Review of mesh generation and load balancing technologies

Nikos Chrisochoides
Northeast Parallel Architectures Center,
Syracuse University, 111, College Place,
Syracuse, NY, 13244-4100

December 11, 1994

Procedure for :

Simulating a Physical Phenom. or Designing a Structure

1.	Define the physical space $\Omega \subset R^n$ and a PDE operator $\mathrm{Lu}(\mathbf{x}) = \mathrm{f}(\mathbf{x}),\ \mathbf{x} \in \Omega$ with B.C's $\mathrm{Bu}(\mathbf{x}) = \mathrm{g}(\mathbf{x}),\ \mathbf{x}$ on $\vartheta\Omega$	1.	Prepare the candidate design
2.	Analyze the behavior of the phenomenon using F.E or F.D	2.	Analyze the design using F.E
3.	Asses the validity of the analytical results	3.	Asses the validity of the analytical results
4.	Repeat steps 1-3 until satisfied form the analysis.	4.	Repeat steps 1-3 until design is acceptable

Automatic Finite Element Analysis System

2.1 ***Discretize the physical space*** Ω , Ω^h, and the PDE operators, $\mathrm{Ax} = \mathrm{b}$.

2.2 Solve the linear system of equations $\mathrm{Ax} = \mathrm{b}$.

2.3 Asses the validity of the numerical results.

2.4 Repeat steps 2.1-2.3 until acceptable numerical results are obtained.

Figure 1: Automatic Finite Element Analysis System

OUTLINE

- Background on Mesh Generation
- Mesh Generation Methods
  - Numerical
  - Geometric
  - Algebraic
- Parallel Mesh Generation
- Adaptive Meshes
- Parallel Adaptive Mesh Generation & Load Balancing
- Sumary

Background on Mesh Generation

DEFINITIONS :

Conforming Mesh in R^2

Let M $= \{t_1, t_2, ..., t_N\}$, where t_i is polygonal element (usually triangle) so that (i) $\cup_{i=1,N} t_i = \Omega^h$ and (ii) for $i \neq j$, $t_i \cap t_j$ is either *emtry* or a common *side* of t_i and t_j, or a common *vertex* of t_i and t_j.

Valid Mesh

- The mesh M is conforming
- No element of M is outside Ω
- All nodes of M are either inside Ω or on the $\vartheta\Omega$
- All edges and vertices of the domain Ω are represented in M.

Structured/Unstructured Mesh

A mesh is called structured if every vertex of the mesh has a constant number of incident edges, otherwise the mesh is called unstructured.

Background on Mesh Generation

MORE DEFINITIONS :

- **Voronoi Diagram**
  For a given set of points S, the Voronoi is the tesselation of the plane into convex polygons, each containing one point such that every point in that region is closer to that point than other given points.
- **Delaunay Triangulation**
  Is the strait-line dual of the Voronoi diagram.

Background on Mesh Generation

EVEN MORE DEFINITIONS :

Automatic Meshing procedure

exhibits the following behavior :

- It is iterative and includes a finite number of submeshes $M_0, M_1, ..., M_n$
- M_{i+1} is uniquely determined by M_i.
- Each of $M_0, M_1, ..., M_n$ is guaranteed to terminate after finite number of numerical operations.

Mesh Smoothing procedure

improves the mesh by repositioning the vertices/edges of the mesh so that the elements satisfy predefined criteria. (Ex. *Laplacian smoothing* seeks to reposition the vertices so that each internal node is at the centroid of the polygon formed by its connected neighbors.)

Vital Issues in Mesh Generation

- Automation
- Robustness
- Element Quality (type, shape)
- Efficiency (time, space)
- Self-Adaptiveness

OUTLINE

- Background on Mesh Generation
- <u>Mesh Generation Methods</u>
  - ⇒ <u>**Numerical**</u>
  - Geometric
  - Algebraic
- Adaptive Meshes
- Parallel Mesh Generation
- Parallel Adaptive Mesh Generation & Load Balancing
- Summary

Mesh Generation Methods

CLASSIFICATION SCHEMES :

- Based on the temporal order of the creation of vertex and element sets of the mesh.
  - i) Front Techniques
  - ii) Advanced Front Techniques
  - iii) Topology Decomposition + Smoothing
  - iv) Mesh Templates

- Based on the nature of the computation is required to generate the mesh
  - i) Numerical
  - ii) Geometrical
  - iii) Algebraic

Numerical Grid Generation

- Basic Procedure
- Numerical Generation Systems
  - Elliptic
  - Parabolic & Hyperbolic
  - Orthogonal
  - Conformal
- Summary

Numerical Grid Generation

Basic Procedure

S1 : Partition the domain Ω into disjoint patches (blocks in 3D) consisting by four (six in 3D) curved sides (surfaces in 3D).

S2 : *For* each patch *Do*
Specify the values of the curvilinear coordinates on the boundary of the patch.

S3 : *For* each patch *Do*
Transform the physical patch into a rectangular computational space.

S4 : *For* each rectangular *Do*
1. Discretized with uniform square grid
2. Assign the values of the Cartesian coordinates of each successive boundary point as boundary values.

S5 : Choose transformation relations (from curvilinear to Cartesian system). Usually of PDE system.

S6 : Discretize the PDE system, (Ax = b).

S7 : Solve the system of equations, Ax = b.

Numerical Grid Generation

Close Look of the Steps 3 & 4

Numerical Grid Generation

Boundary Value Problem

Numerical Grid Generation

Elliptic

Parabolic & Hyperbolic

Orthogonal

Conformal

Numerical Grid Generation

SUMMARY

Aut : Automatic , Rob : Robust

Tef : Time Efficient , Sef : Space Efficient ,

ElQ : Element Quality , SAd : Self-Adaptive

Table 1: Evaluation of the Numeric Grid Generation.

Approach	Aut	Rob	Tef	Sef	ElQ	SAd
Numerical	No	Yes	O(N)	O(N)	Fair	Fair

OUTLINE

- Background on Mesh Generation
- <u>Mesh Generation Methods</u>
  - Numerical
  - ⇒ **<u>Geometric</u>**
  - Algebraic
- Adaptive Meshes
- Parallel Mesh Generation
- Parallel Adaptive Mesh Generation & Load Balancing
- Summary

Geometric Mesh Generation

Approaches

- ### Cut and Mesh

  Based on the partition of the domain into convex and simple-connected subdomains.

- ### Domain Triangulations

  Based either on Node Insertion followed by Domain Triangulation or on Background mesh which iteratively refined using spacing function.

- ### Recursive Spatial Decomposition

  Based on approximating the domain with a union on non-intersecting variably sized cells generated by recursively subdividing a spatial region enclosing the domain.

Summary

Geometric Mesh Generation

Cut and Mesh

- The domain Ω is first partitioned into subdomains, that are convex and simple-connected (i.e. no holes). Each subdomain then is meshed.
- The partitioning of a general domain into convex subdomains is NP-Complete problem, thus it depends upon heuristics.
- The partition of curved polyhedron S requires first the construction of a planar polyhedron PS of S and then apply the cut and mesh algorithm.
  There is no robust algorithm for constructing planar polyhedron PS for a curved polyhedron.

Geometric Mesh Generation

Domain Triangulations

- *Node Insertion Phase :* Nodes are distributed within and on the boundary of the domain using FE density function.
  Domain Triangulation Phase: The nodes are automaticly triangulated.
- A. *Constrained Delaunay Triangulation :* maximizes the minimum angles of the triangles.
- B. Background mesh (by discretizing the boundary of the domain) and iterative improvement based on spacing function.

Note : No need to subdivide the domain into convex polygons.

Geometric Mesh Generation

Recursive Spatial Decomposition

- Initialization of the quadtree, without performing any classification of the quadrants.
- Discretization of boundary edges, intersections with quadrants are calculated and its status changes to boundary terminal (include edge segment) or vertex quadrants.
- Determination of the interior quadrants.
- Improvement of boundary quadrants. Smoothing operations are applied.
- Triangular mesh generation.

Limitation : The mesh is dependent on the orientation and position if the initial enclosing box is not tight.

Numerical Grid Generation

SUMMARY

Aut : Automatic , Rob : Robust
Tef : Time Efficient , Sef : Space Efficient ,
ElQ : Element Quality , SAd : Self-Adaptive

Table 2: Evaluation of the Numeric Grid Generation.

Approach	Aut	Rob	Tef	Sef	ElQ	SAd
Numerical	No	Yes	-	O(N^{ϵ})	$Fair^{+}$	Fair
Cut & Mesh.	Y/N	Fair	-	O(N)	Fair	Fair
Triangulations	Yes	Fair	-	O(N)	Fair	Yes
Rec. Spt. Dec.	Yes	Fair	-	O(N)	$Good^{-}$	Yes

OUTLINE

- Background on Mesh Generation
- Mesh Generation Methods
  - Numerical
  - Geometric
- ⇒ **Parallel Mesh Generation**
- Adaptive Meshes
- Parallel Adaptive Mesh Generation & Load Balancing
- Summary

Parallel Mesh Generation

- Static Single Phase Computation
- Locality Properties
- Parallel Numerical Mesh Generations Methods (See PDEs)
- Parallel Triangulations Based on a Background Mesh
  - Pre-Processing :

    Partitioning + Allocation
  - Parallel-Processing :

    Compute Sub-meshes + Smoothing
  - Post-Processing :

    Assembly Sub-Meshes

Parallel Mesh Generation

Locality Properties

OUTLINE

- Background on Mesh Generation
- Mesh Generation Methods
  Numerical
  Geometric
- Parallel Mesh Generation
- ⇒ **Adaptive Meshes Refinement**
- Parallel Adaptive Mesh Generation & Load Balancing
- Summary

Adaptive Meshes Refinement (AMR)

- Basic Idea
- AMR + PDEs
- AMR Strategies
  - Priori in nature
  - Posteriori in nature
- Data Structures + AMR Algorithms
- Adaptive VS. Quasi-Uniform meshes
- Summary

Adaptive Meshes - Basic Idea

IDEA :
Distribute the mesh points by concentrating them in regions of large variation in the solution of the PDE.

CONFLICTING ISSUES :

- Increase computational effort/iteration
- Increase software complexity
- Improve convergence rate
- Minimize number of points required

Adaptive Mesh Refinement + PDEs

Definitions :

- Let Lu = f in Ω a PDE, where $\Omega \in R^2$.
- Let Ω^h be a tessellation of Ω and
- Let u_h the finite element approximation of u corresponding to Ω^h.
- Define the error e = u - u_h, and denote with $||e||_t$ the error restricted to element t.
- Then $||e||_t \approx C h_t^q$, for some constand C depending on u, PDE, t, and Ω. $q \geq 0$ and h_t diameter of t.
- Let $||e||_{tmax} = \max_{t \in \Omega} ||e||_t$

Adaptive Mesh Refinement + PDEs

Algorithm :

```
k ← 0
solve the PDE on the mesh $\Omega_{h_o}$
estimate erros
while $||e||_{tmax} \geq \tau$ do
    determine the set $S_k$ to refine, $S_k \subseteq \Omega_{h_k}$
    divide $t \in S_k$ and those necessary for
       compatibility to get $\Omega_{h_{k+1}}$
    solve the PDE on tessellation $\Omega_{h_{k+1}}$
    estimate errors
endwhile
```

Tasks to be performed :

- solving the PDE
- dtermining a set of elements to divide
- dividing triangles and maintaining compatibility
- estimating erros

AMR Problem : Criteria

Discretize the physical space so that :

- Represent continuous functions by discrete values with sufficient accuracy.
- Generate **smooth** point distribution.
  (gradual increase/decrease of distance among neighbor points)
- Avoid **skewed** elements.
  Truncation error will increase.
- Control point distribution by error-estimators.
- Map point distribution on simple and efficient data structures.

AMR Strategies

- Priori in Nature
  Based on the estimation of the discretization errors present in a given mesh. Such erros must be computed directly from the mesh.

  Based on the Geometric characteristics (singularities e.t.c)
- Posteriori Nature
  Based on finite estimators derived from the finite element solution.
  There are heuristics for the error estimation based on gradient, residual. e.t.c

Priori AMR Strategies

- Triangulation + Density Functions

- Curvilinear Coordinate Systems

- Modified Quadtree + Geometry

Posteriori AMR Strategies

- Redistribution of a Fixed Number of Points (r-methods)
  - Control Function Approach
  - Variational Approach
  - Attraction - Repulsion Approach
- Restructure of a Fixed Set of Points (h-methods)
  - Dividing Triangles
  - Creating a Nested Hierarchy of Finer Rectangular Subgrids (NHFR)
- Local Increase in Algorithms Order (p-methods)
- hp-methods

AMR Strategies

Redistribution of a Fixed Number of Points

- Control Function Approach

- Variational Approach

- Attraction - Repulsion Approach

AMR Strategies

Restructure of a Fixed Set of Points

1. Dividing Triangles

- Delaunay Method +

$$\begin{pmatrix} Node\ Insertion \\ Edge\ Swapping \end{pmatrix} + Smoothing$$

- Bisection Method

  Local : Search for all non-conforming elements and bisect them to eliminate the non-conforming nodes.

  Global : Repeatedly bisect the longest side of all non-conforming nodes, until a conforming mesh is produced.

Notes

- AMR methods based on Dividing Trianles ensure conforming meshes.
- Bisection Methods ensure smooth meshes.
- Delaunay Methods are fast but not so good in quality.

AMR Strategies

Restructure of a Fixed Set of Points

2. Creating a Nested Hierarchy of Finer Rectangular Subgrids (NHFR)

IDEA : Given a list of flagged grid points, place (rotate) rectangular subgrids so that :

1. each flagged point is interior to a fine grid and
2. the total area of the refined grids is minimum.

ALGORITHM :

- Separate the flagged points into clusters
- Fit a rotated rectangular grid to each cluster
- For each rectangular, evaluate the ratio
  $\mathrm{r} = \frac{\#\ of\ flagged\ grid\ points}{total\ \#\ of\ coarse\ grid\ poins\ in\ the\ new\ fine\ grid}$
  if r $\in [\frac{1}{2}, \frac{3}{4}]$ then stop, otherwise use more expensive clustering algorithm based on MST.

AMR + Data Structures

Table 3: Performance evaluation of the Adaptive Mesh Refinement Approaches

Approach	Data Strt.	Space Cmplx.	Time Cmplx	Pt. Distrb.	Error Eval.

OUTLINE

- Background on Mesh Generation
- Mesh Generation Methods
  - Numerical
  - Geometric
- Adaptive Meshes
- Parallel Mesh Generation
- ⇒ **Parallel AMR & Load Balancing**
- Summary

Parallel AMR + PDEs

Algorithm :

```
k ← 0
solve the PDE on the mesh $\Omega_{h_o,m(i)}$ or $\Omega_{h_o}$
estimate erros
while $||e||_{tmax,m(i)} \geq \tau$ do
    determine the set $S_{k,m(i)}$ to refine, $S_{k,m(i)} \subseteq \Omega_{h_k,m(i)}$
    divide $t \in S_{k,m(i)}$ and those necessary for compatibility and boundary indegrity to get $\Omega_{h_{k+1},m(i)}$
    compute new communication pattern
    solve the PDE on tessellation $\Omega_{h_{k+1},m(i)}$ or $\Omega_{h_{k+1}}$
    estimate errors
endwhile
```

Parallel Adaptive Mesh Refinement + PDEs

Tasks to be performed :

- solving the PDE
- determining a set of elements to divide
- dividing triangles and maintaining compatibility and boundary integrity for the subdomains
- load balancing
- updating communication patterns
- migrating data
- estimating erros

Load Balancing

Goal :

Load Balancing

Statment of the problem :

Load Balancing

Mathematical formulation of the problem :

Load Balancing

Approaches :

- Relaxation or Diffusion
- Centralized
- Decentralized
- Nearest-Neighbor

Load Balancing

Approaches - Evaluation :

Parallel Adaptive Mesh Refinement + PDEs

Some Results :

- Quasi-uniformity instead of adaptivity
- Multigrid + LUMR
- Object-Oriented programming + Data Encapsulation
- Data Migration + Hashe Cache Scheme

OUTLINE

- Background on Mesh Generation
- Mesh Generation Methods
  - Numerical
  - Geometric
- Adaptive Meshes
- Parallel Mesh Generation
- Parallel Adaptive Mesh Generation & Load Balancing
- ⇒ **Summary**

Summary