ARC Simulation Campaign Manual

File naming convention: CamelCase for executables, snake_case for data, Mixed_Case for folders

These simulations use the SI1 unit system: kg mm ms kN GPa K

Preparation

Geometry Preparation

1. SQ/PSQ generation via MATLAB
2. Void generation, *Solidworks* or similar
3. Meshing in *Coreform Cubit*
    a. Approximate sizing ~0.1
    b. Tet mesh, geometry sizing off depending on surface mesh behavior
    c. * Take care not to have more than 3 to 4 SPH nodes per characteristic particle element length!
4. Import to *LSPrePost*.
    a. Detach nodes for standalone elements. Success generally results in visual artifacts at element boundaries.
    b. Renumber nodes and elements to prevent potential overlap with the substrate. 6,000,000~ for element, 8,000,000~ for node by precedent. Renumber part to 2.
    c. Create a nodeset including all particle nodes (assign ID 3).

Z-shift Extraction

1. Extract node coordinates from geometry *.k* file (manually discard all content except coordinates). Results in *geometry_node_coordinates.txt*
2. Execute *NodePositionsToCSV.py* to convert *.txt* to *.csv* file containing coordinates. File Accepts *geometry_node_coordinates.txt*, returns *geometry_node_coordinates.csv*.
3. Execute *ExtractZshifts.m* to generate orientations for campaign and collate associated Z-shifts. Executable accepts *geometry_node_coordinates.csv,* returns *Z_shift_values.csv*

Campaign Generation

1. Prepare campaign master directory.
2. Prepare BASE *.k* and *.sh* files and *PostProcess.sh* file in BASE folder within master directory.
    a. Specify computer resource allocation in *BASE.sh*
    b. Ensure filepath to post-processing scripts specified in *PostProcessing.sh* is clear.
3. Prepare target & projectile *.k* files, *theta_values.csv file*, *Z_shift_matrix.csv* file, and executables *CreateCampaign.sh, RunCampaign.sh, GetCompletionStatus.py,* and *ExtractBINOUT.l2a* in master directory.
    a. Ensure executables have execute authority.
4. Specify impact orientations generated via *ExtractZshifts.m* in *CreateCampaign.sh*, and execute. Results in individual directories with modified *.k* and *.sh* files per impact orientation simulation.
5. Specify number of jobs to be submitted to SLURM in *RunCampaign.sh*.

Campaign

Execution

1. Execute *RunCampaign.sh*. as batch submission.
    a. This will periodically run *PostProcessing.sh* to inspect, process, and terminate jobs once interaction is finished.
        i. Ensure *RunScript.m*, *ExtractData.py*, *ClusterAlgorithmParallelTermination.m* are present in source directory */home/harryjang/Simulations/Campaign_Template_Fracture/Data_Extraction_Processing/*
    b. The directory *Automated_Campaign_Output* will be created in the master directory with log files *submission_script_log.txt, submitted_jobs_history.txt,* etc. for troubleshooting.

Post

Data Extraction

1. Prepare *ExtractBINOUT.sh* in output folder.
    a. *ExtractBINOUT.sh* will collect .glstat and .matsum files from simulations into dedicated folders (*Collected_GLSTAT, Collected_MATSUM*).
2. Prepare *ExtractOutput.sh* in output folder.
    a. *ExtractOutput.sh* will collect cluster algorithm output from *Output_Files* in simulation folders and place them in *Collected_Output*.
3. Place *ExtractGLSTAT.py* in *Collected_GLSTAT* folder and execute.
    a. *ExtractGLSTAT.py* will collect statistics from all simulations into *glstat_summary.csv*.
4. Place *CollectedOutputDataExtraction.m* adjacent to *Collected_Output* and *Collected_GLSTAT* and execute.
    a. *CollectedOutputDataExtraction.m* will generate figures of interest at simulation and ensemble level.

Appendix

Linux Misc

- ls (list directory items)
- cd (navigate to directory)
- mkdir (make directory)
- cp (copy)

Executable

- chmod u+x <executable> (assign execute permission to executable)
- squeue -u <username> (list SLURM jobs)
- sbatch <executable> (submit job to SLURM)
- ./<executable> (run executable)
- module load python3 → python <executable.py> (python script execution)

Screen

- screen (initialize screen)
- Ctrl+A+D (detach screen, return to terminal)

- screen -r <screen identifier> (reattach screen)
- screen -X -S <screen identifier> quit (end screen)