

NEW FOR 2020



**High quality, in depth training on techniques,
equipment and facilities**
with experts in the field

See **Communifire** for sign up details

FARADAY TECH

#skills #techniques #training #equipment #facilities



While you wait...



Say hello in the **#faraday-tech-series** channel on the pybamm slack



Visit www.pybamm.org



Follow [@pybamm_](https://twitter.com/pybamm_) on twitter for updates



PyBaMM

Faraday Tech Series Part 1
02/06/2020





How to Install (Linux and Mac)

Visit www.pybamm.org/getting-started for detailed instructions

We recommend that you install into a virtual environment. On Linux and Mac:

1. `python3 -m venv env`
2. `source env/bin/activate`
3. `pip install pybamm`



How to Install (Windows)

Visit www.pybamm.org/getting-started for detailed instructions

We recommend that you install into a virtual environment. On Windows:

1. `python -m venv env`
2. `env\Scripts\activate.bat`
3. `pip install pybamm`

For advanced features in Windows (like extra solvers) we recommend the use of Windows Subsystem for Linux (instructions online)



What is PyBaMM?

PyBaMM is a framework for building and solving battery models

FAST

FLEXIBLE

MODULAR

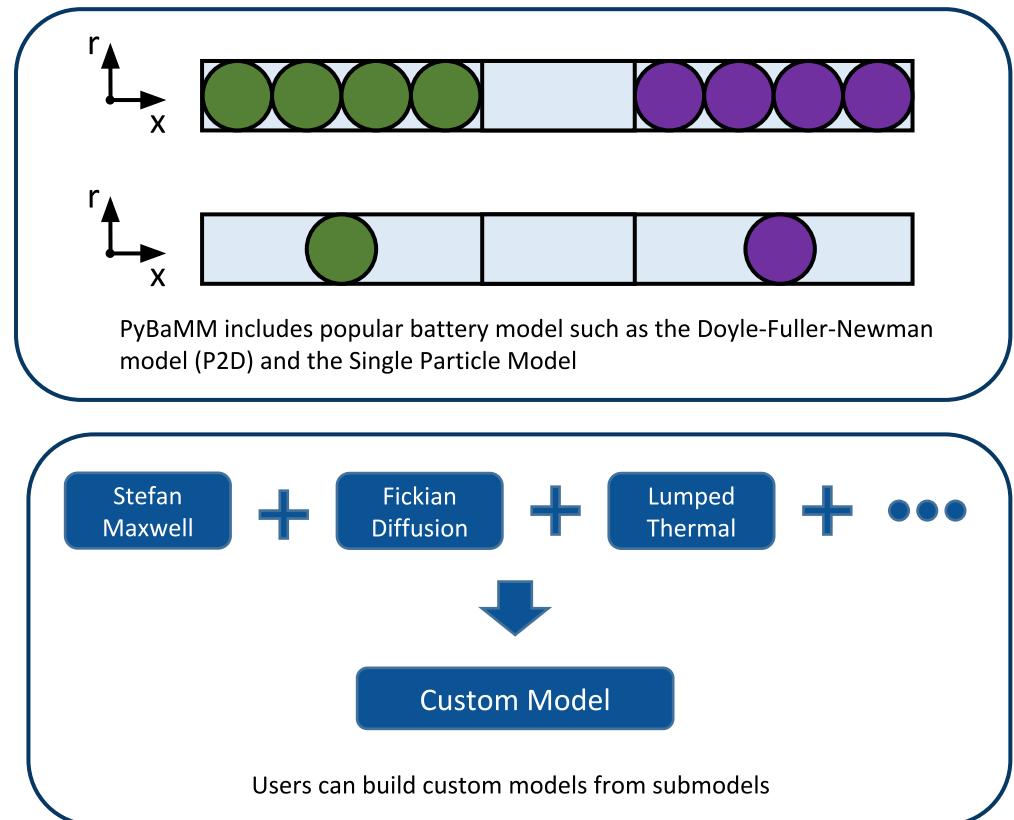
Many standard models already implemented and easy to reuse

- Cell geometries: 1D, 2+1D single layer pouch cells
- Standard lithium-ion models: ODE, SPM, SPMe, DFN plus lead-acid models
- Chemistries: LCO, LiNiCoO₂, NCA, Graphite, LiPF₆
- Fully coupled thermal models
- Experimental suite
 - Any voltage, current or power control (e.g. GITT, PITT, CCCV, drive cycle, etc.)
 - Easy interface to define new protocols
- Nonlinear parameters provided as functional form or as data
- Structured electrodes
- Interface with external software



Flexible Submodel Structure

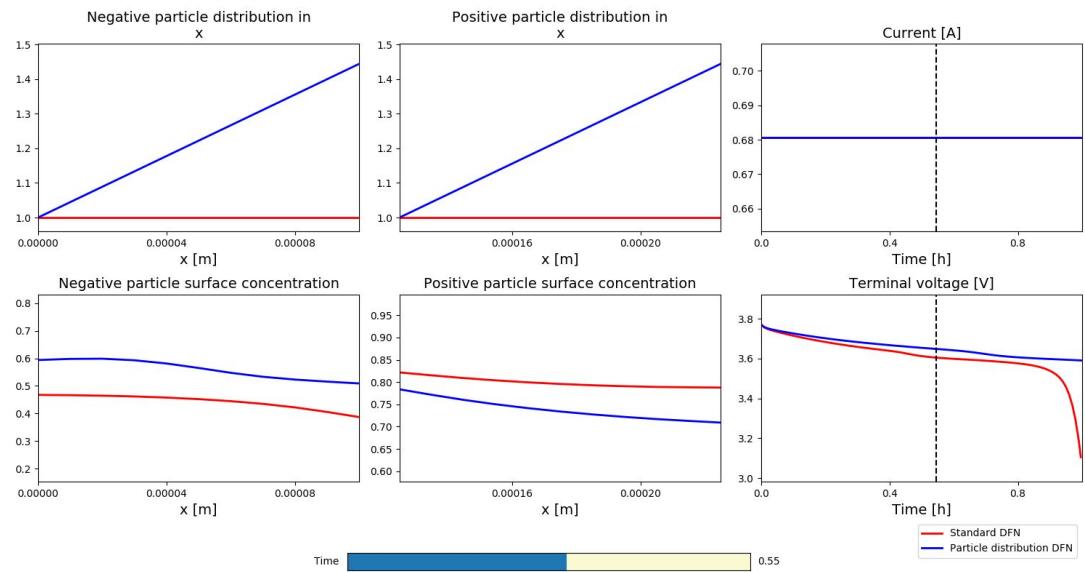
- Battery models are constructed from submodels
- Use or adapt pre-built models
- Create custom models





Flexible Submodel Structure

- Structure allows for “plug and play” physics
- New physics can be quickly implemented in an independent fashion
- Different physical mechanisms are readily included in both full and reduced-order models

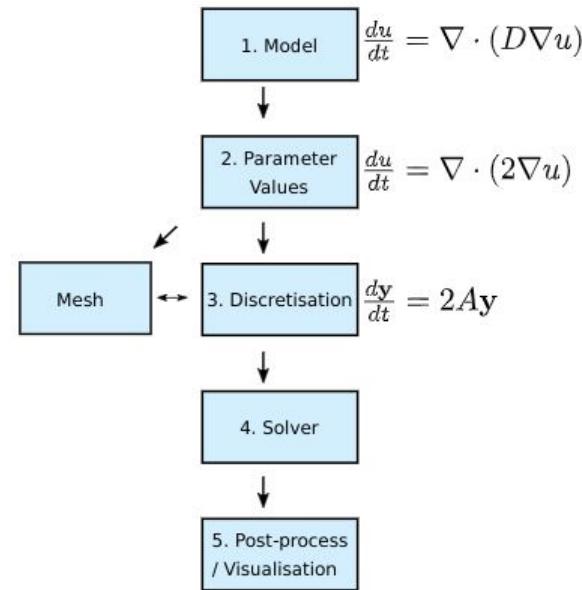


Using PyBaMM it is easy to compare how different physics affects the model solution. Here we investigate the influence of graded electrodes (particle distribution in the through-cell direction x).



Numerical Methods

- Choice of spatial discretisation
 - Finite Volume Method (macroscopic and particles)
 - Finite Element Method in 2D direction (current collectors in pouch cell models)
- Calculation of the analytic Jacobian from expression tree via automatic differentiation (using CasADi)
- Multiple ODE and DAE solvers (e.g. scipy, sundials CVODE and IDA)
- Ability to set up, solve and perform forward and adjoint sensitivity analysis



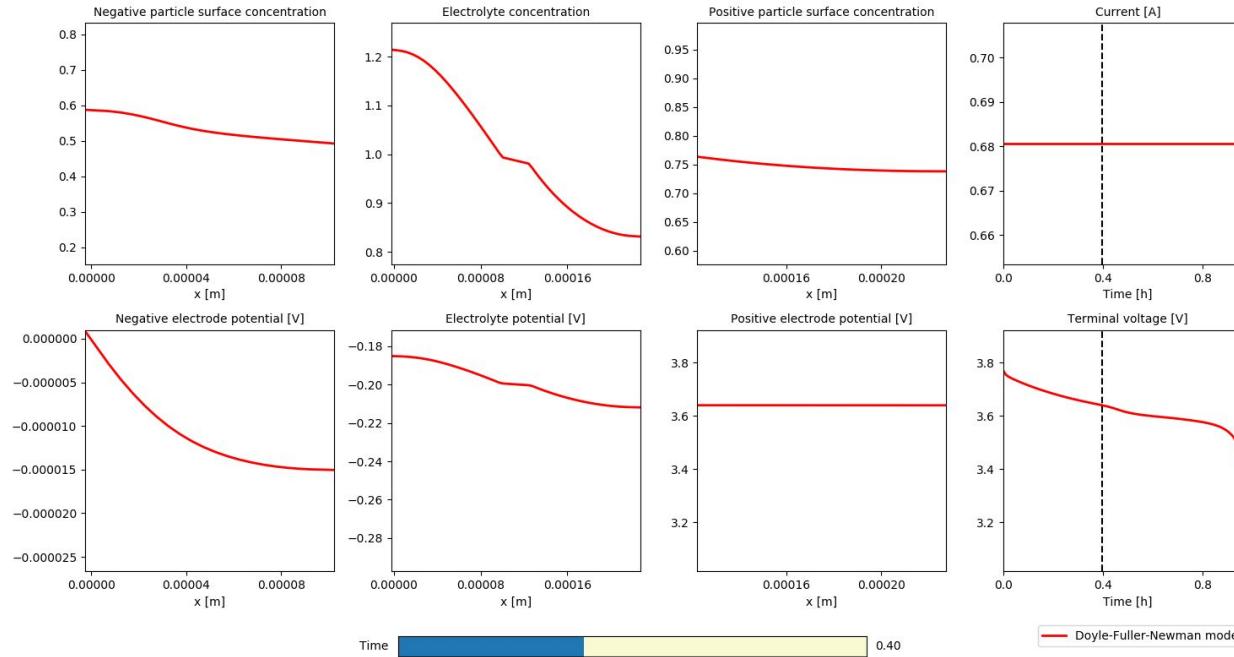
By employing a pipeline process PyBaMM separates the statement of model equations from the inputted parameters, choice of spatial discretization and choice of time stepping algorithm.



Simple Interface: CC Discharge

```
import pybamm
model = pybamm.lithium_ion.DFN()
sim = pybamm.Simulation(model)
sim.solve()
sim.plot()
```

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

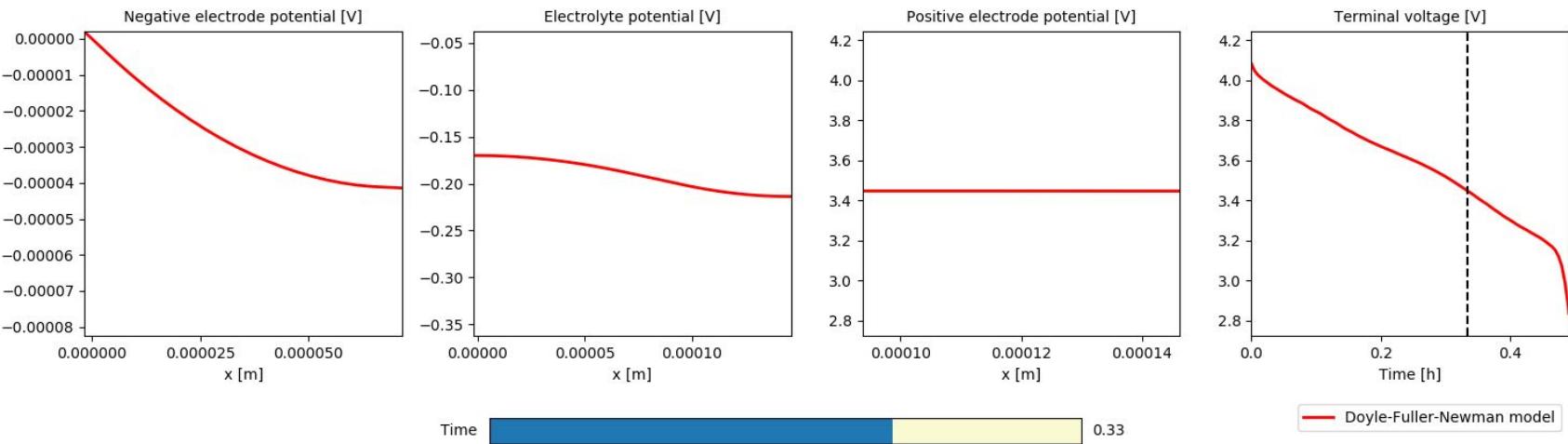




Simple Interface: Change parameters

```
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()
sim.plot()
```

- Easy to change parameters and model options



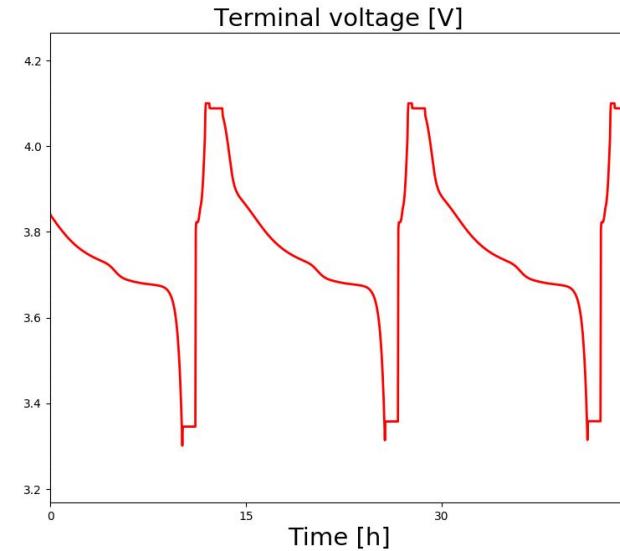


Simple Interface: Experimental protocol

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





Fast Solve Times

A comparison of typical solve times for different models and discretisations. Results are for a 1C constant current discharge using CasADI solver.

Grid points per domain	Single Particle Model		Doyle-Fuller-Newman Model	
	States	Time (milliseconds)	States	Time (milliseconds)
10	21	3	281	50
20	41	3	961	90
40	81	4	3251	250
80	161	5	13441	930
160	321	9	52481	4010

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

- 1) SG Marquis, V Sulzer, R Timms, CP Please and SJ Chapman. "An asymptotic derivation of a single particle model with electrolyte". Journal of The Electrochemical Society, 166(15):A3693–A3706, 2019



Fast Solve Times

A comparison of typical solve times for different solvers. Results are for a 1C constant current discharge.

Grid points per domain	States	CasADI (“fast” mode)	CasADI (“safe” mode)	IDA KLU	Scikits DAE
10	281	0.155s	0.361s	4.860s	2.102s
20	961	0.226s	0.763s	4.776s	10.120s
40	3521	0.566s	2.272s	6.992s	1m 47s
80	13441	1.977s	9.228s	11.642s	-
160	52481	8.865s	44.751s	34.289s	-

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

- 1) M. Ecker, T.K.D. Tran, P. Dechent, S. Kabitz, A. Warnecke, and D.U. Sauer. *Parameterization of a physico-chemical model of a lithium-ion battery. I. Determination of parameters*. Journal of The Electrochemical Society, 162 (2015), pp. A1836-A1848.
- 2) M. Ecker, S. Kabitz, I. Laresgoiti, and D.U. Sauer. *Parameterization of a physico-chemical model of a lithium-ion battery. II. Model validation*. Journal of The Electrochemical Society, 162 (2015), pp. A1849-A1857.



Fast Simulation of Drive Cycles

A comparison of typical solve times for different solvers. Results are based on the US06 drive cycle.

Grid points per domain	States	CasADI (“fast” mode)	CasADI (“safe” mode)	IDA KLU	Scikits DAE
10	281	3.089s	8.860s	40.621s	1m 14s
20	961	5.85s	19.730s	4m 16s	>10m
40	3521	16.729s	1m 8s	>10m	>10m
80	13441	58.220s	5m 9s	>10m	-
160	52481	4m 4s	>10m	>10m	-

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

- 1) M. Ecker, T.K.D. Tran, P. Dechent, S. Kabitz, A. Warnecke, and D.U. Sauer. *Parameterization of a physico-chemical model of a lithium-ion battery. I. Determination of parameters*. Journal of The Electrochemical Society, 162 (2015), pp. A1836-A1848.
- 2) M. Ecker, S. Kabitz, I. Laresgoiti, and D.U. Sauer. *Parameterization of a physico-chemical model of a lithium-ion battery. II. Model validation*. Journal of The Electrochemical Society, 162 (2015), pp. A1849-A1857.



Live Demo

- If you would like to follow along launch a jupyter notebook
- Ask for help on slack and we will try to answer live
- Many more examples online at examples.pybamm.org



Homework

Before the next session look at the examples online

examples.pybamm.org

- Try to modify the examples to do something **you** are interested in. For example:
 - try to run a GITT test (with C-rate and duration of your choice) for two different parameter sets and compare the output
 - investigate how the agreement between the SPM and DFN varies with applied current
- Ask for help on slack if you get stuck
- If you would like a new feature or find a bug create an issue on github

NEW FOR 2020



**High quality, in depth training on techniques,
equipment and facilities**
with experts in the field

See **Communifire** for sign up details

FARADAY TECH

#skills #techniques #training #equipment #facilities



Upcoming sessions:

- 4th June: Working with your IP advisers towards high quality patents
- 9th June: **PyBaMM Part 2**
- 11th June: IP from a commercial perspective