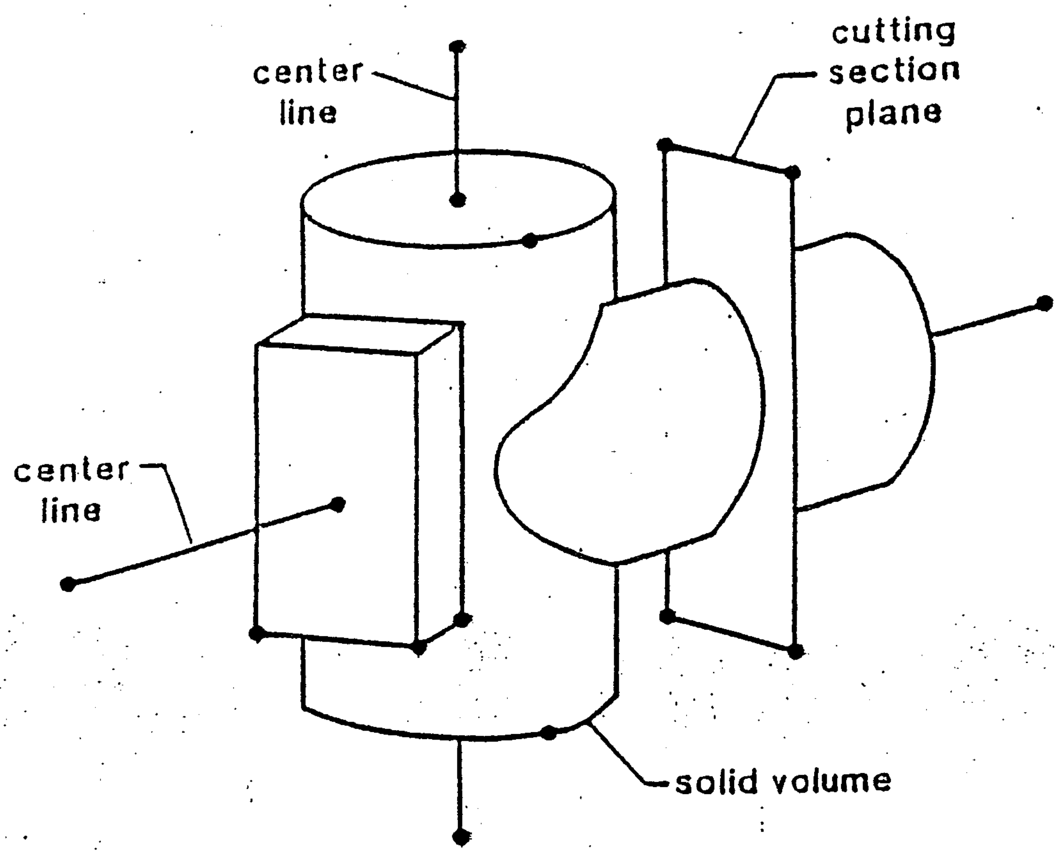
CONSTRUCTIVE SOLID GEOMETRY



BOUNDARY REPRESENTATIONS

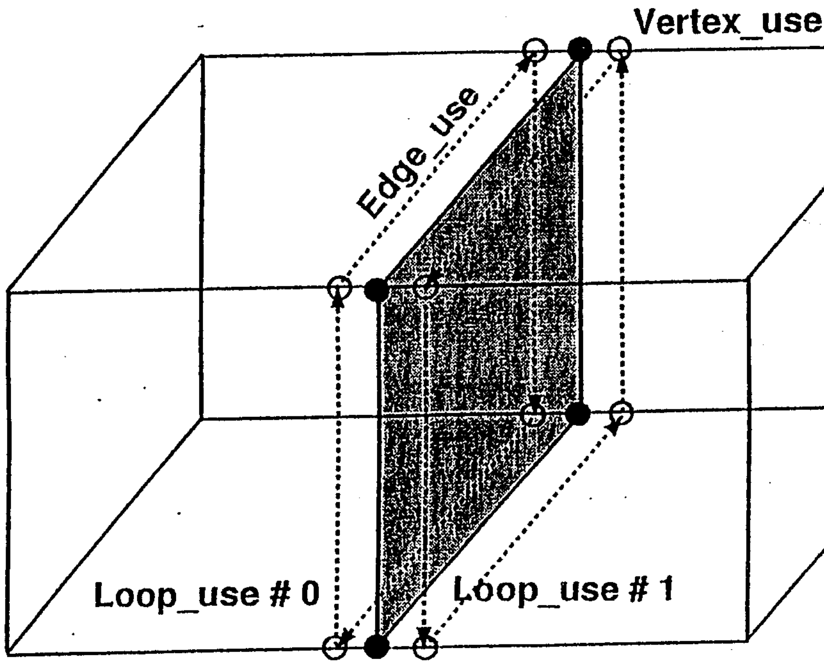


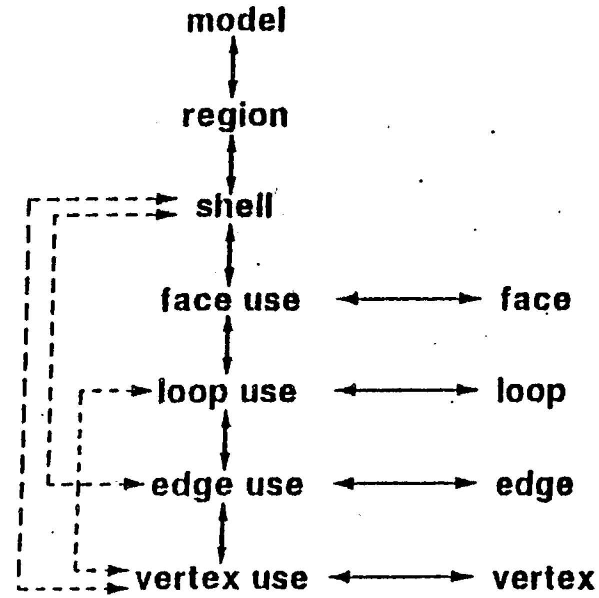
v_1
v_2
v_1
v_1
NON-MANIFOLD POINTS
MANIFOLD POINT
v_1
NON-MANIFOLD RESULT FROM
UNION OF TWO MANIFOLD OBJECTS



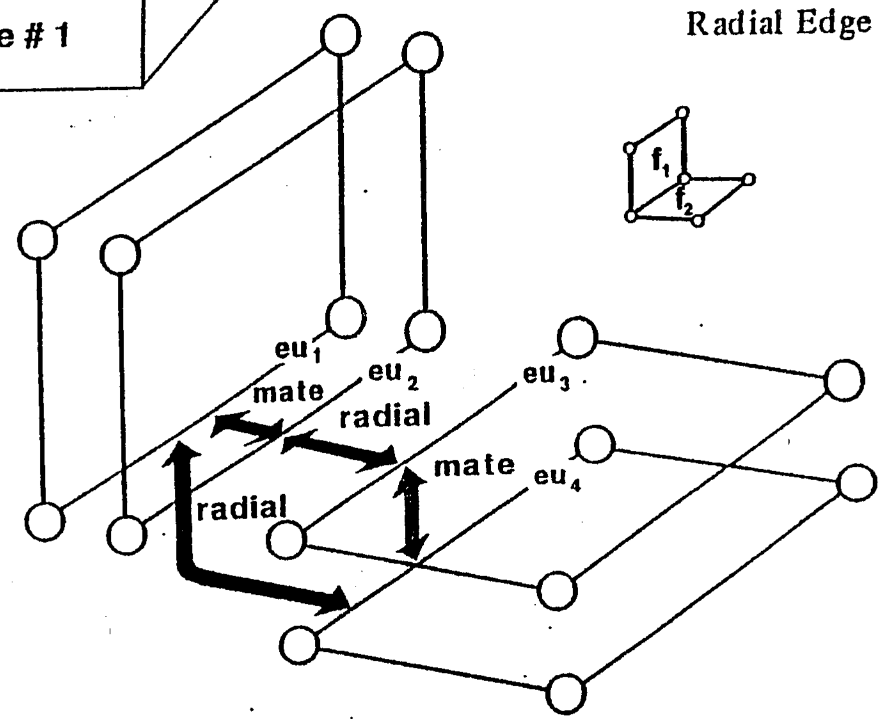
center
line
cutting
section
plane
center
line
solid volume

Non-manifold topological representation





Radial Edge structure relationships



1.1. Nomenclature

Models

Ω_V	domain associated with the model V, $V = G, M$ where G signifies the geometric model and M signifies the mesh model
$\bar{\Omega}_V$	the closure of the domain associated with the model V, $V = G, M$

Topological entities

V_i^d	the ith entity of dimension d in model V. Shorthand for $V\{V^d\}_i$
$\partial(V_i^d)$	the entities on the boundary of V_i^d
$\bar{V}_i^d$	closure of topological entity defined as $V_i^d \cup \partial(V_i^d)$
$\sqsubset$	classification symbol used to indicate the association of one or more entities from the mesh, M, with an entity in the geometric model, G

Groups

$\{V^d\}$	unordered group of topological entities of dimension d in model V
$\lfloor V^d \rfloor$	ordered group of topological entities of dimension d in model V
$[V^d]$	cyclically ordered group of topological entities of dimension d in model V
$\langle V^d \rangle$	a group where the ordering is unspecified (ordering is one of: unordered, ordered or cyclically ordered)
φ_i	ith topological entity in group φ, where φ is any one of the groups above

Adjacency operations

$\varphi\langle V^d \rangle$	the set of entities of dimension d in model V that are adjacent to, or contained in φ. φ may be a single entity, V_i^d or $\langle V^d \rangle_i$, a group of entities, $\langle V^d \rangle$ (possibly a group resulting from another adjacency operation), or a model V
$\varphi\langle V_{\pm}^d \rangle$	an adjacency relation with directional use information associated with each entity. The $\pm$ indicates the directional use of each entity. A $+$ indicates use in the same direction as the entity definition, a $-$ indicates use in the opposite direction

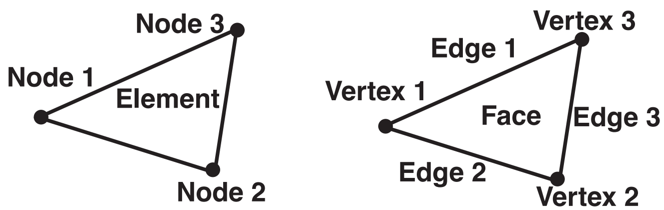
Examples

$V\{V^d\}$	all of the entities of order d in model V
$V_i^{d_i}\{V^{d_j}\}$	the unordered group of topological entities of dimension d_j that are adjacent to the entity $V_i^{d_i}$ in model V
$V_k^{d_i}\{V^{d_j}\}_i$	the ith member of the unordered group of topological entities of dimension d_j that are adjacent to the entity $V_{k'}^{d_i}$ in model V

The adjacency notation is evaluated from left to right, for example:
$V_i^3\{V^0\}\{V^3\}_j$ is found by first finding $\varphi = V_i^3\{V^0\}$ and then the jth member of $\varphi\{V^3\}$

Hierarchical Mesh Representation

- Element-Node connectivity is not sufficient for mesh generation and adaption - Can't be used to verify mesh validity
- Operations on other entities (such as faces or edges) are often more efficient and natural
- Topological hierarchy gives a general, shape-independent abstraction of a mesh



- Also a useful representation for analysis procedures
- Reference: M.W. Beall and M.S. Shephard, "A General Topology-Based Mesh Data Structure," *Int. J. Num. Meth. Engng.*, 40(9):1573-1596, 1997.

Topological Entities

Topology provides an unambiguous, shape independent, abstraction of the mesh

Each topological entity of dimension d, M_i^d, is defined by a set of topological entities of dimension $d-1$, $M_i^d\{M^{d-1}\}$, which form its boundary.

A region is a 3-d entity defined by the set of faces that bound it.

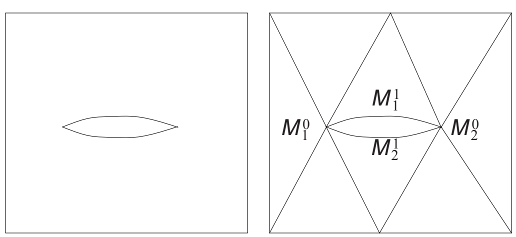
A face is a 2-d entity defined by the set of edges that bound it.

An edge is a 1-d entity defined by the two vertices that bound it.

A vertex is a 0-d entity that is the base of the hierarchy, it has no lower order entities bounding it.

Mesh Topology

1. Regions and faces have no interior holes.
2. Each entity of order d_i in a mesh, M^{d_i}, may use a particular entity of lower order, M^{d_j}, $d_j < d_i$, at most once.
3. For any entity $M_i^{d_i}$ there is a unique set of entities of order $d_i - 1$, $M_i^{d_i}\langle M^{d_i-1}\rangle$ that are on the boundary of $M_i^{d_i}$ if at least one member of $M_i^{d_i}\langle M^{d_i-1}\rangle$ is on the model entity $G_j^{d_j}$ where $d_j \geq d_i$.



(a) Geometric Model (b) Mesh

Mesh Classification

Definition: Mesh Classification Against the Geometric Domain - The unique association of a topological mesh entity to a topological geometric domain entity.

$M_i^{d_i} \sqsubset G_i^{d_i}$ denotes $M_i^{d_i}$ is classified on $G_i^{d_i}$, $(d_i \leq d_j)$

- Multiple $M_i^{d_i}$ can be classified on a $G_i^{d_i}$.

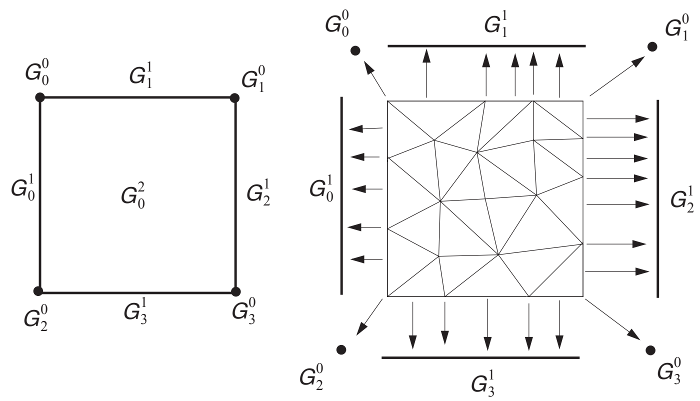
  A mesh region, M_i^3, is classified on a G_i^3

  A mesh face, M_i^2, is classified on a G_i^3, or G_i^2.

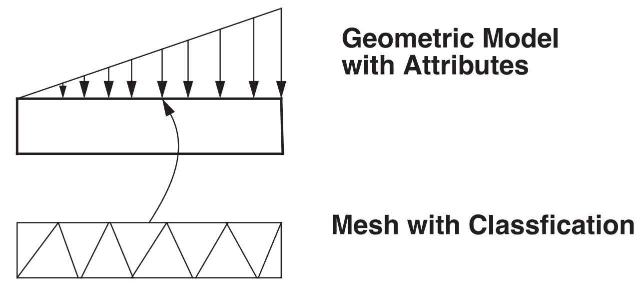
  A mesh edge, M_i^1, is classified on a G_i^3, G_i^2, or G_i^1

  A mesh vertex, M_i^0, is classified on a G_i^3, G_i^2, G_i^1, or G_i^0
- Mesh entities are always classified with respect to the lowest order object entity possible.

Mesh Classification



Classification and Attributes



- Attributes which apply to mesh are retrieved using classification information
- Attributes are never “transferred” to the mesh - always are only defined on geometric model
- When mesh is adapted, attributes do not change

Radial Edge Data Structure

MODEL

Region
↕
Shell
↕
Face use ↔ Face
↕
Loop use ↔ Loop
↕
Edge use ↔ Edge
↕
Vertex use ↔ Vertex

MESH

Region
↕
Face use ↔ Face
↕
Edge use ↔ Edge
↕
Vertex use ↔ Vertex

Application to Analysis Codes

- The hierarchical mesh representation can also be used for analysis
- Rather than having elements and nodes, degrees of freedom are directly associated with mesh entities

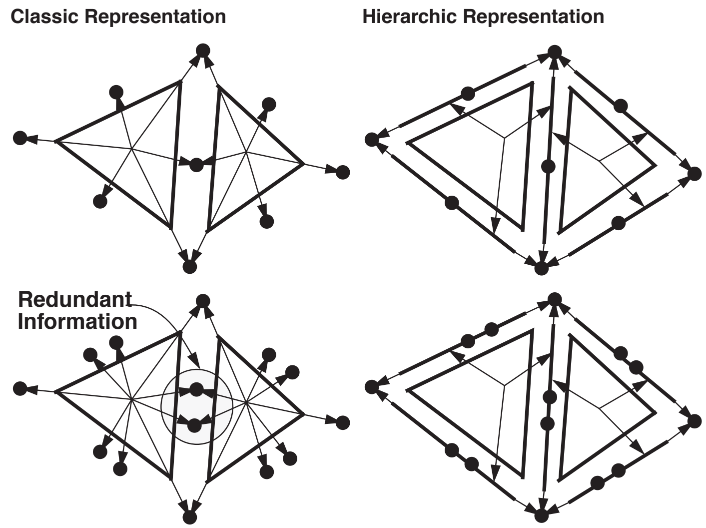
Allows same representation to be used for mesh modification and for analysis - important in adaptive environments

Reduces redundant information storage in higher order formulations

- Multiple nodes on mesh edge or face are pointed to by each element in classic representation
- Hierarchic representation only points to each once
- Very important for variable order p-meshes

Provides links to exact geometry for element integration procedures

Redundant information: Higher-orders nodes on shared edge



First-Order Adjacencies

First-order adjacencies for $M_k^{d_i}$ are the entities, M^{d_j}, $(i \neq j)$ which are either on its closure $(j < i)$ or which it is on the closure of $(j > i)$.

The complete list of these adjacencies is as follows:

Vertex adjacencies: $M_i^0\{M^1\}$, $M_i^0\{M^2\}$, $M_i^0\{M^3\}$

Edge adjacencies: $M_i^1\lfloor M^0 \rfloor$, $M_i^1[M^2]$, $M_i^1[M^3]$

Face adjacencies: $M_i^2[M^0]$, $M_i^2[M_\pm^1]$, $M_i^2\lfloor M^3 \rfloor$

Region adjacencies: $M_i^3\{M^0\}$, $M_i^3\{M^1\}$,$M_i^3\{M_\pm^2\}$

$\lfloor\ \rfloor$ **- ordered list,** $[\ ]$ **- ordered cyclic list,** $\{\ \}$ **- unordered list**

Storing all relations would take up too much space.

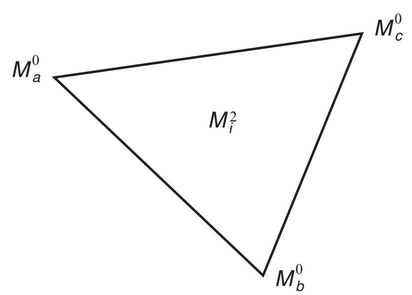
Can derive some of the above relations from the others.

e.g. can derive $M_i^3\{M^1\}$ **from** $M_i^3\{M_\pm^2\}$ **and** $M_i^2[M_\pm^1]$

Three reasonable implementations will be given that satisfy all requirements

First-Order Adjacencies

Example: $M_i^2[M^0]$ **is the circular ordered list of mesh vertices which are on the closure of the mesh face** M_i^2**.**



$$M_i^2[M^0] = [M_a^0, M_b^0, M_c^0]$$

Second-Order Adjacencies

Second-order adjacencies of $M_k^{d_i}$ are all of the entities, M^{d_j}, which share a boundary entity of a given order, d_b with the entity.

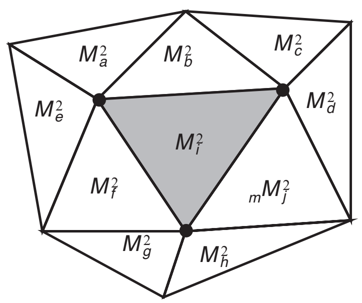
The complete set of unordered second-order adjacencies can be expressed as follows:

$$M_i^j\{M^k\}\{M^l\}, j \neq k, l \neq k$$

- **Second order adjacencies are derivable from first order adjacencies.**
- **Higher order adjacency relations can be expressed in a similar manner.**

Second-Order Adjacencies

Example: $M_i^2\{M^0\}\{M^2\}$ **, which is the set of all faces which share a vertex with** M_i^2 **(such a relationship is useful for element renumbering).**



$$M_i^2\{M^0\}\{M^2\} = \{M_a^2, M_b^2, M_c^2, M_d^2, M_e^2, M_f^2, M_g^2, M_h^2, M_j^2\}$$

·

Implementation Options

One-Level Representation

$M_i^3\{M_\pm^2\}$, $M_i^2[M_\pm^1]$, $M_i^1\lfloor M^0\rfloor$, $M_i^0\{M^1\}$, $M_i^1[M^2]$, $M_i^2\lfloor M^3\rfloor$

- **all relations are easy/fast to obtain**
- **not minimum storage**

$$M^3 \rightleftarrows M^2 \rightleftarrows M^1 \rightleftarrows M^0$$

Circular Representation

$M_i^3\{M_\pm^2\}$, $M_i^2[M_\pm^1]$, $M_i^1\lfloor M^0\rfloor$, $M_i^0\{M^3\}$

- **adjacency relations can be derived**
- **less storage than one-level representation**
- **upward adjacency relations are more costly to obtain than from one-level adjacency**

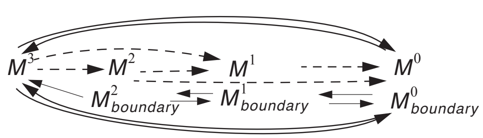
$$\begin{tikzcd} M^3 \arrow[r] & M^2 \arrow[r] & M^1 \arrow[r] & M^0 \arrow[lll, bend right] \end{tikzcd}$$

Reduced Interior Representation

Only $M_i^3\lfloor M^0\rfloor$, $M_i^0\{M^3\}$ **on interior**

$M_i^2[M_\pm^1]$, $M_i^1\lfloor M^0\rfloor$, $M_i^0\{M^1\}$, $M_i^1[M^2]$, $M_i^2\lfloor M^3\rfloor$, $M_i^0\{M^3\}$ **on boundary**

- **interior faces and edges not explicitly represented**
- **ordered region-vertex relation implies interior entities**
- **orientation of interior entities determined by a vertex numbering scheme**
- **less storage than either circular or one-level representations**
- **inefficient for procedures that modify mesh**



Performance Comparison

Operation count to retrieve adjacency relation - tetrahedral mesh

	One-level	Circular	Reduced interior	Classic
$M_i^0\{M^1\}$	1	304	198	n.a.
$M_i^0\{M^2\}$	70	264	219	n.a.
$M_i^0\{M^3\}$	140	1	1	n.a.
$M_i^1\lfloor M^0\rfloor$	1	1	1	1
$M_i^1[M^2]$	1	570	373	n.a.
$M_i^1[M^3]$	10	538	230	n.a.
$M_i^2[M^0]$	3	3	1	1
$M_i^2[M_\pm^1]$	1	1	3	3
$M_i^2\lfloor M^3\rfloor$	1	299	293	n.a.
$M_i^3\{M^0\}$	6	6	1	1
$M_i^3\{M^1\}$	9	9	6	6
$M_i^3\{M_\pm^2\}$	1	1	4	4

n.a. - cannot be obtained without global search

Performance Comparison

Operation count to retrieve adjacency relation - hexahedral mesh

	One-level	Circular	Reduced interior	Classic
$M_i^0\{M^1\}$	1	228	86	n.a.
$M_i^0\{M^2\}$	24	192	116	n.a.
$M_i^0\{M^3\}$	48	1	1	n.a.
$M_i^1\lfloor M^0\rfloor$	1	1	1	1
$M_i^1[M^2]$	1	296	212	n.a.
$M_i^1[M^3]$	8	304	112	n.a.
$M_i^2[M^0]$	4	4	1	1
$M_i^2[M_{\pm}^1]$	1	1	4	4
$M_i^2\lfloor M^3\rfloor$	1	148	176	n.a.
$M_i^3\{M^0\}$	16	16	1	1
$M_i^3\{M^1\}$	20	20	12	12
$M_i^3\{M_{\pm}^2\}$	1	1	8	8

n.a. - cannot be obtained without global search

Size Comparison

Comparison to published adaptive data structures shows hierarchic representation is approximately the same size (and is more general)

For analysis purposes comparison to classic mesh data structure is of interest

Not fair comparison since, classic mesh data structure:

- **is not suited to needs of adaptivity**
  - **no classification information**
  - **insufficient representation of mesh to verify that mesh correctly represents the geometric model**
- **is not well suited for variable p-meshes**
- **needs auxiliary data structures for operations such as node renumbering - hidden cost that can be huge**

Size Comparison

Tetrahedral Meshes

Element Order	Classic	One-Level	% of Classic	Circular	% of Classic	Reduced Interior	% of Classic
Linear	$7N_M^3$ ($9.5N_M^3$)	$35N_M^3$	500% (368%)	$26N_M^3$	371% (274%)	$13N_M^3$	186% (137%)
Quadratic	$20N_M^3$ ($57N_M^3$)	$43N_M^3$	215% (75%)	$34N_M^3$	170% (60%)	$22N_M^3$	110% (39%)
Cubic	$33N_M^3$ ($97N_M^3$)	$52N_M^3$	158% (54%)	$43N_M^3$	130% (44%)	$31N_M^3$	94% (32%)

(parenthesis indicate size with data structures for nodal renumbering)

Size Comparison

Hexahedral Meshes

Element Order	Classic	One-Level	% of Classic	Circular	% of Classic	Reduced Interior	% of Classic
Linear	$17N_M^3$ ($43N_M^3$)	$71N_M^3$	418% (165%)	$55N_M^3$	324% (128%)	$31N_M^3$	182% (72%)
Quadratic	$50N_M^3$ ($280N_M^3$)	$92N_M^3$	184% (33%)	$76N_M^3$	152% (27%)	$52N_M^3$	104% (19%)
Cubic	$83N_M^3$ ($454N_M^3$)	$113N_M^3$	136% (25%)	$91N_M^3$	110% (20%)	$71N_M^3$	86% (16%)

(parenthesis indicate size with data structures for nodal renumbering)

Mesh Information Cost

Total mesh storage/number of nodes (words/node)

Element Order	Classic	One-Level	Circular	Reduced Interior
Tetrahedral Mesh				
Linear	40 (56)	201	153	76
Quadratic	15 (41)	31	25	16
Cubic	13 (37)	20	17	12
Hexahedral Mesh				
Linear	17 (43)	71	55	31
Quadratic	13 (70)	23	19	13
Cubic	12 (65)	16	13	10

Solution Process Information Cost

Element Order	Solution	Element Matrices	Global Stiffness
Tetrahedral Mesh			
Linear	$2n$	$376n^2$	$21n^2$
Quadratic	$2n$	$292n^2$	$41n^2$
Cubic	$2n$	$398n^2$	$64n^2$
Hexahedral Mesh			
Linear	$2n$	$256n^2$	$39n^2$
Quadratic	$2n$	$400n^2$	$86n^2$
Cubic	$2n$	$585n^2$	$132n^2$

n = #d.o.f per node

Solution: values of degrees of freedom

Element Matrices: unassembled element matrices

Global Stiffness: assembled, compressed row storage

Is Mesh Storage Significant?

Example

- **3-d elasticity problem, quadratic tetrahedral elements, iterative solver using compressed row storage for global stiffness.**
  storage for solution process 375 words/node
  (6 words/node for solution, 369 words/node for global matrix)

  Classic mesh data structure adds 15 words/node, total = 390 words/node
  Largest hierarchic data structure (one-level representation) adds 31 words/node, total = 406 words/node
  4% difference in total storage
  Same problem with hexahedral elements results in 1% difference

Extra storage for hierarchic data structure does not seem significant

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

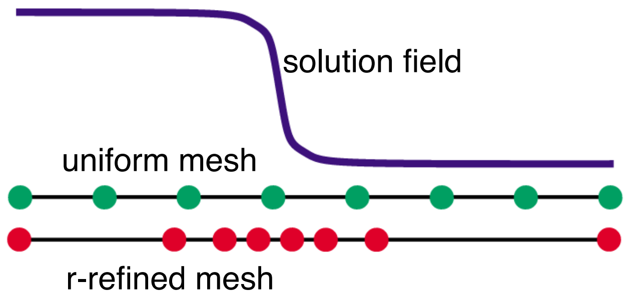
Correction indication
- Include positions of vertices in functional - too expensive
- Add nodal velocities as unknowns with penalty term to maintain mesh validity

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

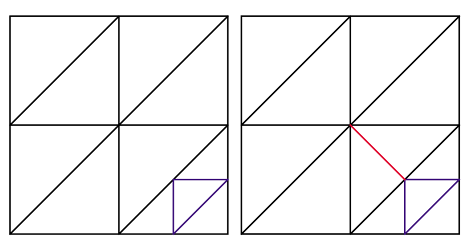


Nested Refinement

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

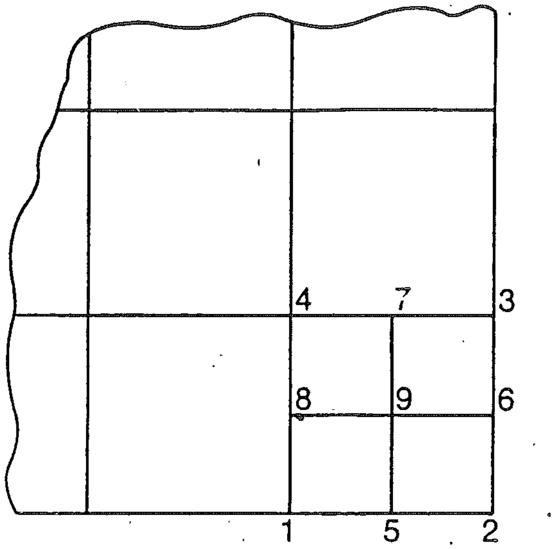
Advantages
- Straight forward for non-conforming case
- *a-priori* control of element shapes
- Allows effective solution transfer processes
- Can obtain level of accuracy desired



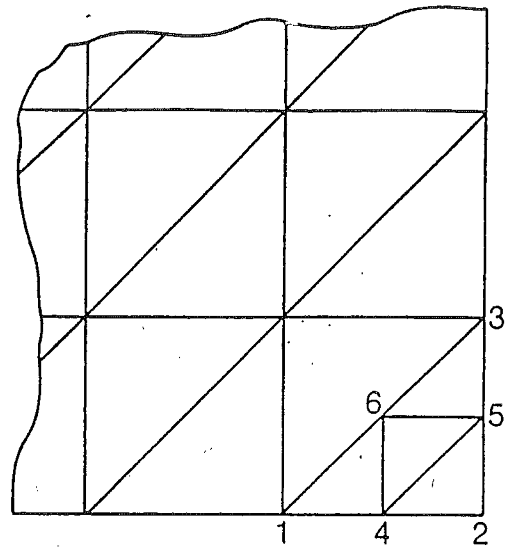
bisection　　bisection with temporary split

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

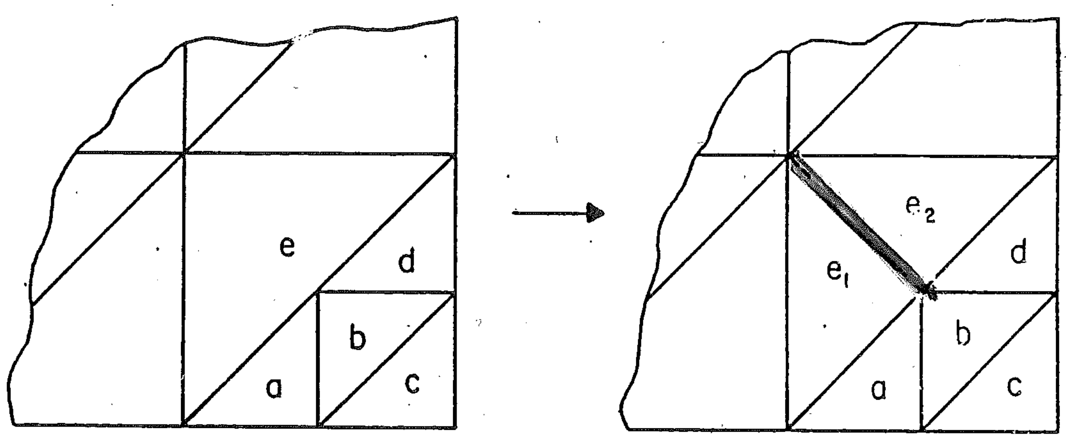
h-REFINEMENT



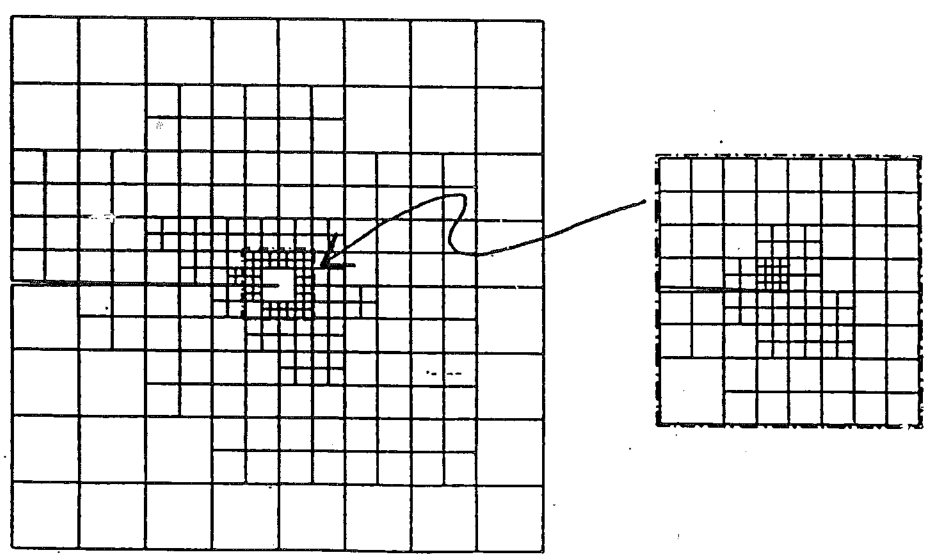
NODES 7 AND 8 NEED CONSTRAINTS



NODE 6 NEEDS CONSTRAINT

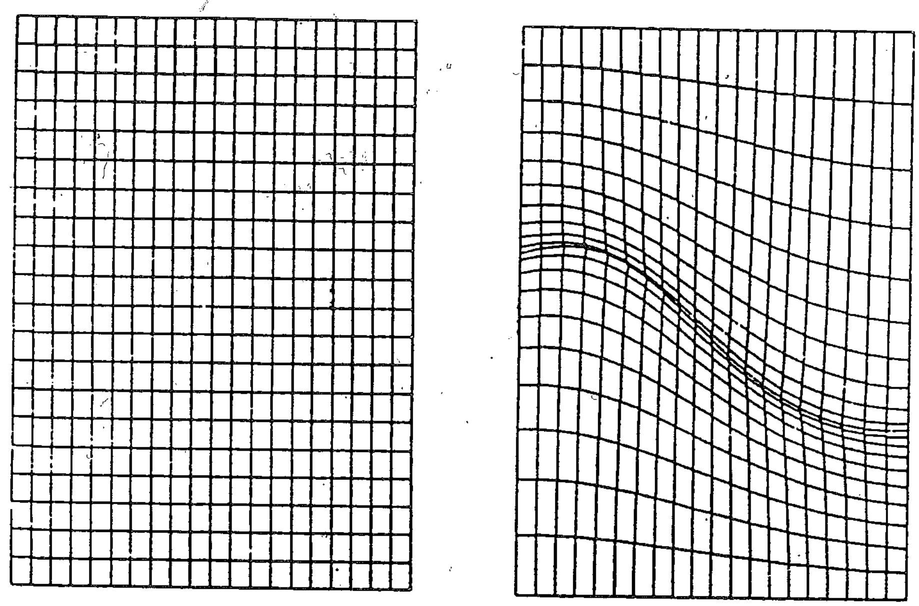


AVOIDING CONSTRAINTS WITH TEMPORARY SPLIT

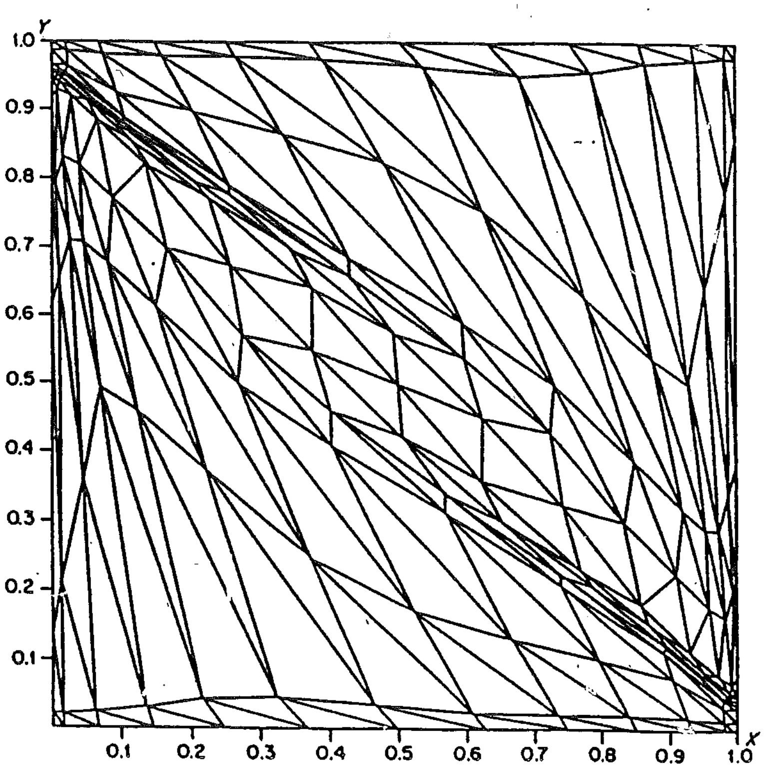


TYPICAL h-REFINEMENT EXAMPLE

r-REFINEMENT



MESH ALIGNING WITH SOLUTION FEATURE



LIMIT OF r-REFINEMENT

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

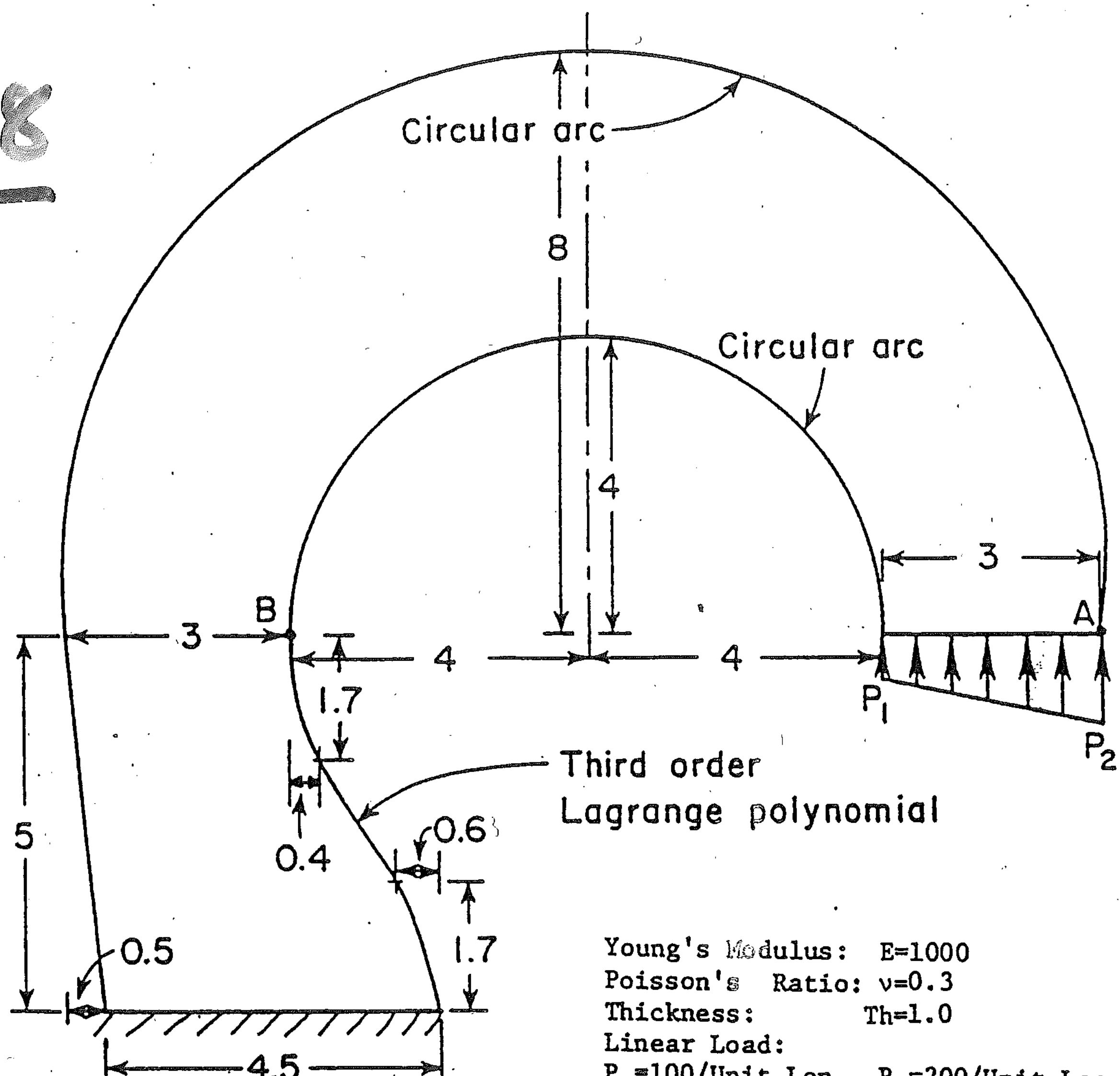
- Supports general changes in mesh size including construction of anisotropic meshes
- Can deal with any level of geometric domain complexity
- Can obtain level of accuracy desired
- Solution transfer can be applied incrementally - may have more control

Disadvantages

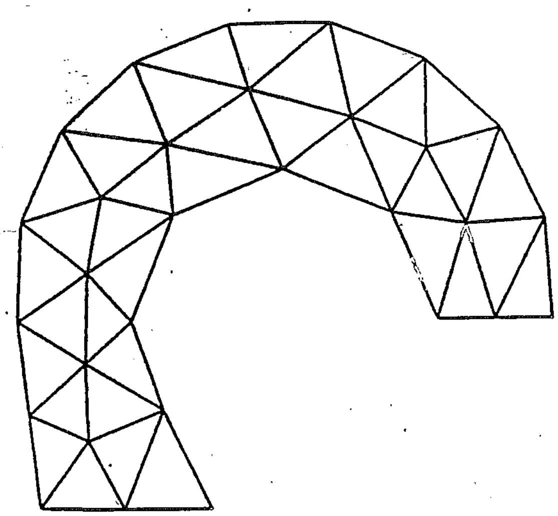
- Nearly as complex as complete mesh generation

1928

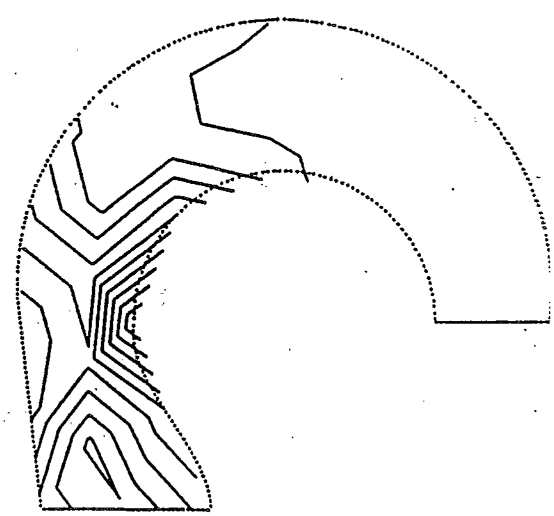
REMESHING EXAMPLE



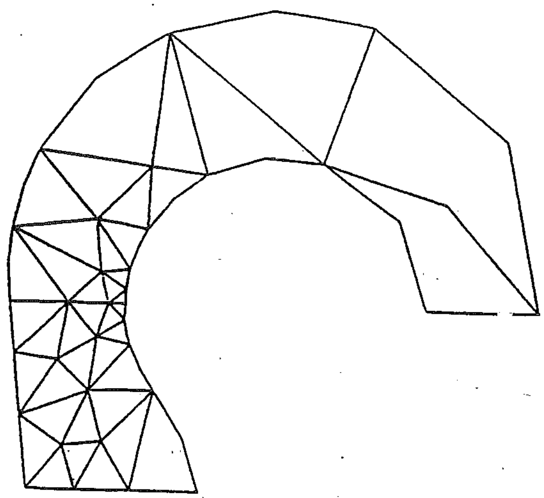
Young's Modulus: E=1000
Poisson's Ratio: ν=0.3
Thickness: Th=1.0
Linear Load:
P_1=100/Unit Len., P_2=200/Unit Len.



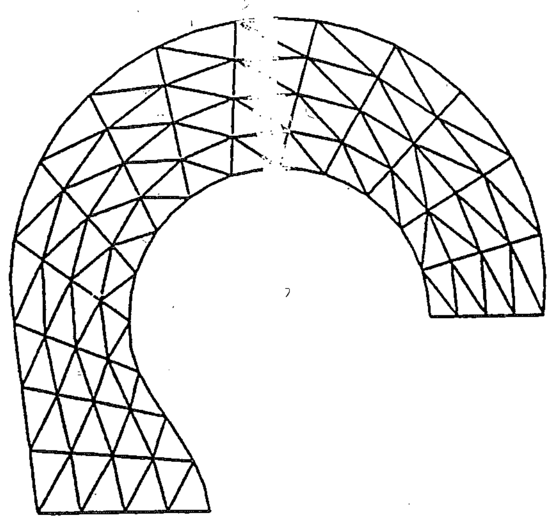
a) Initial Mesh



b) SED Contours



c) Synthesized Mesh



d) Uniform Mesh

Mesh	Number of Unknowns	Half-Bandwidth	Total Strain Energy ($x10^4$)	Maximum Vert. Displacement	Maximum Stress
Initial-CST	54	10	2.455	124.28	1986.1
Synthesized-T-6	192	34	4.395	221.88	4188.4
Uniform-T-6	504	42	4.469	225.55	3980.9

Results for Hook Example

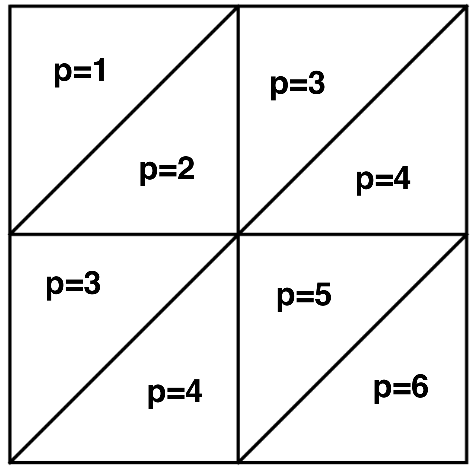
Altering the Order on a Fixed Mesh

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



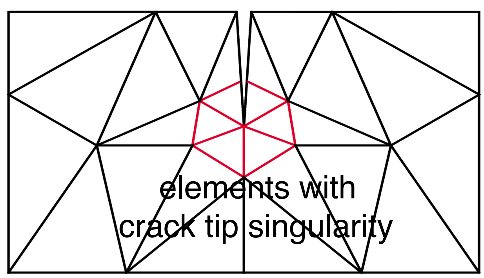
Advantages

- 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

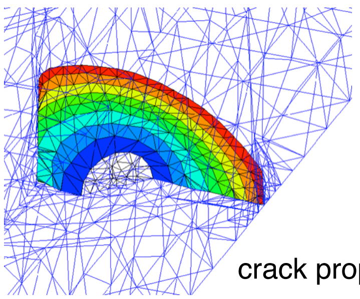
Disadvantages

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

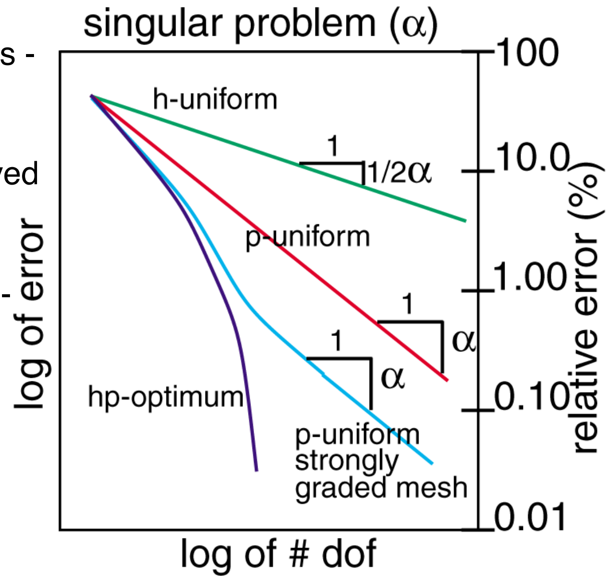
Good option in combination with others when appropriate features are present

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



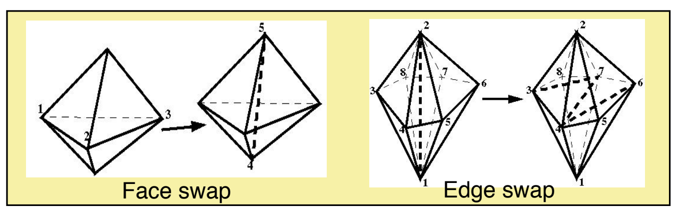
crack propagation



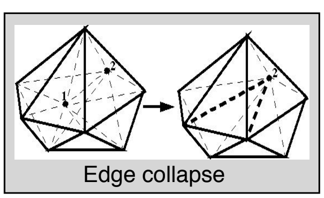
Simplex Element Mesh Modification Operators

Single step operators

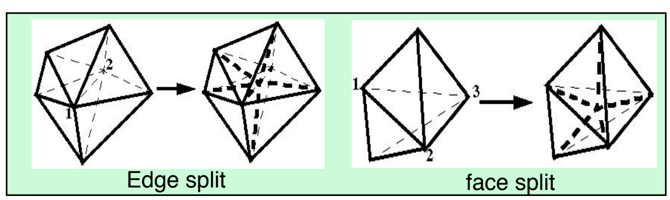
- swap
- collapse
- split
- Vertex motion



Face swap　Edge swap



Edge collapse

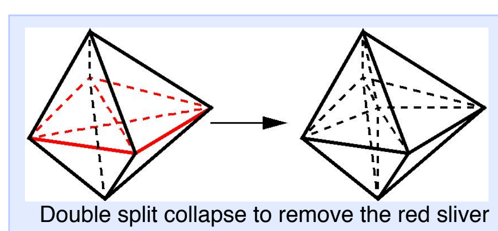


Edge split　face split

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

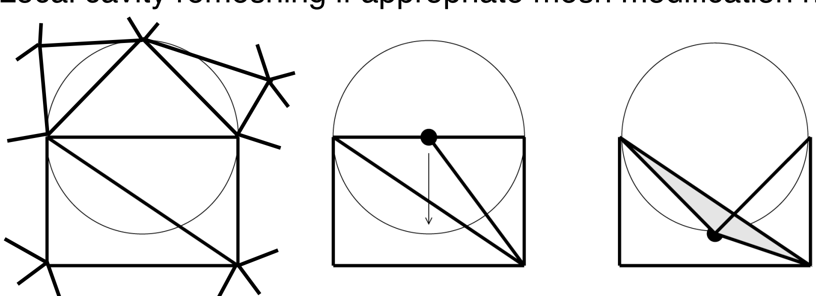
Must improve mesh geometric approximation as mesh is refined

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

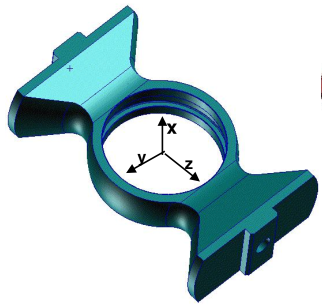
- Simple mesh motion when all connected elements remain valid and of acceptable shape
- Local mesh modification to eliminate the invalid elements created when moving the vertex
- Local cavity remeshing if appropriate mesh modification not satisfactory

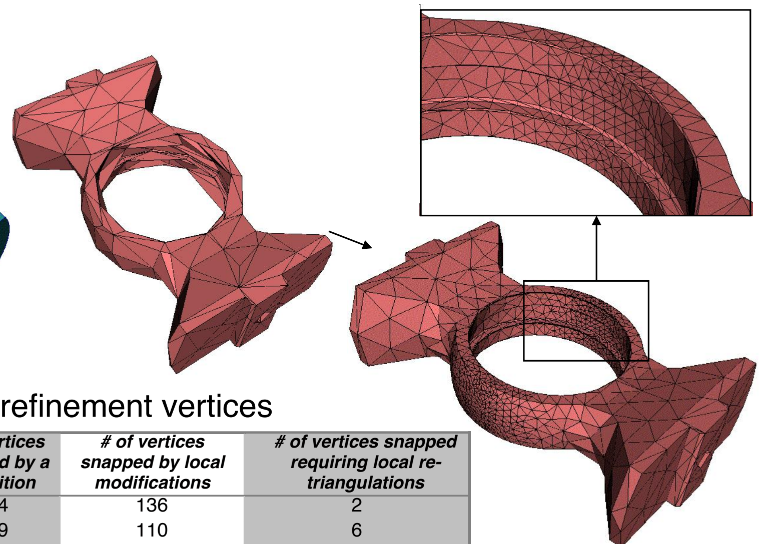


a) b) c)

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

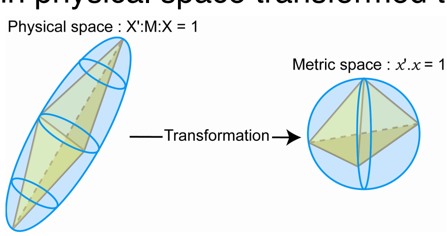
- Transformation matrix field T(x,y,z)

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

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

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

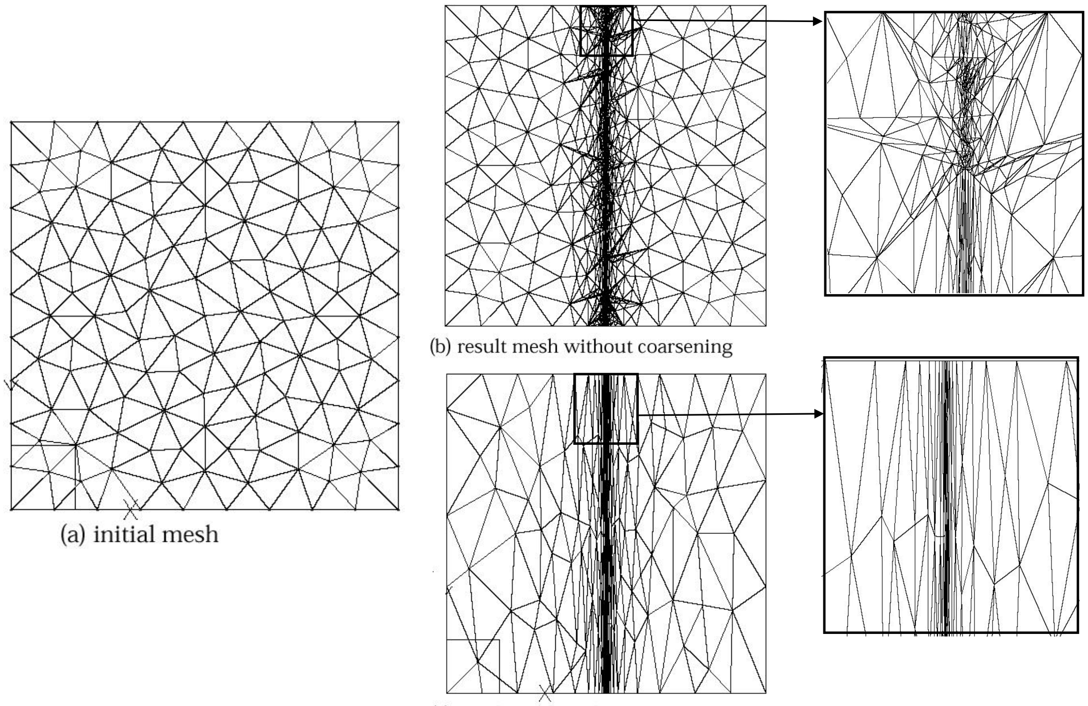
- Ellipsoidal in physical space transformed to normalized sphere



- 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

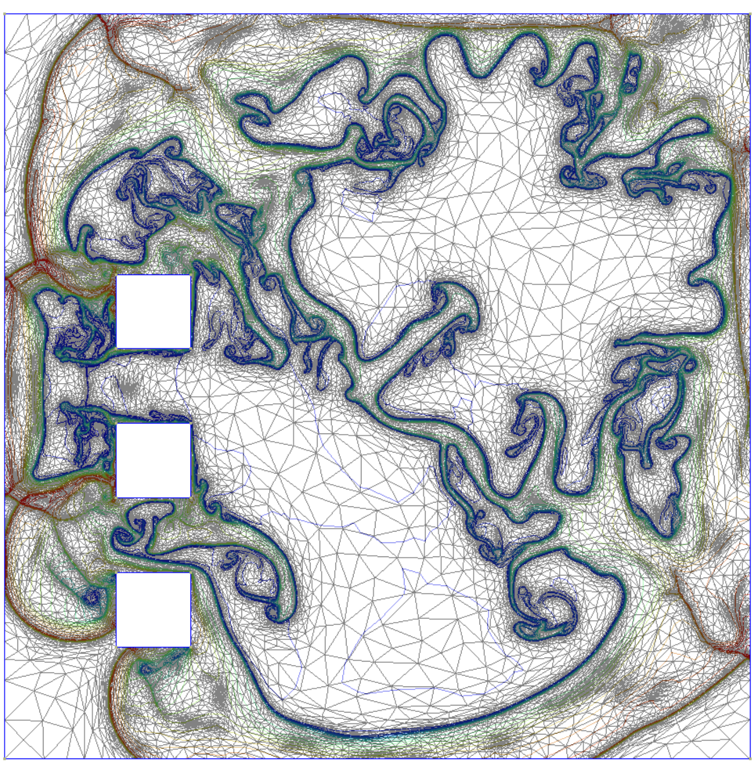


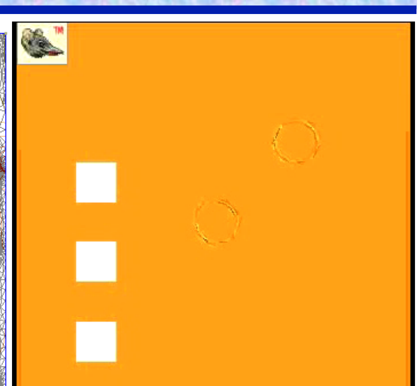
(a) initial mesh

(b) result mesh without coarsening

(c) result mesh with coarsening

Colliding Explosions - 250 refinement steps

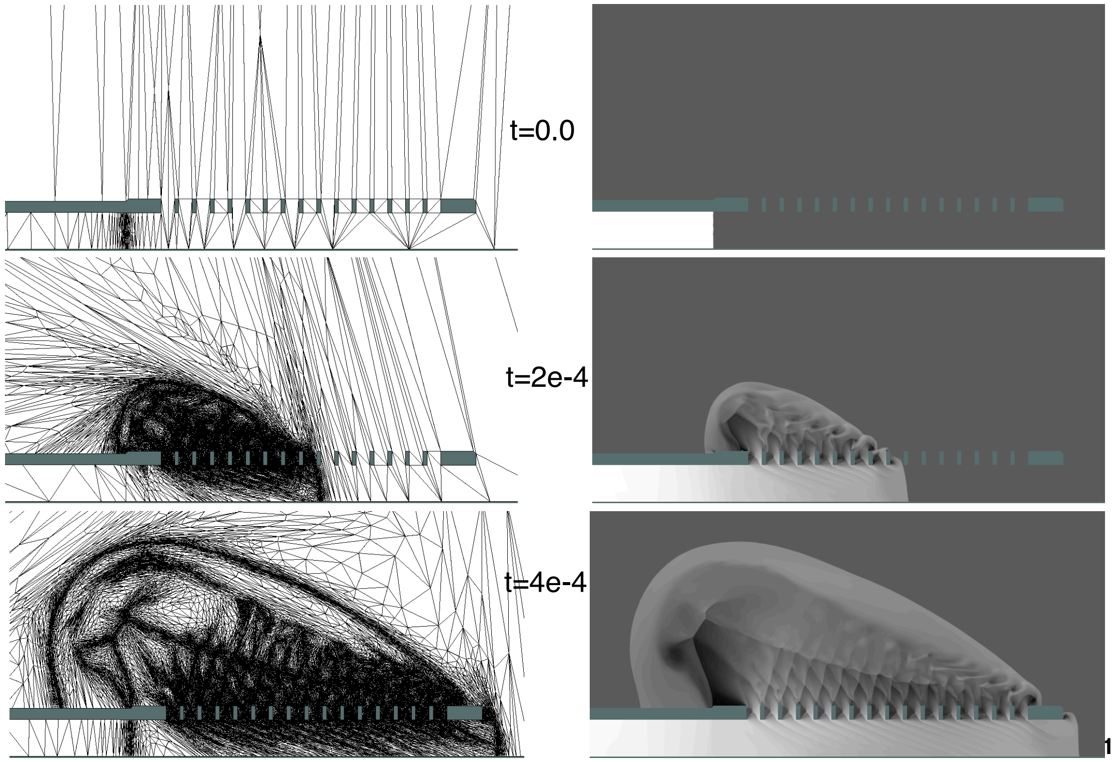




Evolution of density contours

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

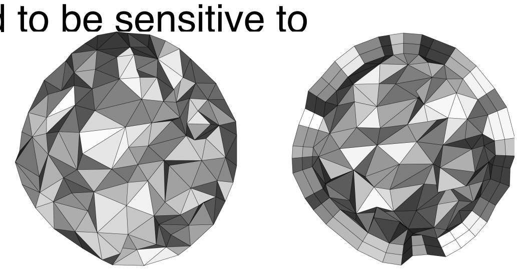
Muzzle Blast



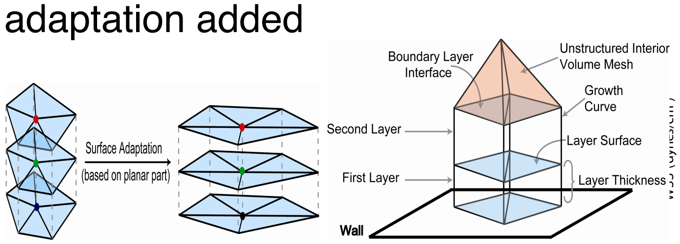
Maintaining “Structure” for Derivative Recovery

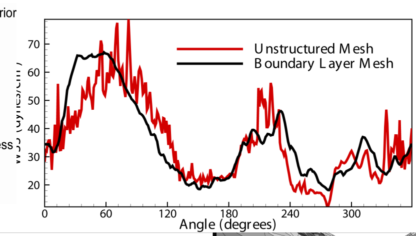
Post-processing procedure for recovering conservative wall shear stress has been observed to be sensitive to near wall mesh “structure”.

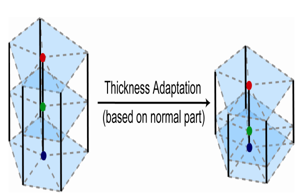
- Coarse example of arterial cross section

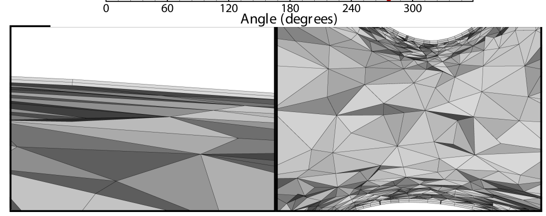


Semi-structured meshes mesh adaptation added



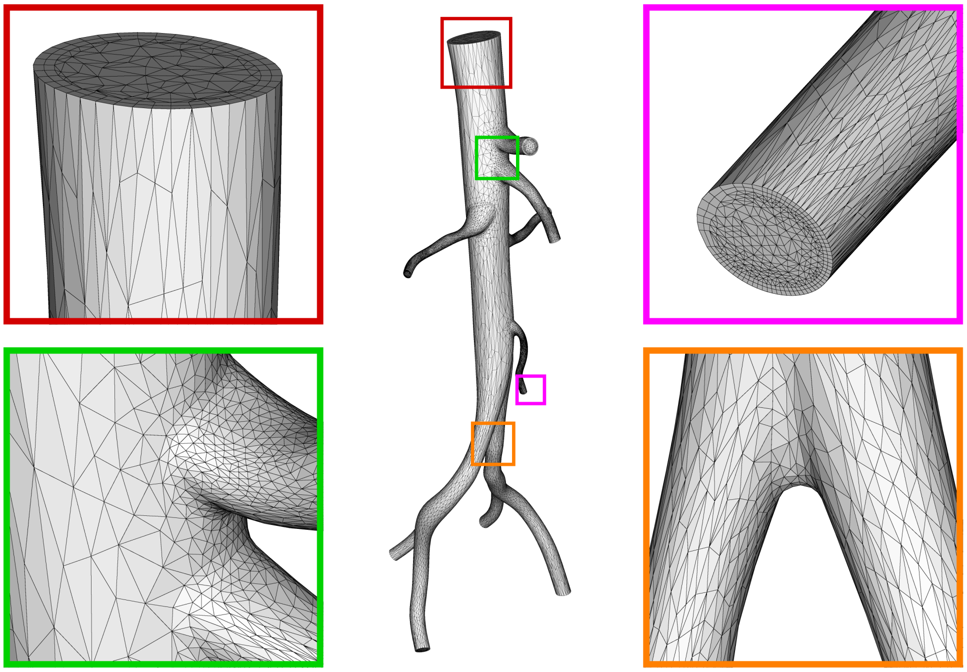






Example

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

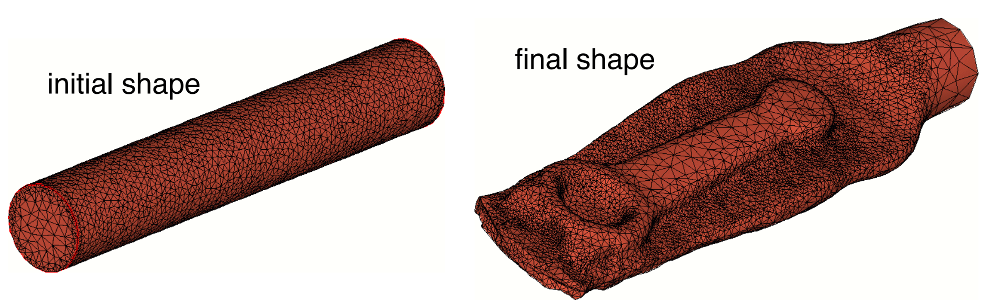


9

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



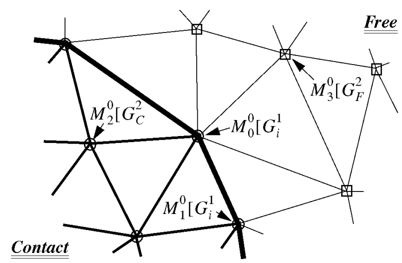
Model Topology Update

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

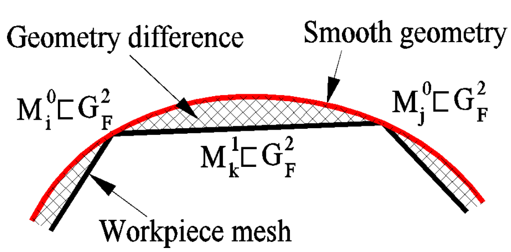
Accuracy degrades due to

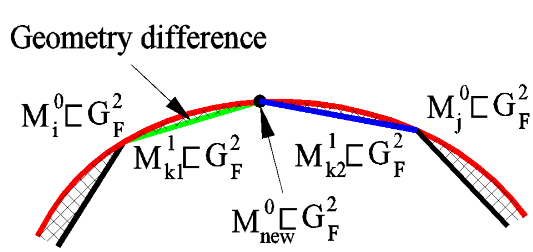
- Poor approximations to the smooth workpiece free surfaces
- Volume change of the workpiece during analysis and mesh updating

- 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

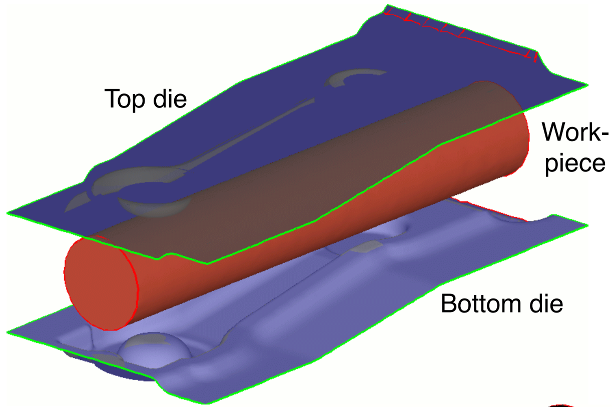
- Nodal velocities
- Nodal temperature
- Elemental effective strains
- Elemental stress tensor (for elasto-plastic materials)

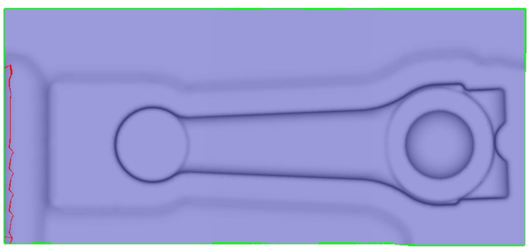
Two approaches for solution transfer

- 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

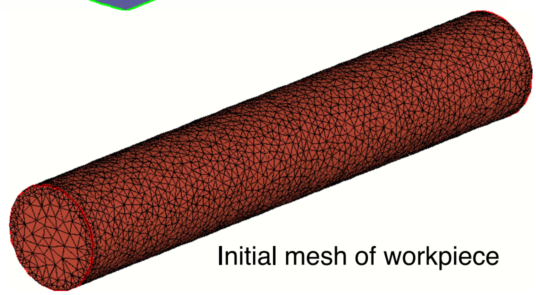
Example Simulation: Steering Link Problem

- 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