PYRAMID: Parallel Unstructured Adaptive Mesh Refinement

John Z. Lou, Charles D. Norton, and Thomas A. Cwik

National Aeronautics and Space Administration

Jet Propulsion Laboratory

California Institute of Technology

http://hpc.jpl.nasa.gov/APPS/AMR

HPCC/Earth and Space Sciences Project

PYRAMID: Parallel Unstructured Adaptive Mesh Refinement

PYRAMID

Modern... Simple... Efficient… Scalable...

Technology Description

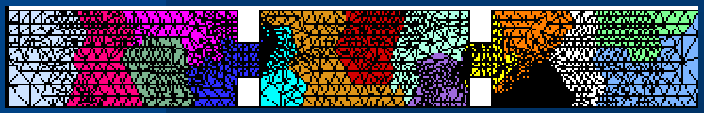
An advanced software library supporting parallel adaptive mesh refinement in large-scale, adaptive scientific & engineering simulations.

State-of-the-Art Design!

- Efficient object-oriented design in Fortran 90 and MPI
- Automatic mesh quality control & dynamic load balancing
- Scalable to hundreds of processors & millions of elements

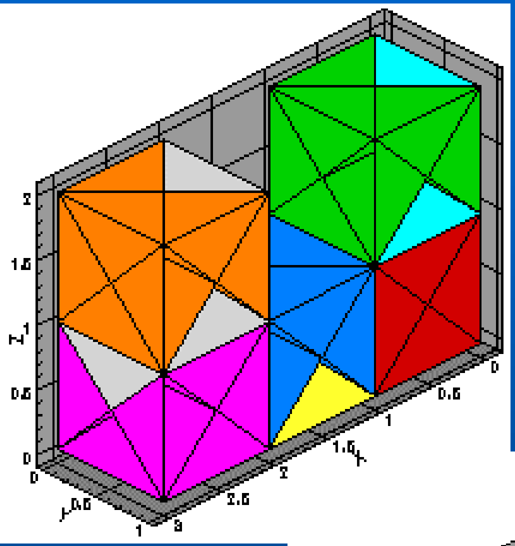
Application Arena

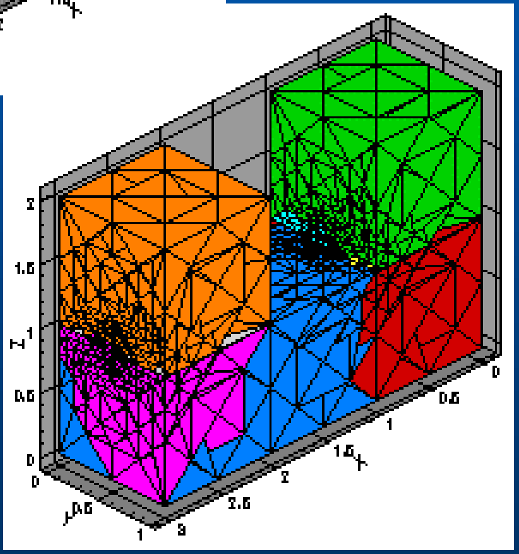
- Computer Modeling & Simulation Applications with complex geometry
- Electromagnetic and semiconductor device modeling
- Structural/Mechanical/Fluid dynamics applications



2D Adaptive Refinement on Waveguide Filter

John Z. Lou, Charles D. Norton, & Thomas A. Cwik
High Performance Computing Systems and Applications Group
http://www-hpc.jpl.nasa.gov/APPS/AMR





3D Successive Parallel Adaptive Refinements on Cray T3E

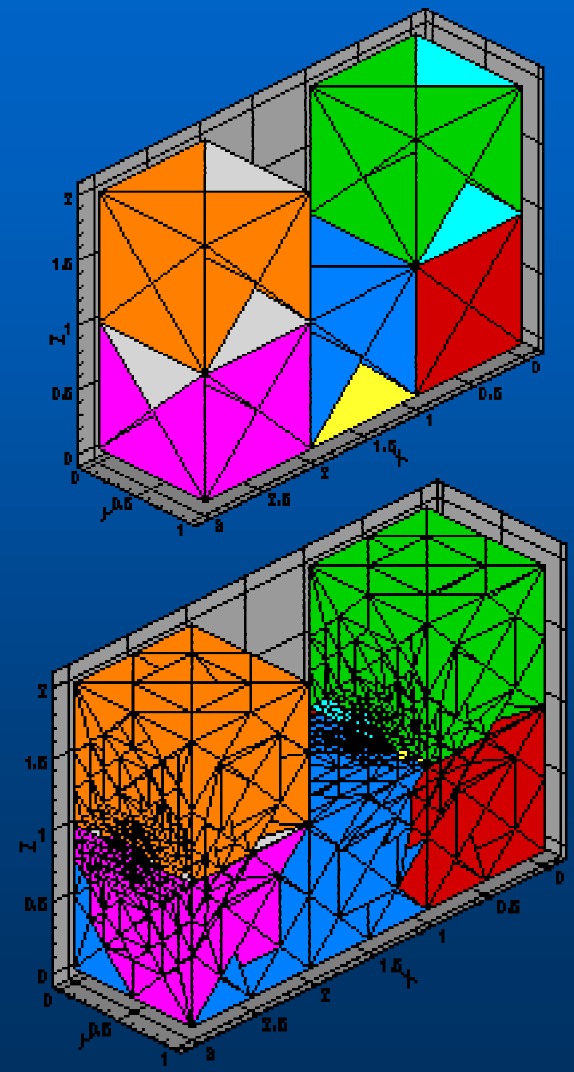
Accomplishments

- 3D library developed and tested on Cray T3E & Beowulf cluster
- System applied to a variety of meshes
- Publications and presentations

JPL

The Pyramid Project

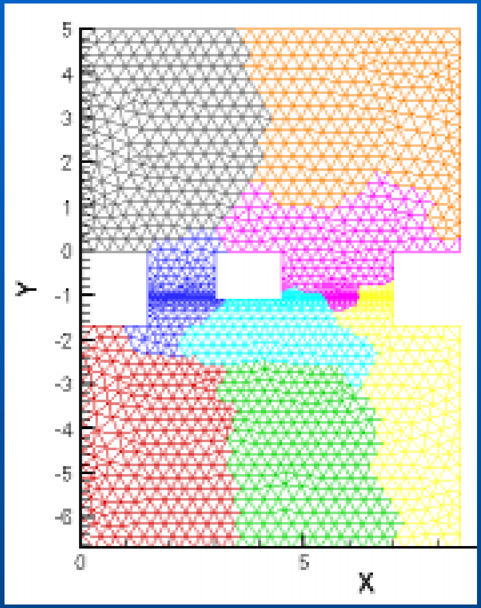
- **Motivation**
  - Develop a portable, robust, parallel AMR tool for NASA applications based on triangular and tetrahedral meshes
- **Applications**
  - Electromagnetic and semiconductor device modeling
  - Earth and space science applications
  - Unstructured parallel finite-volume and finite elements
  - Synthetic aperture radar applications



Demonstration Tetrahedral Mesh
NASA/JPL/Caltech

Pyramid Package Components

- **Components**
  - Parallel mesh I/O, partitioner, logical and physical adaptive refiners, mesh migration, and visualization
- **Development Platform**
  - Fortran 90: Core data structures and internals
  - C: Interface to ParMetis graph partitioner
  - MPI: Distributed-Memory communication
- **Front-end interface to Fortran 77 and C is included**



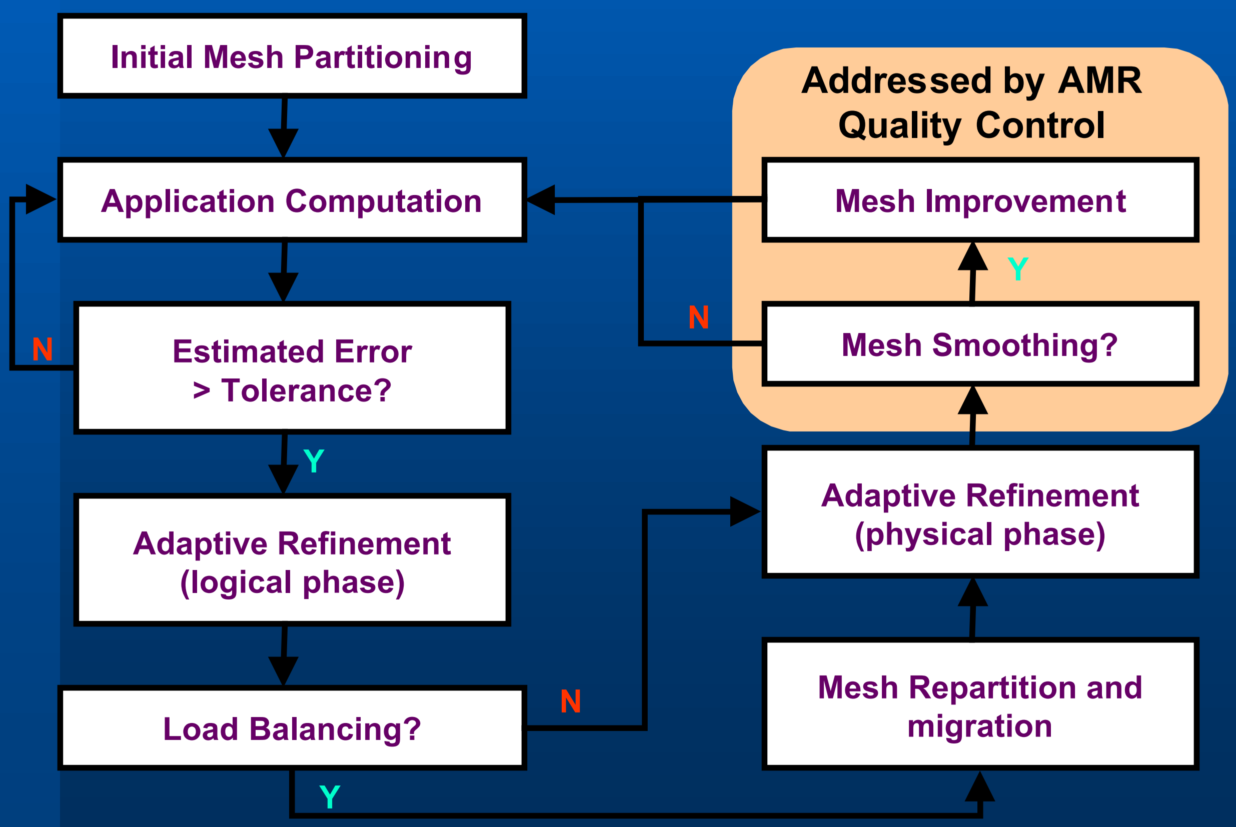
Demonstration Triangular Mesh
NASA/JPL/Caltech

Development Issues

- **Parallel Unstructured AMR Scheme**
  - Logical AMR: Iterative scheme defining refinement pattern (with mesh quality control)
  - Physical AMR: Locally refine coarse elements based on logical refinement
- **Parallel Dynamic Load Balancing Strategy**
  - Repartition weighted logical mesh, migrate coarse elements, and perform local physical refinement
- **Modular Design**
  - Performance and abstraction features of Fortran 90

Our Parallel AMR Process

- **Organization**
  - Partitioning, Adaptive Refinement, Load Balancing, Mesh Migration, and Element Quality Control



Technology

- **Fortran 90 Supports Abstract Data Types**
  - Simplifies program interfaces and hides implementation details
- **Helps build codes from modular pieces written by different authors**
  - Enhances software collaboration
  - Increases program safety
  - Allows code changes without impacting users
  - Encapsulates codes, advancing them to new standards
  - Programs can be expressed in a more natural way

Technology

- **Fortran 90/95 Features Modernize Programming**

Modules
Encapsulate data and routines across program units

Use-Association
Controls access to module content

Generic Interfaces
One call can perform different actions based on types

Derived Types
User-defined types supporting abstractions in programming

Array Syntax
Simplifies whole array, and array subset, operations

Pointers/Allocatable Arrays
Supports flexible/dynamic data structures

Backward compatible with Fortran 77

FOR MORE INFO...

Fortran 90 Programming. Ellis, Philips, & Lahey; Addison Wesley, 1994
http://hpc.jpl.nasa.gov/PEP/nortonc/oof90.html

Technology

- **Data Structures and AMR Relationships**
  - Using abstractions in the initialization section

```
PROGRAM pamr
USE pyramid_module
implicit none
   ! Statements omitted...
   type (mesh) :: in_mesh, out_mesh
   call PAMR_INIT()
   call PAMR_CREATE_INCORE( in_mesh, in_file )
   call PAMR_REPARTITION( in_mesh )
   ! Adaptive refinement loop…
   call PAMR_ELEMENT_COUNT( out_mesh )
   call PAMR_VISUALIZE( in_mesh, “visfile.plt” )
   call PAMR_FINALIZE()
END PROGRAM pamr
```

Technology

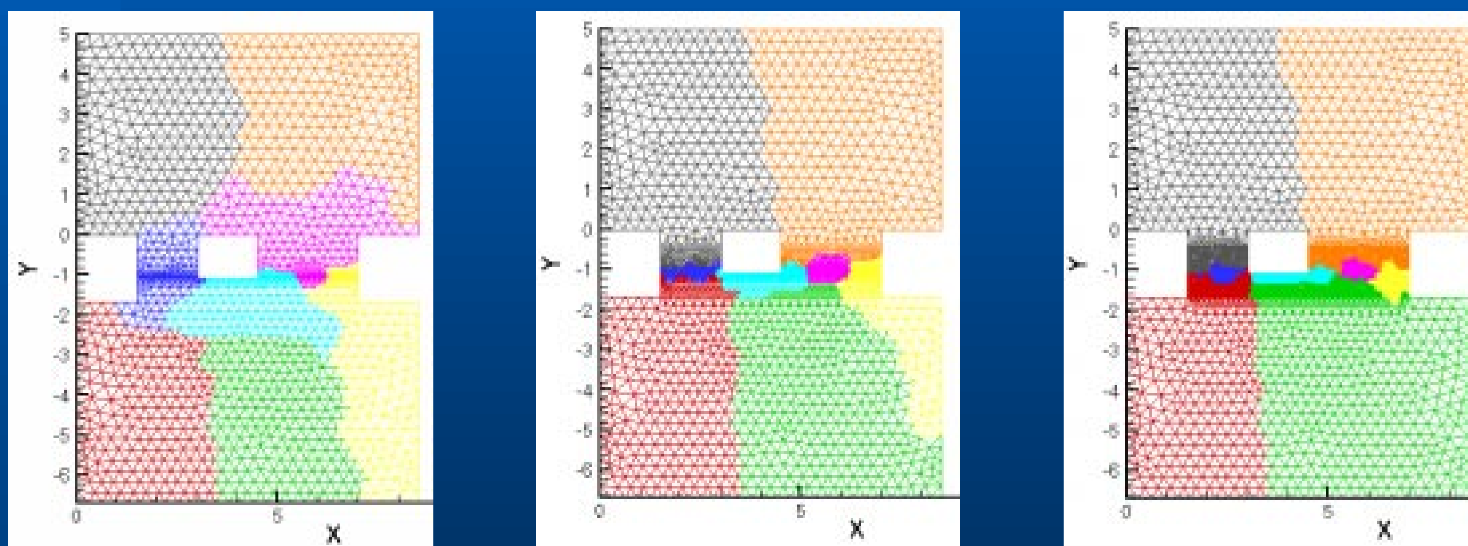
- **Data Structures and AMR Relationships**
  - Using abstractions in the loop section

```
PROGRAM pamr
! Adaptive refinement loop...
   do i = 1, refinement_level
      call PAMR_ERROR_EST( in_mesh, out_mesh )
      call PAMR_LOGICAL_AMR ( in_mesh )
      call PAMR_REPARTITION( in_mesh )
      call PAMR_PHYSICAL_AMR ( in_mesh, out_mesh )
   end do
END PROGRAM pamr
```

  - Users must specify their error estimation method

Technology

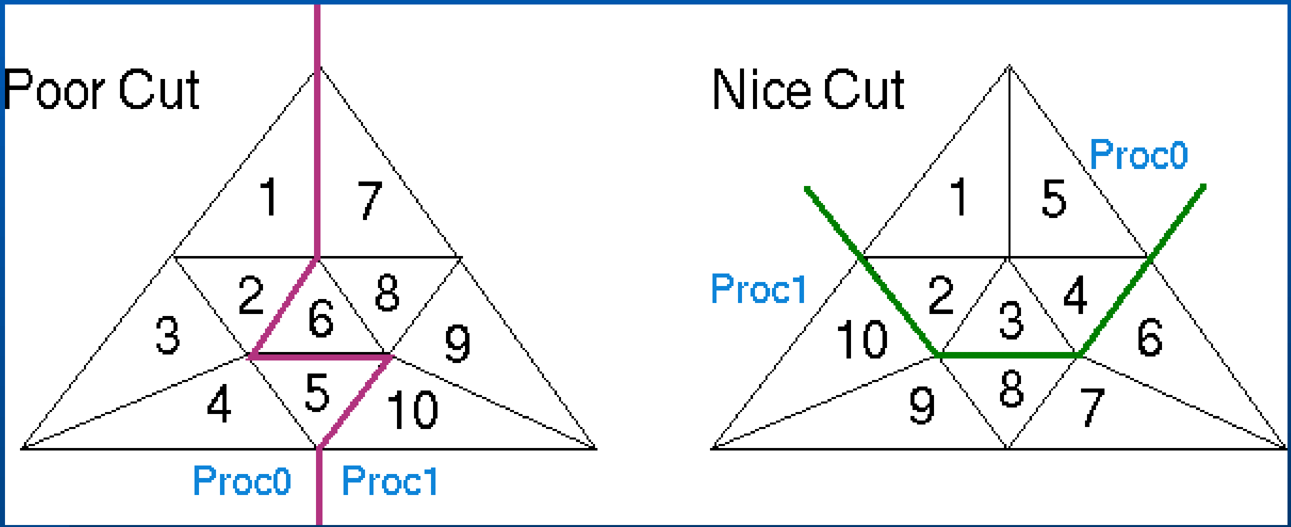
- **Dynamic Load Balancing with ParMetis**
  - Construct ParMetis dual-graph (F90/C interoperability)
  - Migration relocates elements, using new partitioning and reconstructs the mesh efficiently, and in parallel



  - Migration handles irregular communication patterns with an efficient non-blocking algorithm

Technology

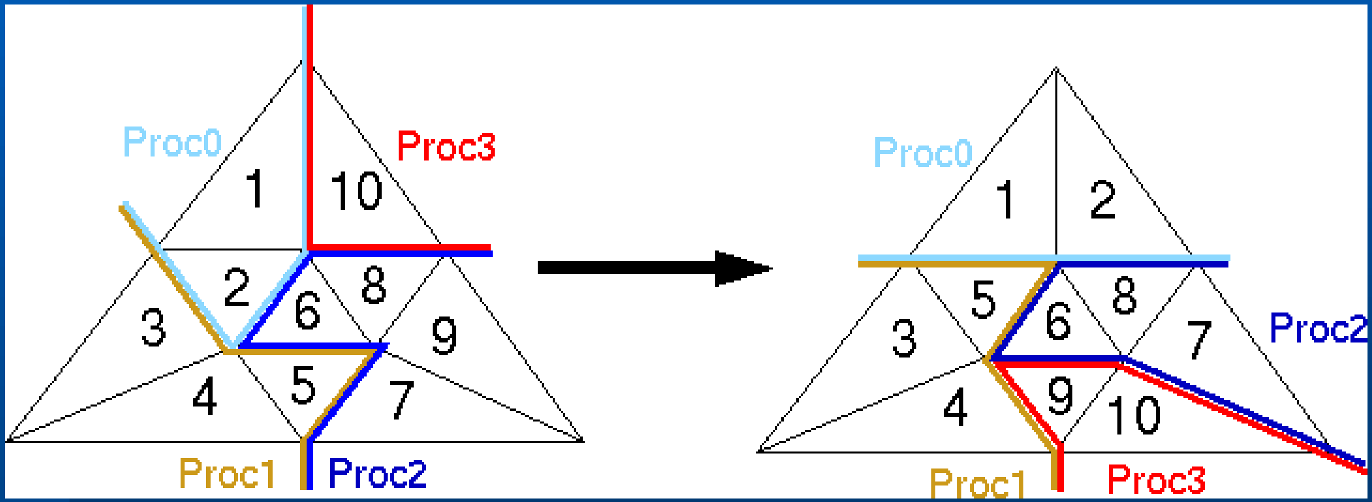
- **Dynamic Load Balancing with ParMetis**
  - ParMetis tries to minimize edge-cut length and data movement among partitions



- ParMetis requires that elements are ordered based on the processor ordering

Technology

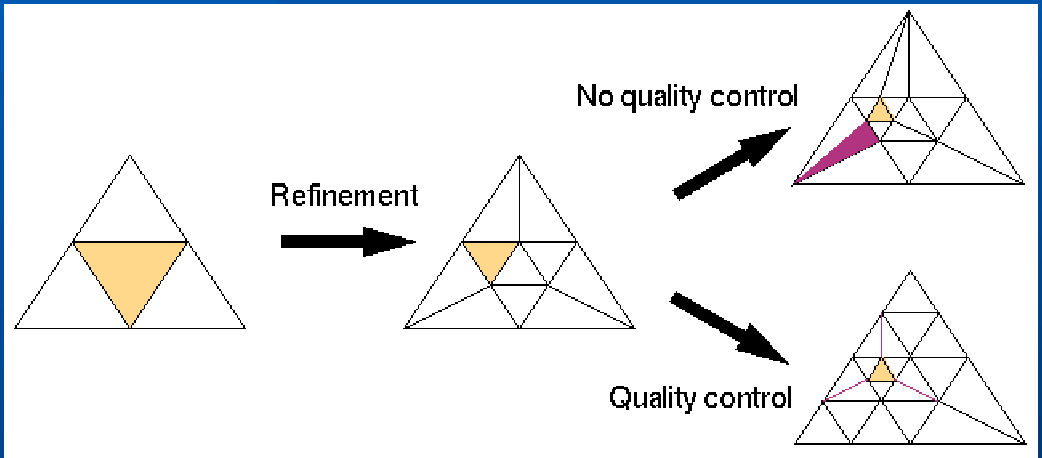
- **Dynamic Load Balancing with ParMetis**
  - Asynchronous communication algorithm is applied since data movement is irregular

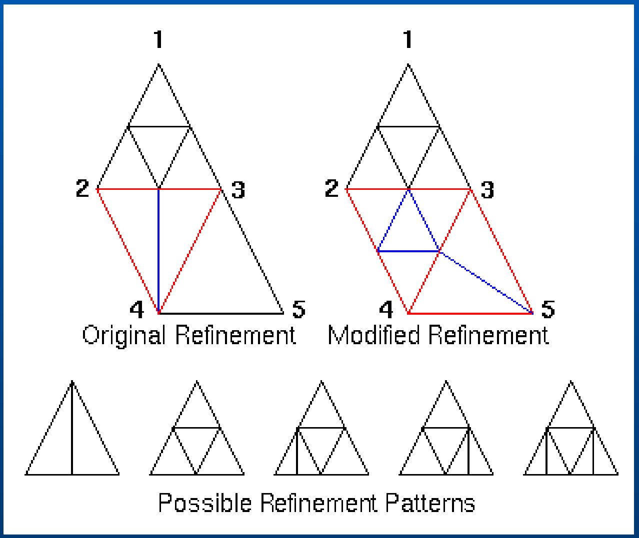


- Mesh is rebuilt efficiently, due to ParMetis requirements, by heapsorting methods

Technology

- **Automatic Mesh Quality Control**
  - Modify coarse element refinement if successive refinements cause poor aspect ratios

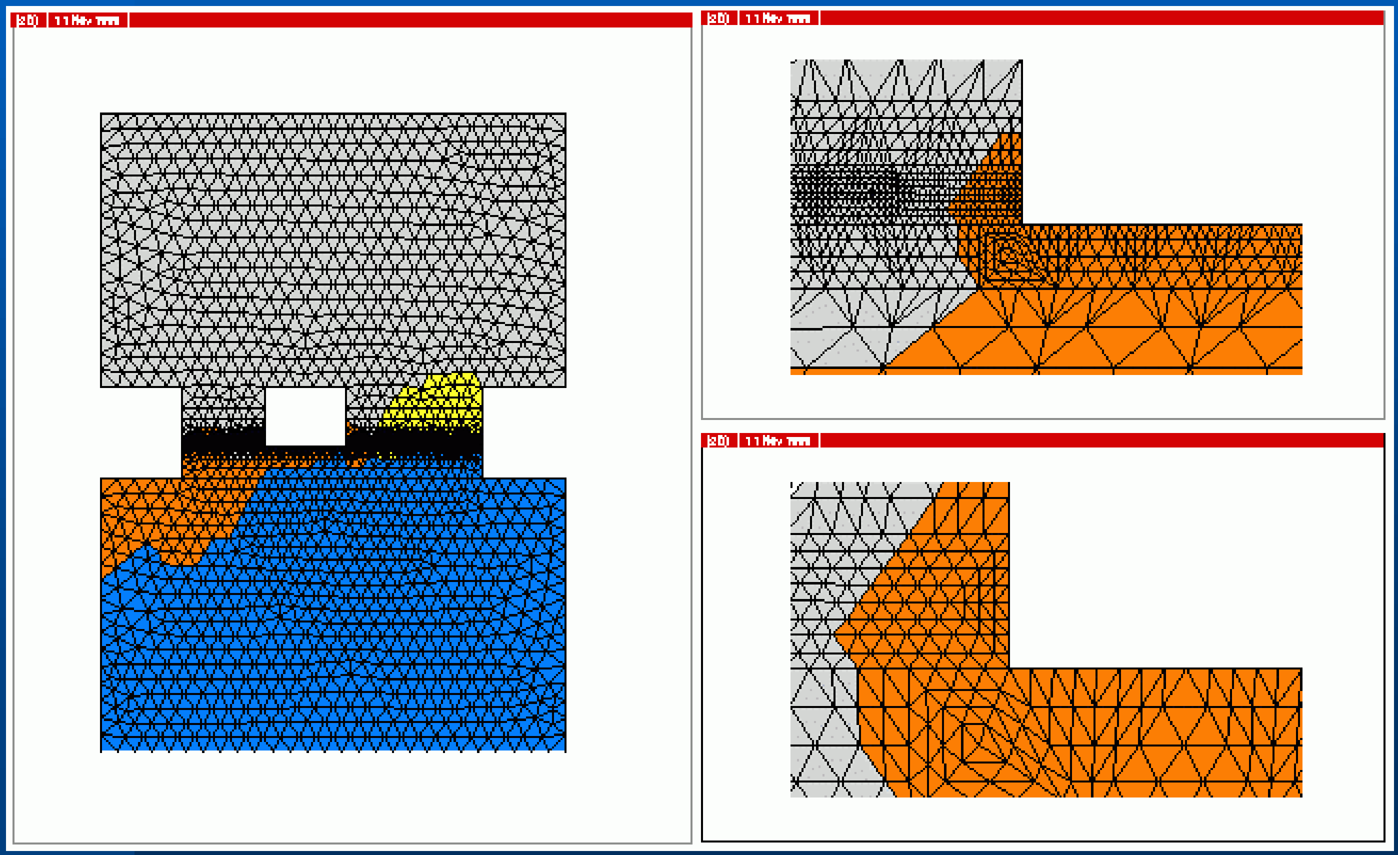




  - Controls quality at the expense of additional elements

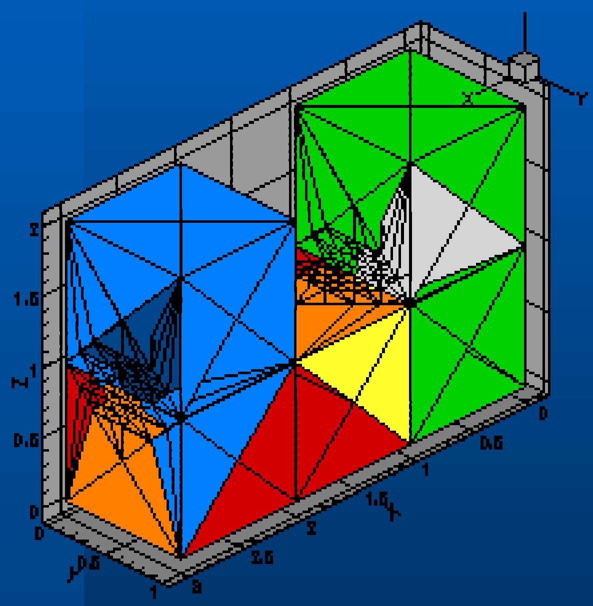
Technology

- **Automatic Mesh Quality Control**
  - Benefit of quality control applied to triangular elements

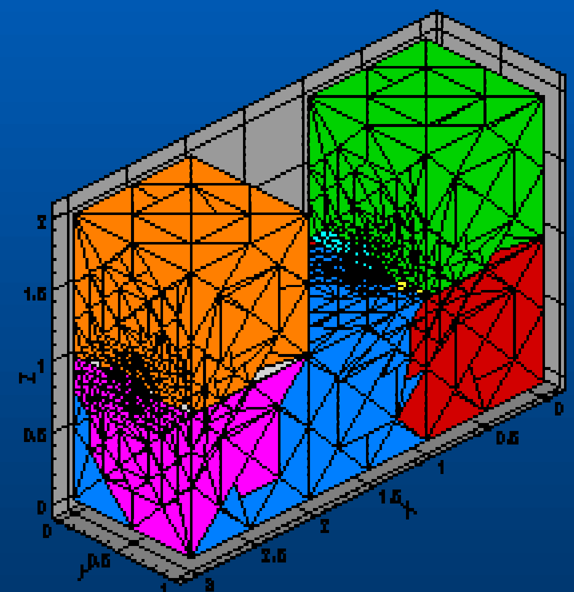


Technology

- **Automatic Mesh Quality Control**
  - Benefit of quality control applied to tetrahedral elements



Poor Mesh Elements Without Quality Control



Good Mesh Elements With Quality Control

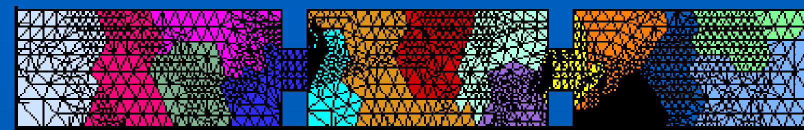
Performance & Scalability

- **Commentary**
  - Measurements can, at times, be awkward as this varies based on how the mesh is refined and load balanced
- **Empirical Lower Bound Limits**
  - We estimate that a lower bound of ~280,000 elements per processor is achievable on a 512 MB per node processor

Machine	PE Memory	Elements per PE	Minimum # of PEs	Largest Mesh
Cray T3E	128 MB	~60,000	64	~3,500,000
Beowulf	512 MB	~140,000	4	~554,000

Performance on Cray T3E

- **2D Scalability For Fixed Size Waveguide Filter Mesh**
  - Three levels of refinement with increasing numbers of processors



Number of Processors	AMR Time	Load Balancing (Migration) Time	Number of Elements
32	57.34	15.36	292,612
64	13.55	3.75	295,405
128	2.93	1.65	305,221
256	0.54	1.51	335,527
512	0.27	1.86	397,145

- Element number varies since refinement randomly chooses 1/2 of the elements per partition where mesh consistency must be maintained

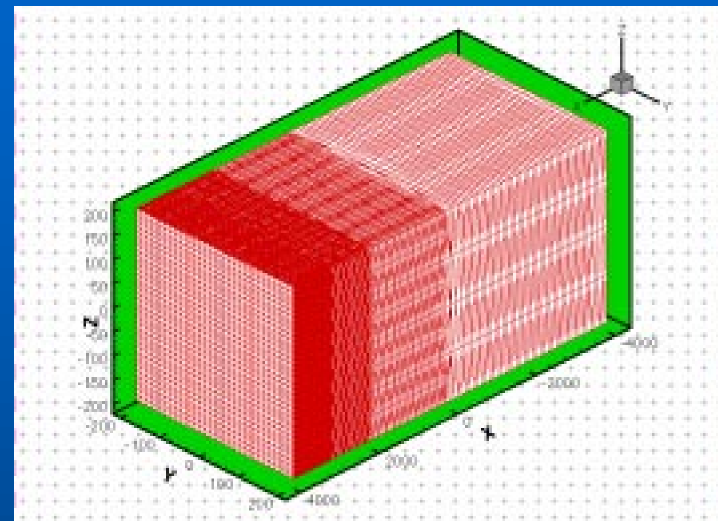
Performance on Cray T3E

- **3D Scalability For Fixed Size U-Mesh Problem**
  - Initial number of elements: 2648

Processors	Level 1	Level 2	Level 3
Elements at AMR Level	~19,338	~133,434	~570,012
8	3.22	45.59	818.70
16	2.27	34.57	484.82
32	1.32	19.71	260.97
64	1.22	11.38	116.73
128	1.20	4.56	58.27
256	1.58	3.47	25.63

Performance on Cray T3E

- **3D Scalability For Fixed Size Earthquake Mesh Problem**
  - Initial number of elements: 1316
  - Three levels of refinement on 4 PEs exceeds memory limitation



Processors	Level 1	Level 2	Level 3
Elements at AMR Level	~10,528	~84,224	~673,792
4	1.35	15.83	N/A
8	0.89	9.49	340.71
16	0.67	5.70	162.64

Performance on Beowulf

- **3D Scalability For Fixed Size Earthquake Mesh Problem**
  - Initial number of elements: 1316
  - Small 8 nodes of 14 available on Beowulf cluster

Processors	Level 1	Level 2	Level 3
Elements at AMR Level	~10,528	~84,224	~554,141
4	2.93	27.77	2229.76
8	2.51	17.67	1297.23

Performance on Cray T3E

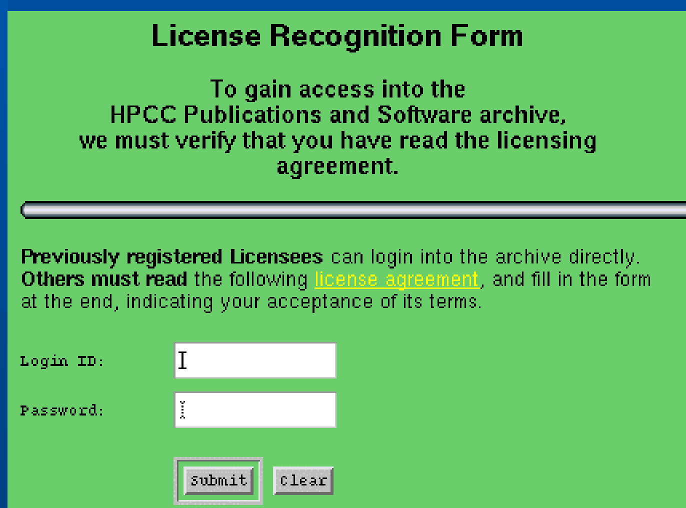
- **3D Scalability For Scaled Size Earthquake Mesh Problem (PEs and Problem Size)**
  - Initial number of elements: 1316
  - Scalability exists, but not proportional to the number of processors.

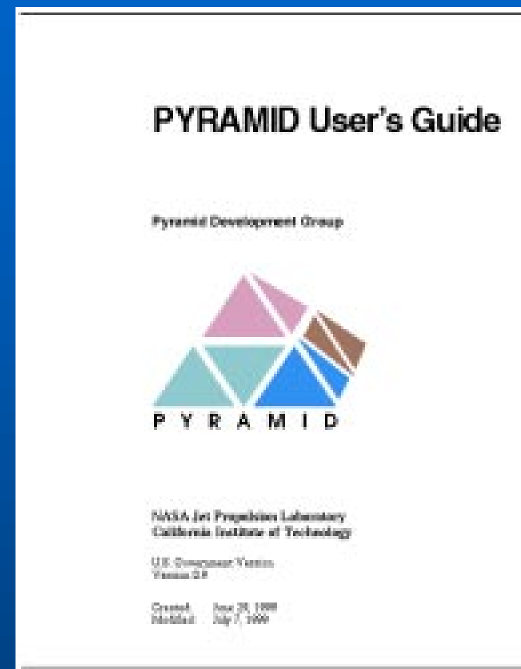
Processors	Level 1	Level 2	Level 3	Level 4
Elements at AMR Level	~10,528	~84,224	~673,792	~5,390,336
4	1.25	15.66		
32	0.67	4.10	83.23	
256	0.93	1.95	18.80	512.84

  - A non-fixed partitioning will cause the load balancing scheme to do unnecessary work
  - Resolving new element identifiers requires communication, although the partitioning is fixed

Software Availability

- **Demonstration Release from JPL HPCC Repository**
  - http://hpc.jpl.nasa.gov (Software Exchange)
  - http://hpc.jpl.nasa.gov/APPS/AMR





  - Completing a license registration form is required

  - Future work includes mesh coarsening and additional platform availability

HPCC/Earth and Space Sciences Project

PYRAMID: Parallel Unstructured Adaptive Mesh Refinement

PYRAMID

Modern... Simple... Efficient… Scalable...

Technology Description

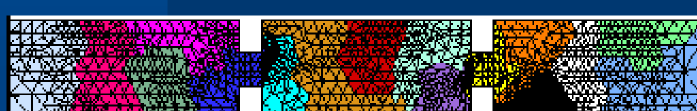
An advanced software library supporting parallel adaptive mesh refinement in large-scale, adaptive scientific & engineering simulations.

State-of-the-Art Design!

- Efficient object-oriented design in Fortran 90 and MPI
- Automatic mesh quality control & dynamic load balancing
- Scalable to hundreds of processors & millions of elements

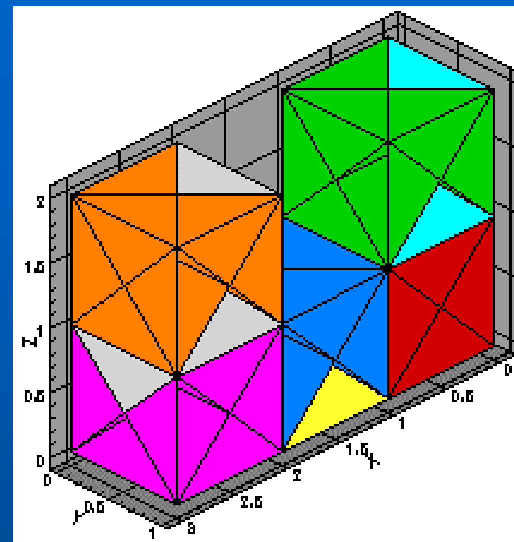
Application Arena

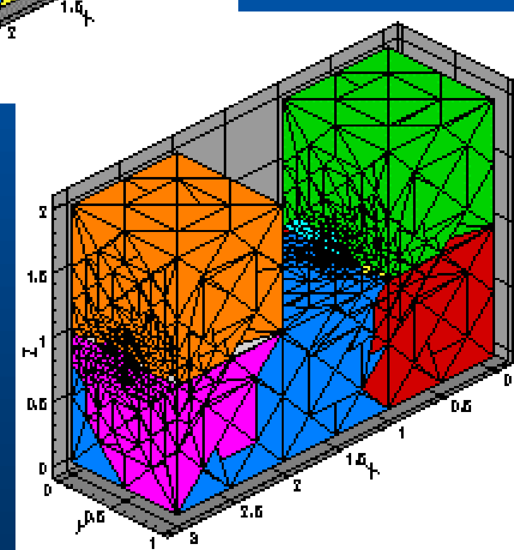
- Computer Modeling & Simulation Applications with complex geometry
- Electromagnetic and semiconductor device modeling
- Structural/Mechanical/Fluid dynamics applications



2D Adaptive Refinement on Waveguide Filter

John Z. Lou, Charles D. Norton, & Thomas A. Cwik
High Performance Computing Systems and Applications Group
http://www-hpc.jpl.nasa.gov/APPS/AMR





3D Successive Parallel Adaptive Refinements on Cray T3E

Accomplishments

- 3D library developed and tested on Cray T3E & Beowulf cluster
- System applied to a variety of meshes
- Publications and presentations

JPL