Contributing to PyBaMM

User Workshop
1st April 2020

Valentin Sulzer[a], Scott G Marquis[b], Robert Timms[b,e], Tom Tranter[c,e], Ferran Brosa Planella[d,e], Jacqueline Edge[e], Martin Robinson[f], Thibault Lestang[f], S Jon Chapman[b,e]

[a]Department of Mechanical Engineering, University of Michigan, 1231 Beal Ave, Ann Arbor, MI 48109, United States

[b]Mathematical Institute, University of Oxford, Andrew Wiles Building, Woodstock Road, Oxford, UK, OX2 6GG, UK

[c]EIL, Department of Chemical Engineering, University College London, London, WC1E 7JE, UK

[d]WMG, University of Warwick, Gibbet Hill Road, Coventry, CV4 7AL, UK

[e]The Faraday Institution, Quad One, Becquerel Avenue, Harwell Campus, Didcot, OX11 0RA, UK

[f]Department of Computer Science, University of Oxford, 15 Parks Rd, Oxford OX1 3QD, UK

Contributing to PyBaMM: Workflow

Create an issue where new proposals can be discussed before any coding is done

Create a branch of PyBaMM on your own fork where all changes will be made

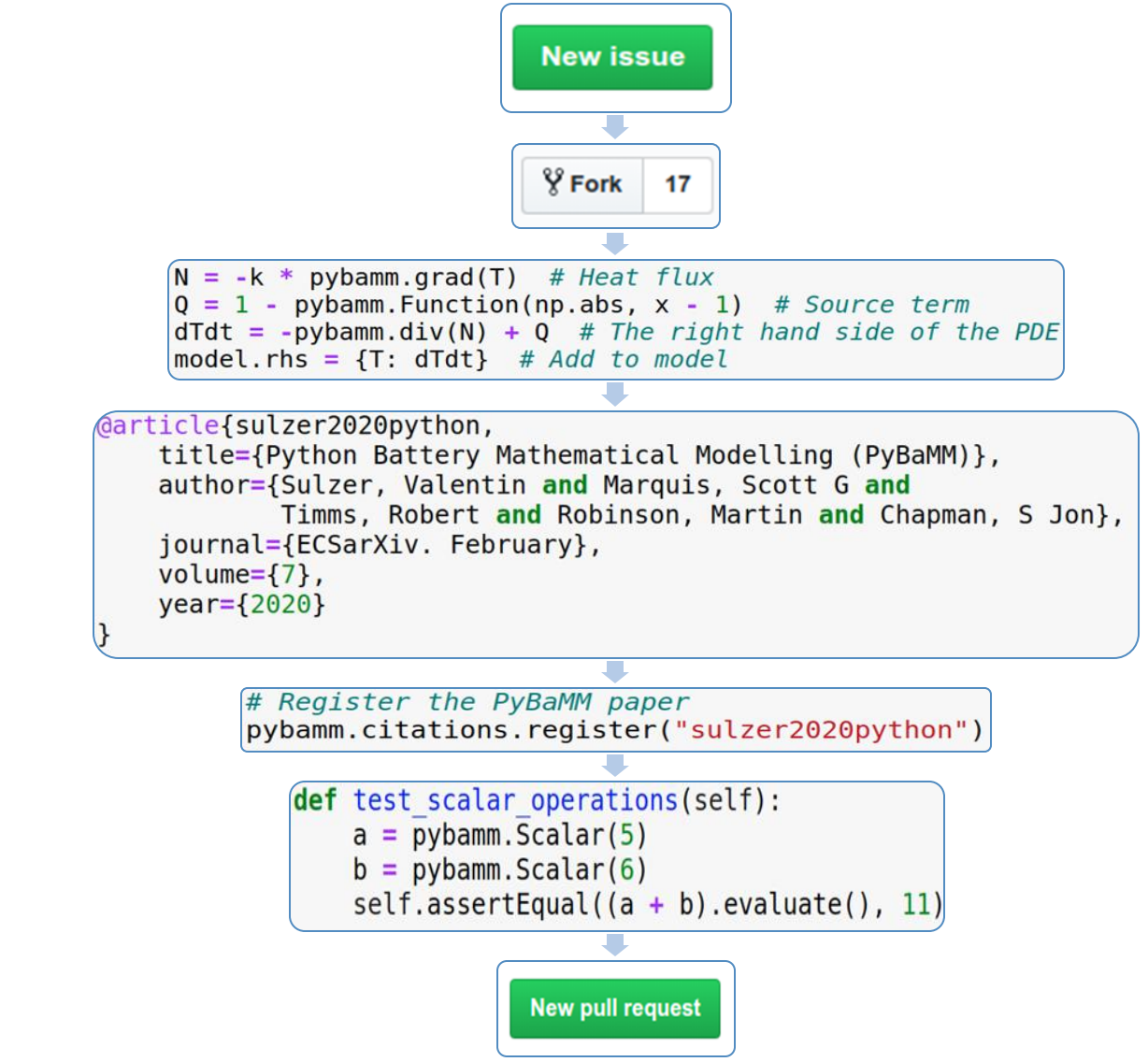
Implement your work (ask for help in the issue or on slack!)

Add relevant references to CITATIONS.txt

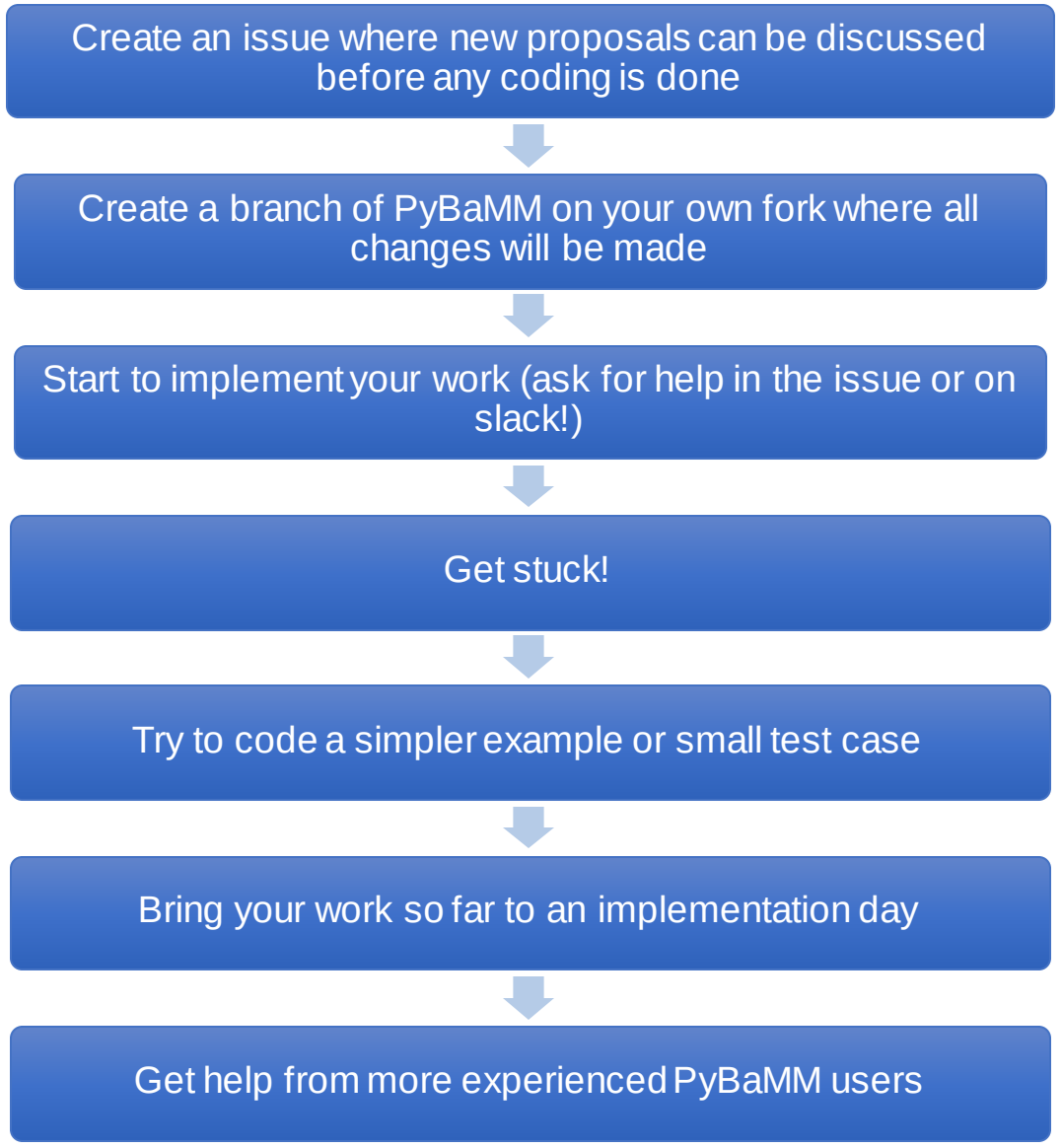
Register these in the parts of the code that use your paper

Test your code!

Create a pull request to merge into the main code when you are ready for your code to be publicly available



Contributing to PyBaMM: Implementation Days



Drop-in sessions

PyBaMM team on hand to help

Organised in response to contributors who approach us

How to cite PyBaMM, and ensuring your contribution is credited

Example: Ferran Brosa, WMG

Added LGM50 parameter set on his own fork

Added reference to relevant paper

Merged into main code

When the parameters set “Chen2020” is used, PyBaMM returns the citation

```python
import pybamm

model = pybamm.lithium_ion.SPMe()
param = pybamm.ParameterValues(chemistry=pybamm.parameter_sets.Chen2020)
sim = pybamm.Simulation(model, parameter_values=param)
sim.solve()

pybamm.print_citations()
```

```python
Chen2020 = {
    "chemistry": "lithium-ion",
    "cell": "LGM50_Chen2020",
    "anode": "graphite_Chen2020",
    "separator": "separator_Chen2020",
    "cathode": "nmc_Chen2020",
    "electrolyte": "lipf6_Nyman2008",
    "experiment": "1C_discharge_from_full_Chen2020",
    "citation": "Chen2020",
}
```

```python
# register citations
if "citation" in chemistry:
    citation = chemistry["citation"]
    pybamm.citations.register(citation)
```

```
@article{Chen2020,
author = {Chen, Chang-Hui and Brosa Planella, Ferran and O'Regan, Kieran and Gastol, Dominika and Widanage, W. Dhammika and Kendrick, Emma},
title = {{Development of Experimental Techniques for Parameterization of Multi-scale Lithium-ion Battery Models}},
journal = {Submitted for publication},
year = {2020}
}
```

PyBaMM generates BibTeX entries for the papers cited in the script.

pybamm-team / PyBaMM

Latest release V0.2.1

Lower barrier to entry

Submodel structure

- New models BasicSPM and BasicDFN
- Defines models in a single script
- No need to understand submodel structure
- PyBaMM core team will help integrate your work into submodel structure later

Installation

- New simple pip install
- Windows, Mac, Linux
- Jupyter hub allows access through a browser with no install

PyBaMM ongoing projects

- EIS and nonlinear EIS
- Degradation mechanisms (SEI, Li plating)
- Mechanics coupled to temperature and lithiation
- Other geometries (jelly rolls, packs, multiple layer cells)
- Other chemistries (LiS) Spatially structured electrodes
- Parameter estimation, sensitivity analysis
- Tools to analyse experimental data and characterise batteries
- Improvements to framework