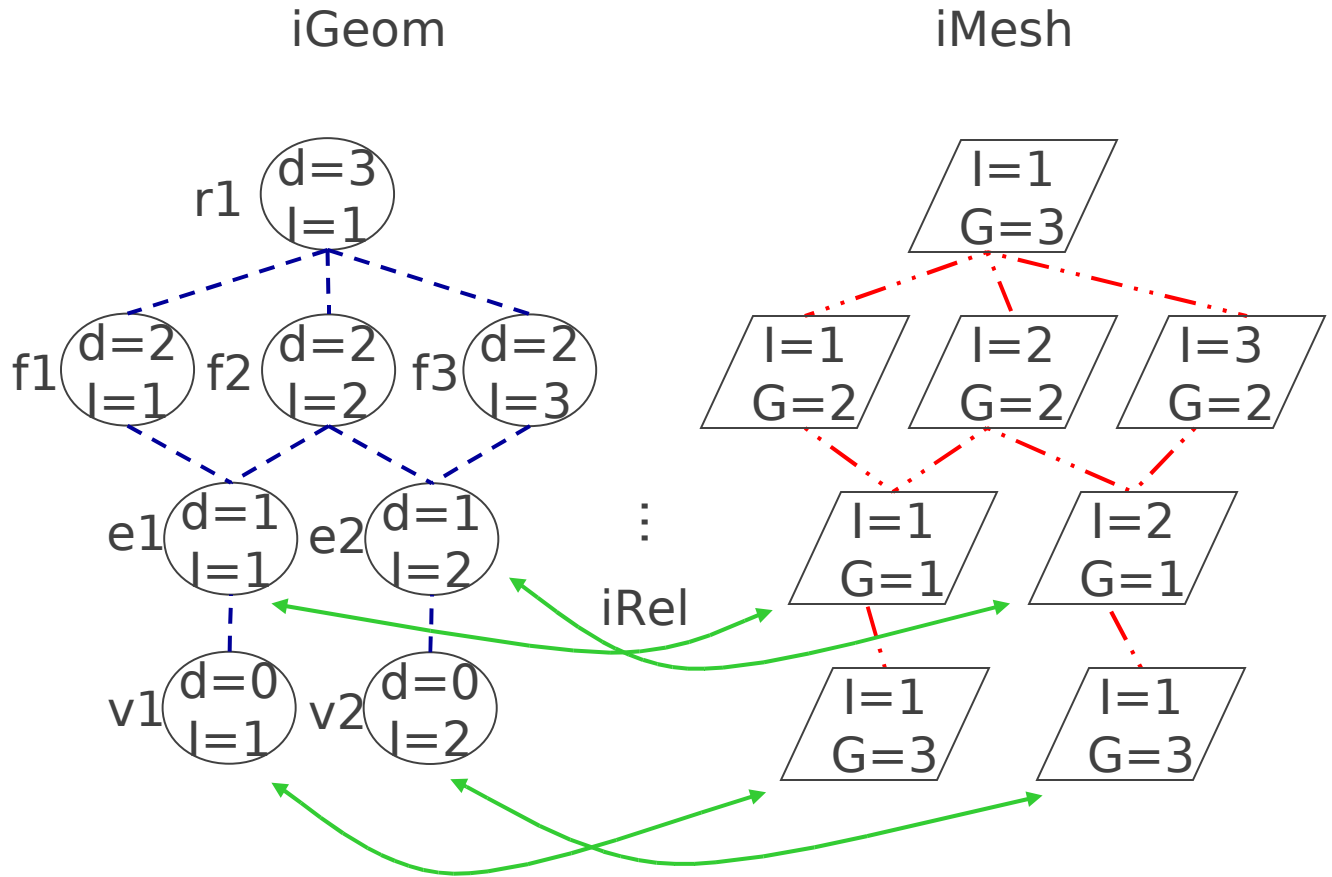
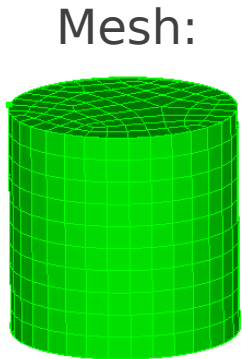
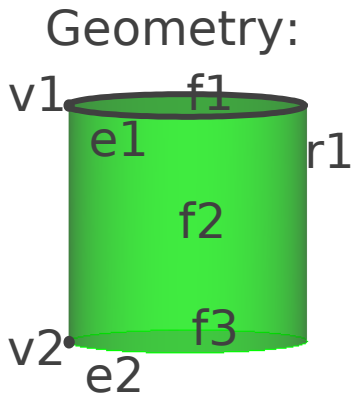


MeshKit Design & Coding Practices

Geometry-Mesh-Relations



= entity d = iGeom entity type G = "GEOM_DIMENSION" tag
 = entity set l = "GLOBAL_ID" tag
- - - = iGeom adjacency . . . = parent/child set relationship



MeshKit Algorithm Execution

- Pre-mesh
 - Mesh my boundary
- Mesh
 - Gather mesh from boundary
 - Generate mesh (a)
 - Generate mesh (b)
 - ...
- Post-mesh
 - Global smooth
 - Assign to iMesh*

* - to use other tools, e.g. smoothing, mesh should already be in iMesh; maybe this step binds it to the GEOM_DIMENSION set



MeshKit Algorithm Implementation

- Input/output should **ALWAYS** be in terms of iMesh sets
 - Even during initial development
 - Even when using external meshing algorithm implementation
 - If you must input from file to 3rd party tool, implement a writer that writes to a temporary file that is then read by the tool (called at the start of the tool execution)
 - If you must output to a file from a 3rd party tool, implement a reader that then reads the mesh from the temporary file into MOAB (called at the end of the tool execution)
 - Makes it possible to call as a sequence of operations operating on MOAB/iMesh state
 - read from facets - decimate - instantiate iGeom - quad mesh - smooth - read into analysis
 - edge mesh - trimesh with Triangle - quad mesh with CAMAL - extrude
 - edge mesh - trimesh with CAMAL - quad mesh with JAAL- extrude
- Implement as class, driver program
 - Class:
 - Constructor: input iGeom/entity, iMesh/entity set handles
 - Execute(): performs algorithm execution
 - Algorithm-specific settings and input using member functions and variables
 - Driver program:
 - Handle file-based IO here, NOT IN EXECUTE FUNCTION
 - Options that input to class member functions/variables for algorithm options



MeshKit Algorithm Implementation (cont)

- Eventually: handle pre-defined sets
 - Copy sets: entities in these sets have copies put in new sets
 - Expand sets: entities in these sets have copies added to these sets
 - Extrude sets: entities in these sets have higher-dimensional extrusions put into new sets
 - Refine sets? Or use copy/expand sets?
 - See CopyMesh class for example usage, handling
- Build system
 - If algorithm is just a couple of classes, put in top-level algs subdirectory in MeshKit
 - If not, make subdirectory under algs and put files there
 - Either way, add files to algs/Makefile.am, in libMKalgs_la_SOURCES and libMKalgs_la_HEADERS
 - Driver program can be either in main implementation file, segregated by a `#define`, or in a separate driver program file
 - See CopyMesh.cpp for example of `#define`, CutCellMesh_test.cpp for separate driver
 - Add Makefile.am input for test program too
- In process of defining better overall class/library design, stay tuned

