Trilinos Pamgen Parallel Meshing Library: Enabling Billion Element Simulations

Pamgen, a library within the Trilinos Project, produces, in parallel, during simulation code execution, multi-billion element hexahedral or quadrilateral meshes of simple topologies. It facilitates easy setup and execution of performance, scaling, and convergence studies for massively parallel (more than 1000 processors) unstructured finite element simulations. Because it generates and decomposes meshes in parallel during execution, it circumvents the limitations and burdens that serial mesh generation imposes on simulations. Pamgen supports meshes having up to 2^{64-1} (9,223,372,036,854,775,807) nodes.

The Pamgen library provides functions to mimic the process of reading mesh data from a processor specific mesh file through a file access API. During a parallel simulation, each processor provides Pamgen with a terse specification of the desired finite element mesh, the total number of processors, and the index of the local processor. At that time, the Pamgen library calculates the mesh for the local processor and its relation to mesh on adjacent processors. The processor may then call functions that return descriptions of its local the finite element mesh and the relationship of that mesh to the mesh on adjacent processors. The specification of the finite element mesh is generally read from an input file by the client code and passed to Pamgen as a "**c**". The access functions which return mesh descriptions are a subset of those provided by the ExodusII and NemesisII mesh data file APIs with "im_" (for inline mesh) added as a prefix to the function names.

The Pamgen library relies on several assumptions in order to generate meshes in parallel.

All Processor are Identical (except for their ids): If it can be assumed that each processor will produce the same results for identical calculations then the local processor can calculate anything it may need to know concerning mesh on a neighboring processor.

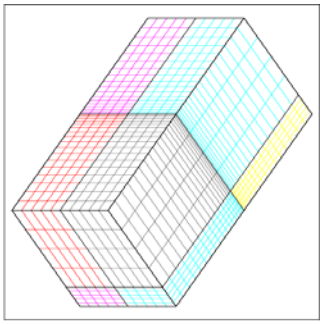
Communication is Unavailable: Assuming communication is unavailable guarantees the parallel scalability of the library. All calculations are made locally without resorting to inter-processor queries.

Meshes Consist of One or More Structured Block Topologies: Limiting the topologies of the meshes that can be generated enables algebraic calculations of all mesh topologies .

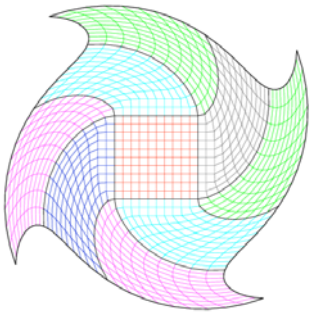
Although the Pamgen library supports a limited number of mesh topologies (cubes, hollow cylinders, and solid cylinders) it provides significant flexibility in geometry, boundary condition, and decomposition specification. It allows boundary condition specification on sub-regions of the surface and interior of the mesh. These are provided to the client code as lists of element faces or nodes. It allows multiple "blocks" of elements accessible by sequential ids. It allows flexible manipulation of the mesh geometry by evaluating a user provided "**c**" routine to calculate the coordinates of each node based on its "original" coordinates. It supports several decomposition strategies including a random strategy that produces decompositions to test communication subsystems. It allows several methods for adjusting or "grading" the mesh resolution in locations of the mesh.

The first example shows a brick topology that consists of two “blocks” of mesh in each topological direction. The mesh size within the blocks has been specified to produce an area of refinement between the two sets of four blocks. The second example illustrates the flexibility of the user supplied geometric transformation that allows for geometric complexity of the resulting mesh despite the requirement of topological simplicity.

```
mesh
 brick
  numz 2
  zblock 1 2. interval 5
  zblock 2 8. interval 4
  numx 2
  xblock 1 5.0 interval 5
  xblock 2 5.0 interval 5
  numy 2
  yblock 1 10. first size 1. last size .1
  yblock 2 10. first size .1 last size 1.
 end
end
```



```
mesh
 rectilinear
  nx = 10
  ny = 10
  bx = 3
  by = 3
end
 user defined geometry transformation
  "
  double r = sqrt(inxcoord*inxcoord
+inycoord*inycoord);
  double theta = atan2(inycoord,inxcoord);
  if(r > 0.5)
  {
  theta = theta + (3.14159 / 4.0)*((r-0.5)/0.5);
  outxcoord = r*cos(theta);
  outycoord = r*sin(theta);
  }
  "
 end
end
```



verview of Trilinos, SAND2003-2927, Sandia National

Hensinger, D. M, and Drake, R. R. et al. (2008). *Pamgen, a Library for Parallel Generation of Simple Finite Element Meshes*, SAND2008-1933, Sandia National Laboratories, Albuquerque, NM.

L. A. Schoof, and V. R. Yarberry. (1992) *EXODUS II: A Finite Element Data Model*. SAND1992-2137, Sandia National Laboratories, Albuquerque, NM, November 1995.

G. L. Hennigan, M. St. John, and J. N. Shadid. (1998) *NEMESIS I: A set of functions for describing unstructured finite-element data on parallel computers*. Technical report, Sandia National Laboratories, Albuquerque, NM, May 1998.