Computational Mechanics: HDF Parallel I/O Implementation In Warp3D

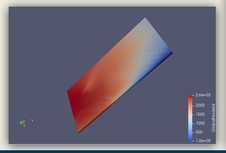
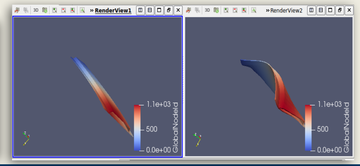
Daniel Pledger (UTK), **Carlos Estrada** (NMSU), **Rocco Febbo** (UTK)
Mentors: **Tim Truster** (UTK), **Kwai Wong** (UTK)

Warp3D

Description: Open source code for 3D nonlinear analysis of solids primarily for fatigue and fracture simulations with static, impact, dynamic and thermal loadings

Purpose: Analyze material mechanics under stresses to improve designs with OpenMP Symmetric MultiProcessing (SMP)

Code: Written largely in Fortran (Late 1980's - Cur.)

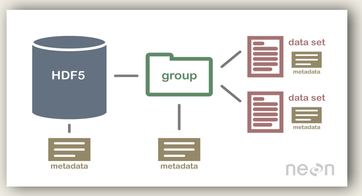




HDF5

HDF - Hierarchical Data Format

Description: File Type .h5

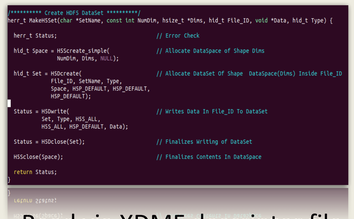
Parallelism: Attributes are naturally separated



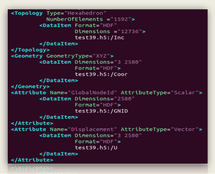
Purpose: Store data by efficient and compact means using a hierarchical format (similar to Unix)

Code: Written in C, C++, Fortran

Team HDF5 Implementation -->



<-- Read via XDMF descriptor file



Project Overview

Goal: Use front end programs to add complexity for more accurate-to-life models and utilize HDF5 functions and format to store output from Warp3D after each computational step to be rendered by Paraview in parallel with OpenDIEL working with python GUI

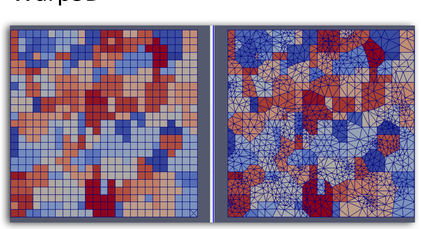
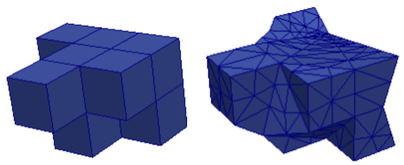
Milestones:

- Familiarize team with front and back end user interface
- Research program documentations and source codes
- Develop additional functions within source code
- Create complex mesh and surface elements in simulated objects
- Implement parallelism in I/O processing via HDF5 data storage format

Front End

Measures Taken:

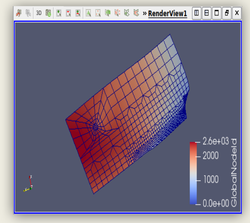
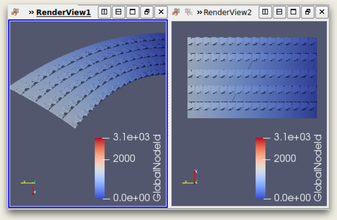
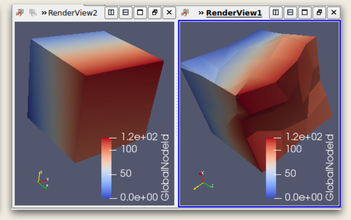
- Setup and ran simulations using Warp3D
- Developed workflow to input GMSH mesh files into Dream3D to output voxel grain structures
- Created more realistic grain structures with tetrahedral meshes using Voxel2Tet
- Developed workflow to view models in Paraview
- Wrote Matlab code that reads Voxel2Tet model files for simulation using Warp3D

Application

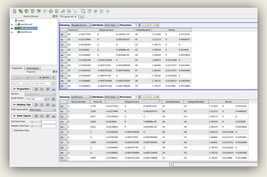
Striving for processed simulations as accurate to "real" world physics as possible to include:

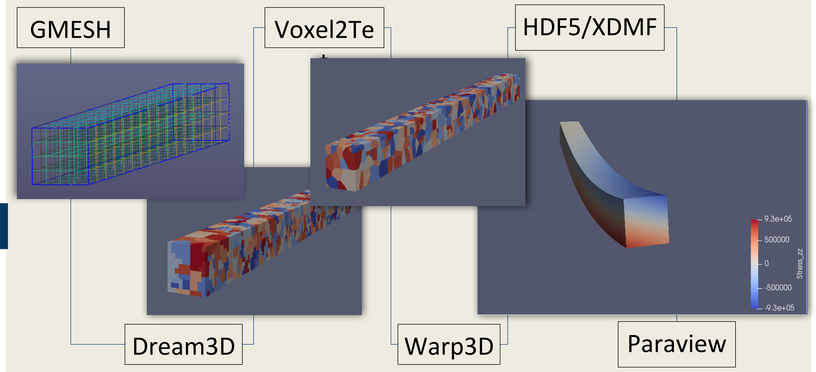
- Fracture analysis
- Crack propagation
- Object deformation
- Grain structure analysis
- Internal interactions and movements of grains
- Nodal connections (Element incidences)

Motivation:

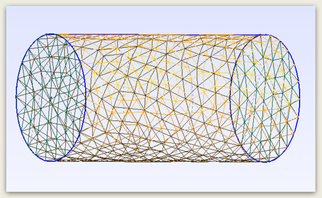
- More effective nonlinear analysis
- Parallelism leading to quicker processing
- Higher definition of simulated models
- Accurate studies of material sciences
- Simple workflow for easy user interface

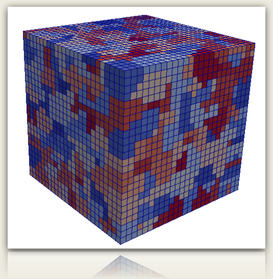
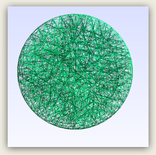




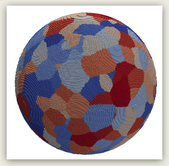
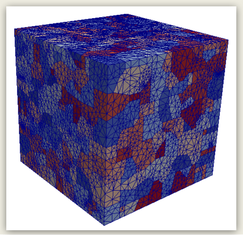
Rendering Software

GMESH: CAD software that self-fabricates multidimensional meshes useful for FEA



Dream3D: Data analysis tool that creates voxel grain structures in a CAD modeled geometry

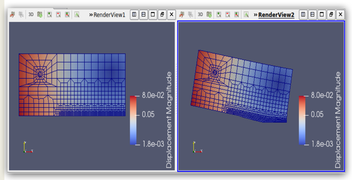
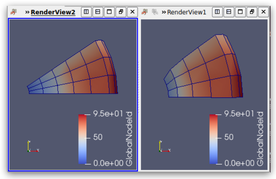
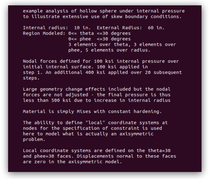



Voxel2Tet: Converts voxel representations to tetrahedral mesh with smooth interfaces

Back End

Measures Taken:

- Developed new subroutine for parsing and processing user commands within a Warp3D input file
- Created interface between Fortran source code and team-written C functions
- Manipulated compilation process to successfully implement Fortran to C interface with HDF5 functions used in C function
- Wrote XDMF format descriptor file to accompany HDF5 file for Paraview to successfully render simulated object with Warp3D computed stress, strain, displacement, etc.

Future Work

- Persist in achieving more accurately simulated models for optimal data most precise to the physical world
- Continue developing parallelism in input and output processing

Acknowledgements

- This research was supported with contributions made by NSF, ORNL, UTK and JICS.
- Mentors, Kwai Wong and Tim Truster, were crucial to the success of this project.
- A special thanks to the developers of all open-source programs used.