Generation and Refinement of Anisotropic Tetrahedral and Hybrid Grids in Three Dimensions

Carlo L. Bottasso and Davide Detomi

AnisGrid02
June 21, 2002
Politecnico di Milano, Italy
Presentation Outline

- Introduction and motivation;
- Boundary layer meshing;
- Anisotropic layer insertion;
- Implementation issues: data structures, mesh motion, handling corners, curving growth curves, local mesh optimization, etc.;
- Examples;
- Concluding remarks.
Introduction and Motivation

- Isotropic grids are **highly inefficient** for resolving certain solution features.
- Good quality anisotropic grids are **difficult** to obtain through the use of subdivision patterns and local retriangulations.

**Example**: Onera M6, Mach=0.8395, Alpha=3.06 deg.
Introduction and Motivation

Experimental data: T.W. Purcell.

Test case: tip Mach=0.95, pitch=0 deg.

Final (isotropic) adapted grid after 4 refinement steps (575K tets).
Introduction and Motivation

Experimental data: T.W. Purcell.

Test case: tip Mach=0.95, pitch=0 deg.

Pressure isocontour lines.

Acoustic pressures at r/R=1.053.
1) **Grow** anisotropic layers of prisms from the model faces inside the domain (possibly split the prisms into tets);

2) **Fix** cross-overs and self intersections;

3) **Fill** the rest of the domain with an isotropic grid using an advancing front mesh generator.


- Fixing of cross-overs **difficult in complex geometries** and time consuming.
- Isotropic meshing necessarily **limited** to the advancing front method.
- **Curving growth curves** not yet satisfactorily demonstrated.

Most difficulties can be traced back to the fact that the **grid is grown in a void**.
Proposed procedure:

1) **Detach** the grid from the model faces where a boundary layer mesh is needed;
2) **March** the grid away from the boundary by using a mesh motion algorithm based on fictitious springs;
3) **Fill** the resulting void with prisms, and (optionally) split all the prisms into tets.

- Can handle arbitrarily **complex** situations.
- No need to use an advancing front generator as in the **advancing layers method**.
- Automatically handles multiple body problems, reentrant corners and other **problematic configurations**.
Proposed procedure:

1) **Cut** along the feature (creates a triangulated surface within the mesh);

2) Improve locally the mesh (collapse small edges, etc.);

3) The two sides of the cut are **marched away from each other**, using a mesh motion algorithm based on fictitious springs;

4) **Fill** the resulting void with prisms, then (optionally) split the prisms into tets.
Implementation Issues

- Data structures for complex topologies;
- Mesh motion algorithm for deforming the isotropic grid;
- Local mesh optimization;
- Consistent tetrahedronization;
- Other issues.
Entitites in the data structure:

- Elements $E_i$
- Nodes $N_j$

Implementation:

$E_i \rightarrow \{N_j\}$

$N_k \rightarrow P_k$

$P_k = (x_k, y_k, z_k)$

+ Efficient and effective for solving PDE’s (residuals, Jacobians, assembly, etc.) on fixed meshes.

– Incomplete for generating and modifying an existing mesh.
Data Structures

Goal:
Define a data structure capable of supporting all the analysis phases: problem definition, mesh generation, solution, error estimation, adaptive grid modification (Beall & Shephard 1997).

Topological entities:
a BRep (Boundary Representation) can be used, as for the geometric model:

- Region \( _M T^3 \): a 3-D entity defined by its bounding faces.
- Face \( _M T^2 \): a 2-D entity defined by its bounding edges.
- Edge \( _M T^1 \): a 1-D entity defined by its bounding vertices.
- Vertex \( _M T^0 \): a 0-D entity that is at the hierarchy base.
Data Structures

Hierarchical representation of the topological entities in a mesh.

Ingredients: regions $\_M T^3$, faces $\_M T^2$, edges $\_M T^1$ and vertices $\_M T^0$.

Adjacencies: describe how the various entities are related to one another.

First order adjacencies: for a given entity $\_M T^d_i$, give all the entities $\_M T^d_j$ ($j \neq i$) which are on its closure ($j<i$), or which it is on the closure of ($j>i$):

Implementation: compromise between adjacency representation and computation (memory vs. speed).
Data Structures

1-level representation:

Circular representation:
**Data Structures**

**Classification** of mesh entities wrt entities of the CAD solid model:

- Ensure mesh validity.
- Automatic inheritance of the analysis attributes from the geometric model to the mesh.
- Mesh adaption in curved domains ("snapping"):

1) $M_T^1 \subset G_T^2$
2) $M_T^2 \subset G_T^2$
3) $M_T^0 \subset G_T^2$

$M_T^1$ will be split.

Snapping
Implementation Issues

• Data structures for complex topologies;
• Mesh motion algorithm for deforming the isotropic grid;
• Local mesh optimization;
• Consistent tetrahedronization;
• Other issues.
Improved Mesh Motion

Mesh motion based on **fictitious elasticity problem** (edge-springs, Batina 1989), using Gauss-Seidel relaxation and **force** (not displacement) control.

Requirements:

- Need to handle **large displacements** without creating invalid elements;
- Need vertex **ordering** to speed-up relaxation.

Add fictitious vertex-face springs:

- **Prevent** a mesh vertex from crossing a neighboring face;
- **Improve** grid quality;
- **Linear** algorithm.
Mesh Ordering

A modified greedy scheme.

Ordering levels around submarine.
Handling Closely-Spaced Faces

- Guarantee vertex visibility;
- **Curved** growth curves;
- **Tangential** relaxation.
Implementation Issues

- Data structures for complex topologies;
- Mesh motion algorithm for deforming the isotropic grid;
- Local mesh optimization:
  - Optimization by repositioning;
  - Optimization by local retriangulation.
- Consistent tetrahedronization;
- Other issues.
Optimization by Repositioning

Centroid of concave region yields invalid mesh.

Centroid of “regularized” region yields valid mesh.
Optimization by Local Retriangulation

Refinement by edge splitting based on subdivision patterns:

Coarsening by edge collapsing:
Optimization by Local Retriangulation

- Edge swap:

- Edge removal:

- Edge, face, region splitting:
Implementation Issues

- Data structures for complex topologies;
- Mesh motion algorithm for deforming the isotropic grid;
- Local mesh optimization;
- Consistent tetrahedronization;
- Other issues.
Consistent Tetrahedronization

Splitting templates.

Splitting algorithm.
Implementation Issues

- Data structures for complex topologies;
- Mesh motion algorithm for deforming the isotropic grid;
- Local mesh optimization;
- Consistent tetrahedronization;
- Other issues.
Ending layers:

Boundary layer ending at a model edge.

Boundary layer ending at a model face.
Other Issues

Growth directions with visibility constraint:

- **Normal to vertex-manifold** (Kallinderis & Ward, 1993);
- Nominal normals *smoothed* by distance-weighted averaging;
- Apply only to transition triangulation (internal layer vertices are free to deviate, automatic *curving of growth curves*).
Other Issues

Mesh reclassification:
**BL Meshing Examples**

**Engine nacelle:** 272K tets in 10 layers.

**Max dihedral angles**

**r/R**
BL Meshing Examples

**Submarine**: 238K tets in 6 layers.

Max dihedral angles

![Graph showing dihedral angles distribution.](image)
BL Meshing Examples

Growth curves at problem locations:
Anisotropic Layer Insertion
Anisotropic Layer Insertion
Conclusions

A software program consisting of:

• A **generator** of highly stretched tetrahedral meshes;
• Tools for mesh **motion** and mesh **cutting**;
• Tools for mesh **optimization**.

Highlights:

• Any **mesh generation** algorithm can be used for the isotropic grid;
• No post-processing required for correcting **cross-overs** and **self-intersections**;
• Integration with **CAD** assures mesh validity and no losses of information and precision;
• Handling of **arbitrarily complex 3D geometries** and multiple body configurations.