Documentation on wrapper of *"Mesoscale analysis of Li-ion battery microstructure using sequential coupling of discrete element and finite element method"*

Pramod Kumbhar*[1], Narasimhan Swaminathan†[1], and Ratna Kumar Annabattula‡[1]

[1]Department of Mechanical Engineering, Indian Institute of Technology Madras, Chennai 600036, India

This report acts as the documentation for the developed wrapper which is made open-source for the benefit of the readers. This wrapper is used to communicate the information between the two modules - FEM (Finite Element Method) and the DEM (Discrete Element Method) and automating certain tasks such as sub-domain formations, meshing the sub-domains etc. For the FEM and DEM, Abaqus [1] and open source C++ based MercuryDPM [2] is used, respectively. The wrapper is developed in the PYTHON language. For the more details about the wrapper and the developed framework, the readers are requested to refer to the accompanying paper in the **International Journal of Energy Research** titled "Mesoscale analysis of Li-ion battery microstructure using sequential coupling of discrete element and finite element method".

The wrapper consists of the following routines:

- abaqus_exec.py
- boundary.py
- compileandexecute.py
- expansion.py
- modify_inp.py
- read_data_output_group_contactarea.py
- read_group_data_make_MDPM_script.py
- read_MDPM_into_ABAQUS.py
- run_UEL.py
- tecplot_combine.py

*pramodkumbhar789@gmail.com
†Corresponding Author
‡Corresponding Author

The objectives of these routines are as follow:

1. abaqus_exec.py - To automate the meshing of the sub-domains using Abaqus.
2. boundary.py - To identify the nodes on the boundary of the sub-domains. The nodal coordinates and element connectivity is provided as the input to this routine.
3. compileandexecute.py - Automation of the compilation and execution of the DEM C++ script in `MercuryDPM`.
4. expansion.py - To compute the changed area of the sub-domains. It needs nodal coordinates, element connectivity etc. Based on the deformations, the new area is computed using the Gauss quadrature rule. Also, this function writes the deformation, normalised concentration, stresses, and strains for the subsequent visualisation in TecPlot.
5. modify_inp.py - To automate the creation of the Abaqus input file to carry out the FE simulations. This file imposes the initial and boundary conditions too. The necessary material properties, time increment, end time are specified through this file.
6. read_data_output_group_contactarea.py - This routine reads the `MercuryDPM` file and splits into the sub-domains based on if the particles are contacting each other or not. The non-contacting particles (island particles) are written into the seperate files.
7. read_group_data_make_MDPM_script.py - Based on the split groups, a C++ file is generated for the `MercuryDPM` simulation. Here, contact stiffness, coefficient of friction, time-steps necessary for the DEM simulation should be provided.
8. run_UEL.py - This is necessary to execute the FE and its allied codes.
9. tecplot_combine.py - This function creates a `.dat` file that can be read by the `TecPlot` software for the visualisation.

Prerequisites: A Ubuntu OS enabled system with Abaqus and MercuryDPM pre-installed. Moreover, all the PYTHON libraries specified in each file. In case they are not pre-installed, they can be installed using the PYTHON package manager - *pip*.

If using this code for research or industrial purposes, please cite the paper. For any queries, discussions and collaborations, please feel free to mail to pramodkumbhar789@gmail.com.

References

[1] Abaqus user manual, v6.12. 2012.

[2] http://www.mercurydpm.org/. v 0.10.