The Journal of Open Source Software

BattMoApp: A Web-Based application for running cell-level battery simulations

Lorena Hendrix[1], **Simon Clark***[1], **Oscar Bolzinger**[1], **and Eibar Flores**[1]

1 SINTEF Industry, Norway

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

BattMoApp is a web-based application built upon the command-line based battery modelling software, BattMo (Raynaud et al., 2024). It features a user-friendly graphical interface that simplifies the simulation of battery cells. The development of BattMoApp has been centered on accessibility, intuitiveness, and usability, with the aim of making it a practical and valuable tool for both educational and research purposes in the battery field. Its design allows users to simulate, obtain, analyze, and compare results within just a few minutes. While BattMoApp leverages a small yet crucial portion of BattMo's capabilities, its intuitive and explanatory design also makes it an ideal starting point for those looking to explore the more comprehensive and complex BattMo software.

Statement of need

The Battery Modelling Toolbox (BattMo) is a framework for continuum modeling of electrochemical devices. Built primarily in MATLAB (MATLAB, 2023), it offers a pseudo X-dimensional (PXD) framework for the Doyle-Fuller-Newman model of lithium-ion battery cells. Additionally, extensions for other battery chemistries and hydrogen systems are in development. BattMo provides a flexible framework for creating fully coupled electrochemical-thermal simulations of electrochemical devices using 1D, 2D, or 3D geometries. Besides the MATLAB toolbox, the framework is also being developed in Julia (Bezanson et al., 2017) to leverage increased simulation speed and the non-proprietary nature of Julia.

The primary objective of BattMoApp is to unlock access to battery simulations for users without coding experience. Most researchers who can benefit from battery simulations are not familier with any scripting language. A simple to use graphical user interface brings battery simulations to laboratory engineers and battery scientists who can use the results to inform their cell design and development activities.

BattMoApp builds upon the P2D model implemented in the Julia version of BattMo, see Raynaud et al. (2024). The development of BattMoApp has focused on accessibility, intuitiveness, and usability. Users can quickly and easily obtain results using the default input parameter sets available in the application or input their own values in a straightforward manner. The results can be easily analyzed and compared using the predefined plots that can handle multiple sets of simulation results. Users can also download their results and later upload them back into the application to review. Furthermore, significant effort has been made to ensure the parameters are realistic for both computational research and lab use, making it easier for experimentalists to fulfill the necessary inputs.

Another important aspect that was kept in mind during the development of BattMoApp is interoperability. To ensure that the input data of the simulation is inter-operable, the selected data format adheres to the FAIR principles (Wilkinson et al., 2016) and the 5-star open data guidelines (Berners-Lee, 2006). The data entered by the user is automatically formatted into a JSON (*JSON*, 2017) Linked Data (LD) format which includes all the semantic metadata along with the actual data. This semantic data connects the actual data to the ontology documentations, EMMO (EMMO, 2024) and BattINFO (Clark et al., 2022), which contain descriptions of the linked data definitions. If the user wishes to publish their results, they can include the JSON LD file in their publication, allowing anyone seeking to replicate the results to simply upload the JSON LD file into BattMoApp and obtain the anticipated results.

The documentation of BattMoApp includes an overview on what the application has to offer and a troubleshooting section that provides insights into the relationship between input parameters and results. As BattMoApp has a graphical and explanatory nature, it can also be a powerful tool for educational purposes, helping students understand batteries, battery modelling, and the impact of material and cell design parameters on battery cell performance.

Technical setup

The application consists of two main components: the graphical user interface (GUI), which includes the frontend, a database, and the backend that provides the frontend's functionality, and the application programming interface (API) which runs the BattMo software on a web socket server. These two components are isolated from each other, each running in its own Docker (Docker, 2024) container.

BattMo GUI

The frontend is Python-based (Rossum & Python Development Team, 2024) and developed using the Streamlit framework (Allaire et al., 2024). Streamlit was chosen due to its user-friendly framework that greatly accelerates the development process. The database that supports the frontend is created using the sqlite3 Python package (Hipp, 2024).

BattMo API

The web socket API runs BattMo.jl. Integrating Julia, a pre-compiled language, with Python, a runtime language, to form a smoothly running and stable application turned out to be complex. Therefore, a Julia-based web socket API was created and containerized within a separate Docker container, isolating the Julia environment from the python environment. The web socket API has been created using the Julia package HTTP (JuliaWeb, 2024) and includes a system image of BattMo's pre-compilation to ensure an instantanious API response.

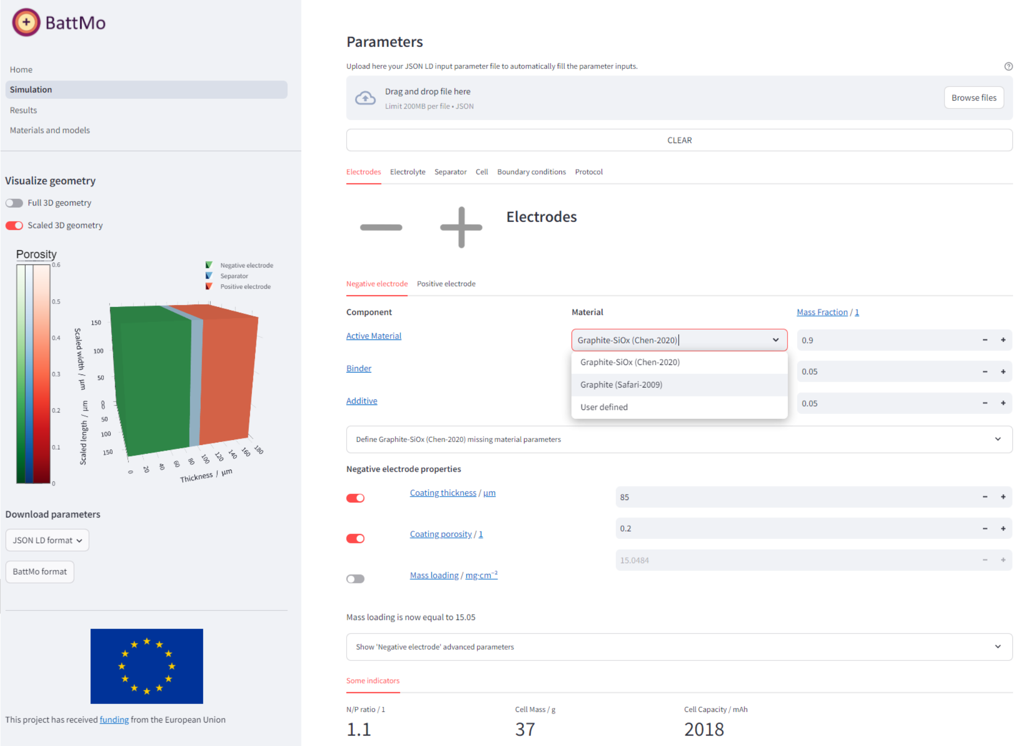
Examples

The application provides a list of features:

- Access to parameter sets from literature, and customization of these sets.
- Download input parameter values as a BattMo formatted JSON file or as a linked data formatted JSON file.
- Upload previously downloaded input parameters to quickly review and alter previous simulations.
- Visualization of 3D grids.
- P2D model.
- Quick calculations of key indicators.
- Interactive plots that zoom, hover, and downloads PNGs.

- Visualization of not only voltage curves but also internal states to provide further insight into gradients of concentrations and potentials.
- Comparison of multiple simulation results.
- Download full results in HDF5 format.
- Upload previously downloaded results to quickly review them.

The following figures display screenshots of the application's 'Simulation' and 'Results' pages. On the 'Simulation' page, Figure 1, users can define input parameters, visualize their cell geometry, and initiate a simulation. The 'Results' page then allows for the visualization of simulation outcomes using predefined plots. In Figure 2, the results of two simulations are visualized to show an example. The two simulations were conducted using the default parameter sets retrieved from Chen et al. (2020) with the electrode coating thicknesses varied to illustrate a comparison of results. Both pages also present key indicators of the battery cell, such as capacities, cell energy, and round-trip efficiency.



Upload here your JSON LD input parameter file to automatically fill the parameter inputs.

Component	Material	Mass Fraction / 1
Active Material	Graphite-SiOx (Chen-2020)	0.9
Binder		0.05
Additive		0.05

Define Graphite-SiOx (Chen-2020) missing material parameters

Negative electrode properties

●	Coating thickness / µm	85
●	Coating porosity / 1	0.2
○	Mass loading / mg·cm^{-2}	15.0484

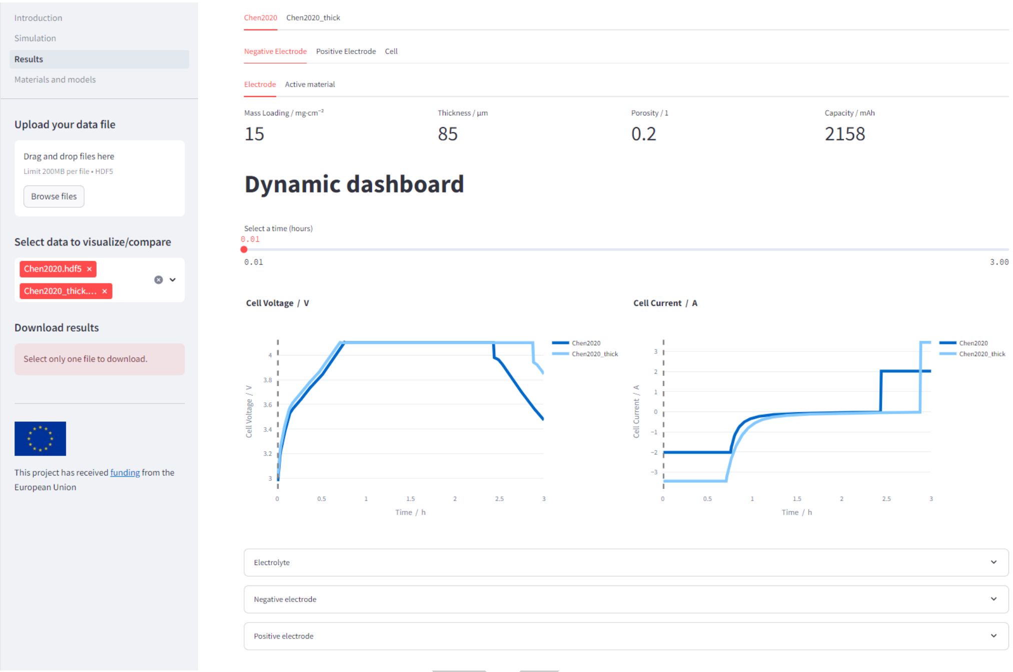
Mass loading is now equal to 15.05

Show 'Negative electrode' advanced parameters

N/P ratio / 1	Cell Mass / g	Cell Capacity / mAh
1.1	37	2018

This project has received funding from the European Union

Figure 1: A screenshot of the Simulation page of BattMoApp.



Upload your data file

Drag and drop files here

Limit 200MB per file • HDF5

Select data to visualize/compare

Download results

Select only one file to download.

This project has received funding from the European Union

Mass Loading / mg·cm^{-2}	Thickness / µm	Porosity / 1	Capacity / mAh
15	85	0.2	2158

Dynamic dashboard

Select a time (hours): 0.01 (range 0.01 – 3.00)

Figure 2: A screenshot of the Results page of BattMoApp.

Future work

While BattMoApp has reached a mature state and offers a valuable platform for P2D simulations, there are still countless possibilities for further development. Its evolution will continue in parallel with BattMo.jl, allowing for the future integration of additional simulation models and features like parameterization. BattMoApp will continue seeking feedback from its target audience to enhance usability and practicality. Additionally, more effort will be dedicated to improving the performance of the BattMo GUI to improve user interactivity.

Installation

`BattMoApp` can easily be used online at the following address: app.batterymodel.com. Furthermore, it can be installed locally using Docker. The Docker images and a detailed instruction on how to install `BattMoApp` locally can be found in the Github repository.

Acknowledgements

The authors acknowledge Francesca Watson and Olav Møyner for their guidance and support. BattMoApp has received funding from the European Union's Horizon 2020 innovation program under grant agreement no. 875527 (HYDRA), no. 957189 (BIG-MAP), no. 101069765 (IntelLiGent), no. 101104013 (BATMAX), and no. 101103997 (DigiBatt).

References

Allaire, A., Brondex, A., Plotkin, R., & others. (2024). *Streamlit* (Version 1.39.0). Streamlit Inc. https://www.streamlit.io/

Berners-Lee, T. (2006). *Star open data*. https://www.w3.org/DesignIssues/LinkedData.html

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Chen, C.-H., Planella, F. B., O'regan, K., Gastol, D., Widanage, W. D., & Kendrick, E. (2020). Development of experimental techniques for parameterization of multi-scale lithium-ion battery models. *Journal of The Electrochemical Society*, *167*(8), 080534.

Clark, S., Bleken, F. L., Stier, S., Flores, E., Andersen, C. W., Marcinek, M., Szczesna-Chrzan, A., Gaberscek, M., Palacin, M. R., Uhrin, M., & Friis, J. (2022, May). *Toward a unified description of battery data*. https://doi.org/10.1002/aenm.202102702

Docker, Inc. (2024). *Docker* (Version v4.36.0). https://www.docker.com

EMMO. (2024). *Elementary multiperspective material ontology (EMMO)*. https://github.com/emmo-repo/

Hipp, D. R. (2024). *SQLite* (Version v3.45.3). https://sqlite.org/

JSON. (2017). https://www.json.org/

JuliaWeb. (2024). *HTTP.jl* (Version v1.10.14). https://doi.org/10.5281/zenodo.11032263

MATLAB. (2023). *MATLAB* (Version R2023b). The MathWorks Inc. https://se.mathworks.com/

Raynaud, X., Johansson, A., Nilsen, H. M., Watson, F., Clark, S., & Erasdna. (2024). *BattMoTeam/BattMo: v0.3.0* (Version v0.3.0). Zenodo. https://doi.org/10.5281/zenodo.10633682

Rossum, G. van, & Python Development Team, the. (2024). *Python* (Version v3.12.0). https://www.python.org/

Wilkinson, M. D., Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., Silva Santos, L. B. da, Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., … Mons, B. (2016). The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, *3*, 160018. https://doi.org/10.1038/sdata.2016.18