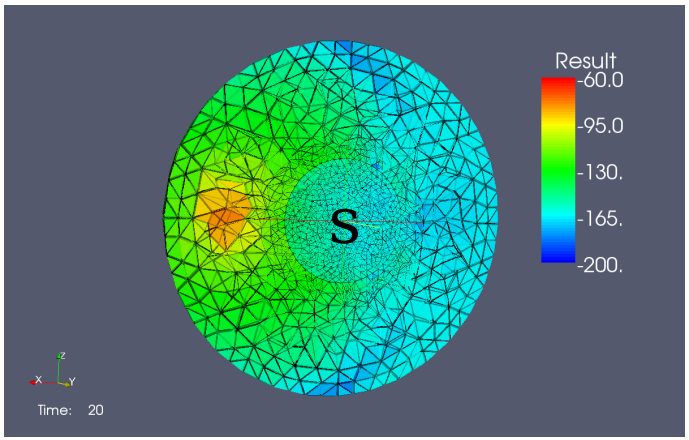
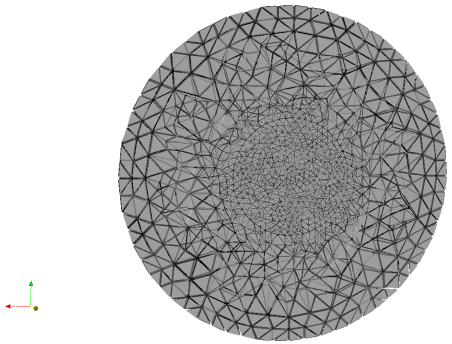
H5FED – HDF5 Based Finite Element Data Storage

Novel, Open Source, Unrestricted Usability

Benedikt Oswald (GFA, PSI) and Achim Gsell (AIT, PSI)





PAUL SCHERRER INSTITUT

PSI

From: B. Oswald, P. Leidenberger. *Electromagnetic fields scattered by subwavelength-sized tip - Finite element time domain (FETD) model with a dispersive Drude dielectric (Invited).* In 4th Workshop on Numerical Methods for Optical Nano Structures,Swiss Federal Institute of Technology, Zurich (ETHZ)

37th Speedup Workshop, ETH Zurich, 10-11 September, 2008

Some preliminary notes

- This talk presents work in progress
- The idea is to focus on very recent results and future plans
- Some of them will probably need some polishing in the future
- Based on H5Part & H5Block
- http://wwwvis.lbl.gov/Research/AcceleratorSAPP/index.html
- Originated within the AMAS group, Large Research Facilities

Contents

- **Motivation**
  - Common problem of all finite element codes: storage & retrieval of data
  - Application areas: finite element methods, visualization, geometry storage
- **Objectives**
- **Data Components**
  - Topological entities of a tetrahedral mesh
- **H5Fed Features**
  - Available & planned
- **API**
  - General, status, mesh, boundary
- **Internal File Organization**
  - HDF5 groups and datasets
- **Roadmap**
  - Features, availability

Motivation (1)

- **Common problem to all finite element codes**
- how to store & retrieve:
- Meshes, hierarchies thereof, degrees of freedom (DoF) ?
- *Time dependent* data, boundaries internal & external ?
- Material properties, boundary conditions ?
- Other data associated with the mesh & its components ?

Motivation (2)

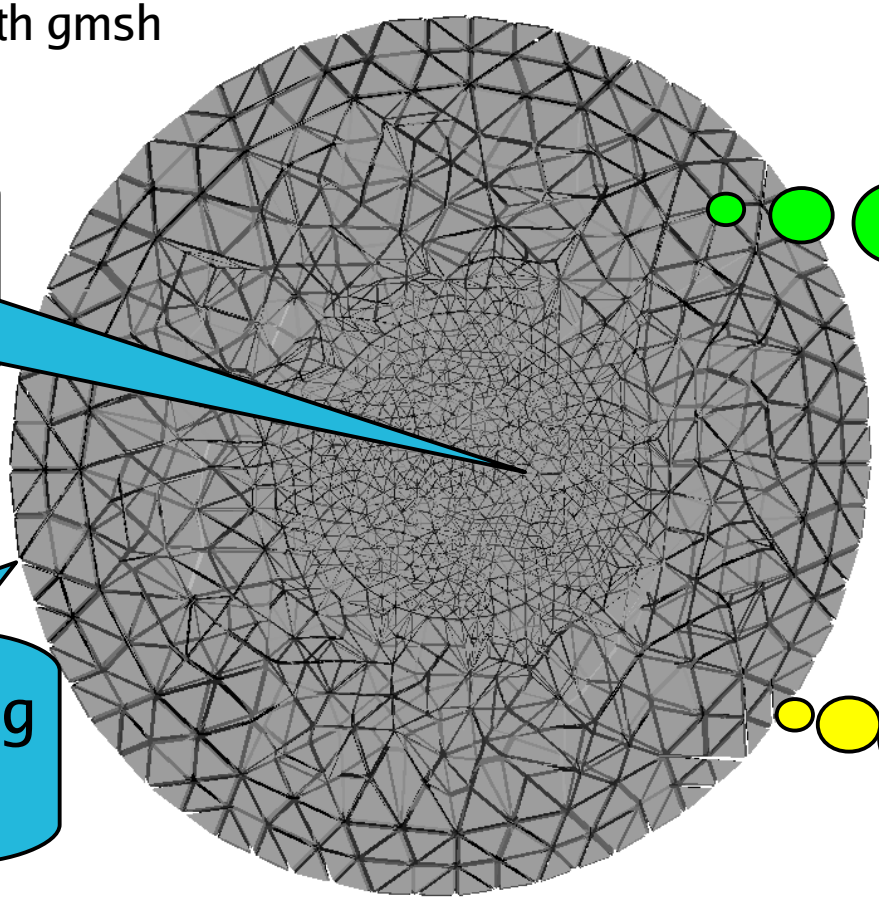
- **Avoid proprietary, code specific solutions**
- **Provide a solution acceptable to a larger community**
- **No knowledge of HDF5 and MPI required**

Existing Approaches ?

- **There are other approaches (Ensight)**
- **Concept and features not according to our needs**

Motivation (2) – storing & retrieving a tetrahedral mesh

- Gold sphere contained within vacuum, ca.80 kTet total
- mesh generated with gmsh



Radius of inner sphere = 70 nm

Radius of intermediate sphere=170nm

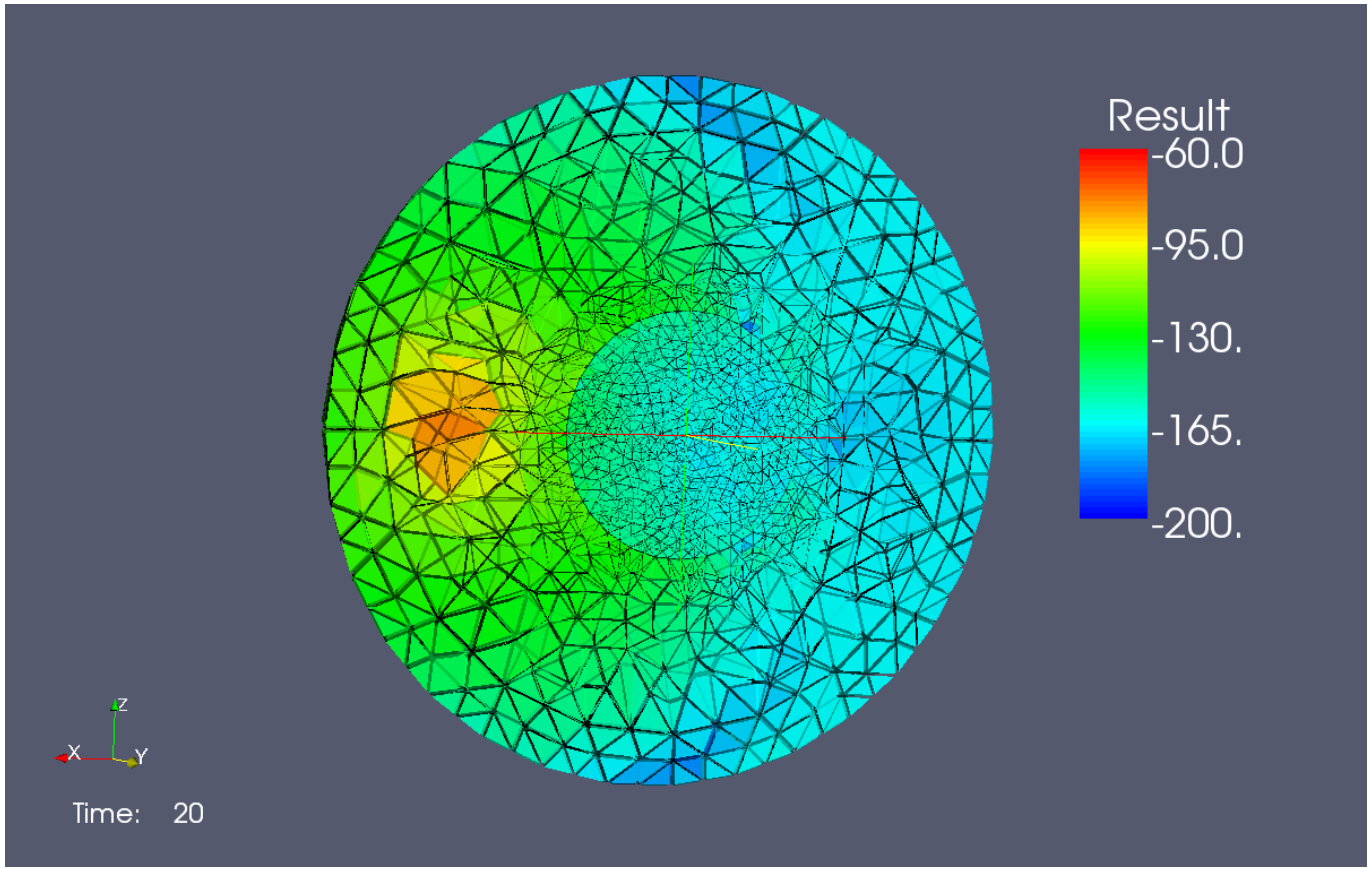
Radius of bounding sphere = 200nm

Boundary condition at outer boundary

From: B. Oswald, P. Leidenberger. *Electromagnetic fields scattered by subwavelength-sized tip - Finite element time domain (FETD) model with a dispersive Drude dielectric (Invited)*. In 4th Workshop on Numerical Methods for Optical Nano Structures,Swiss Federal Institute of Technology, Zurich (ETHZ)

Motivation (3) – storage of computed solution

Computed solution, expressed via degrees of freedom attached to tetrahedral edges, evaluated at different timesteps

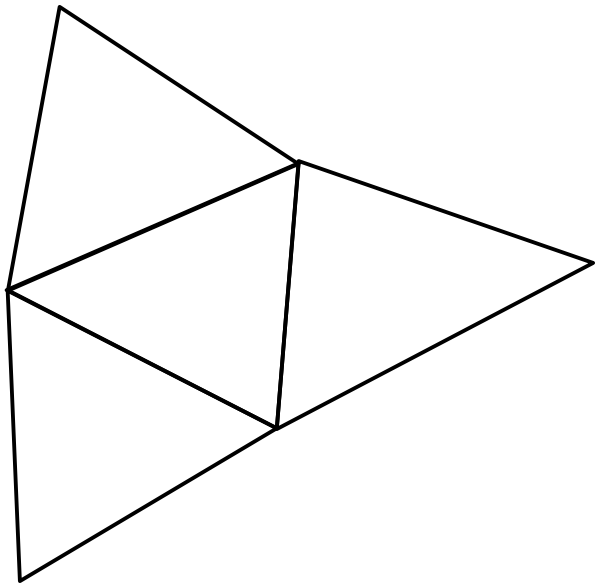


From: B. Oswald, P. Leidenberger. *Electromagnetic fields scattered by subwavelength-sized tip - Finite element time domain (FETD) model with a dispersive Drude dielectric (Invited).* In 4th Workshop on Numerical Methods for Optical Nano Structures, Swiss Federal Institute of Technology, Zurich (ETHZ)

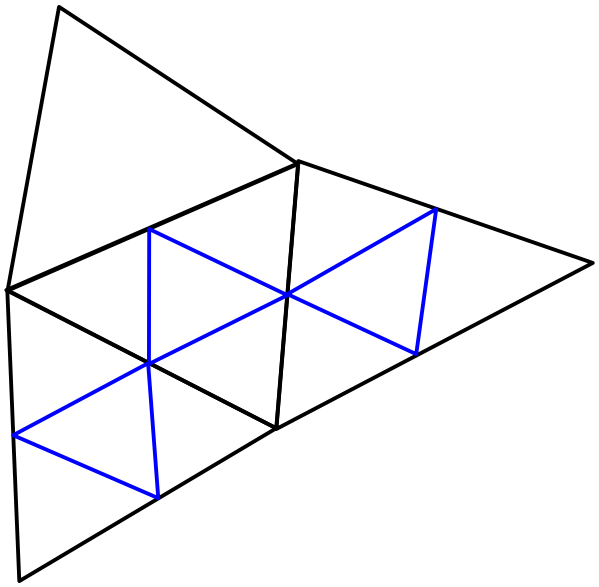
Objectives

- Provide the software library H5Fed for storing finite element data
- Base the library on the HDF5 standard (www.hdfgroup.org)
- Make sure H5Fed runs on Unix™ like platforms, including MacOS X
- Provide H5Fed for Cray/XT3/XT4, IBM/Bluegene, all MPI based cluster systems
- Seamless serial and parallel operation
- Establish H5Fed as an Open Source library
- Allow usage by both commercial companies and research establishments

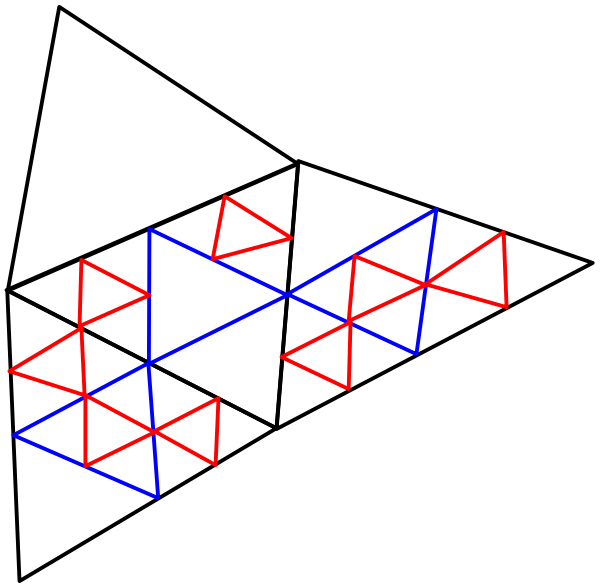
H5Fed Features (1) – Hierarchy of meshes



Level 0 (macro grid)



Level 1



Level 2

FE codes often use the concept of adaptive mesh refinement leading to a hierarchically nested meshes on different levels

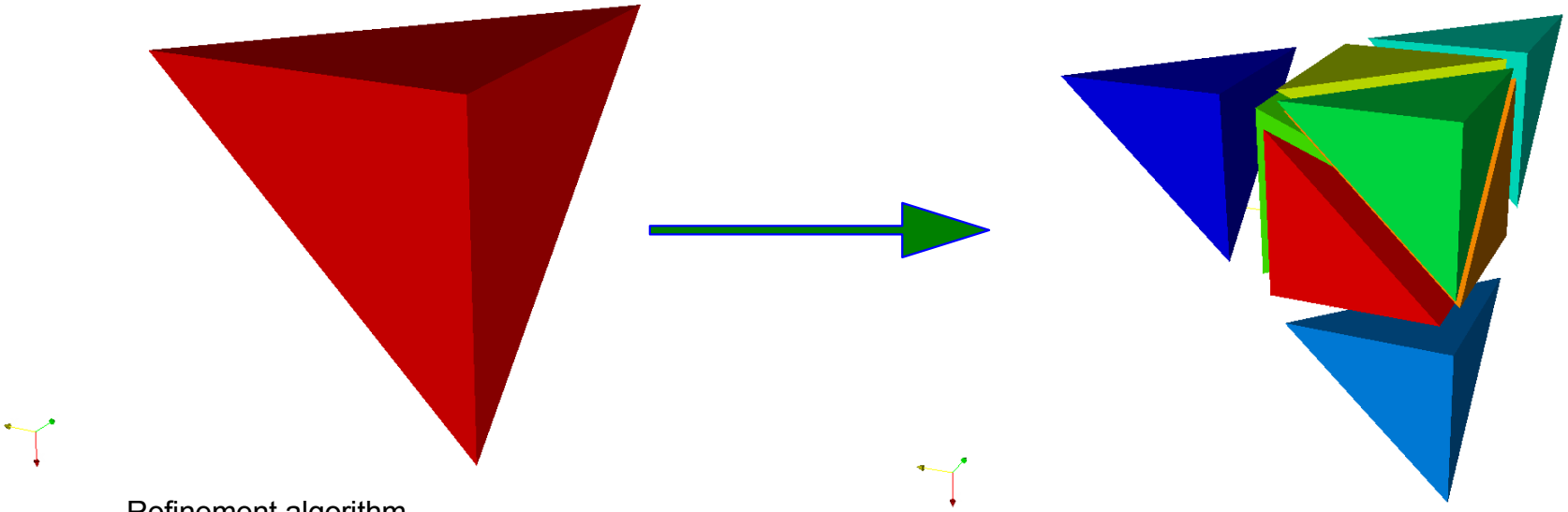
Infinite number of levels can be stored.

Coarsening is implemented on users request.

H5Fed Features (2) - Refinement

tetrahedron refinement through insertion of novel vertices

at the midpoints of the edges



Refinement algorithm

(1) create novel vertices at the midpoints of all edges
(2) every edge on level *k* becomes the father of two new edges on level *k+1*
(3) every face on level *k* becomes the father of 4 new faces on level *k+1*
(4) the tetrahedron on level *k* becomes the father of 8 new tetrahedra on level *k+1*
(5) build downward and upward adjacencies on level *k+1,* use a priori information available due to refinement algorithm

Nota bene: tetrahedra can be refined on a per element basis, i.e. hanging nodes can exist in the mesh

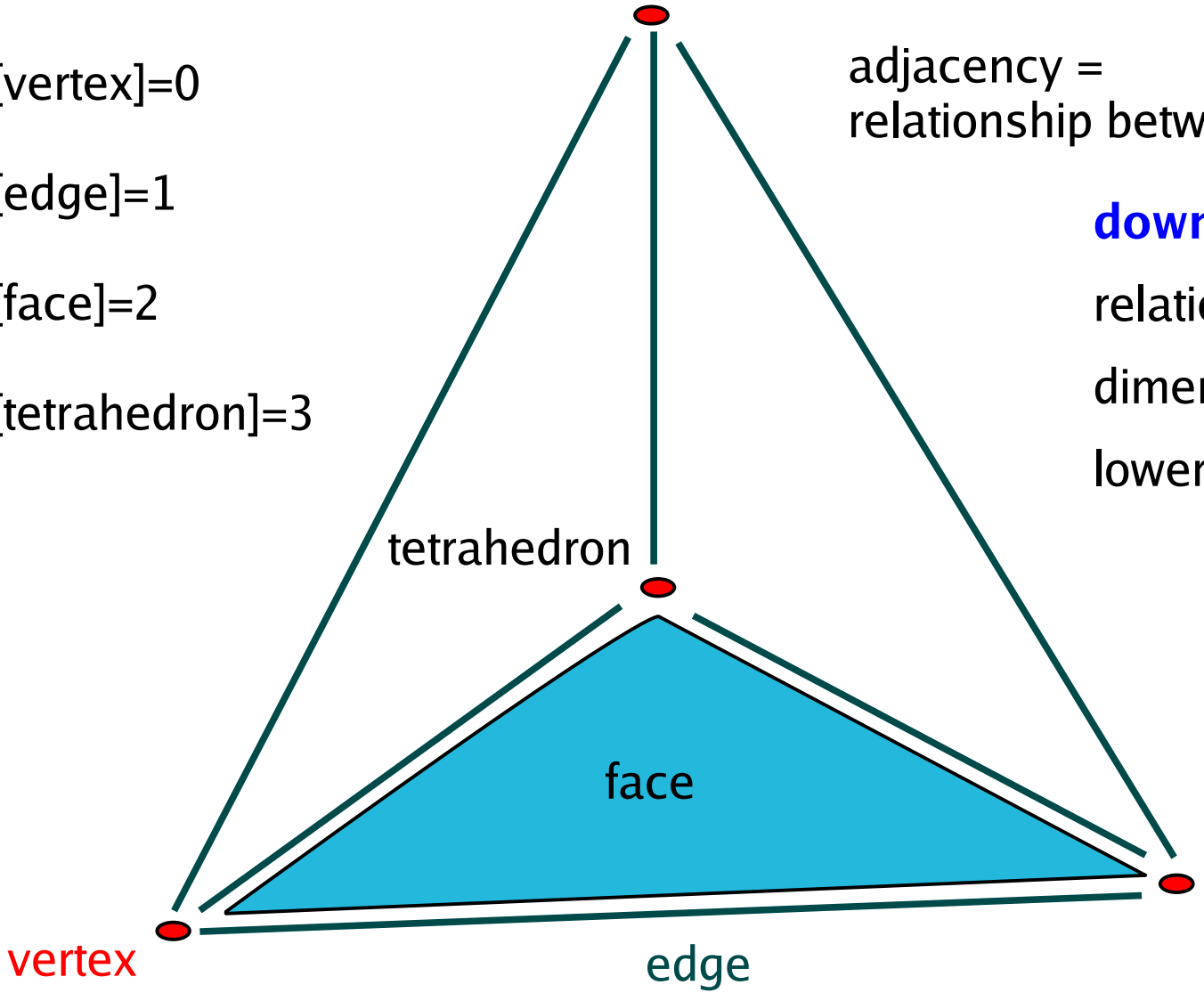
Data Components (1) – Topological Entities

dim[vertex]=0

dim[edge]=1

dim[face]=2

dim[tetrahedron]=3



adjacency =
relationship between topological entities

downward adjacency:
relationship from higher dimensional entity to lower dimensional top. ent.

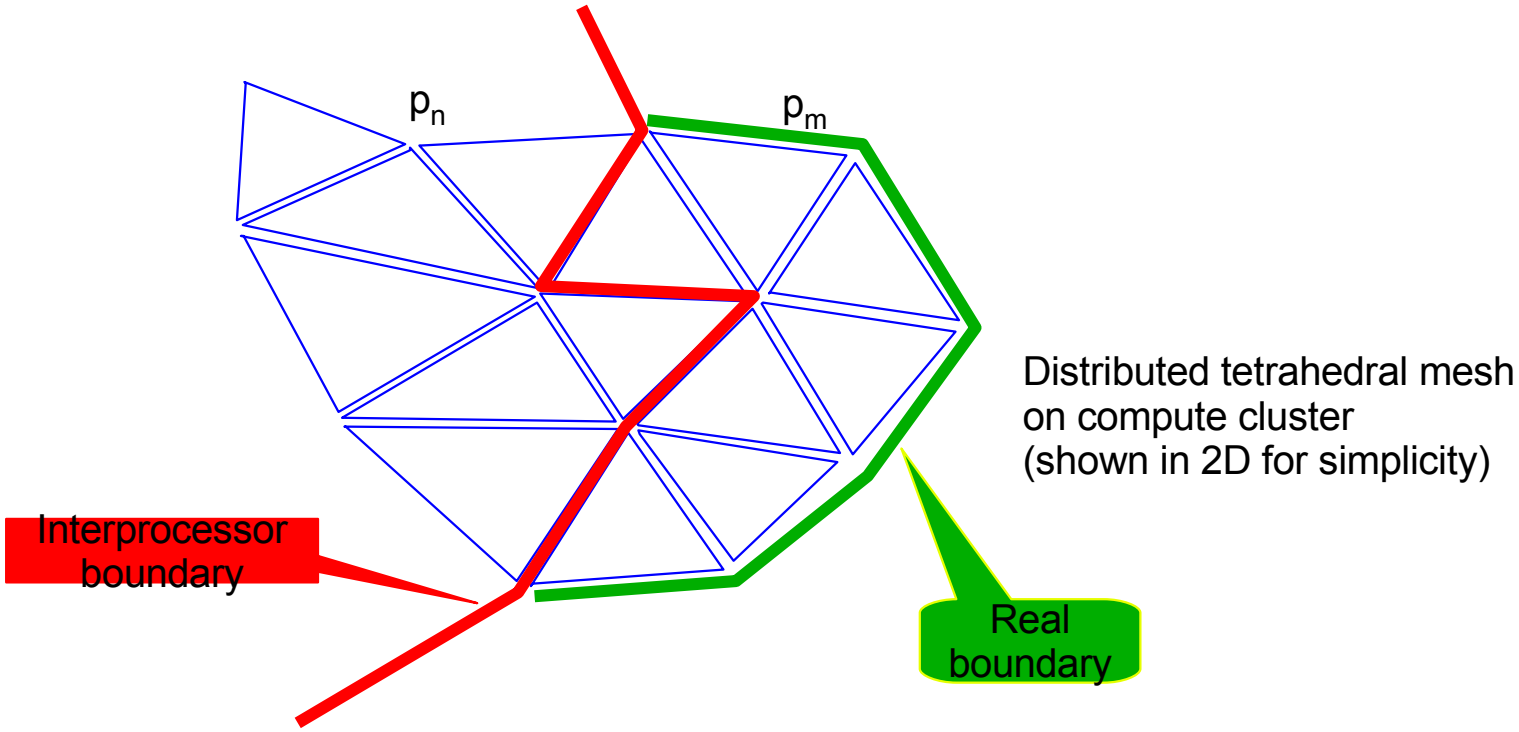
upward adjacency:
relationship from lower dimensional to higher dimensional top. ent.

H5Fed Features (3) – Data Associated with Topology

- Data associated with topology = tags
- Data types associated with topological entities of the mesh:
- Integer, double, complex (represented by 64 bit base type)
- Scalar, vector, tensor: technically realized as 1D arrays
- Interpretation of tag semantics left to the application

H5Fed Features (4) – Parallel Partitioning

- Partitioning of tetrahedral & triangle meshes at any time
- No restrictions on number of CPU's for which partitioning is calculated
- Use cases: compute on **many** CPU's, visualize on a **few** CPU's



H5Fed Features (5)

- Infinite number of meshes
- Infinite number of levels
- Infinite number of boundary meshes
- Each step is associated with a mesh and meshes can change from step to step
- Example: mesh refinement coupled to propagating pulse; mesh is coarsened after pulse

H5Fed – Restrictions

- No hanging nodes on level 0 grid (macrogrid)
- New vertices are always on existing edges
- Restricted to tetrahedral and triangle meshes
- Refinement of edges, faces and tetrahedra only via midpoints of edges
- The number of tets is limited to 268'435'455
- It is not possible to open more than one mesh at the same time, ditto for boundaries
- External and internal boundaries can not be geometrically changed in refinement levels
- As a consequence boundaries can be defined for the initial mesh only

API

1 General functions

2 Status inquiry functions

3.1 Mesh - general functions

3.2 Mesh - inquiry functions

3.3 Mesh - mapping functions

3.4 Mesh - definition functions

3.5 Mesh – retrieve & query functions

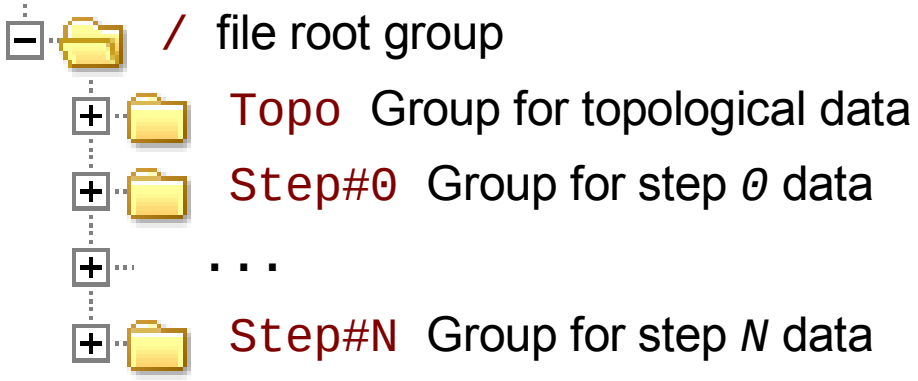
3.6 Mesh - boundaries

3.7 Adjacency query functions

3.8 Tag functions

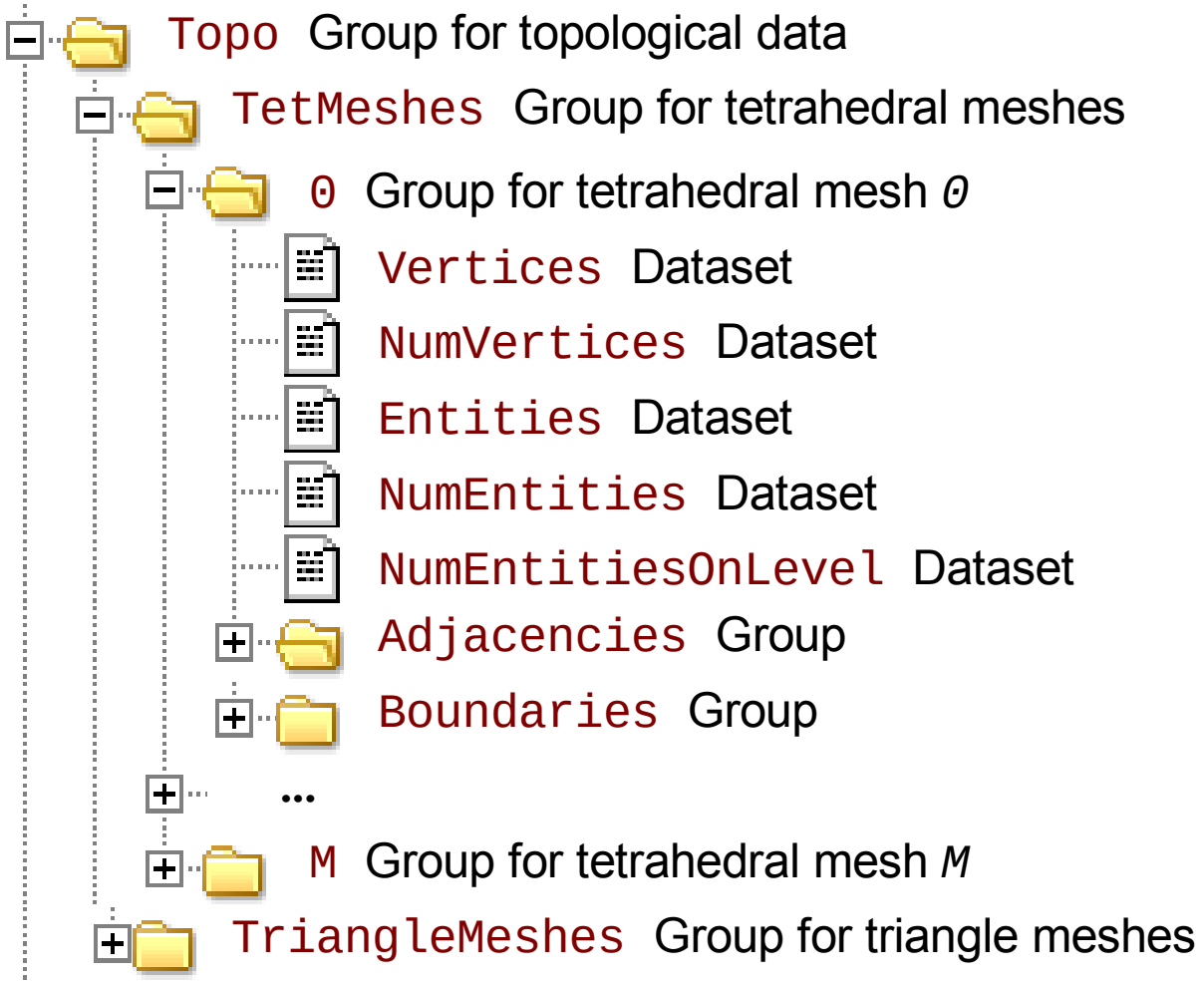
Internal HDF5 File Organization (1)

- **Root group**



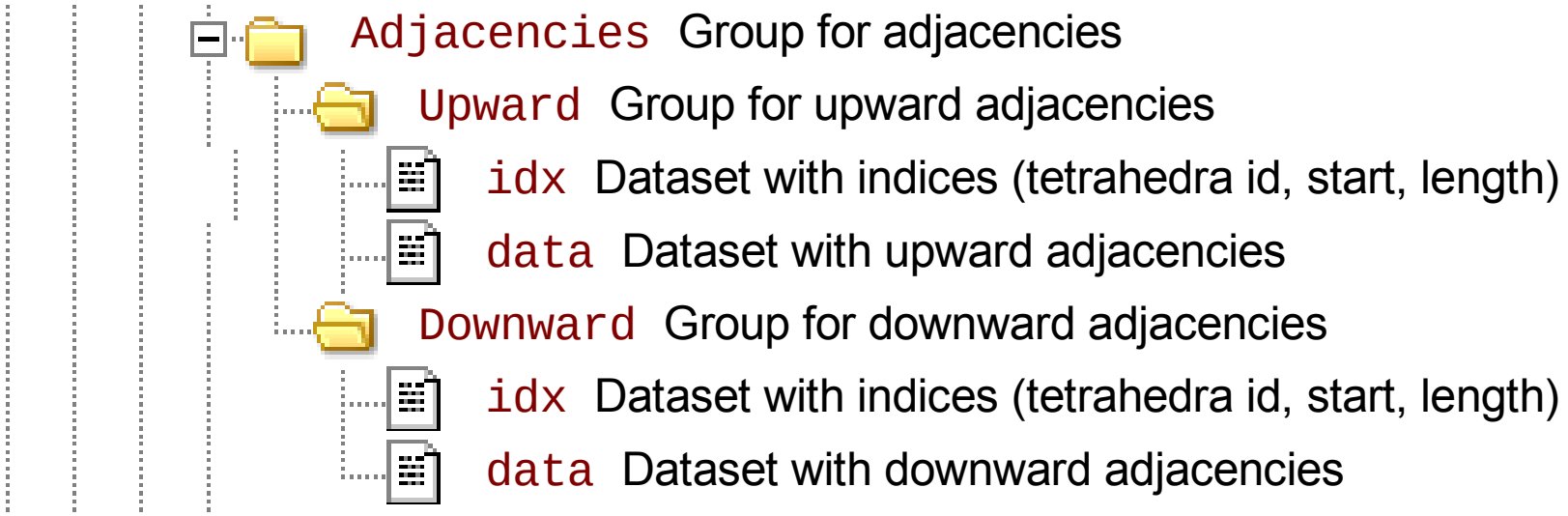
Internal HDF5 File Organization (2)

- **Definition of mesh hierachies**

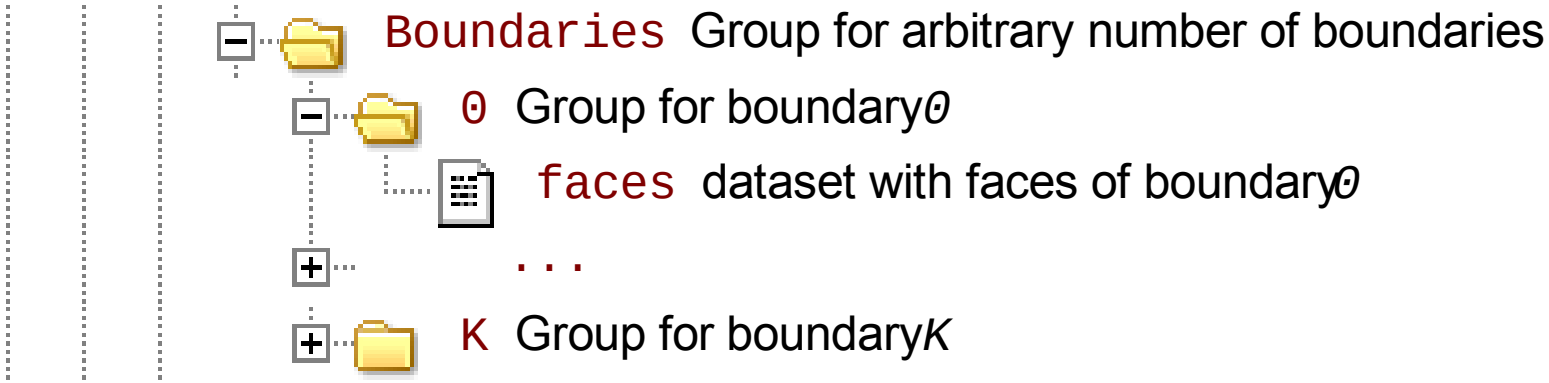


Internal HDF5 File Organization (3)

- **Up- and downward adjacencies**

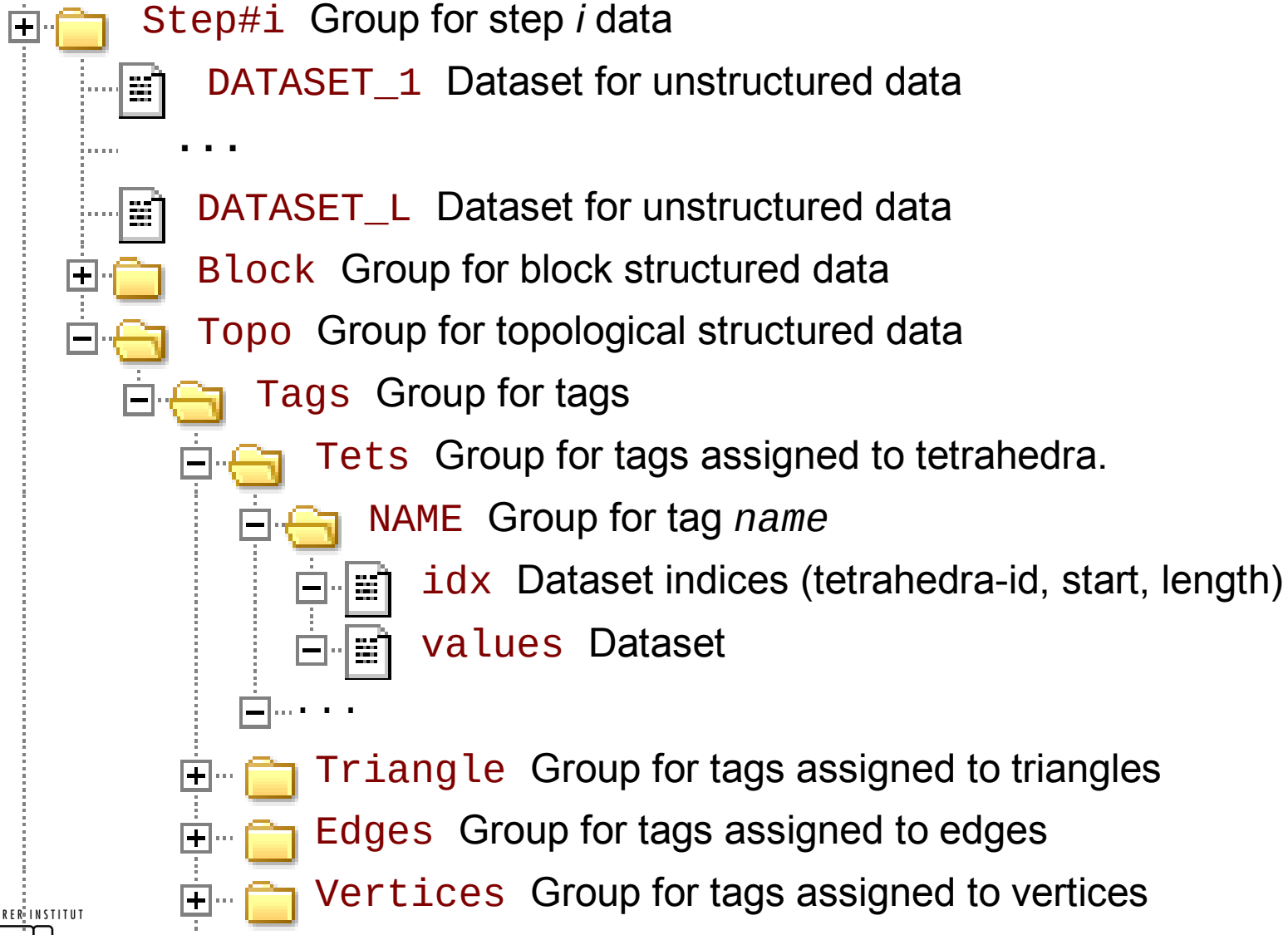


- **Boundary definitions**



Internal HDF5 File Organization (4)

- **(Time-)Step data**



Roadmap

1. serial writing and reading tetrahedral and triangle meshes
2. boundary meshes, read/write, serial: ongoing
3. tags, read/write
4. parallel write of tetrahedra and triangle meshes
5. parallel read of tetrahedra and triangle meshes
6. parallel write of tags
7. parallel read of tags
8. repartition mesh
9. adjacencies, serial
10. adjacencies, parallel
11. computation of boundaries (interprocessor, external)
12. parallel write of boundary meshes
13. parallel read of boundary meshes

Acknowledgements

- The **Speedup** workshop organizers for the opportunity to have this talk
- The inventors and creators of H5Part: http://wwwvis.lbl.gov/Research/AcceleratorSAPP/index.html

Thanks for your attention!

API (1)

1 General functions

H5OpenFile, Open H5 file, H5CloseFile, Close H5 file

H5DefineStepNameFormat, H5GetStepNameFormat,

H5SetStep, H5GetStep, H5StartTraverseSteps,

H5TraverseSteps

2 Status inquiry functions

H5GetNumNodes, H5GetNumSteps, H5HasStep

3.1 Mesh - general functions

H5FedOpenMesh, H5FedCloseMesh,

H5FedSetMeshLevel, H5FedAssignMeshToStep

API (2)

3.2 Mesh - inquiry functions

H5FedHasMesh, H5FedGetNumMeshes,H5FedGetNumLevels,

H5FedGetLevel, H5FedGetNumTets, H5FedGetNumTetsCNode

H5FedGetNumTetsTotal, H5FedGetNumTriangles,

H5FedGetNumTrianglesCNode, H5FedGetNumTrianglesTotal,

H5FedGetNumEdges, H5FedGetNumEdgesCNode,

H5FedGetNumEdgesTotal, H5FedGetNumVertices,

H5FedGetNumVerticesCNode, H5FedGetNumVerticesTotal

3.3 Mesh - mapping functions

H5FedMapTet2GlobalID, H5FedMapTriangle2GlobalID, H5FedMapEdge2GlobalID,

H5FedMapVertex2GlobalID, H5FedMapTet2LocalID, H5FedMapTriangle2LocalID,

H5FedMapEdge2LocalID, H5FedMapVertex2LocalID, H5FedMapTetGlobalID2LocalID,

H5FedMapTriangleGlobalID2LocalID, H5FedMapEdgeGlobalID2LocalID,

H5FedMapVertexGlobalID2LocalID

API (3)

3.4 Mesh - definition functions

H5FedAddMesh, H5FedAddLevel,

H5FedAddNumEntities, H5FedAddNumVertices,

H5FedStoreTet, H5FedStoreTriangle,

H5FedStoreVertex

3.5 Mesh – retrieve & query functions

H5FedGetTet, H5FedGetTriangle, H5FedGetEdge, H5FedGetVertex,

H5FedGetLocalTet, H5FedGetLocalTriangle, H5FedGetLocalEdge,

H5FedGetLocalVertex, H5FedStartTraverseTets, H5FedTraverseTets,

H5FedStartTraverseTriangle, H5FedTraverseTriangles, H5FedStartTraverseEdges,

H5FedTraverseEdges, H5FedStartTraverseVertices, H5FedTraverseVertices,

H5FedGetVertex

API (4)

3.6 Mesh - boundaries

H5FedAddBoundary, H5FedOpenBoundary,

H5FedCloseBoundary, H5FedAddNumBoundaryFaces,

H5FedGetNumBoundaryFaces, H5FedGetNumBoundaryFacesCNode,

H5FedGetNumBoundaryFacesTotal, H5FedStoreBoundaryFace,

H5FedStoreBoundaryFaceGlobalID, H5FedStoreBoundaryFaceLocalID,

H5FedStartTraverseBoundaryFaces, H5FedTraverseBoundaryFaces