

Meeting Minutes 07/04/2020

Group 1

May 14, 2020

1 RSEP Tutorial 1

- Project is about using open source FE packages, image analysis packages, to run any sort of physics simulation e.g. diffusion, thermal mechanics). Whatever we like. Deliberately vague.
- FE packages have some examples, can recreate what they do
- Additionally to the FE examples, import images that are relevant to our PhD projects.
- Split the entire project into several tasks
- First task: taking an image, cleaning it up, segmentation, whatever preprocessing step you can think of. Task is built around converting an image into something that you can import into an FE simulation. Image analysis of a bunch of images + generate a mesh for the FE simulation
- Second task: taking the cleaned up image and assigning different material IDs/boundary conditions to different regions, can see why the segmentation is important so material IDs can be assigned. Figuring out how to use some python FE software, can be any package, it is just about being able to use the package for our particular problem
- FE physics simulation e.g. deformation/heat flow in a composite material
- Take an FE package and create some examples, run FE simulation on example microstructures
- Create test cases, a simple case where you know the solution e.g. 1D heat conduction, a source on one end. Open up a FE textbook and create an FE simulation for these test cases
- Compare tests to textbook answer, if solution is completely different, then something has gone wrong in the simulation
- Link sent to us for FE packages: skimage, scipy, Google search FE packages in python
- Grading is based upon how the project is managed and how we are collaborating on Git, how issues are raised, how we communicate
- Quality of technical job is not critical
- Different repositories for different tasks, each should have a test case folder (a suite of test cases should be included to ensure the code is running as expected) adding a level of trust
- Developing examples for your particular task
- Test case should be quick, easy to run and the solution should be obvious. Should have example files showcasing for the project

- Document the code and the examples correctly. Document everything you do in the code e.g. comment lines
- Do not create a Word report, do it on a Git repository (text file, which you can markup and see differences). Commits should show that the whole file has not just been committed. Need to add section by section.
- Do not upload a word file as it will be seen in Git as a binary file and not a gradual/step-by-step process.
- Perfect = Use Latex, easier to manage on Git and to grade
- No page limit
- Graded on Git log
- When committing make sure the message is descriptive and try not to mix too many commits together. Commit needs to be atomic i.e. if a commit needs to be reverted and only one functionality causes an error/broken, then the whole commit will need to be reverted regardless of whether the other functionalities within the commit cause an error or not.
- Each functionalities will then be added separately, not a nice way of doing things.
- Feature = set of commits that make up a functionality
- Make sure features are created in separate branches and merge features back into into a main branch
- When merging, need to raise a pull request and someone else needs to approve of the pull request
- Need someone to approve of changes, otherwise changes would be biased. Always good to have someone else review the changes created and approve the merge.
- Links for good practice in using Git will be provided