

While you wait



Follow us on twitter - [@pybamm_](https://twitter.com/pybamm_)



Star us on GitHub - [PyBaMM](https://github.com/PyBaMM)



Fill out the PyBaMM Survey - survey.pybamm.org



Try our online examples - colab.pybamm.org



PyBaMM

Python Battery Mathematical Modelling



Why model batteries?



- Batteries are complicated!

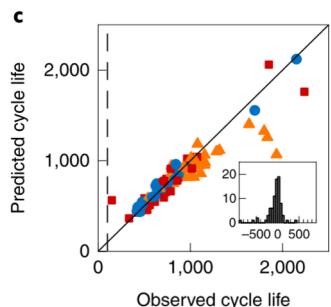
$$\begin{aligned}\frac{\partial}{\partial t}(c_k) &= -\frac{1}{C_e} \nabla \cdot \mathbf{N}_k - \sum_{r \in \mathcal{R}} s_r^k a^r j^r, \\ \mathbf{N}_k &= -D_k D_k^{\text{eff}} (\nabla c_k - \pi_{os} \psi_k \nabla p) + C_e t_k^e \mathbf{i}_e + C_e c_k \mathbf{v}^\square, \\ \frac{\partial \epsilon}{\partial t} &= -\beta^{\text{surf}} a^S j^S, \\ \nabla \cdot \mathbf{i}_e &= \sum_{r \in \mathcal{R}} a^r j^r + a^{\text{dl}} C^{\text{eff}} \frac{\partial}{\partial t} (\Phi_s - \Phi_e), \\ C_e \dot{\mathbf{i}}_e &= \kappa^{\text{eff}} \left(\chi_e \nabla \ln(c_e) - \nabla \Phi_e + \pi_{os} \left(\frac{M_{\text{e}} c_e^{\max}}{\rho^{\max} \rho} - \frac{\chi_e \psi_e}{c_e} \right) \nabla p \right), \\ \nabla \cdot \mathbf{v}^\square &= \sum_{r \in \mathcal{R}} \beta^r a^r j^r, \\ \mathbf{v} &= -\frac{K}{\mu} \nabla p, \\ \rho (\mathbf{v} - \mathbf{v}^\square) &= \frac{\omega_e D_e^{\text{eff}}}{C_e} (\nabla c_e - \pi_{os} \psi_e \nabla p) + \omega_i \dot{\mathbf{i}}_e, \\ \nabla \cdot \mathbf{i}_s &= -\sum_{r \in \mathcal{R}} a^r j^r - a^{\text{dl}} C^{\text{eff}} \frac{\partial}{\partial t} (\Phi_s - \Phi_e),\end{aligned}$$

- Not *that* much battery data (+ curse of dimensionality)

- Electrochemical models can:
 - Provide physical insight
 - Be generalized
 - Regularize data-driven methods

- Need to be:
 - Fast (enough)
 - Easy to use, share, understand
 - Reliable

- Ultimately, will need hybrid approach



Data-driven prediction of battery cycle life before capacity degradation. K.A. Severson et al., *Nature Energy*, 2019

Why PyBaMM?



PyBaMM's mission is to **accelerate battery modelling research** by providing an **open-source framework** for multi-institutional, interdisciplinary collaboration.

PyBaMM Goals



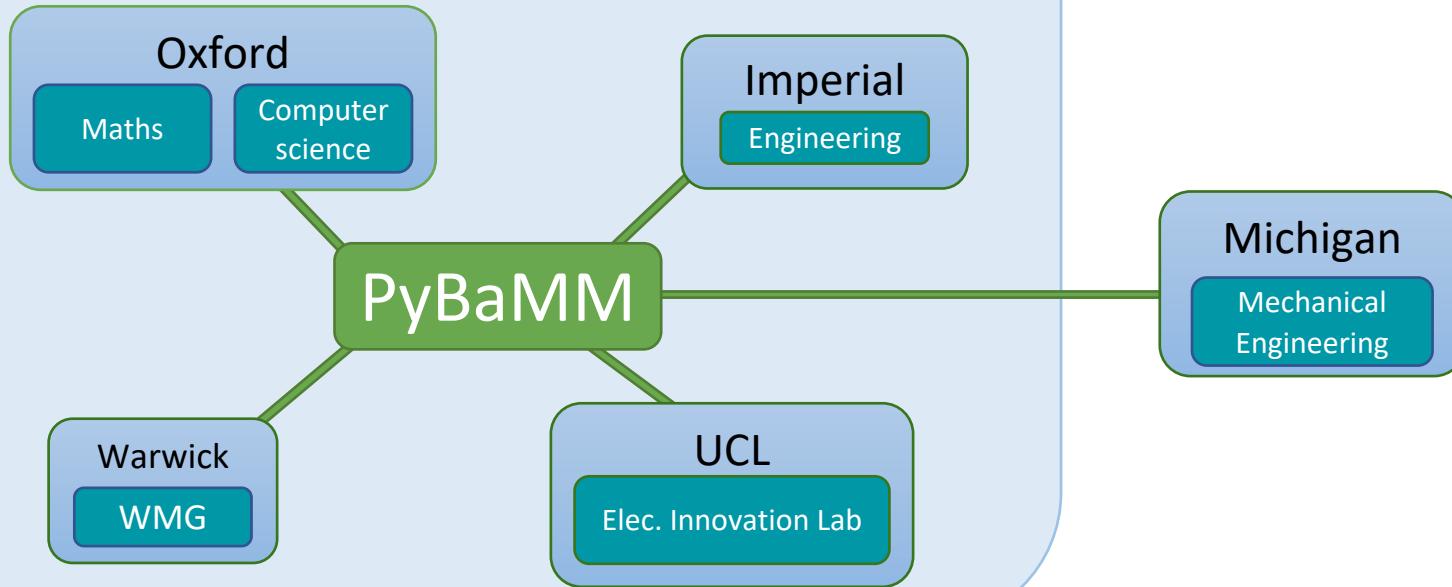
- Provide fast, reliable battery simulations
- Grow battery modelling user-base
- Facilitate development of new battery models
- Improve reproducibility of research

Who is PyBaMM?

Core Developers



Faraday Institution





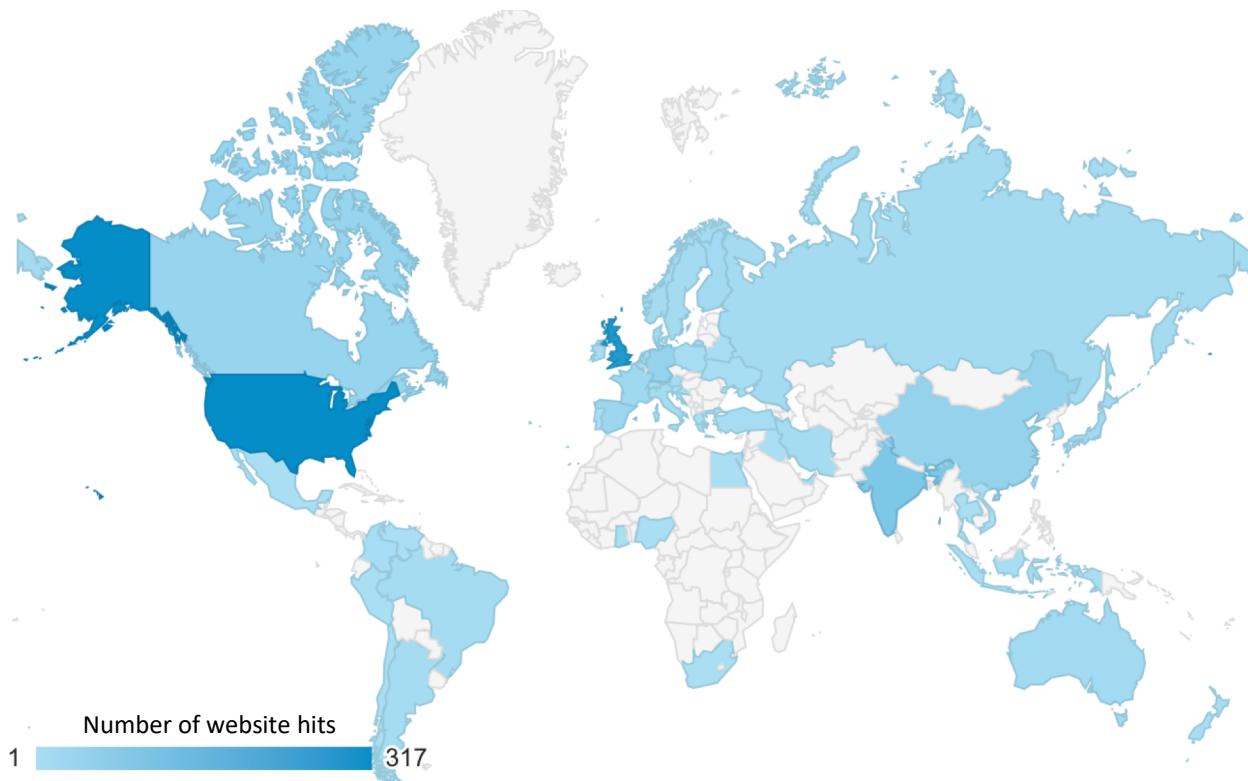
Who is PyBaMM?

Active and growing community



Who is PyBaMM?

Active and growing community



PYBAMM BASICS

What is PyBaMM?



PyBaMM is a framework for building and solving battery models

FAST

FLEXIBLE

RELIABLE

Models

- Lithium-ion models (full and reduced-order)
- Lead-acid models
- Fully coupled thermal models
- Degradation mechanisms (SEI)

Experimental suite

- Any voltage, current or power control, for example:
 - GITT, PITT, CCCV, drive cycle
- Easy interface to define custom protocols

Library of Chemistries / Parameters

- LCO, LiNiCoO₂, NCA, NMC
- Graphite
- LiPF₆
- Easy to add your own parameter sets

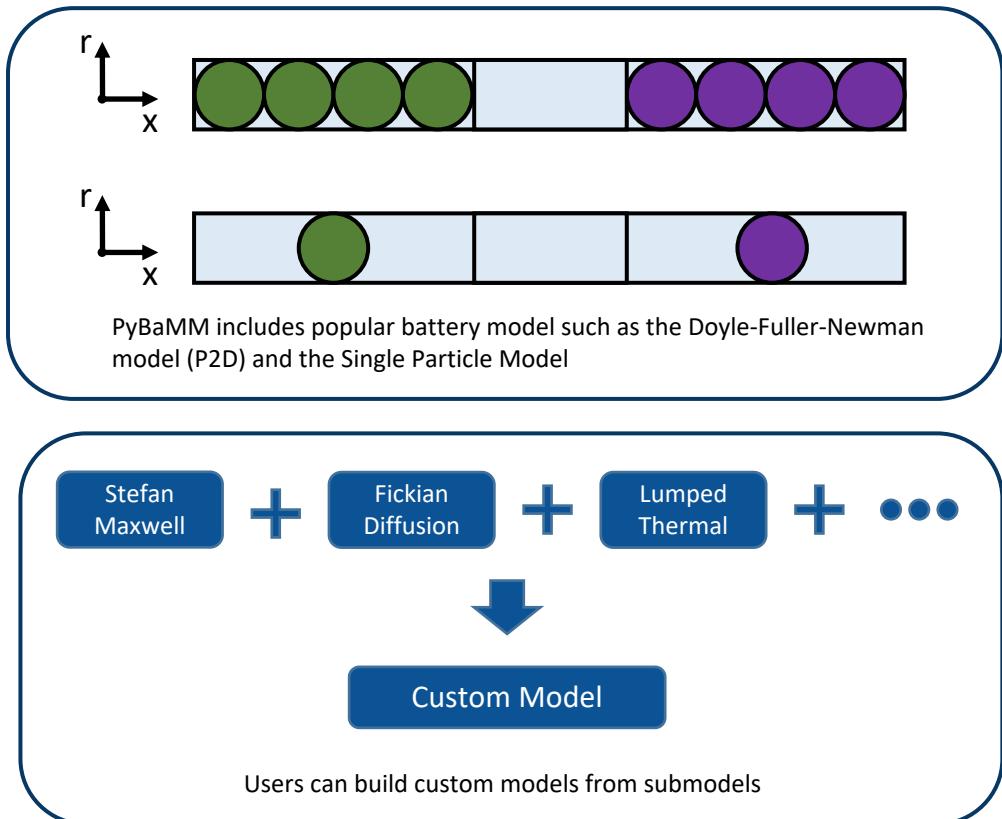
Functionality

- Range of numerical methods for solving models
- Choice of cell geometries: 1D, 2+1D single layer pouch cells
- Interface with external software

Flexible Submodel Structure



- Battery models are constructed from submodels
- Use or adapt pre-built models
- Create custom models



Flexible Submodel Structure



- Structure allows for “plug and play” physics

- New physics can be implemented in independent fast



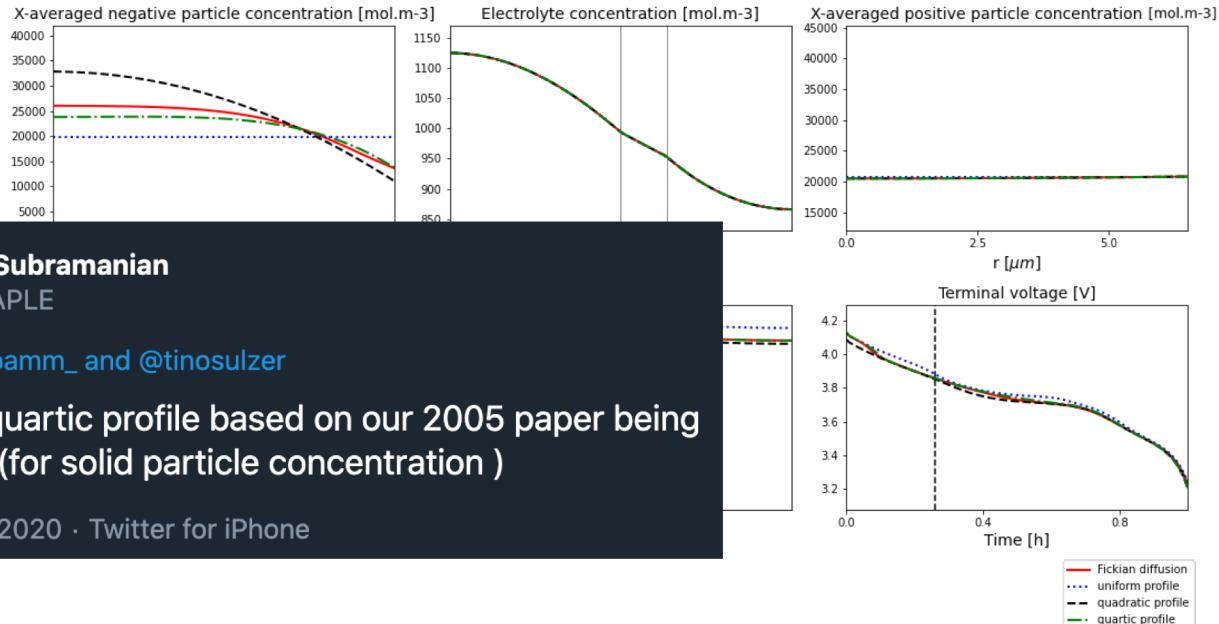
Venkat Subramanian
@UT_MAPLE

Replies to @pybamm_ and @tinosulzer

Great to see quartic profile based on our 2005 paper being implemented (for solid particle concentration)

6:45 PM · Sep 7, 2020 · Twitter for iPhone

1 retweet

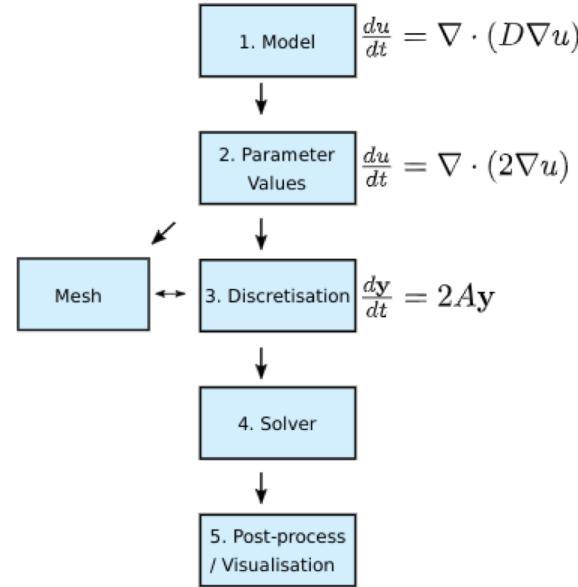


Using PyBaMM it is easy to compare how different physics affects the model solution. Here we investigate the influence of different models for the particles (Fickian diffusion, and three reduced-order polynomial approximations [Subramanian et al., 2005]).

PyBaMM Pipeline



- Choice of spatial discretisation
 - Finite Volume Method
 - Spectral Volume Method
 - Finite Element Method
- Multiple ODE and DAE solvers (e.g. scipy, sundials CVODE and IDA)
- Obtain Jacobians via automatic differentiation
- [Work in progress] Sensitivity analysis via automatic differentiation



Martin Robinson,
Oxford RSE

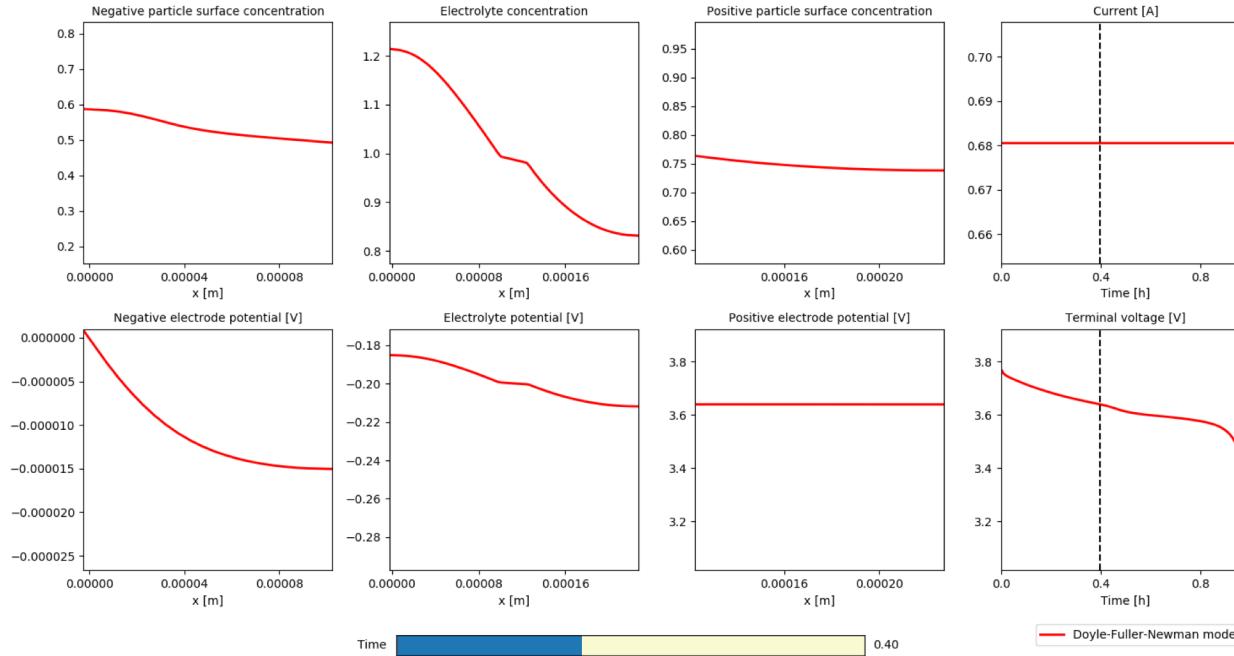
By employing a pipeline process PyBaMM separates the statement of model equations from the inputted parameters, choice of spatial discretization and choice of time stepping algorithm.

Simple Interface: CC Discharge



```
import pybamm
model = pybamm.lithium_ion.DFN()
sim = pybamm.Simulation(model)
sim.solve([0, 3600])
sim.plot()
```

- Easy interface for running simulations
- Both model and simulation can be customised

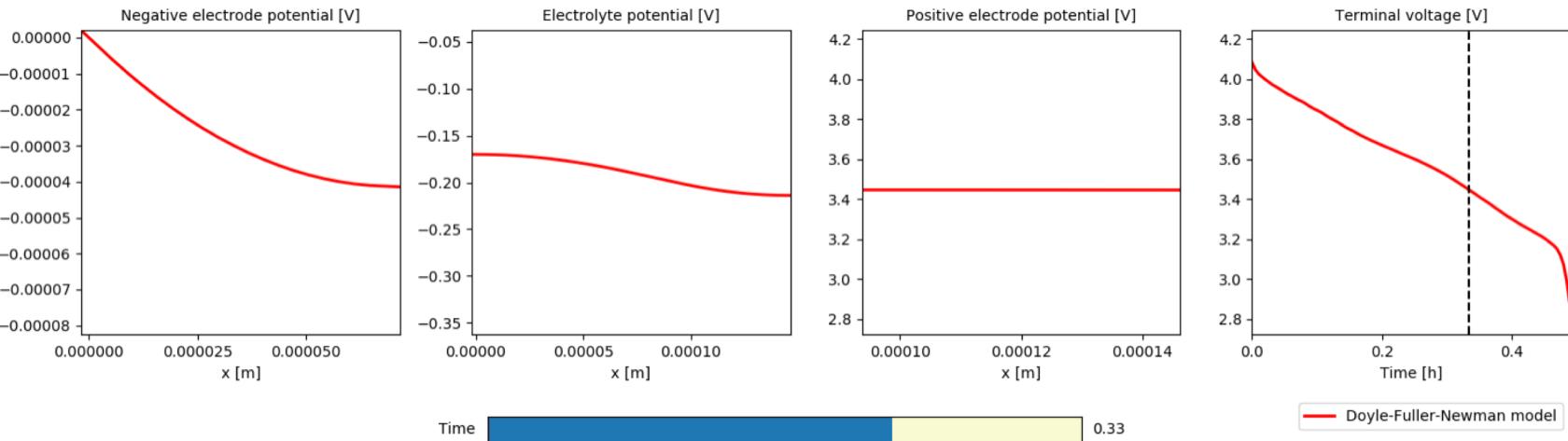


Simple Interface: Change parameters



```
import pybamm
model = pybamm.lithium_ion.DFN()
chemistry = pybamm.parameter_sets.NCA_Kim2011
parameter_values = pybamm.ParameterValues(chemistry=chemistry)
sim = pybamm.Simulation(model, parameter_values=parameter_values, C_rate=2)
sim.solve([0, 1800])
sim.plot()
```

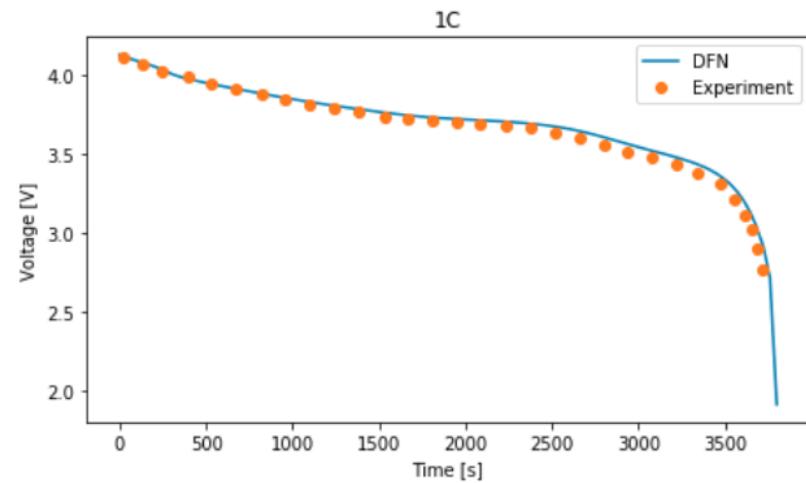
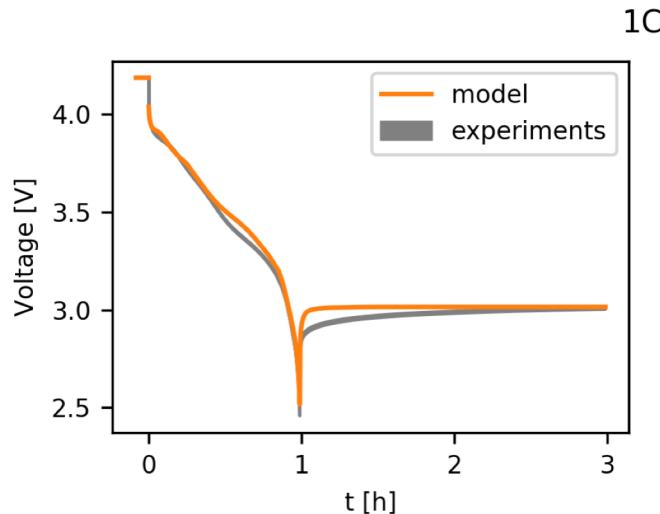
Easy to change
parameters and
model options



Parameter Library



- Library of parameter values including popular chemistries such as LiNiCo02, NMC, Graphite, LiPF6
- Choose by component (anode, cathode, etc.) or pick a complete parametrization from a given reference
- Easy to “tweak” existing parameter sets or add your own from scratch



`chemistry = pybamm.parameter_sets.Chen2020`



Development of Experimental Techniques for Parameterization of Multiscale Lithium-ion Battery Models, *Chang-Hui Chen, Ferran Brosa Planella, Kieran O'Regan, Dominika Gastol, W. Dhammika Widanage and Emma Kendrick*, Journal of The Electrochemical Society, 2020
Parameterization of a physico-chemical model of a lithium-ion battery II. Model validation, *Madeleine Ecker, et al*, Journal of The Electrochemical Society, 2015

`chemistry = pybamm.parameter_sets.Ecker2015`

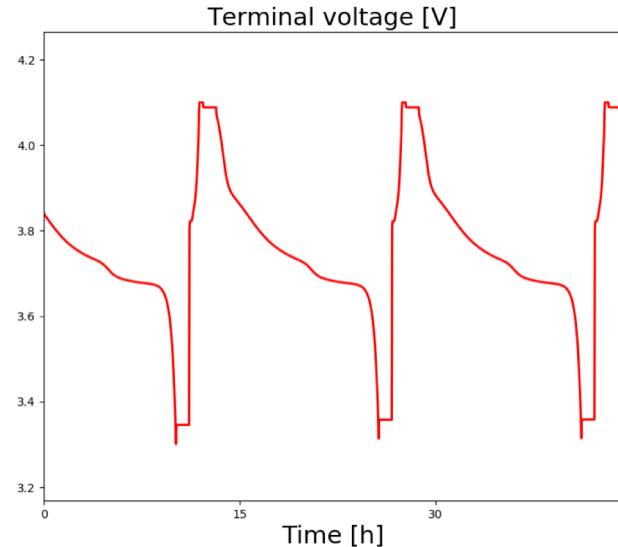
Simple Interface: Experimental protocol



PyBaMM allows users to specify experimental protocols using keyword strings

```
experiment = pybamm.Experiment(
    [
        "Discharge at C/10 for 13 hours or until 3.3 V",
        "Rest for 1 hour",
        "Charge at 1 A until 4.1 V",
        "Hold at 4.1 V until 50 mA",
        "Rest for 1 hour",
    ]
    * 3,
)
model = pybamm.lithium_ion.DFN()
sim = pybamm.Simulation(model, experiment=experiment)
sim.solve()
```

- Can easily define custom protocols
- Specify current, power or voltage



Simple Interface: Experimental protocols & drive cycles



```
experiment = pybamm.Experiment(
```

```
[  
    "Discharge at C/20 for 1 hour",  
    "Rest for 1 hour"  
] * 20,
```

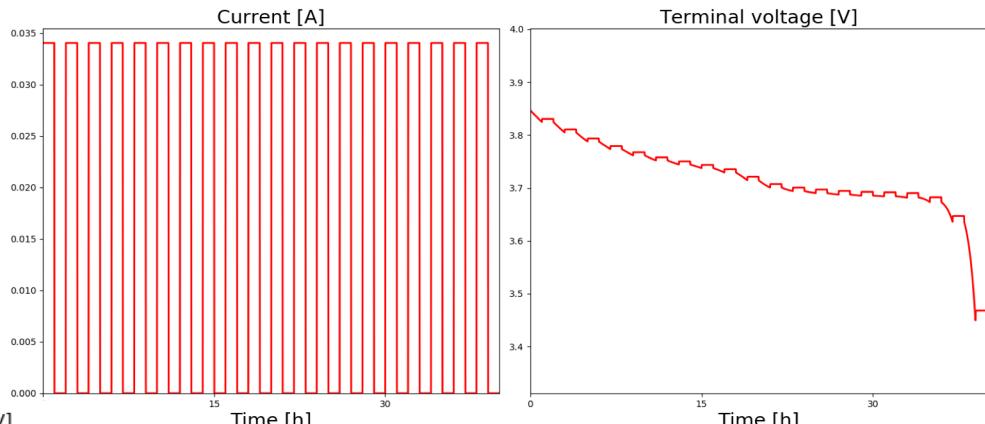
```
)
```

```
model = pybamm.lithium_ion.DFN()
```

```
sim = pybamm.Simulation(model, experiment=experiment)
```

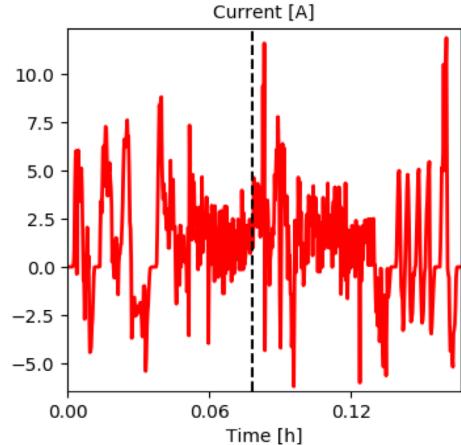
```
sim.solve()
```

```
sim.plot()
```



Current [A]

Terminal voltage [V]



```
model = pybamm.lithium_ion.DFN()  
param = model.default_parameter_values  
param["Current function [A]"] = "[current data]US06"  
sim = pybamm.Simulation(model, parameter_values=param)  
sim.solve()  
sim.plot()
```

Fast Solve Times



A comparison of typical solve times for different models and discretisations for a **1C constant current discharge**.

Grid points per domain	Single Particle Model		Doyle-Fuller-Newman Model	
	States	Time (milliseconds)	States	Time (milliseconds)
10	21	3	281	72
20	41	3	961	149
40	81	4	3251	384
80	161	5	13441	1293
160	321	9	52481	7689

All simulation performed on a laptop computer (i5, 2.1 GHz, 16 Gb of RAM) using the CasADi solver with both absolute and relative tolerances of 10^{-6} . The parameters are from the reference:

An asymptotic derivation of a single particle model with electrolyte, S.G. Marquis, V.Sulzer, R. Timms, C.P. Please and S.J. Chapman, Journal of The Electrochemical Society 2019

Fast Solve Times



A comparison of typical solve times for different models and discretisations for the **US06 drive cycle**

Grid points per domain	Single Particle Model		Doyle-Fuller-Newman Model	
	States	Time (seconds)	States	Time (seconds)
10	21	0.03	281	4.01
20	41	0.03	961	7.53
40	81	0.04	3251	20.30
80	161	0.07	13441	66.19
160	321	0.10	52481	247.15

All simulation performed on a laptop computer (i5, 2.1 GHz, 16 Gb of RAM) using the CasADi solver with both absolute and relative tolerances of 10^{-6} . The parameters are from the reference:

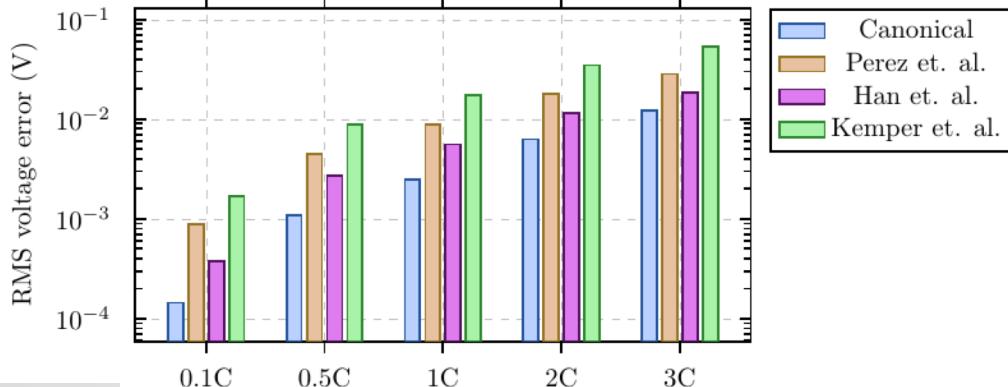
An asymptotic derivation of a single particle model with electrolyte, S.G. Marquis, V.Sulzer, R. Timms, C.P. Please and S.J. Chapman, Journal of The Electrochemical Society 2019

CASE STUDIES

Case Study: Single Particle Model with electrolyte (SPMe)



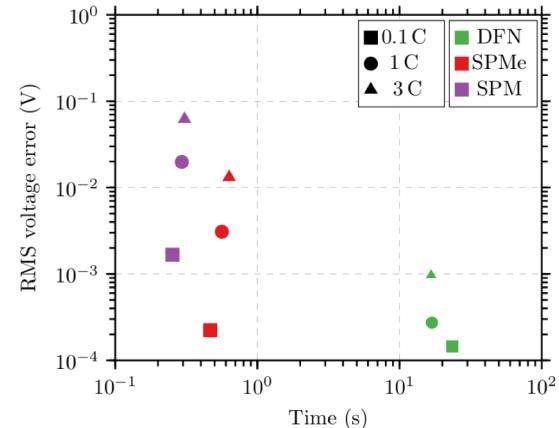
- Systematic derivation of SPMe using asymptotic methods
 - Electrolyte diffusion time \ll discharge time
 - Large electrode and electrolyte conductivity
- Outperforms other (ad hoc) models in literature of similar complexity
- Achieves order of magnitude reduction in computation cost at small loss of accuracy
- Extends the validity of SPM to higher C-rates.



```
models = [
    pybamm.lithium_ion.SPM(),
    pybamm.lithium_ion.SPMe(),
    pybamm.lithium_ion.DFN(),
]
solutions = [None] * 3

for i, model in enumerate(models):
    sim = pybamm.Simulation(model)
    sim.solve()
    solutions[i] = sim.solution

plot = pybamm.QuickPlot(solutions)
plot.dynamic_plot()
```

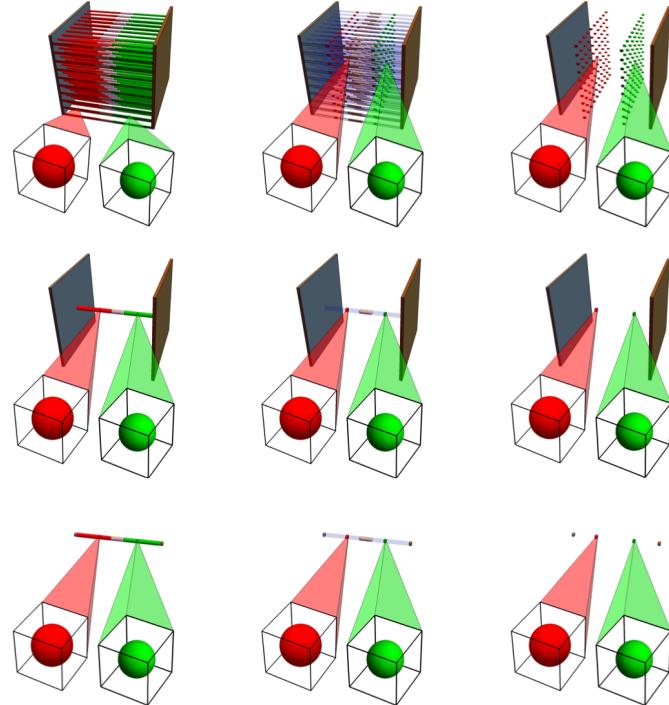


An asymptotic derivation of a single particle model with electrolyte, S.G. Marquis, V.Sulzer, R. Timms, C.P. Please and S.J. Chapman, Journal of The Electrochemical Society 2019

Case Study: Reduced order pouch cell modelling

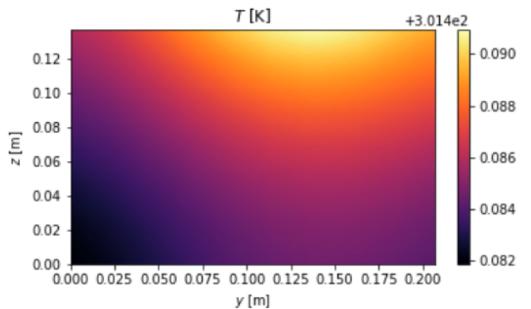
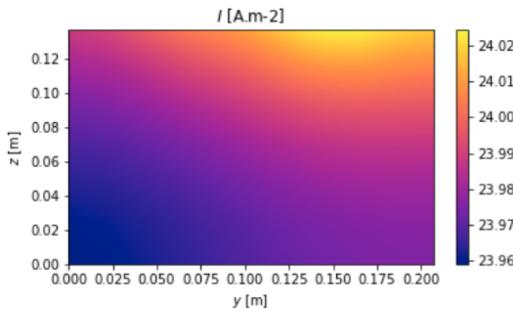
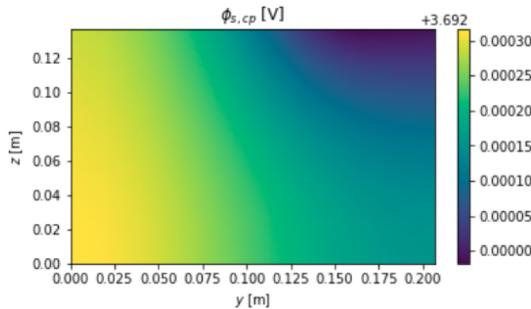
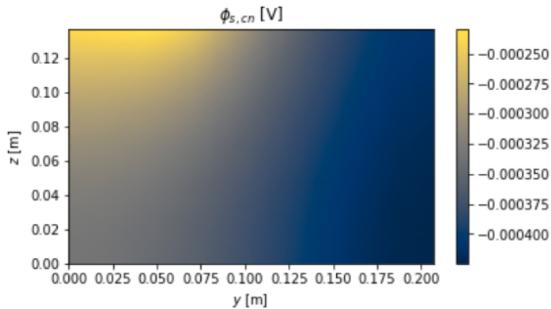


- Larger batteries exhibit non-uniform behaviour which may adversely affect battery performance and lifetime
- Exploit the geometry to systematically derive 2+1D model
- Combine with other limits to derive a hierarchy of reduced-order models
- Influence of cell geometry can be investigated efficiently
- Numerical validation against “full” model in COMSOL



Asymptotic Reduction of a Lithium-ion Pouch Cell Model, R. Timms, S.G. Marquis, V.Sulzer, C.P. Please and S.J. Chapman, Submitted to SIAM Journal on Applied Mathematics
A Suite of Reduced-Order Models of a Single-Layer Lithium-ion Pouch Cell, S.G. Marquis, R. Timms, V.Sulzer, C.P. Please and S.J. Chapman, Submitted to Journal of the Electrochemical Society

Case Study: Reduced order pouch cell modelling



```
options = {
    "current collector": "potential pair",
    "dimensionality": 2,
    "thermal": "x-lumped",
}
model = pybamm.lithium_ion.SPM(options)
sim = pybamm.Simulation(model)
sim.solve([0, 3600])
```

Example result: 10x10 grid of Single Particle Models with fully coupled thermal effects. Approximate solve time: 9s



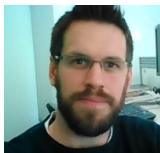
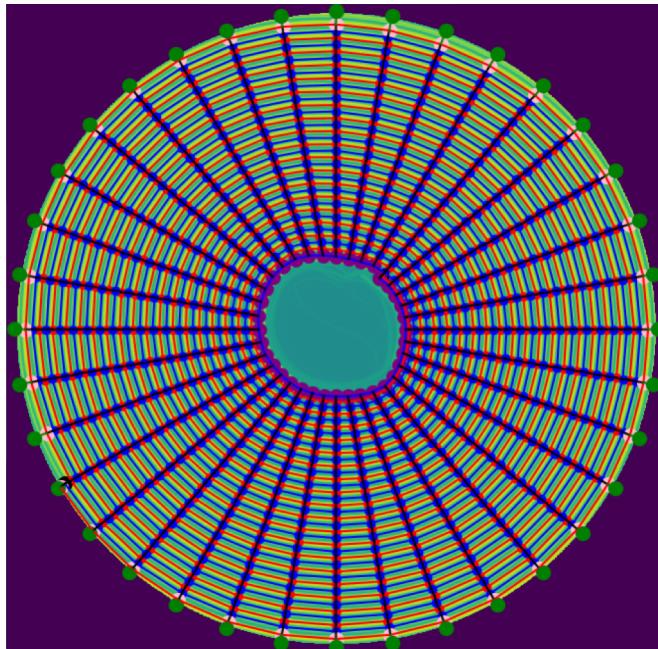
Asymptotic Reduction of a Lithium-ion Pouch Cell Model, R. Timms, S.G. Marquis, V. Sulzer, C.P. Please and S.J. Chapman, Submitted to SIAM Journal on Applied Mathematics

A Suite of Reduced-Order Models of a Single-Layer Lithium-ion Pouch Cell, S.G. Marquis, R. Timms, V. Sulzer, C.P. Please and S.J. Chapman, Submitted to Journal of the Electrochemical Society

Case Study: A coupled multiscale model of electrochemistry and thermal transport for Li-ion cells

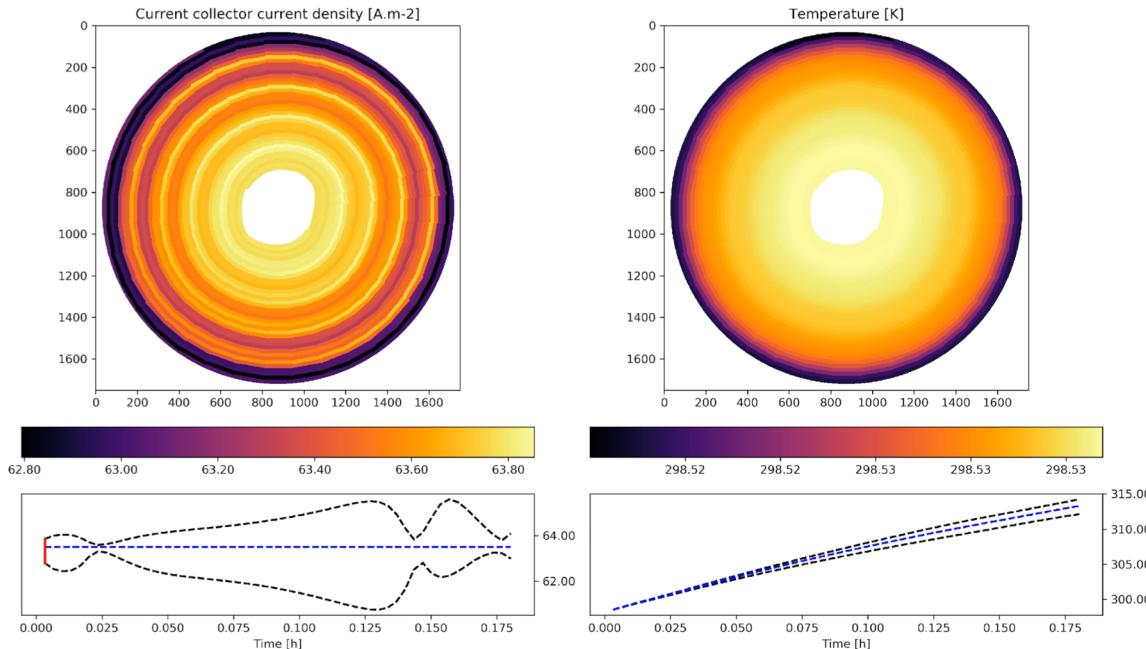


- Internal temperature differences lead to non-uniformities in current flow, state of charge, particle stress and levels of degradation
- Li-ion battery heat generation and transport problem coupling:
 - particle scale electrochemical modelling (PyBaMM)
 - cell level electrical and thermal modelling (OpenPNM)
- Example: a 2D heat-transfer problem for spirally rolled cylindrical batteries using tomography data



Probing heterogeneity in Li-ion batteries with coupled multiscale models of electrochemistry and thermal transport using tomographic domains, T.G. Tranter, R. Timms, T.M.M. Heenan, S.G. Marquis, V. Sulzer, A. Jnawali, M.D.R. Kok, C.P. Please, S.J. Chapman, P.R. Shearing, D.J. L. Brett, Journal of The Electrochemical Society, 2020

Case Study: A coupled multiscale model of electrochemistry and thermal transport for Li-ion cells

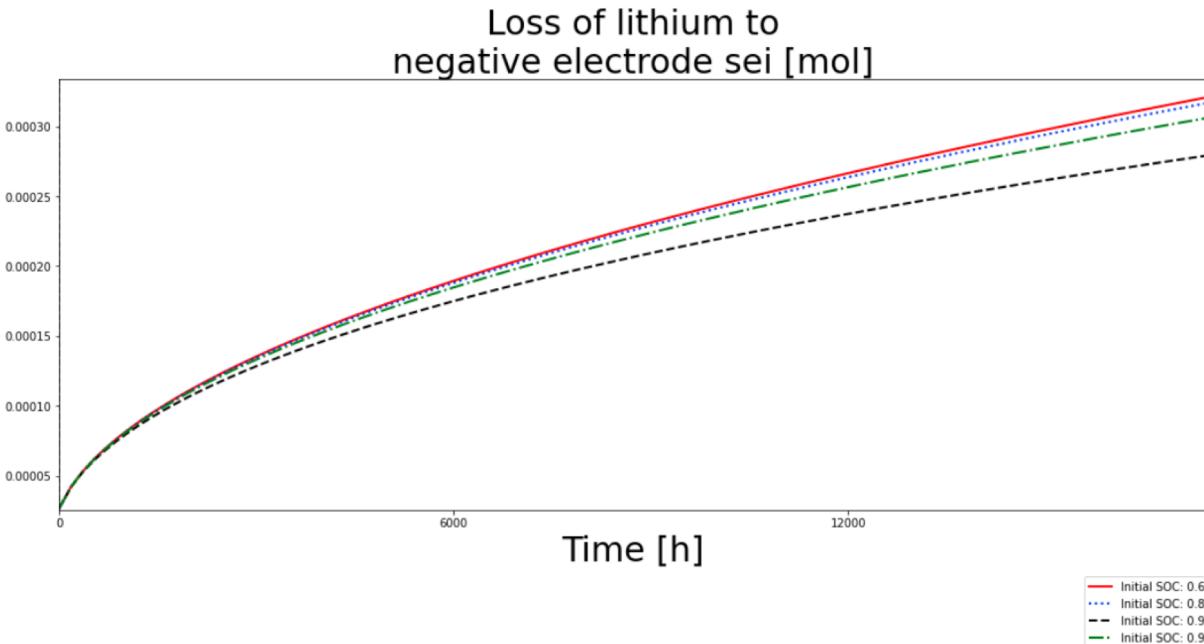


$HTC = 5 \text{ [W.m-2]}$
 $\sigma = 3e7 \text{ [S.m-1]}$
 $\#tabs = 5$
Current = 5 [A]



Probing heterogeneity in Li-ion batteries with coupled multiscale models of electrochemistry and thermal transport using tomographic domains, T.G. Tranter, R. Timms, T.M.M. Heenan, S.G. Marquis, V. Sulzer, A. Jnawali, M.D.R. Kok, C.P. Please, S.J. Chapman, P.R. Shearing, D.J. L. Brett, Journal of The Electrochemical Society, 2020

Case Study: SEI models



SEI models for:

- constant SEI thickness
- reaction limited
- solvent-diffusion limited
- electron-migration limited
- interstitial-diffusion limited
- EC-reaction limited

Example: loss of lithium to negative electrode SEI using the interstitial-diffusion limited model with different initial SOCs

Case Study: Thermal runaway under external short-circuit

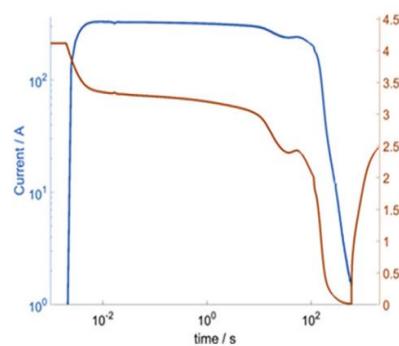


- Simulate external short-circuit by imposing resistance $R_{ext} = 11m\Omega$
- Add decomposition of SEI, anode, and cathode to SPM
- Lumped thermal model
- Good agreement with experiments
- Model provides additional physical insight

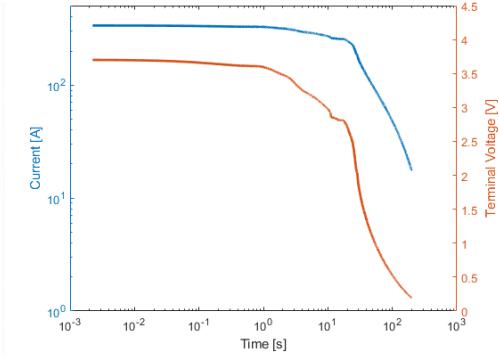


External short circuit performance of Graphite-LiNi_{1/3}Co_{1/3}Mn_{1/3}O₂ and Graphite-LiNi_{0.8}Co_{0.15}Al_{0.05}O₂ cells at different external resistances, A. Kriston et al, *Journal of Power Sources*, 2017
For more information, contact Vivian Tran, Department of Mechanical Engineering, University of Michigan

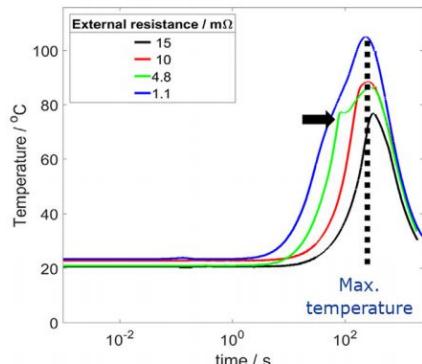
Experiment (Kriston et al)



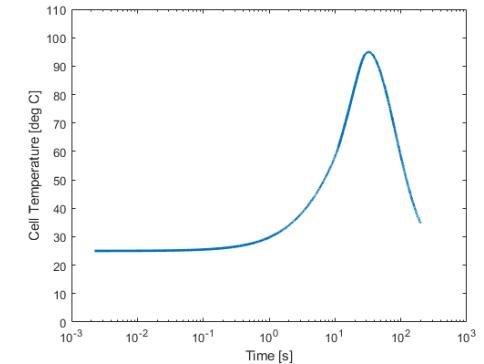
Simulation



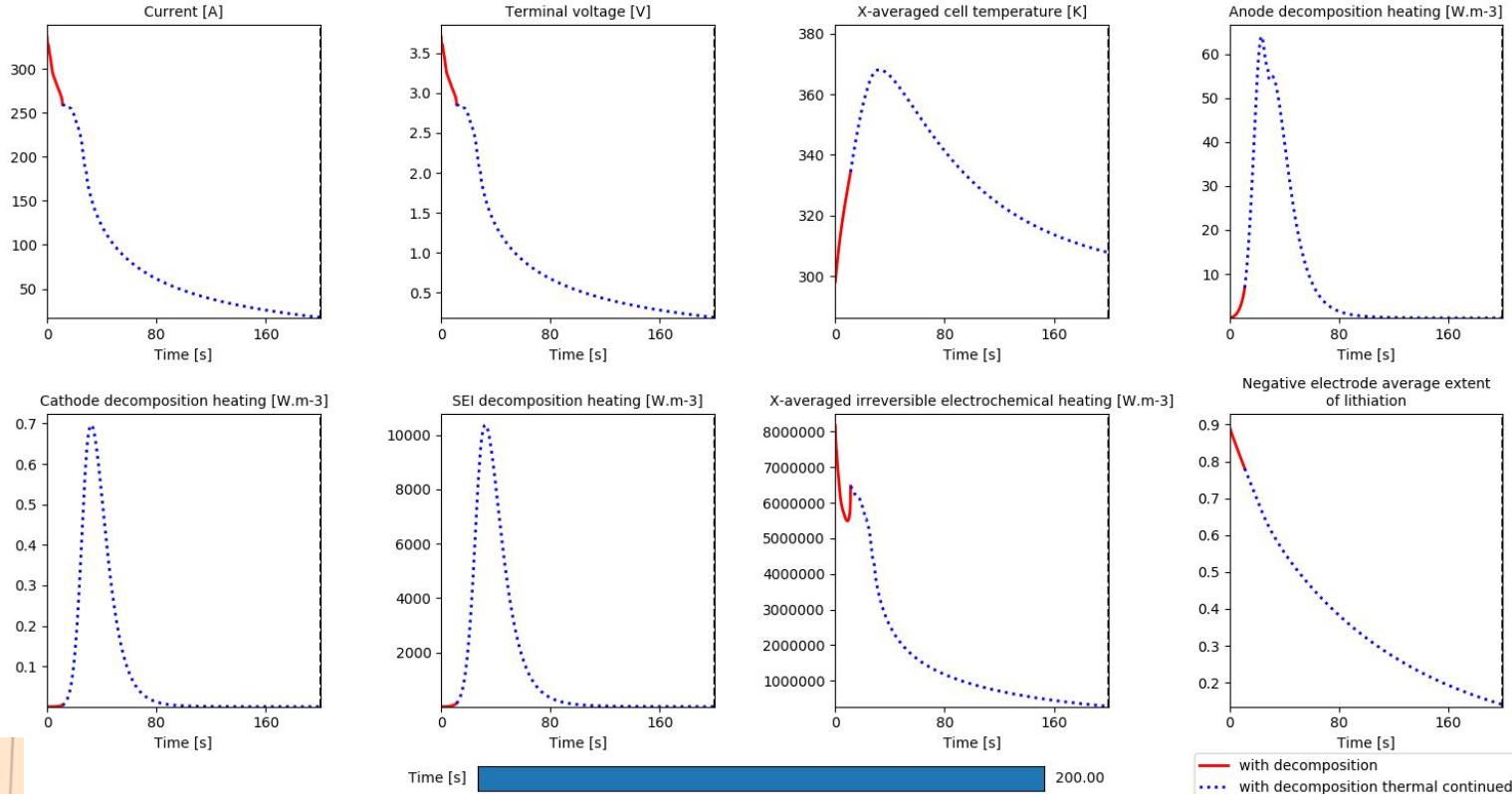
Experiment (Kriston et al)



Simulation



Case Study: Thermal runaway under external short-circuit

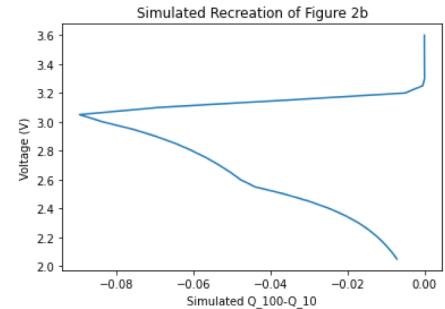
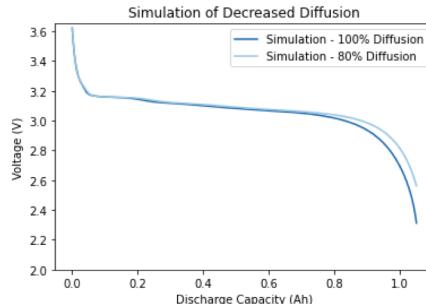
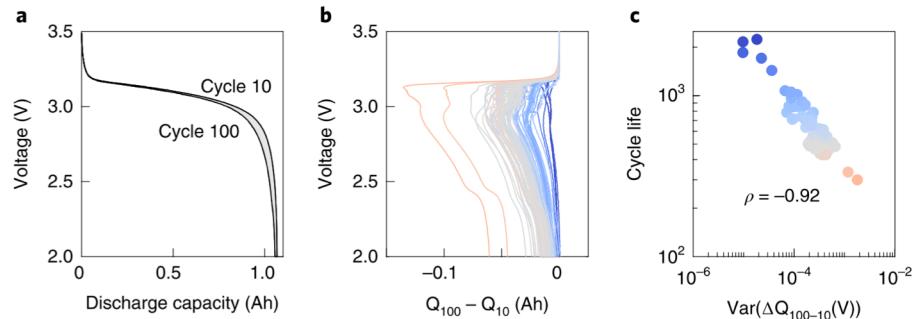
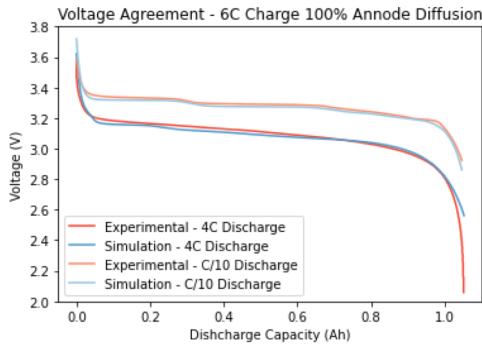


For more information, contact Vivian Tran, Department of Mechanical Engineering, University of Michigan

Case Study: Understanding LFP degradation



- Recent work by Severson et al. shows that features from discharge voltage curves can be used to predict cycle life
- Use PyBaMM to fit DFN model to battery data, giving excellent agreement
- Changing the anode diffusion coefficient produces features as seen in the data



Data-driven prediction of battery cycle life before capacity degradation, K.A. Severson et al., Nature Energy, 2019

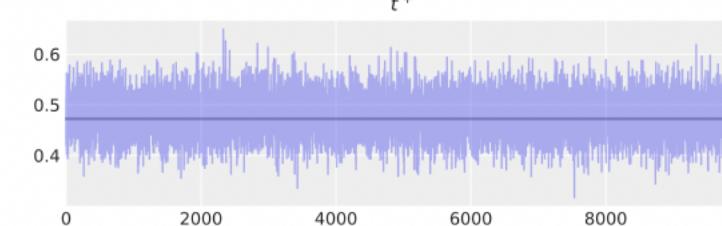
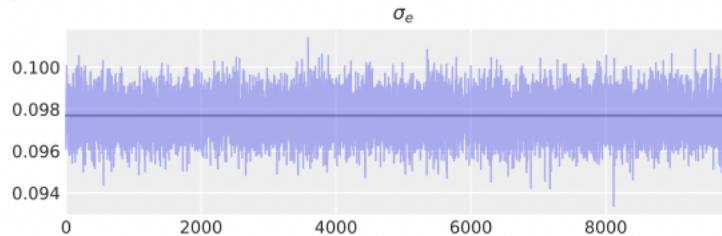
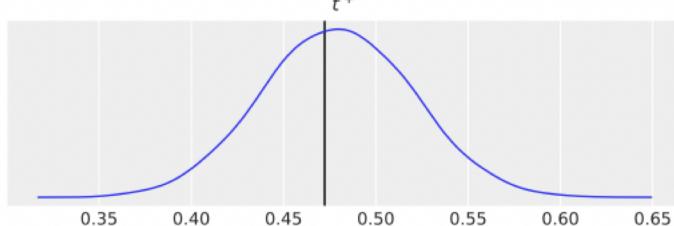
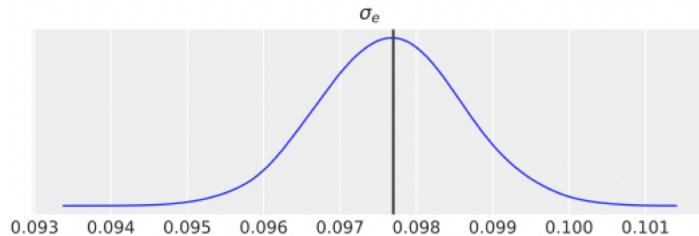
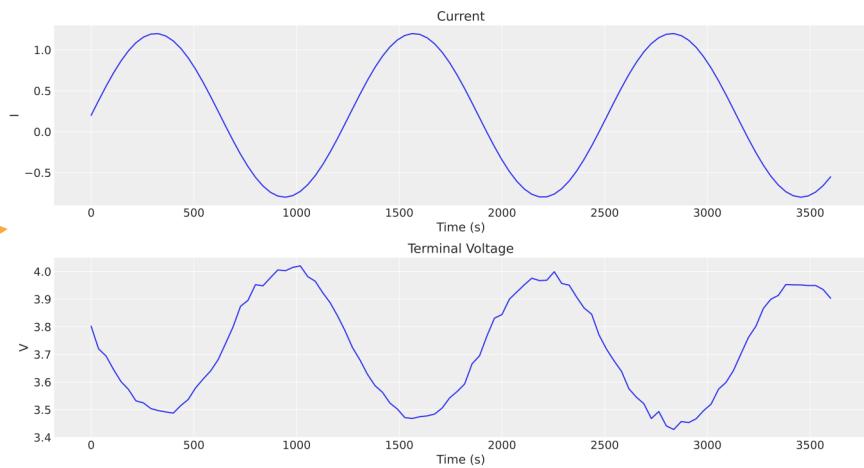


Work led by Daniel Albamonte, University of Michigan. For more information contact Dr V Sulzer, Department of Mechanical Engineering, University of Michigan

Case Study: Variational Bayes

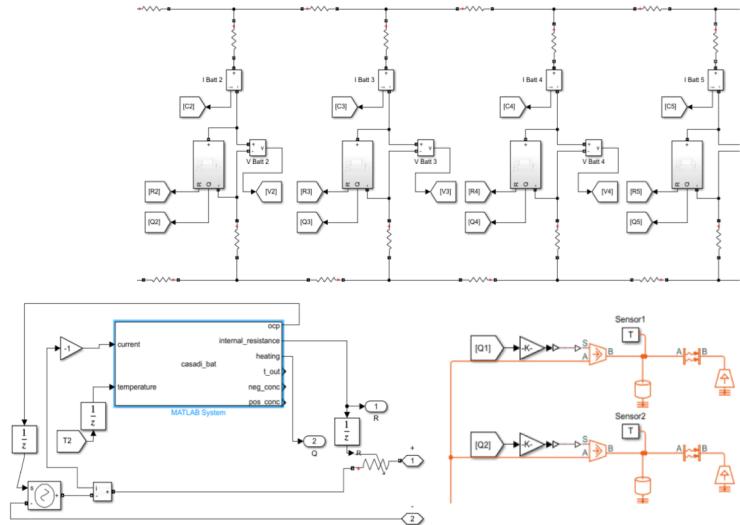


- Wrapping pymc3, a Bayesian library, around PyBaMM's SPMe
- Generate synthetic data with noise
- Use Variational Bayes to recover probability distributions for electrolyte conductivity and transference number

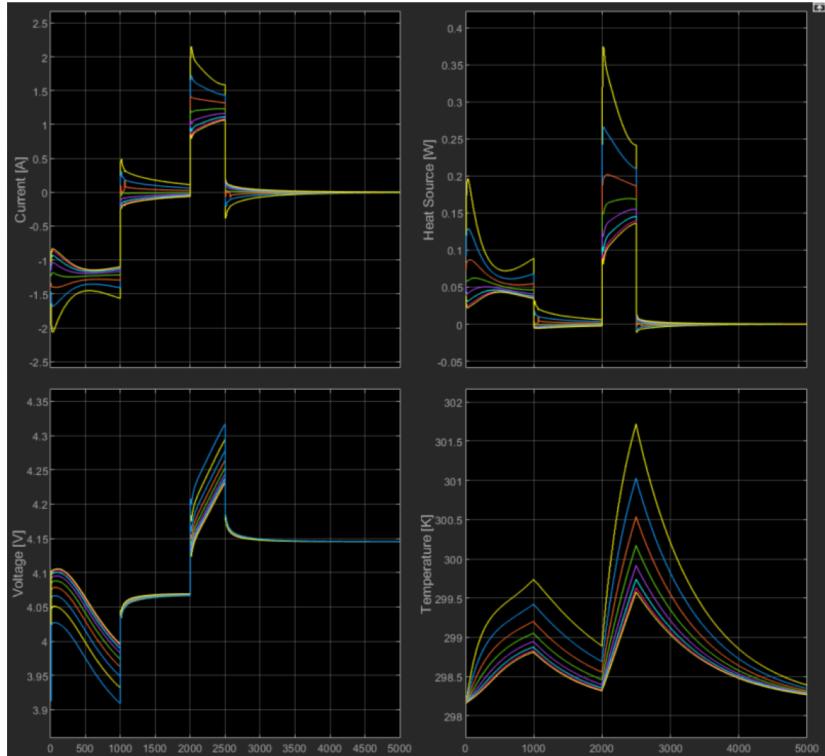


For more information
contact Antti Aitio,
Battery Intelligence Lab,
University of Oxford

Highlighted upcoming feature: pack modelling using PyBaMM and Simulink

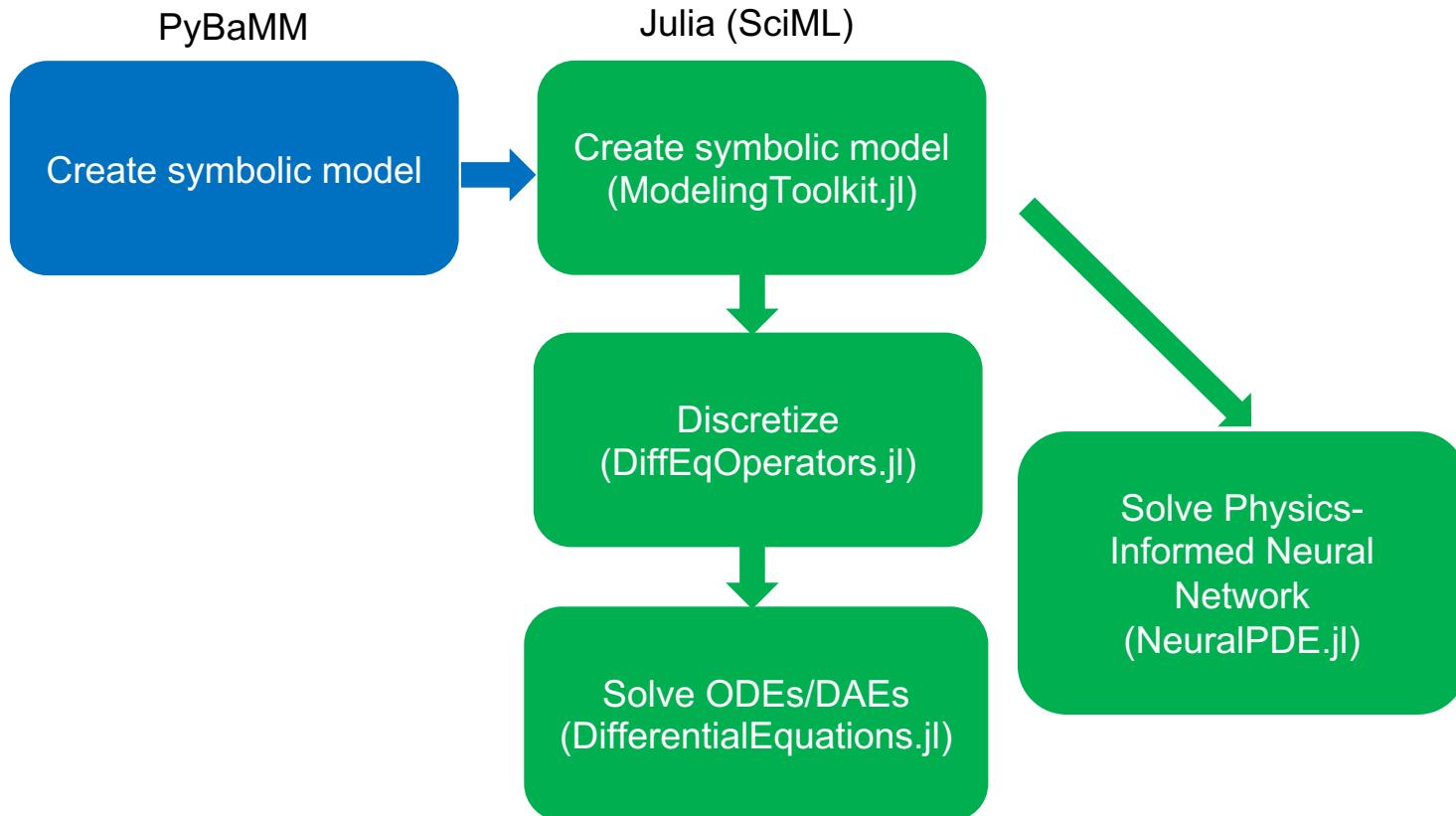


- Black box PyBaMM model put in Matlab system
- Linked in circuit with busbar and connection resistances
- Pack level electrical and thermal problem solved in Simulink
- Casadi solver used to efficiently call battery functions
- Input currents and temperatures
- Output heat and voltage sources and internal resistance calculated with physics based models

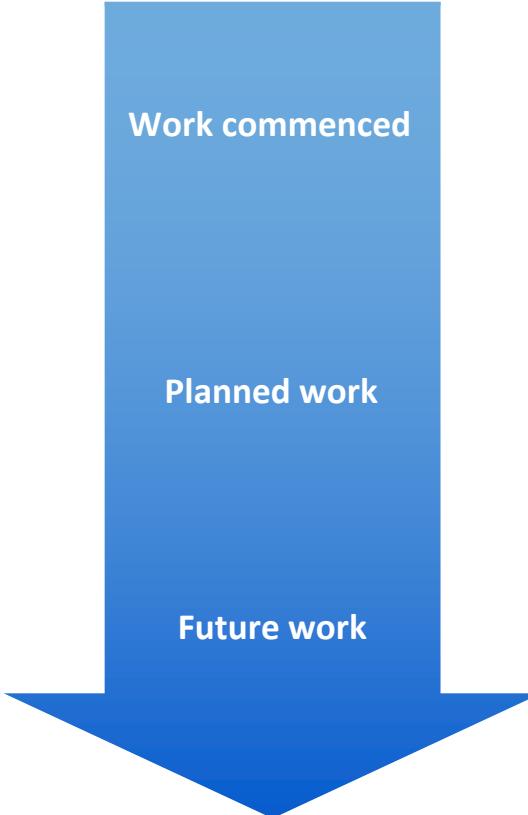


For more information contact Dr T Tranter, UCL Electrochemical Innovation Lab

Highlighted upcoming feature: Integration with Julia



Upcoming features



- Sensitivity analysis & parameter estimation
 - Simulink interface
 - Particle size distribution
 - Other chemistries (LiS)
 - Different particle shapes
 - Lithium plating
 - Fuel cells
-
- Mechanics coupled to temperature and lithiation
 - Solid-state battery models
 - Improvements to framework
-
- Other geometries (jelly rolls, multiple layer cells, etc.)
 - Integration with open battery database, parameter database
-
- ... your ideas?

Further information – pybamm.org



PyBaMM

News

Getting started

Contact

Examples

Docs

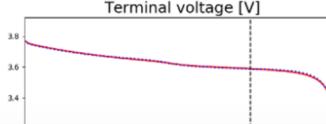
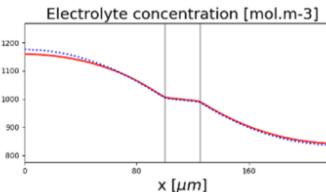
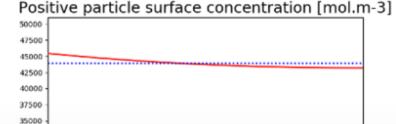
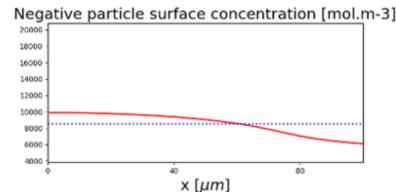
Source code

Publications



PyBaMM

PyBaMM can efficiently solve and compare standard battery electrochemical models such as the Doyle-Fuller-Newman model and the Single Particle Model.



USING PYBAMM

How to use PyBaMM?

Ways to engage



Try online

Use PyBaMM in your browser without any installation.



Install locally

Install PyBaMM on your computer.



Develop

Clone git repository to contribute code.

How to use PyBaMM?



Training resources

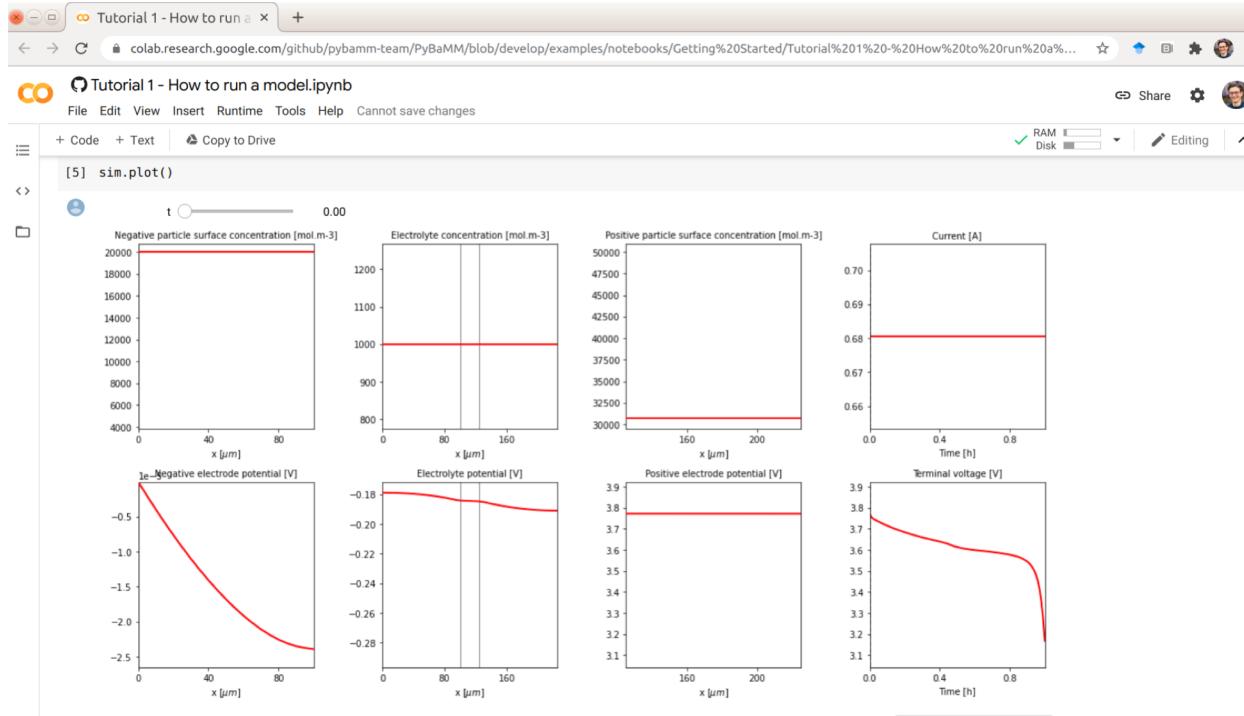
Lots of new resources online at

pybamm.org/getting-started

- Training delivered via Faraday masterclass and tech series training
 - Recordings and resources available online
 - Over 50 attendees
- Getting Started notebooks
 - Run in your browser via Google Colab -- no need to install PyBaMM!



Google Colab



Online tutorials run interactively in your browser using Google Colab, removing the need to install PyBaMM locally

Open in Colab

- [Tutorial 1: How to run a model](#)
- [Tutorial 2: Compare models](#)
- [Tutorial 3: Basic plotting](#)
- [Tutorial 4: Setting parameter values](#)
- [Tutorial 5: Run experiments](#)
- [Tutorial 6: Managing simulation outputs](#)
- [Tutorial 7: Model options](#)



Contributing



If you would like to work
on something in private
please get in touch!

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

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

New issue

Fork 17

```
N = -k * pybamm.grad(T) # Heat flux
Q = 1 - pybamm.Function(np.abs, x - 1) # Source term
dTdt = -pybamm.div(N) + Q # The right hand side of the PDE
model.rhs = {T: dTdt} # Add to model
```

```
@article{sulzer2020python,
  title={Python Battery Mathematical Modelling (PyBaMM)},
  author={Sulzer, Valentin and Marquis, Scott G and
          Timms, Robert and Robinson, Martin and Chapman, S Jon},
  journal={ECSarXiv. February},
  volume={7},
  year={2020}
}
```

```
# Register the PyBaMM paper
pybamm.citations.register("sulzer2020python")
```

```
def test_scalar_operations(self):
    a = pybamm.Scalar(5)
    b = pybamm.Scalar(6)
    self.assertEqual((a + b).evaluate(), 11)
```

New pull request

How to ensure your contribution is credited



Example: Ferran Brosa Planella, 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

```
import pybamm

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

pybamm.print_citations()

@article{Chen2020,
author = {Chen, Chang-Hui and Brosa Planella, Ferran and O'Regan, Kieran and Gastol, Dominika and Widanage, W. Dhammadika 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