PyBOP: Python Battery Model Optimisation and Parameterisation

Brady Planden, Nicola Courtier, David Howey

Engineering Science
University of Oxford

Materials Research Society, Spring Meeting 2024

Battery Intelligence Lab

A framework for parameterisation and design optimisation

- Parameterise, optimise, and export your parameter sets within a single framework
- Any PyBaMM model, across various cost functions and optimisation algorithms

Model	DFN	SPMe	SPM
Time	4.5s	1.5s	1.3s

A framework for parameterisation and design optimisation

- Parameterise, optimise, and export your parameter sets within a single framework
- Any PyBaMM model, across various cost functions and optimisation algorithms

Data quality   Excitation   Data points

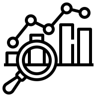
Motivation: Battery models are useful for

Physical Insights

How does the model predict future evolution?

How sensitivity is the battery to manufacturing variations?

Motivation: Battery models are useful for



Physical Insights

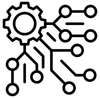
How does the model predict future evolution?

How sensitivity is the battery to manufacturing variations?

State Estimation

What is the current SoX of the cell?

How fast can I charge the cell?



Motivation: Battery models are useful for

Physical Insights
How does the model predict future evolution?
How sensitivity is the battery to manufacturing variations?

State Estimation
What is the current SoX of the cell?
How fast can I charge the cell?

Design Optimisation
How energy dense can the cell be?
What is the best design for this operation?

That sounds great, what's the catch?

That sounds great, what's the catch?

Parameter values

Parameter uniqueness is unknown
(in principle or in practice).

Conventionally requires cell teardown

Excitation signals are not obvious from
gathered test data

That sounds great, what's the catch?

Parameter values

Parameter uniqueness is unknown (in principle or in practice).

Conventionally requires cell teardown

Excitation signals are not obvious from gathered test data

Domain expertise

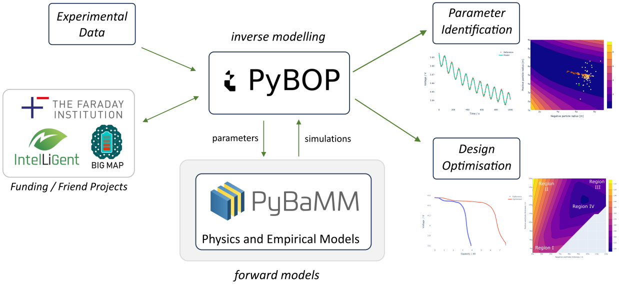
Is a more complex model needed?

Where should the complexity be added?

PyBOP is a platform to solve these challenges

Provides workflows for parameter identification and exploitation:

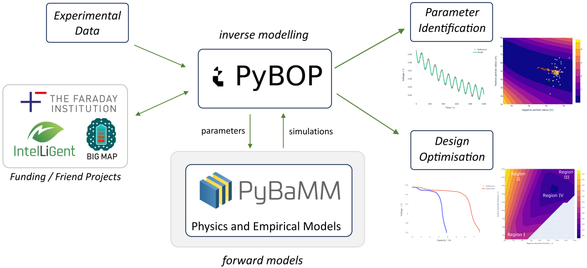
- Physics-based and empirical parameter identification
- Design optimisation capabilities
- Gradient & non-gradient optimisers
- Frequentist and Bayesian methods
- Multi-threaded computing



Multiprocessing enables this to be performant

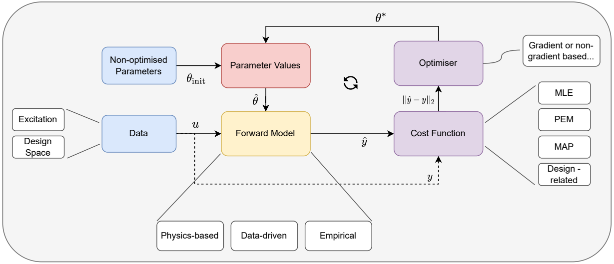
Multiprocessing is built into the package to speed up large parameter space optimisation

- $\sim$1.7x improvement for 4 cores
- $\sim$6.6x improvement for 12 cores



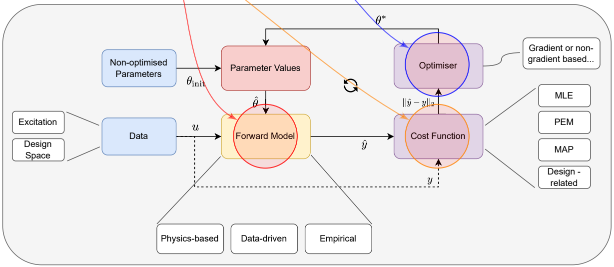
This is achieved through concise, intuitive design

```python
# Generate problem, cost function, and optimisation class
problem = pybop.Problem(model, parameters, dataset)
cost = pybop.SumSquaredError(problem)
optim = pybop.Optimisation(cost, optimiser=pybop.GradientDescent)
x, final_cost = optim.run()
```



This is achieved through concise, intuitive design

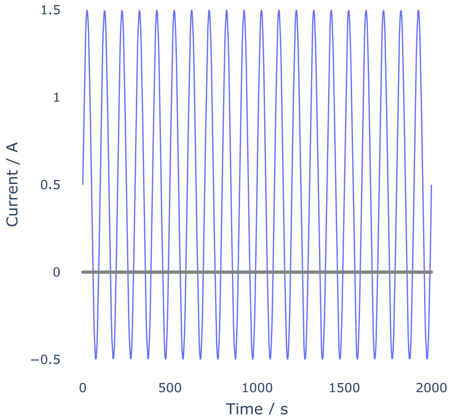
```python
# Generate problem, cost function, and optimisation class
problem = pybop.Problem(model, parameters, dataset)
cost = pybop.SumSquaredError(problem)
optim = pybop.Optimisation(cost, optimiser=pybop.GradientDescent)
x, final_cost = optim.run()
```



Parameterisation operating conditions

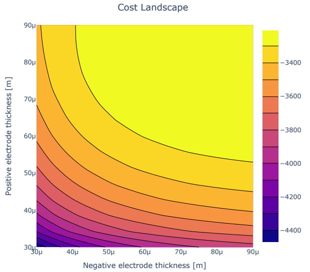
For the upcoming parameterisation examples,

- Biased Sinusoid Applied Current
- Synthetic data from Many-Particle DFN
- $T_0 = 298.15$ K, $SOC_0 = 100\%$
- NMC811–Gr/SiOx Chemistry

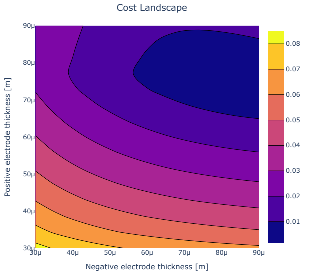


PyBOP provides multiple cost functions for parameter inference

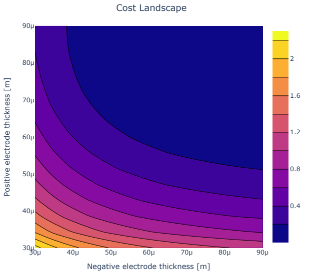
Likelihood-based



Root Mean Squared Error

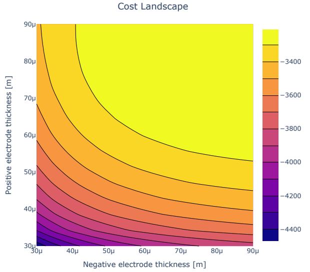


Sum of Squared Error

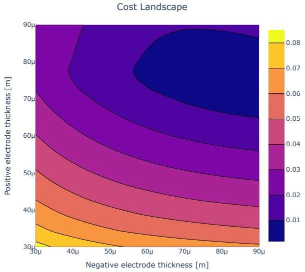


PyBOP provides multiple cost functions for parameter inference

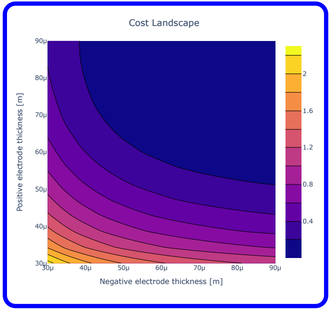
Likelihood-based



Root Mean Squared Error



Sum of Squared Error



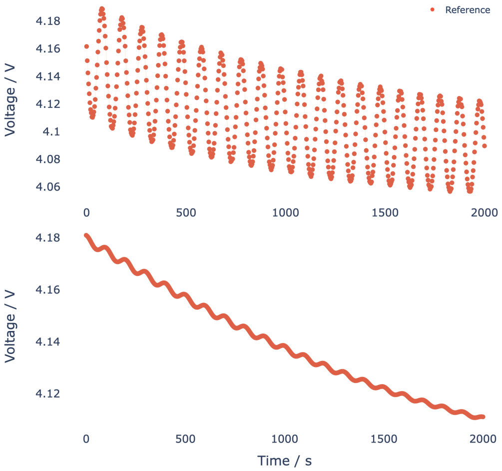
Example: Multi-signal parameter identification

Active Material Volume Fractions (Mass Loading) for each electrode[a] are identified from synthetic MP-DFN data.

- Fitting signals: Terminal voltage and predicted OCV
- Initial conditions $\sim \mathcal{N}(\mu, \sigma^2)$
- Maximum iters: 150

Ground truth parameter values:

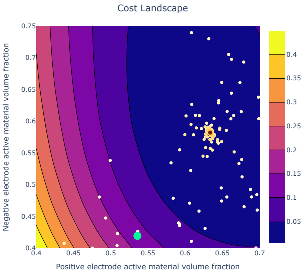
$$[\varepsilon_\mathrm{n} = 0.75, \ \varepsilon_\mathrm{p} = 0.665]$$



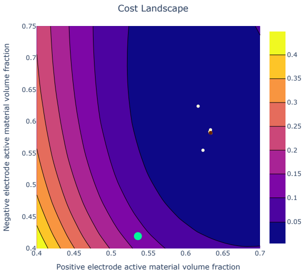
[a]All models are tested daily on the cloud!

Verfiying optimality with convergence plots

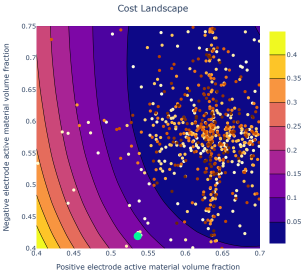
CMA-ES



Differential Evolution

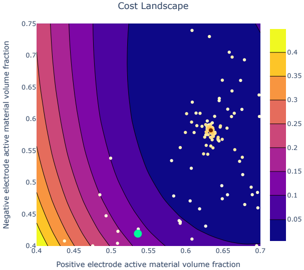


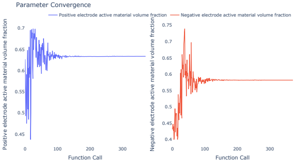
Particle Swarm



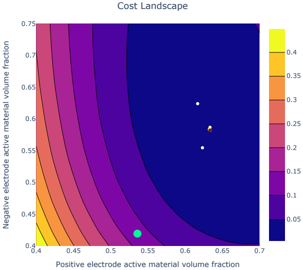
Verfiying optimality with convergence plots

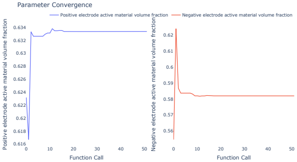
CMA-ES



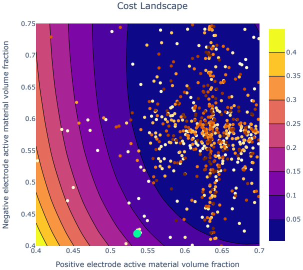


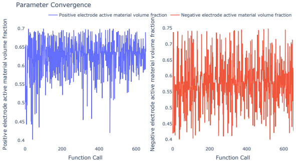
Differential Evolution





Particle Swarm

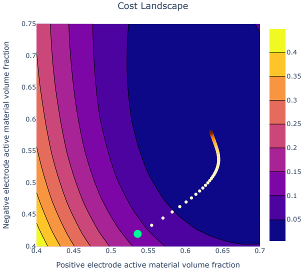


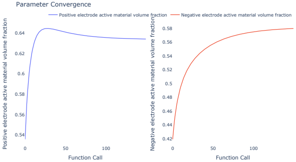


Gradient-based optimisers offer a deterministic solution

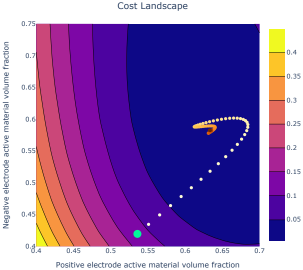
UNIVERSITY OF OXFORD

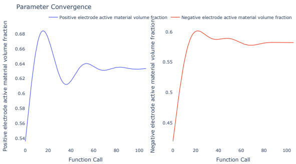
Gradient Descent



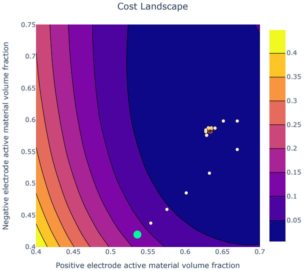


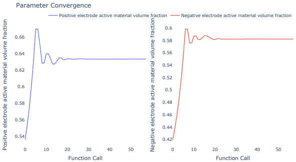
Adam





IRPropMin





Example: Multi-signal parameter identification

With the correct excitation, parameter identification can be fast and robust. However, challenges still exist,

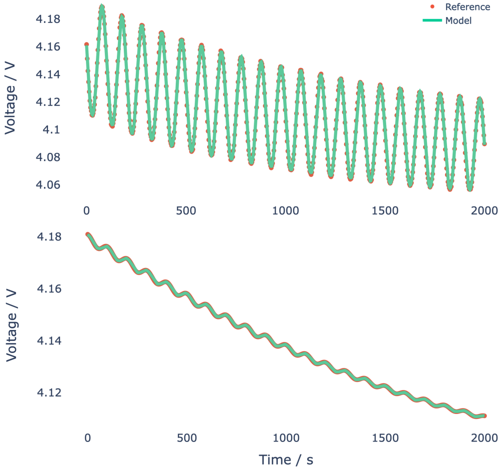
- Missing physics in forward model
- Metadata quality

Parameter values:

True: $[\varepsilon_{\mathrm{n}} = 0.75,\ \varepsilon_{\mathrm{p}} = 0.665]$

SPMe: $[\varepsilon_{\mathrm{n}} = 0.75,\ \varepsilon_{\mathrm{p}} = 0.656]$

SPM: $[\varepsilon_{\mathrm{n}} = 0.595,\ \varepsilon_{\mathrm{p}} = 0.631]$



Example: Physics-based design optimisation

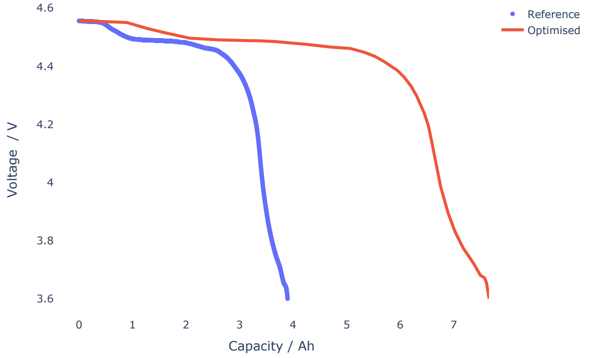
Chemistry: LNMO-Graphite/SiOx

Target: Maximize cell-level specific energy density

Parameters: Electrode coating thickness

Conditions: 1C cycling

Optimal:
$[L_n = 97\text{e-}06, \; L_p = 126\text{e-}06]$



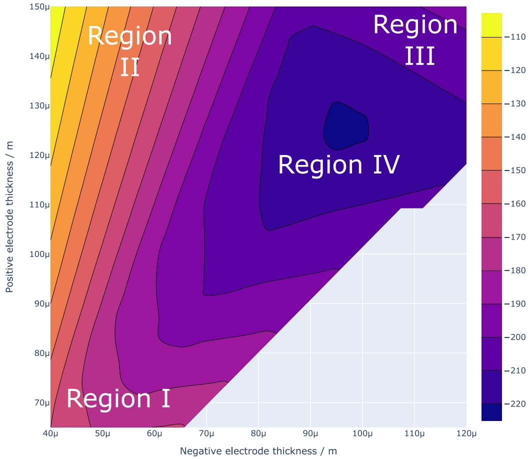
PyBOP enables physical insights into design optimisation

Region I: Electrodes too thin compared to inactive material

Region II: Poor N:P balancing

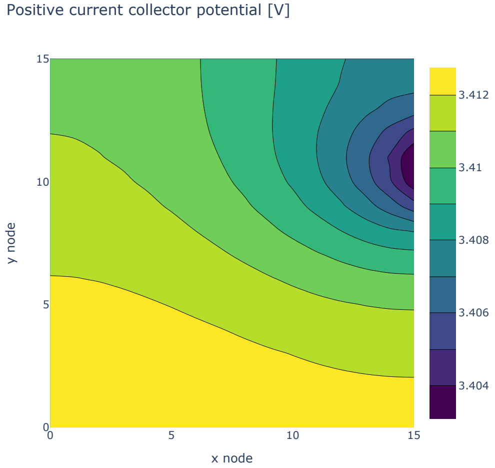
Region III: Electrodes too thick and cause transport limitations

Region IV: Ideal N:P balance and optimised wrt active material and transport



Spatial models enable larger-scale effects

- Spatial models[b] are available for application-specific design optimisation
- Enables design optimisation (tab placement, electrolyte composition, etc.) workflows from spatial variables (thermal, lithiation, etc.)



[b] Marquis et al. J. Electrochem. Soc., 2020

Conclusion and Outlook

- We presented a framework for optimisation and parameterisation of battery models
- This framework offers many optimisers, cost functions, and battery models
- Example workflows were presented covering parameterisation and optimisation

Conclusion and Outlook

- We presented a framework for optimisation and parameterisation of battery models
- This framework offers many optimisers, cost functions, and battery models
- Example workflows were presented covering parameterisation and optimisation

Future Development:

- Monte Carlo and Bayesian Quadrature
- Integration of EIS predictions
- Improved workflows for battery relevant model optimisation
- v24.5 upcoming!

Get involved and become a developer and/or user!

GitHub Repository:

pybop-team/pybop

Installation:

```
pip install pybop
```

Examples **Docs** **Benchmarks**

Contact Info:

bradyplanden

bradyplanden