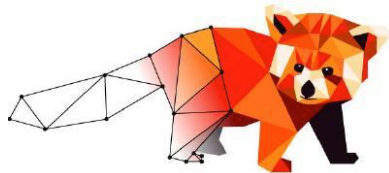


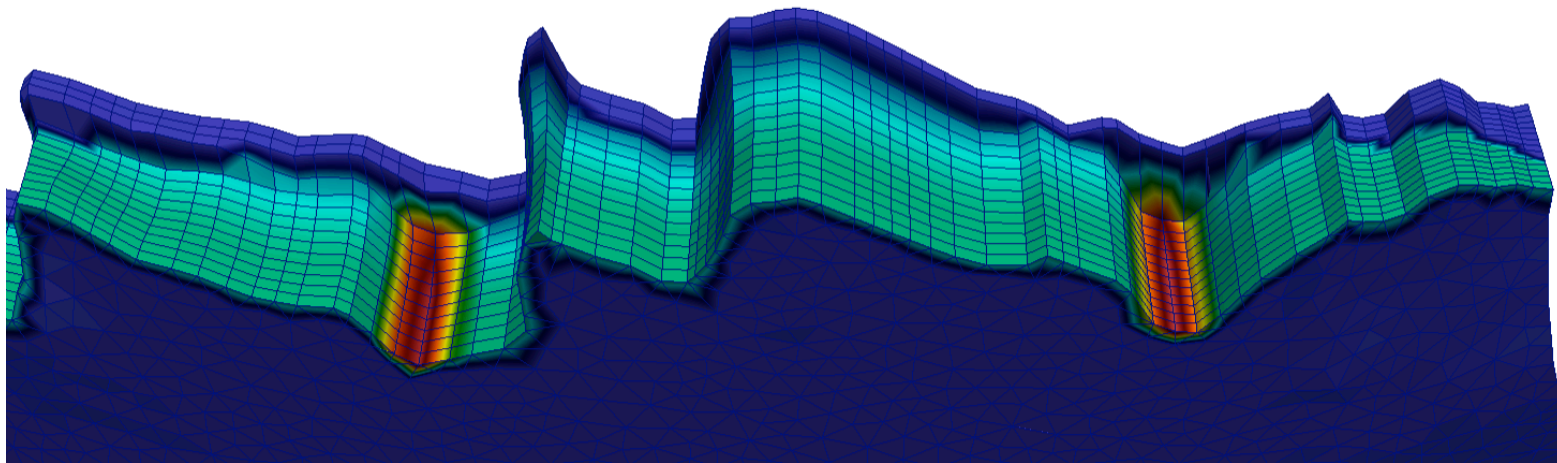
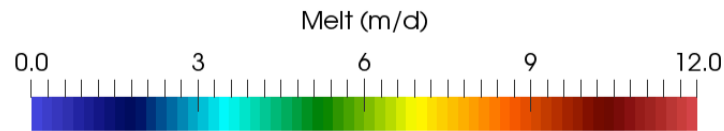
3D Remeshing using MMG & Zoltan



Upgrade
your meshes

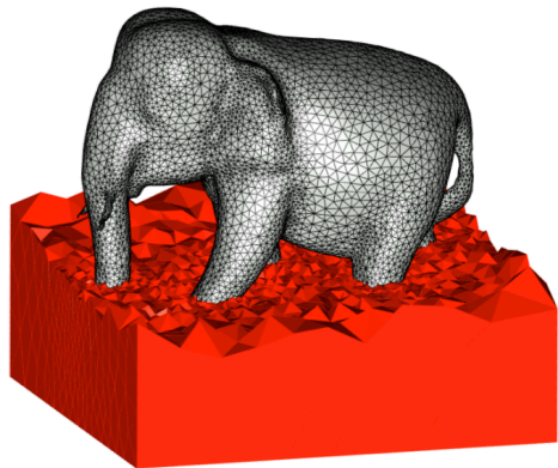
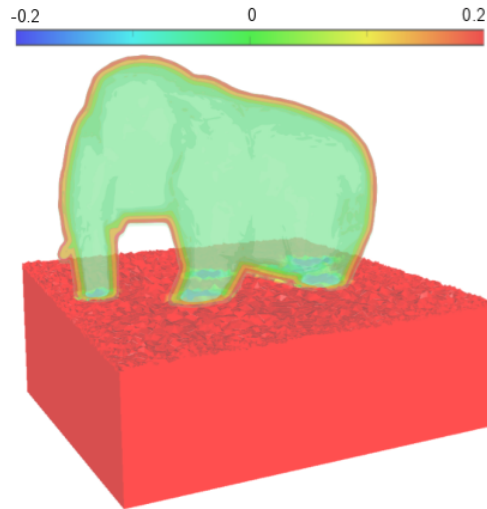
Motivation

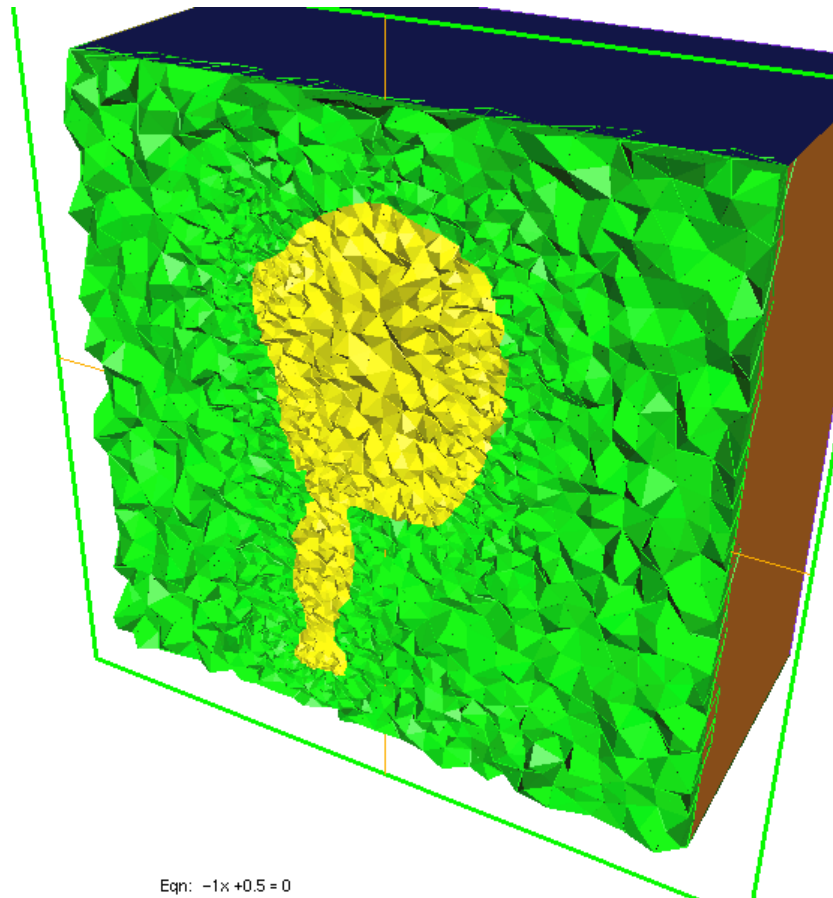
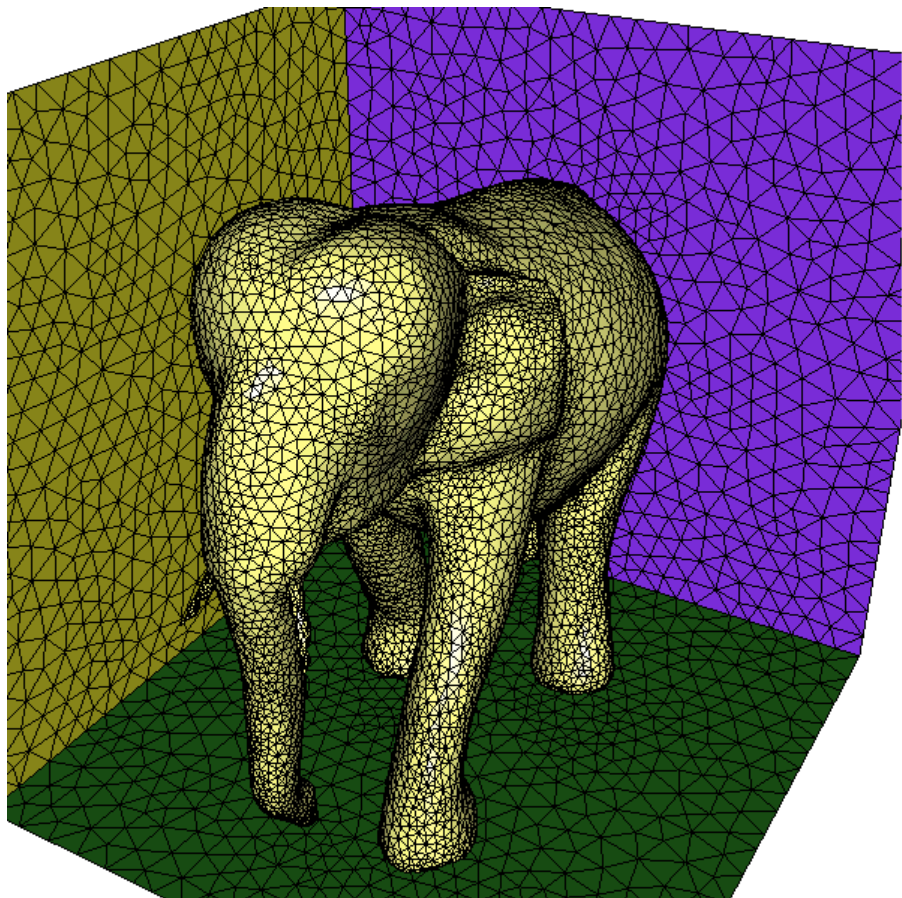
- Elmer/Ice 3D depends on vertically extruded meshes
- This can be severely limiting...
- ... particularly when BCs are non-vertical
- But without vertical extrusion, (re)meshing is more complex.



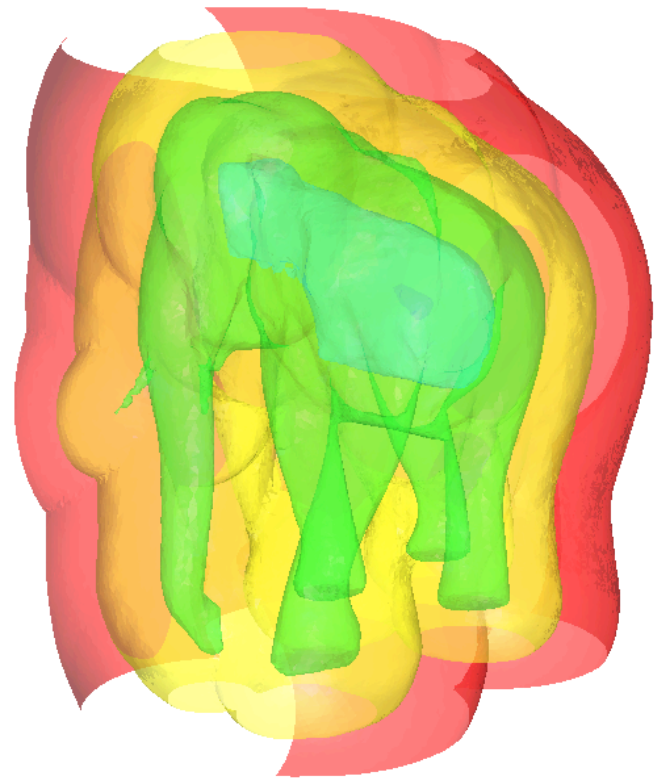
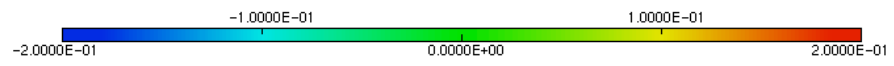
Capabilities of MMG3D

- Anisotropic adaptation - *grounding line, adaptive remeshing. (Elmer/Ice depends on anisotropic meshing!)*
- ‘Implicit surface’ meshing - *calving events*
- Lagrangian deformation - *MeshUpdate++*
- Elephant generation





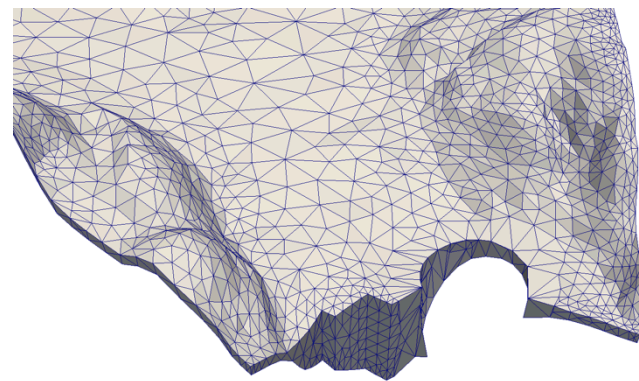
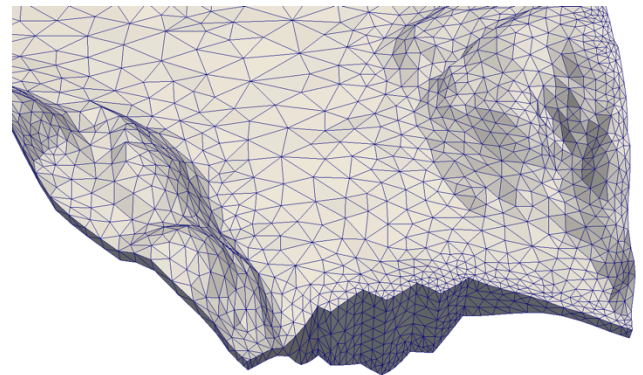
Eqn: $-1x + 0.5 = 0$



Implementing Calving using MMG3D

Designed to be a drop-in replacement for the CalvingRemesh subroutine (CalvingRemesh.F90)

1. Define a level-set surface (a 3D zero contour) for the calving event
2. Pass the mesh, some parameters & the level set to MMG3D
3. MMG3D cuts out the calving event and refines the mesh near the new surface
4. Repartition the mesh (Zoltan), interpolate field variables & continue.



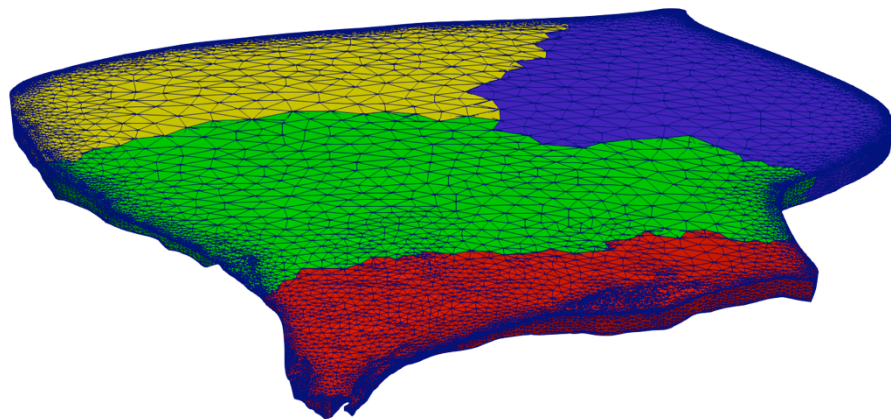
Zoltan - boring but important

Mesh adaptation -> loss of load balance

Zoltan: parallel repartitioning - parallel partitioning improvement in parallel...

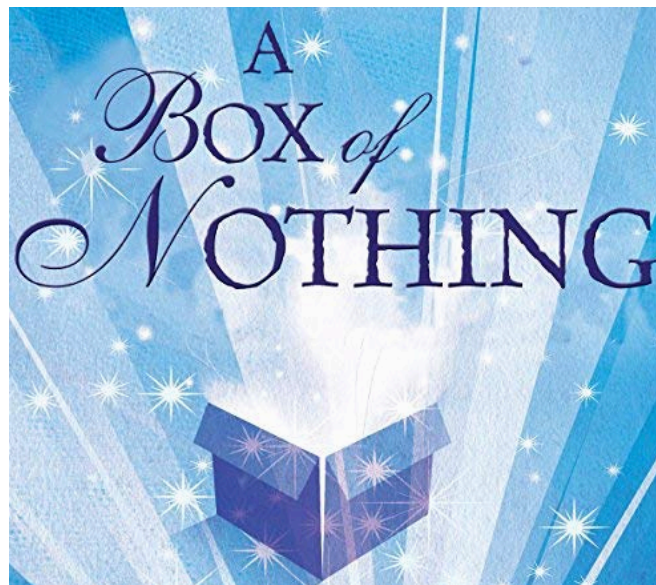
Due to a bug in OpenMPI 2.0, you might get weird errors. Options:

- Use libzoltan from Trilinos
- Don't use OpenMPI
- Get in touch with Joe for Zoltan patch
- Elmer distributes patched Zoltan?



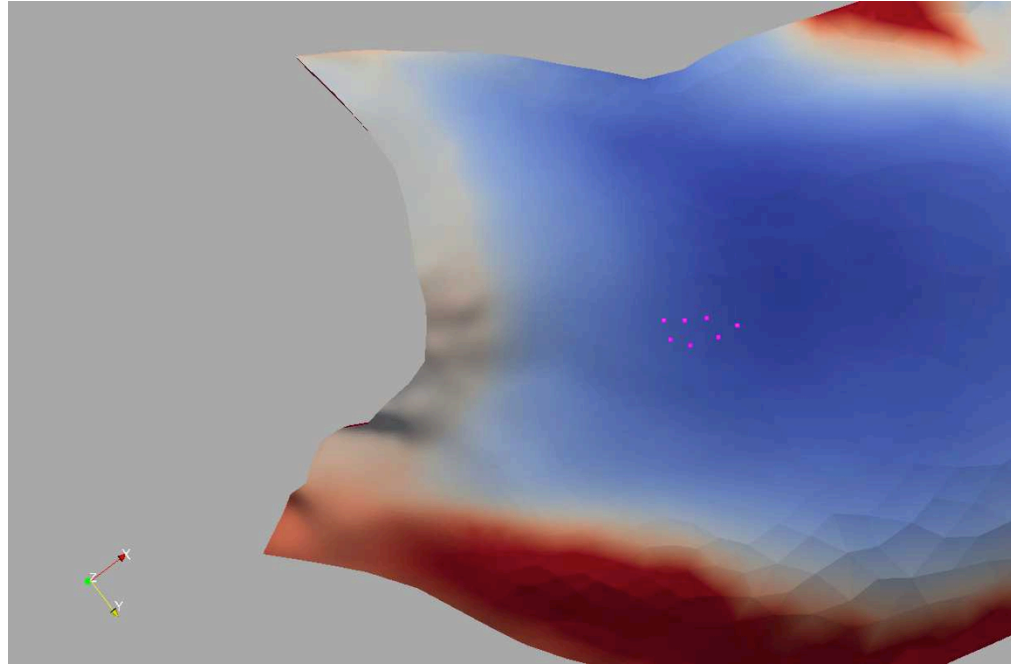
Final Remarks

- Out-the-box, *nothing* is currently properly implemented, but much of the important code is written
- Joe will work on this in 2019, but for now, do it yourself
- MMG can do much more than just calving! (see Fabien's work, for example)



Inversion Mesh (In)dependence

- By default, contributions to DJDBeta are highly dependent on element size
- 'Mesh Independent = Logical True' in optimize_m1qn3 solver section
- Added code presents the possibility for user-provided weights (but this isn't yet implemented)



Non-determinism

- Many (all?) parallel solvers suffer very minor non-deterministic behaviour to varying degrees
- Any repeated remeshing operation acts to amplify initially tiny changes
- Particularly evident in calving simulations
- Is this a problem? Is it worth trying to develop a solution?