**How-To-Use**

**mainmodel.py:** ’Main’ program which initializes Packs or Cells – runs simulation, cycling and any plotting/post-processing routines

* *See the VARIABLES IN ALL CAPS AT THE TOP this program*

**params.py:** Defines cell-level parameters (see ‘VariedParameter’)

* POS\_ELEC\_POROSITY = VariatedParameter.from\_percent(0.50, 5)
  + *The positive electrode’s porosity will be 0.5 +/- 5% across all cells in pack*

**Top Down Look:**

**pack.py:** Builds X Cells in series, Y Cells in parallel (X, Y are input params).

* Builds the DAE for the pack – all algebraic equations for coupling
* Includes discretization method and cycling protocol

**cell.py:** Couples 2 Single\_Particles (which consistute one ‘Cell’)

* Cathode called Positive Electrode (+)
* Anode called Negative Electrode (-)

**single\_particle.py:** Defines model of a single electrode.

* Diffusion ODEs, boundary/initial conditions for Lithium concentration, and Butler-Volmer calculations for electrode potential
* Global meshing geometry, parameter structures are updated
* *Note that this program can be run standalone to validate the ‘charge’ / ‘discharge’ of a single particle.*

**Some Footnotes:**

1. This repository/API for pack-modeling is meant to be parameter-set-agnostic. We are using ‘Marquis-2019’ – a set provided by PyBAMM for Li-Ion coin cells. Results from my code have been cross-checked by PyBAMM’s simulation output (for this parameter set). We can potentially validate against other sets to ensure robustness.