The Journal of Open Source Software

EchemFEM: A Firedrake-based Python package for electrochemical transport

Thomas Roy [1,¶], **Julian Andrej** [1], **Aymeric Antimes** [2], **Victor A. Beck** [1], **Victoria Ehlinger** [1], **Florian Euzenat** [2], **Nitish Govindarajan** [1], **Jack Guo** [1], **Tiras Y. Lin** [1], and **Thomas Moore** [3]

1 Lawrence Livermore National Laboratory, CA, USA **2** TotalEnergies OneTech, France **3** Queensland University of Technology, Australia ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 01 March 2024
Published: unpublished

Summary

The transition from fossil fuels to renewable energy has brought about a rapid increase in the availability of clean electricity. However, electricity generated from sources such as wind and solar are limited to intermittent operation due to daily and seasonal variation. One solution is to utilize electrochemical devices in energy storage and electrochemical manufacturing applications, where they can harness surplus energy and decarbonize chemical industries traditionally reliant on petrochemical feedstocks. Managing the growing prevalence of renewable energy underscores the importance of developing and scaling up these technologies, which can in turn facilitate the achievement of carbon emission reduction commitments of companies and developed economies. Likewise, the electrification of transport creates an increasing need for energy-dense electrochemical energy storage devices such as batteries and supercapacitors. Naturally, simulation tools are required to assist in the design of efficient and industrial-scale electrochemical devices.

Modeling and simulation are used extensively to describe the physics of the electrochemical and transport mechanisms in electrochemical devices. These devices have many applications, from miniaturized lithium-ion batteries for medical devices up to industrial-scale hydrogen fuel cells for backup power generation. Energy storage devices include batteries and supercapacitors, as well as flow batteries, which utilize a flowing electrolyte instead of a stationary liquid or polymer electrolyte. Electrolyzers are devices that use electrical energy to perform electrochemical reactions. Some current industrial applications for electrolysis include the chlor-alkali process for the production of chlorine gas and the Hall-Héroult process for aluminum production. Active areas of research include the development of electrolyzers that transform carbon dioxide into useful chemicals such as base molecules for sustainable aviation fuels or the chemical industry, as well as electrolyzers that create hydrogen from water. In the reverse process, fuel cells use fuels such as hydrogen to generate electricity. While electrochemical devices span many scales and industries, the governing equations and underlying physical phenomena remain similar.

The transport of charged chemical species in a fluid is often modeled using the Nernst-Planck equation, which includes the usual advection and diffusion transport as well as *electromigration*, where charged species are transported by an electric field. Often, these species are also undergoing reactions either in the bulk fluid or on the boundaries.

EchemFEM provides a high-level user interface for a finite element implementation of the Nernst-Planck equation. The user is simply required to provide physical parameters as well as functions describing the chemical reactions (charge-transfer or bulk reactions). The mesh can be defined using either built-in functions for simple geometries, or imported from external packages, such as Gmsh (Geuzaine & Remacle, 2009), for more complex geometries. Then, the desired transport physics are selected using keyword arguments. Ionic charge can be modeled using either the Poisson equation or the electroneutrality approximation. The simulated devices can have resolved electrolyte-electrode interfaces or homogenized porous electrodes, in which case electron conduction is also modeled. Additionally, finite size effects are available, which includes models such as Generalized Modified Poisson-Nernst-Planck (GMPNP) (Wang et al., 2013). Lastly, a fluid flow solver for the incompressible Navier-Stokes and Navier-Stokes-Brinkman equations is provided.

EchemFEM is based on Firedrake (Ham et al., 2023), an open-source finite element package, enabling straightforward implementation of the governing equations in Python. Firedake has access to scalable, customizable, solvers through its interface with PETSc (Balay et al., 2023b, 2023a), allowing for parallelization and scalability on computing clusters. This balance between usability and scalability permits a seamless transition from prototyping to large-scale simulation. EchemFEM leverages Firedrake's capabilities while further increasing the ease-of-use. Indeed, since the governing equations are already implemented, little to no knowledge of Firedrake and the finite element method is required to use EchemFEM.

The repository includes several examples of electrochemical devices such as flow reactors, flow batteries, and CO2 electrolyzers.

Statement of need

Electrochemical phenomena are highly complex, making characterization of electrochemical devices through in-operando experiments challenging. Simulation is an important tool for predicting the performance of electrochemical devices, as well as assisting in their design. As technologies get scaled up from the laboratory scale to industrial scale, experiments become less tractable and therefore simulation increasingly important. Naturally, the scalability of simulators is crucial. Furthermore, many existing models and codes are just one dimensional. To capture the effects of fluid flow and non-monolithic, architected electrodes, higher-dimensional effects do matter. For three-dimensional systems, iterative methods and appropriate preconditioners are required to maintain scalability.

Currently, commercial software are the most commonly used codes for electrochemistry simulations. COMSOL Multiphysics®, with its detailed electrochemistry module, is the most popular, while Simcenter™ STAR-CCM+™ is also used commonly for flowing systems. These programs provide simple graphical user interfaces (GUI), which allow users to quickly set up new simulations. Additionally, other physics modules such as fluid dynamics are available and can usually be coupled with the electrochemistry simulation. However, there are several drawbacks to using such commercial software. For instance, license fees can be prohibitively expensive, therefore limiting collaboration. Furthermore, the closed nature of the source code limits the flexibility of the software. Indeed, it is not possible to implement new discretization schemes and preconditioning approaches that may be required for numerical stability or scalability, respectively. Finally, since everything needs to be set up through the GUI, scripting and coupling to other software are difficult tasks.

There is a growing number of open-source software for electrochemistry, especially Python-based packages (Zheng, 2023), many of which are specialized for specific applications, notably batteries. One such package, PyBaMM (Sulzer et al., 2021), is a battery modelling code with a flexible implementation, allowing for new models and numerical methods to be tested. OpenFOAM (*OpenFOAM*, 2024) is a popular tool that is mainly used for computational fluid dynamics, but implementation of custom transport mechanisms, such as those from electrochemistry, can have a steep learning curve.

EchemFEM provides a general framework for simulating electrochemical transport: it is not specific to an application. Since it is based on Firedrake, any additional physics that can be implemented in a finite element framework can be coupled to EchemFEM. In one of the demos, the incompressible Navier-Stokes equations are solved in a reactor with an irregular surface, providing a velocity field for the transport equations. Similarly, in a flow battery example, the Navier-Stokes-Brinkman equations are solved.

In some cases, for example for fast flows, stabilization schemes that are not offered in other software may be required. For continuous Galerkin (CG) elements, a streamline-upwind Petrov-Galerkin (SUPG) method for the Nernst-Planck equation is provided. For discontinuous Galerkin (DG), a custom upwind scheme for the Nernst-Planck equation is used (Roy et al., 2023). In both cases, the upwinding considers the combined advection-migration "velocity".

As opposed to commercial software, custom scalable solvers are available in Firedrake. A plethora of solver options are available through simple PETSc keywords and custom operators for preconditioning can be defined using Firedrake (Mitusch et al., 2019). In Roy et al. (2023), scalable block preconditioners were developed for the electroneutral Nernst-Planck equations with DG and implemented in EchemFEM.

Combining EchemFEM with other Python packages is rather simple. In Govindarajan et al. (2023), multi-scale simulations for CO2 reduction in flow reactors are performed by coupling a microkinetics model from CatMAP (Medford et al., 2015) with the GMPNP transport model from EchemFEM. The simulations are two orders of magnitude faster than a previous implementation where the transport is done in COMSOL Multiphysics®, due to the tedious interface between the commercial software and CatMAP.

Firedrake's automatic adjoint capabilities facilitate the straightforward solution of PDE-constrained optimization problems (Mitusch et al., 2019), already employed in electrochemical applications (Reale Batista et al., 2023; Roy et al., 2022). We are currently investigating optimization problems using EchemFEM.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory (LLNL) under Contract DE-AC52-07NA27344, and was partially supported by a Cooperative Research and Development Agreement (CRADA) between LLNL and TotalEnergies American Services, Inc. (affiliate of TotalEnergies SE) under agreement number TC02307 and Laboratory Directed Research and Development (LDRD) funding under projects 19-ERD-035 and 22-SI-006. LLNL Release Number LLNL-JRNL-860653-DRAFT.

References

Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E. M., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., … Zhang, J. (2023a). *PETSc Web page*. https://petsc.org/. https://petsc.org/

Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., … Zhang, J. (2023b). *PETSc/TAO users manual* (ANL-21/39 - Revision 3.20). Argonne National Laboratory. https://doi.org/10.2172/1968587

Geuzaine, C., & Remacle, J.-F. (2009). Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International Journal for Numerical Methods in Engineering*, *79*(11), 1309–1331.

Govindarajan, N., Lin, T. Y., Roy, T., Hahn, C., & Varley, J. B. (2023). Coupling microkinetics with continuum transport models to understand electrochemical CO_2 reduction in flow reactors. *PRX Energy*, *2*(3), 033010. https://doi.org/10.1103/PRXEnergy.2.033010

Ham, D. A., Kelly, P. H. J., Mitchell, L., Cotter, C. J., Kirby, R. C., Sagiyama, K., Bouziani, N., Vorderwuelbecke, S., Gregory, T. J., Betteridge, J., Shapero, D. R., Nixon-Hill, R. W., Ward, C. J., Farrell, P. E., Brubeck, P. D., Marsden, I., Gibson, T. H., Homolya, M., Sun, T., … Markall, G. R. (2023). *Firedrake user manual* (First edition). Imperial College London; University of Oxford; Baylor University; University of Washington. https://doi.org/10.25561/104839

Medford, A. J., Shi, C., Hoffmann, M. J., Lausche, A. C., Fitzgibbon, S. R., Bligaard, T., & Nørskov, J. K. (2015). CatMAP: A software package for descriptor-based microkinetic mapping of catalytic trends. *Catal. Lett.*, *145*(3), 794–807. https://doi.org/10.1007/s10562-015-1495-6

Mitusch, S. K., Funke, S. W., & Dokken, J. S. (2019). Dolfin-adjoint 2018.1: Automated adjoints for FEniCS and Firedrake. *Journal of Open Source Software*, *4*(38), 1292. https://doi.org/10.21105/joss.01292

OpenFOAM. (2024). www.openfoam.org

Reale Batista, M. D., Chandrasekaran, S., Moran, B. D., Salazar de Troya, M., Pinongcos, A., Wang, Z., Hensleigh, R., Carleton, A., Zeng, M., Roy, T., Lin, D., Xue, X., Beck, V. A., Tortorelli, D. A., Stadermann, M., Zheng, R., Li, Y., & Worsley, M. A. (2023). Design and additive manufacturing of optimized electrodes for energy storage applications. *Carbon*, *205*, 262–269. https://doi.org/10.1016/j.carbon.2023.01.044

Roy, T., Andrej, J., & Beck, V. A. (2023). A scalable DG solver for the electroneutral Nernst-Planck equations. *Journal of Computational Physics*, *475*, 111859. https://doi.org/10.1016/j.jcp.2022.111859

Roy, T., Salazar de Troya, M. A., Worsley, M. A., & Beck, V. A. (2022). Topology optimization for the design of porous electrodes. *Structural and Multidisciplinary Optimization*, *65*(6), 171. https://doi.org/10.1007/s00158-022-03249-2

Sulzer, V., Marquis, S. G., Timms, R., Robinson, M., & Chapman, S. J. (2021). Python battery mathematical modelling (PyBaMM). *Journal of Open Research Software*, *9*(1).

Wang, H., Thiele, A., & Pilon, L. (2013). Simulations of cyclic voltammetry for electric double layers in asymmetric electrolytes: A generalized modified Poisson–Nernst–Planck model. *The Journal of Physical Chemistry C*, *117*(36), 18286–18297. https://doi.org/10.1021/jp402181e

Zheng, W. (2023). Python for electrochemistry: A free and all-in-one toolset. *ECS Advances*, *2*(4), 040502. https://doi.org/10.1149/2754-2734/acff0b