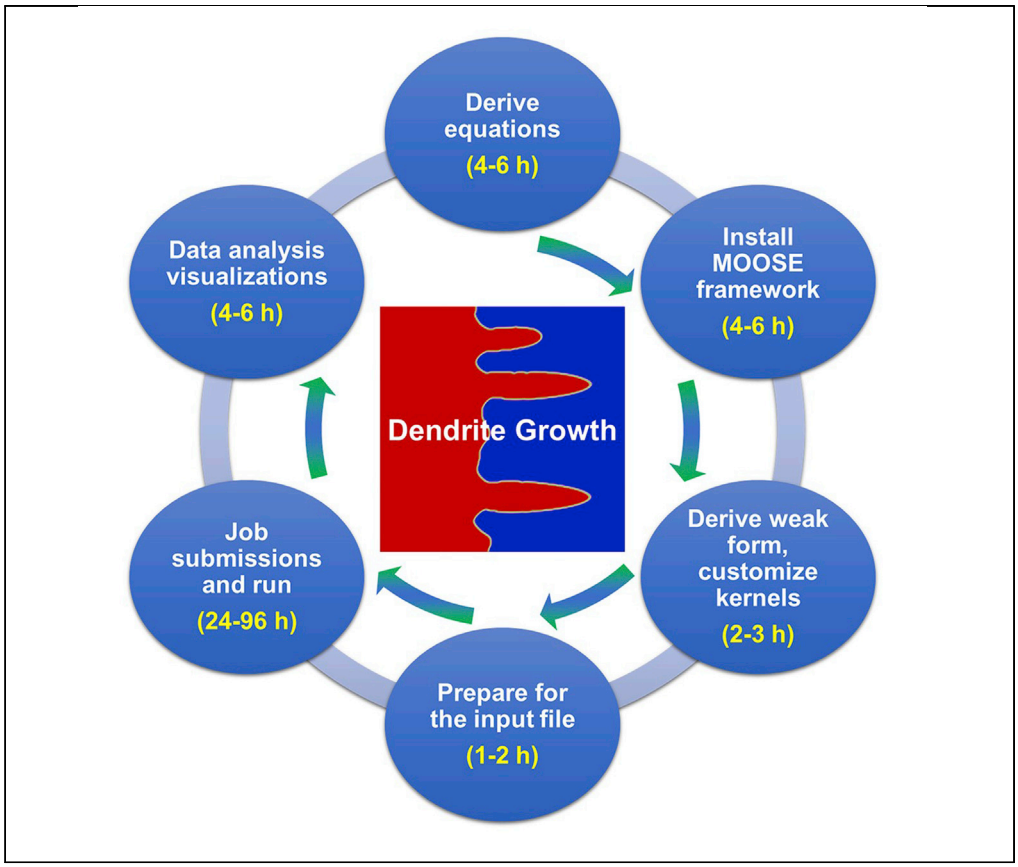
STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol for phase-field simulations of lithium dendrite growth with MOOSE framework



Phase-field simulation is a powerful tool for understanding lithium metal electrodeposition. This protocol outlines the process of numerically solving the phase-field equations using the MOOSE framework. Here, we describe steps to obtain the spatiotemporal distribution of major physical characteristics such as phase-field, ion concentration, overpotential, and driving force. Such an approach may help to reveal the underlying physics and kinetics of dendrite growth, while also providing design principles for suppressing lithium dendrites.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Shoutong Jin, Yongjun Wu, Hui Yang, Yuhui Huang, Zijian Hong

yongjunwu@zju.edu.cn (Y.W.)
hongzijian100@zju.edu.cn (Z.H.)

Highlights

A protocol to simulate lithium electrodeposition using the phase-field model

Uses fully open-source software MOOSE for the phase-field simulations

Reveals the underlying physics of dendrite growth during electrodeposition

Provides theoretical design principles for suppressing lithium dendrites

Jin et al., STAR Protocols 3, 101713
December 16, 2022
https://doi.org/10.1016/j.xpro.2022.101713

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol for phase-field simulations of lithium dendrite growth with MOOSE framework

Shoutong Jin,[1] Yongjun Wu,[1,2,*] Hui Yang,[1] Yuhui Huang,[1] and Zijian Hong[1,2,3,4,*]

[1]School of Materials Science and Engineering, Zhejiang University, Hangzhou, Zhejiang 310027, China

[2]Cyrus Tang Center for Sensor Materials and Applications, State Key Laboratory of Silicon Materials, Zhejiang University, Hangzhou, Zhejiang 310027, China

[3]Technical contact

[4]Lead contact

*Correspondence: yongjunwu@zju.edu.cn (Y.W.), hongzijian100@zju.edu.cn (Z.H.)
https://doi.org/10.1016/j.xpro.2022.101713

SUMMARY

Phase-field simulation is a powerful tool for understanding lithium metal electrodeposition. This protocol outlines the process of numerically solving the phase-field equations using the MOOSE framework. Here, we describe steps to obtain the spatiotemporal distribution of major physical characteristics such as phase-field, ion concentration, overpotential, and driving force. Such an approach may help to reveal the underlying physics and kinetics of dendrite growth, while also providing design principles for suppressing lithium dendrites.

For complete details on the use and execution of this protocol, please refer to Hong and Viswanathan (2018).

BEFORE YOU BEGIN

Lithium metal battery has been regarded as the "holy grail" for batteries (Kerman et al., 2017; Cheng et al., 2017; Xu et al., 2014; Lin et al., 2017). The main challenge of the Li metal anodes stems from the uncontrolled nonlinear dendrite growth during electrodeposition (Monroe and Newman, 2003, 2004, 2005). Phase-field simulation is a powerful theoretical tool for understanding the dendrite growth kinetics and the underline physical mechanisms (Yurkiv et al., 2018, 2020; Hong and Viswanathan, 2019; Hong et al., 2020; Ahmad et al., 2020). Herein, we propose a protocol to numerically solve the hybrid grand potential-based nonlinear phase-field model using the fully open-source MOOSE (Multiphysics Object-Oriented Simulation Environment) framework (Gaston et al., 2009; Tonks et al., 2012; Permann et al., 2020), following the work by Hong et al. (2018) (Hong and Viswanathan, 2018). The MOOSE package described in this protocol is installed on Linux while installing MOOSE directly on a Windows system is experimental and not yet supported.

Derivation of the phase-field equations for lithium metal electrodeposition

Timing: 4–6 h

In a phase-field model for lithium metal electrodeposition, three coupled variables are required, namely the phase-field variable ξ, chemical potential μ, and overpotential ϕ. The governing equations for the three variables are listed below:

1. Phase-field variable ξ.

In the phase-field model for metal electrodeposition, a non-conserved order parameter, namely the phase-field variable ξ is used, which is defined as $\xi = 1$ and 0, to describe the pure electrode and

electrolyte phases, respectively. The temporal evolution of the order parameter ξ can be expressed as: [8]

$$\frac{\partial \xi}{\partial t} = -L_o\left(g'(\xi) - k\nabla^2\xi\right) - L_\eta h'(\xi)\left\{\exp\left[\frac{(1-\alpha)nF\eta_a}{RT}\right] - \frac{c_{Li^+}}{c_0}\exp\left[\frac{-\alpha nF\eta_\alpha}{RT}\right]\right\} \quad \text{(Equation 1)}$$

Note: Details of the symbols can be referred to Hong et al. (Hong and Viswanathan, 2018) The phase-field variable ξ depends on the chemical potential μ through the local lithium-ion molar fraction c_{Li^+}, which can be written as:

$$c_{Li^+} = c^l(1 - h(\xi)) = \frac{\exp\left[\frac{(\mu - \varepsilon^l)}{RT}\right]}{1 + \exp\left[\frac{(\mu - \varepsilon^l)}{RT}\right]}(1 - h(\xi)) \quad \text{(Equation 2)}$$

While the activation overpotential η_α is related to the overpotential ϕ: $\eta_\alpha = \phi - E^\theta$.

2. Chemical potential μ. The chemical potential can be obtained by solving the modified diffusion equation:

$$\chi\frac{\partial \mu}{\partial t} = \nabla\cdot\frac{Dc_{Li^+}}{RT}[\nabla\mu + nF\nabla\phi] - \frac{\partial h(\xi)}{\partial t}\left[c^s\frac{C_m^s}{C_m^l} - c^l\right] \quad \text{(Equation 3)}$$

Note: the susceptibility can be calculated by:

$$\chi = \frac{\partial c^l}{\partial \mu}[1 - h(\xi)] + \frac{\partial c^s}{\partial \mu}h(\xi)\frac{C_m^s}{C_m^l} \quad \text{(Equation 4)}$$

From Equation 3, it can be inferred that the chemical potential μ depends on the overpotential ϕ and the phase-field variable ξ.

3. Overpotential ϕ. The spatial distribution of the electrical overpotential ϕ can be obtained by solving the conduction equation:

$$\nabla\sigma\nabla\phi = nFC_m^s\frac{\partial \xi}{\partial t} \quad \text{(Equation 5)}$$

Note: The effective conductivity σ is related to the conductivity of the electrode phase σ^s and electrolyte phase σ^l, i.e., $\sigma = \sigma^s h(\xi) + \sigma^l[1 - h(\xi)]$. Here the overpotential ϕ. Only depends on the phase-field variable ξ.

⚠ CRITICAL: In MOOSE, the Equations 1–5 are solved together by minimizing the overall residual in a weak form. The normalization of the parameters is required to ensure that the coupled variables weighed almost equally. Details of the normalization are given in a previous report (Hong and Viswanathan, 2018).

Install MOOSE framework

Timing: 4–6 h

4. Install Pre-Requirements. Supporting packages are needed before installing MOOSE.
   a. CMake. Install CMake 3.6.0 or the latest version.
   b. Python. Install Python 3.0 or the latest version.
   c. Paraview. Install Paraview for data analysis and visualization.

⚠ CRITICAL: Ensure that the supporting packages with the correct version are installed before installing MOOSE, an older version of the packages could result in fatal error during installation.

5. Install MOOSE. Follow the instructions from the MOOSE official website (https://mooseframework.inl.gov/getting_started/installation/conda.html).
   a. Install MOOSE Conda packages.
   b. Cloning MOOSE from GitHub: git clone https://github.com/idaholab/moose.git.
   c. Compile and run tests. In this step, type "cd ~/projects/moose/test; make -j 4; ./run_tests -j 4". There are many test examples integrated in the software to test the reliability of the applications, please refer to the MOOSE official website "https://mooseframework.inl.gov/python/TestHarness.html" for details. If the installation was successful, you should see most of the tests passing. If a fetal error occurs, please contact the MOOSE developers or post a message on the MOOSE google group.
   d. (Optional) Uninstall Conda MOOSE Environment if you wish. To remove the moose environment, the following commands are needed: conda deactivate; conda env remove -n moose.

Note: To ensure a reasonable computational speed, a supercomputer system or a cluster with at least 10 nodes and a minimum memory of 2 Gb/core is needed.

Derive the weak form and prepare for the customized kernels

Timing: 2–3 h

6. Follow the derivations from the MOOSE weak form online introduction, https://mooseframework.inl.gov/getting_started/examples_and_tutorials/tutorial01_app_development/step04_weak_form.html. Derive the weak form for Equations 1–5.
7. Use the official kernels first in the phase-field module for those that have been well developed. Add additional kernels (including the *.C and *.h files) if needed and include them in the corresponding positions in the phase-field folder.

Note: Put the *.C and *.h files in the phase_field/src/kernels and phase_field/include/kernels folders, respectively. The additional customized *.C and *.h files are included in the supplementary files of a previous report (Hong and Viswanathan, 2018).

8. Re-compile MOOSE by typing: make -j 4.

⚠ CRITICAL: Double check that the .C and .h files are included in the correct folder before re-compile, and make sure that MOOSE has been re-compiled after you updated the kernels.

Prepare for the input file

Timing: 1–2 h

9. Follow the official input format in MOOSE and generate a *.i file.

Note: Define the different segments, including Mesh, Variables, Functions, initial conditions (ICs), boundary conditions (BCs), Kernels, Constants, Preconditioning, Executioner, and Outputs.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and algorithms		
Cmake	Kitware Inc., National Library of Medicine, etc.	https://cmake.org/
MOOSE	Idaho National Laboratory	http://mooseframework.inl.gov
Linux	Linus Benedict Torvalds	www.Linux.org
C++	Alcatel-Lucent Bell Laboratories	http://isocpp.org
Paraview	Kitware Inc., Sandia National Laboratories, and CSimSoft	https://www.paraview.org/
Python	Python Software Foundation	https://www.python.org/
GCC	GCC steering committee	https://gcc.gnu.org/
MPICH	Argonne National Laboratory and Mississippi State University	https://www.mpich.org/

STEP-BY-STEP METHOD DETAILS

In this session, the major steps for the phase-field simulations of lithium dendrite growth using the MOOSE framework are described. We begin by illustrating how to edit and generate the input file, followed by job submissions. Then, the data analysis and visualizations are presented. Finally, the model limitations and trouble shootings are discussed.

Edit and generate the input file

Timing: 0.5 h

1. Open the *.i file.
2. Customize the "Meshes" according to the simulation system (optional).

Note: The details for customizing the mesh system can be found in https://mooseframework.inl.gov/syntax/Mesh/. For instance, if you want to generate mesh with the aspect ratio of a given image, the "ImageMeshGenerator" can be used (https://mooseframework.inl.gov/source/meshgenerators/ImageMeshGenerator.html).

3. Edit the "ICs" session for different initial conditions.

Note: The initial conditions can be customized based on the target initial structure for the simulations. The supported "ICs" functions can be found in https://mooseframework.inl.gov/syntax/ICs/index.html. For instance, if a circular structure is used, one can apply the "SmoothCircleIC (https://mooseframework.inl.gov/source/ics/SmoothCircleIC.html). The "ICs" can also be customized by defining a specific equation in the [Functions] module, and refer to this function in the "ICs" session. One example is given below:

```
>[Functions]

>[./ic_func_eta] # Define a function called ic_func_eta.

>type = ParsedFunction #The function type is ParsedFunction.

>value = 0.5*(1.0-1.0*tanh((x-20)*2)) # The value of the function depends on the horizontal axis x through this equation.

>[../]

>[]
```

```
>[ICs]
>[./eta] # Define the initial conditions for eta.
>variable = eta # Define the name of the variable.
>type = FunctionIC # Call the Functions module to set up ICs for eta.
>function = ic_func_eta # call the ic_func_eta as defined above.
[../]
[]
```

Change the "Constants" session to model different electrolytes (optional).

4. Optimize the "Executioner" for better convergence and higher speed (optional).
5. Change the boundary conditions according to the physical model (optional).

Note: The boundary conditions can be customized by changing the "BCs" module, details can be found in https://mooseframework.inl.gov/syntax/BCs/index.html. A sample input file (in *.i format) can be found in the supplementary files of Hong et al. (2018).

Job submissions and run

Timing: 24–96 h

6. Generate a bash file according to the server job queue command.

A sample job queue script is given below:

```
>[Functions]
>#!/bin/bash
>#BSUB -J Samplejob (Assigns the specified name to the job)
>#BSUB -q score (Submits job to one of the specified queues)
>#BSUB -n 10 (Specifies the maximum number of processors that are required for a parallel job)
>#BSUB -R "span[ptile=32]" (Specifies host resource requirements)
>#BSUB -o %J.out (Appends the standard output to a file)
>#BSUB -e %J.err (Appends the standard error output to a file)
>#Get present work directory
>CURDIR=$PWD
>#Generate nodelist
>rm -f ${CURDIR}/nodelist >& /dev/null
>for i in `echo $LSB_HOSTS`
>do
>echo $i >> ${CURDIR}/nodelist
>done
>#Get the number of processors in nodelist
```

```
>NP=`cat ${CURDIR}/nodelist |wc -l`

>#Run mpi program with input file using processors in nodelist

>mpirun -np $NP ./phase_field-opt -i input.i
```

7. Submit the job. The submission command depends on the job queue system. A sample command with bsub: bsub <sample.lsf.
8. Check the job status and the output file (the *.e file) regularly, kill the job once the front of the anode reaches 180 μm.

   Note: This cutoff distance is determined by the simulation system size. Typically, the cutoff distance is less than 90% of the simulation size to ensure the validity of the boundary conditions.

   ⚠ CRITICAL: Double check that the designed initial structure and the boundary conditions are correct after you get the first output. Re-exam the input file if the output doesn't match with the design.

9. Optimize the time step, and do scaling tests to find the optimal number of nodes. (Optional).

Data analysis and visualization

Timing: 4–6 h

10. Download the *.e file to a local folder, open with Paraview. Click on "Variables", and then select "Apply", you can get the following picture (Figure 1).
11. Click on the "vtkBlockColors" button, and various variable options will show up, corresponding to the different variables in the equations.

    Note: Taking the order parameter ξ as an example, after we select "eta", the interface shown in Figure 2 will appear. The time evolution of ξ can be obtained after clicking on the right triangle on the top panel, as marked by the dashed red box.

12. In addition to eta, there are various variables in the model such as c: solution concentration distribution, pot: over potential, G: interface driving force, and so on. Through these variables, we can get the changes in the physical properties during electrodeposition.

    Note: Take the concentration field as an example. Click on the "vtkBlockColors: button, and then select the "c" option, adjust "Time" to the target time, and tune the color mapping parameters for visual pleasure. An example is shown in Figure 3.

13. For further data analysis, line plots are required.
    a. Select "Filters" in the menu bar, then click on "Data Analysis", and enter "Plot over line".
    b. To plot the concentration distribution along the horizontal center line, adjust the coordinates of Point 1 to (0, 100, 0), and the coordinates of Point 2 to (200, 100, 0).
    c. Click "Apply", and select c in the variable, Figure 4 will be generated. (Optional).
14. To output the data in excel, select "File" in the menu bar, and click on "save data". The raw data will be saved to your folder for further analysis. (Optional).

EXPECTED OUTCOMES

The morphological evolution of the lithium metal anode during electrodeposition with different applied overpotentials can be obtained. The spatial distributions of the physical quantities such as Lithium-ion

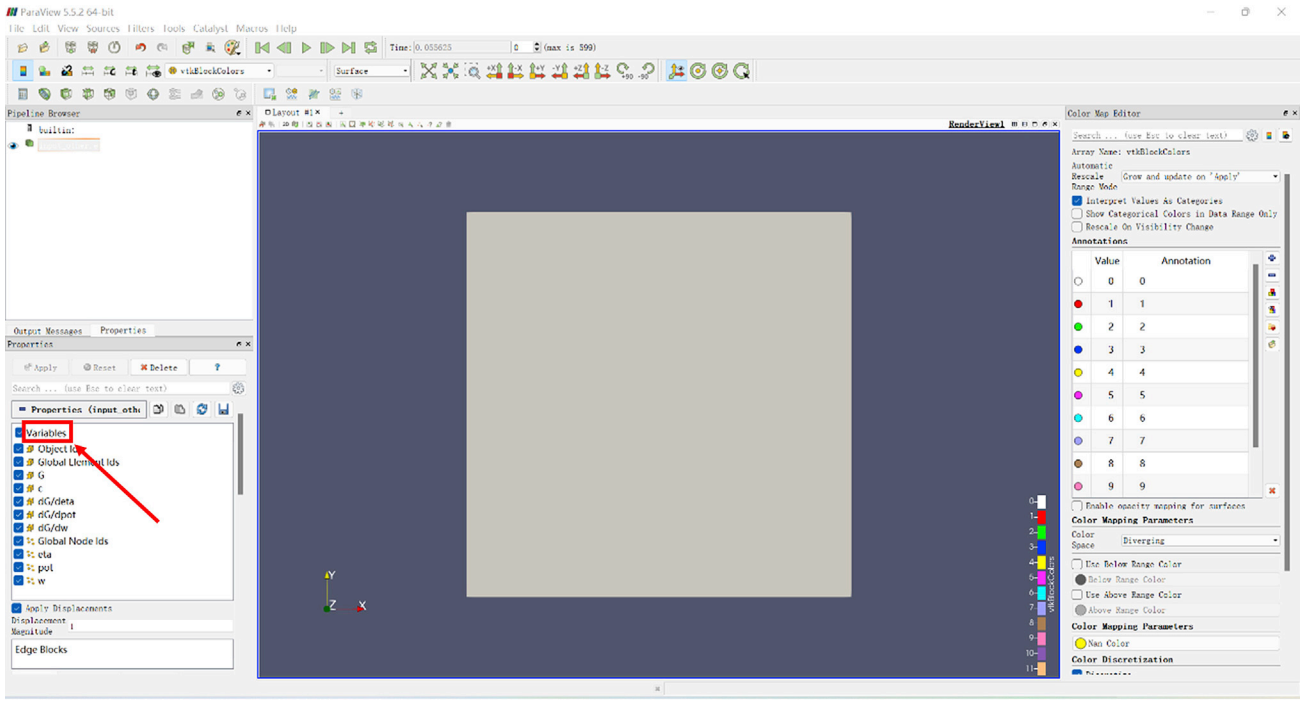


Figure 1. Paraview graphic interface after loading the output .e file

concentration, interface growth velocity, overpotential, etc. can be mapped out, as shown in Figure 5. This will enable an in-depth understanding of the underline physical mechanisms of the dendrite growth.

Moreover, the solid mechanics, thermal transport, and fluid mechanics modules can be further coupled to the current model, enabling the model of solid electrolytes, as well as operation under stack pressure, thermal heating/cooling, and electrolyte flow conditions.

LIMITATIONS

Currently, it is still challenging to perform a full 3-D model on a large scale. This is limited by both the calculation speed and memory.

SEI is not considered in the model due to the large length scale gap (the grid size is 1 μm, which is 2–3 orders of magnitude larger than the SEI thickness.

Only a half-cell is modeled in this work, a full cell with both cathode, anode, and separator is still difficult. This requires a larger length scale and a more complicated model.

TROUBLESHOOTING

Problem 1

You may encounter problems when installing MOOSE for the first time (before you begin, steps 4 and 5). Problems include not all the test jobs can run successfully.

Potential solution

- Make sure Conda is installed properly.
- Follow the installation tutorial step-by-step, and check whether the required software (GCC 7.5.0 or greater, Cmake 3.6.0 or greater, and MPICH 3.4.2 or greater) is installed successfully.

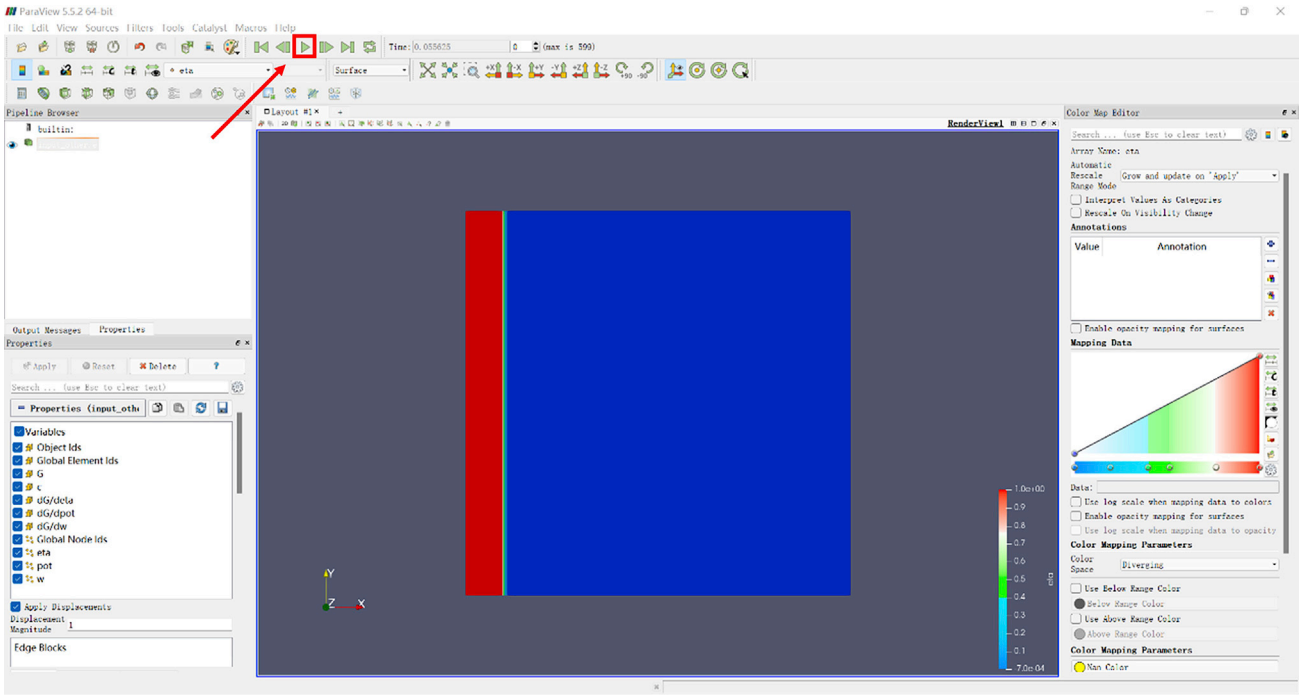


Figure 2. Schematic representation of eta in Paraview

- Read the error message and check the "troubleshooting" webpage on the MOOSE website (https://mooseframework.inl.gov/help/troubleshooting.html#condaissues) to follow the instructions.
- Send an email to the MOOSE developers and post your problem on the MOOSE Google group.

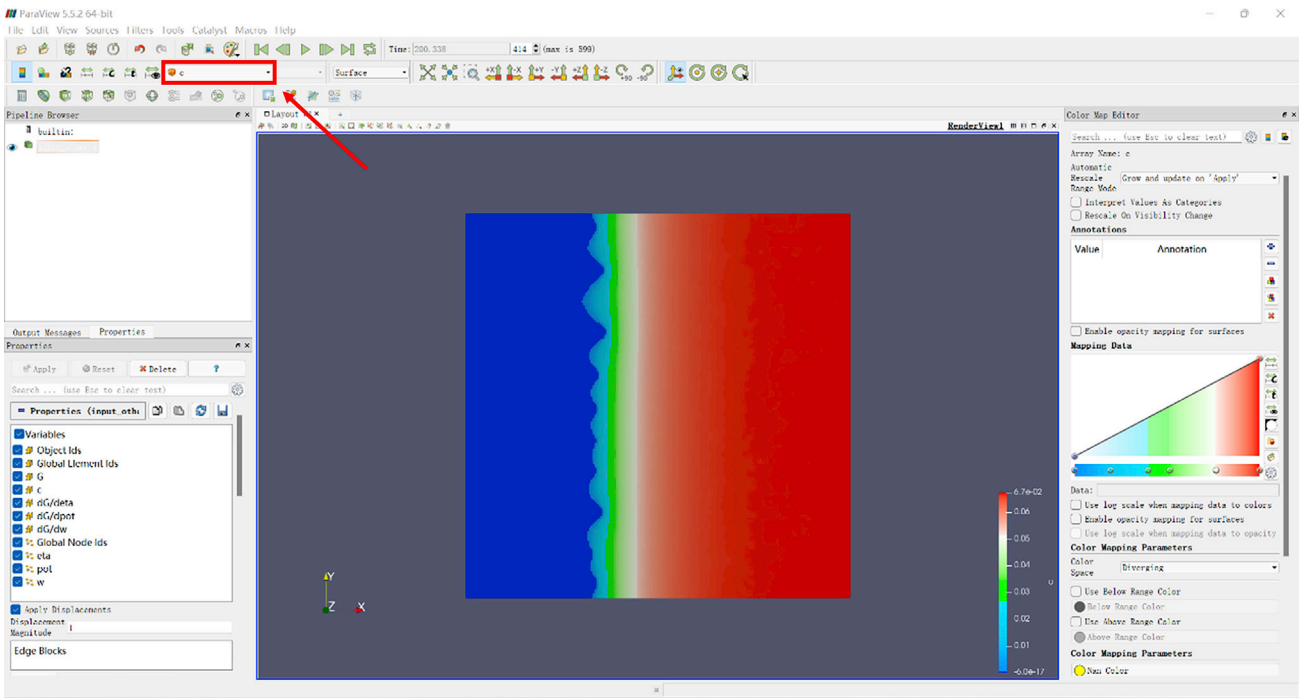


Figure 3. Concentration distribution of system solution at 200 s

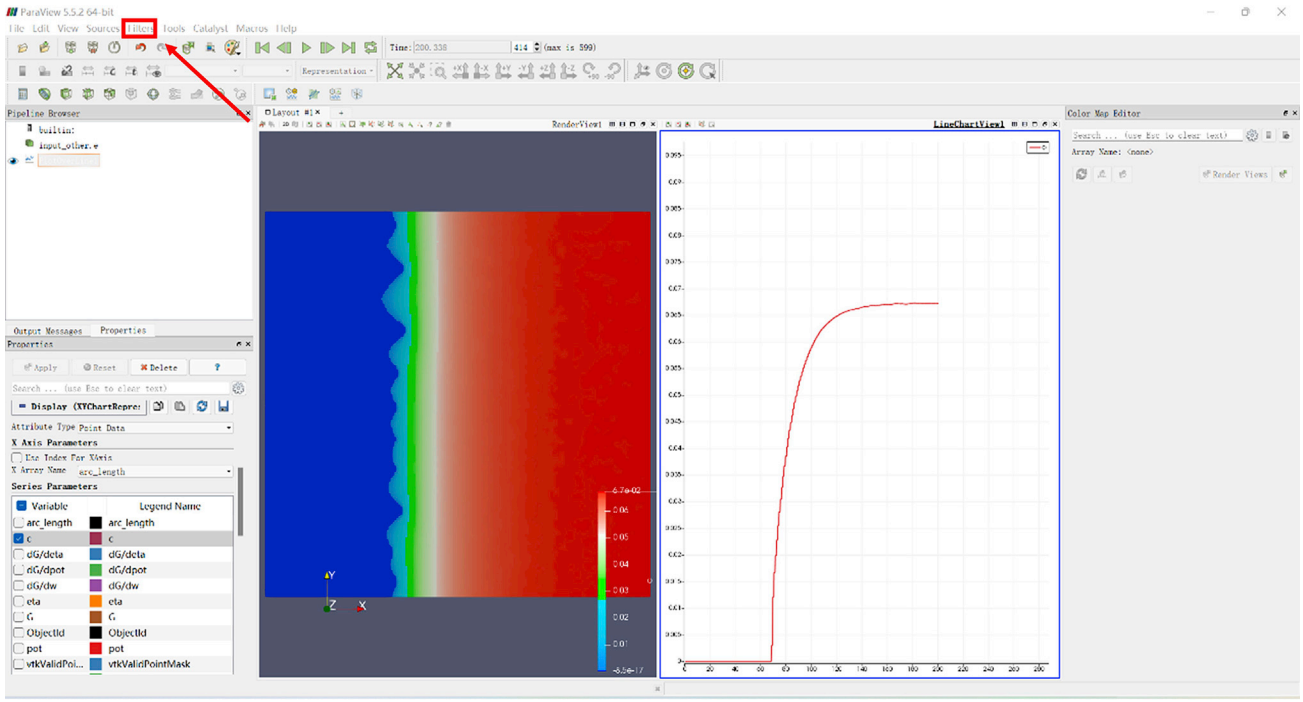


Figure 4. Line distribution of lithium-ion concentration along the horizontal direction

Problem 2

May encounter convergence problem during iteration (step 7). This is a very common problem when developing new modules or starting with a new initial structure.

Potential solution

Adjust and optimize the solver in the "Executioner". Decrease the timestep. Decrease the sharpness of the structure, especially with a triangle or tip in the initial structure.

Problem 3

May encounter sudden crashes of the program while running (steps 7 and 8).

Potential solution

- Double check if it is due to the failure of the infrastructure (e.g., out-of-power or out-of-memory for the supercomputers).
- Read the output and error message (.out and .err files), search for possible reason and solution from the message.
- Double check whether it is due to the convergence problem. Follow the trouble shooting for problem 2 if it is a convergence issue.

Problem 4

"Paraview" doesn't work in your laptop (step 10).

Potential solution

Alternative visualization software can be used, including but not limited to:

- Peacock (https://mooseframework.inl.gov/application_usage/peacock.html), which is a software that allows the user to build or modify an input file, execute the application and view the results all within one package.

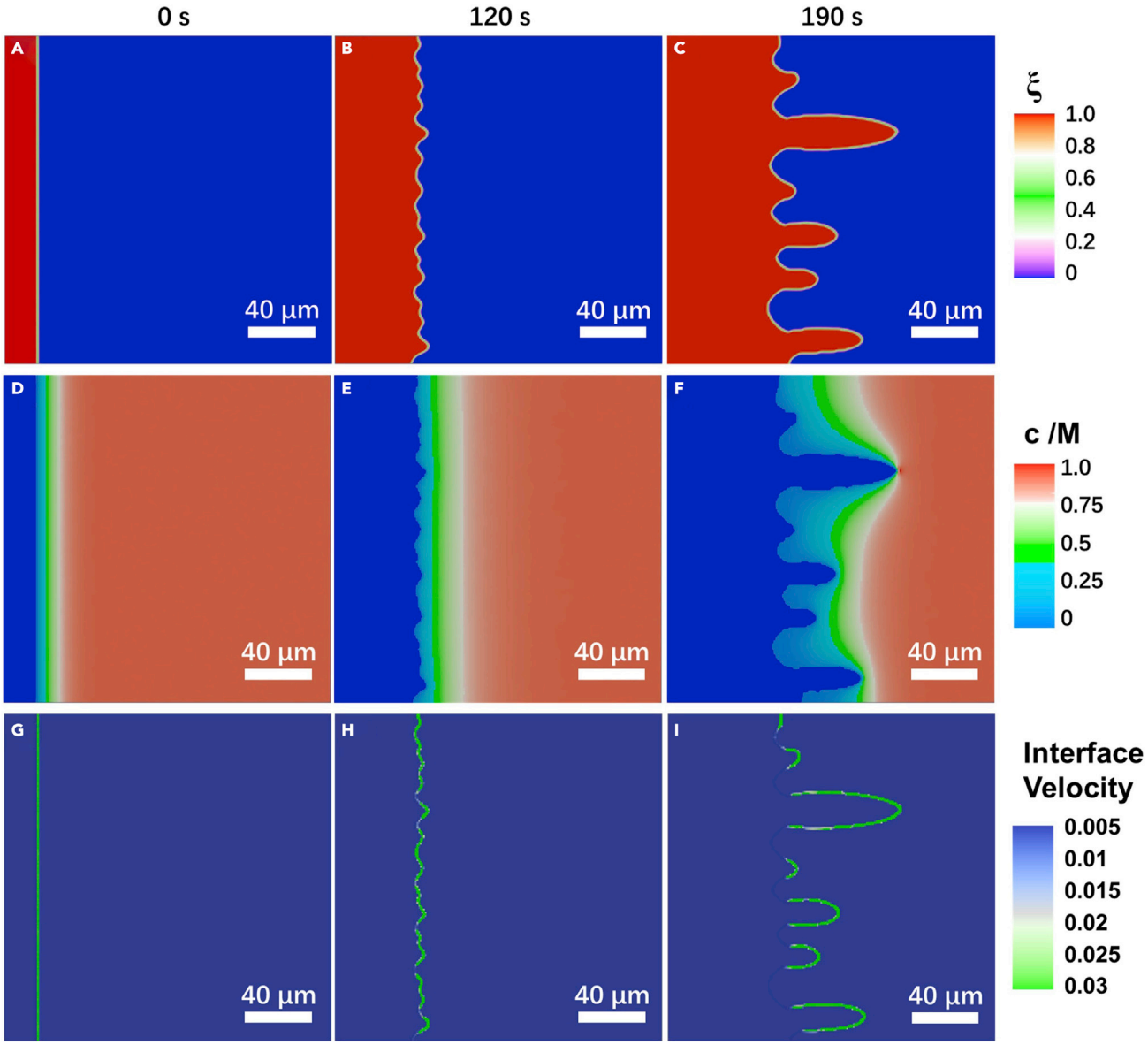


Figure 5. Dendrite growth, concentration, and interface velocity evolution with an applied overpotential of −0.25 V

(A–C) Morphology evolution after lithium electrodeposition of 0, 120, and 190 s.

(D–F) Corresponding evolution of the lithium-ion molar ratio.

(G–I) Corresponding evolution of the interface velocity evolution.

- VisIt (https://visit-dav.github.io/visit-website/index.html). VisIt is a visualization tool that supports the .e file extension. Please refer to the official website for details.

Problem 5

The calculation speed could be very slow (step 8).

Potential solution

- Do the scaling test and increase the number of processors to a reasonable value.
- Decrease the system size.
- Optimize the timestep dt.
- Optimize the convergence criteria and the solver.

RESOURCE AVAILABILITY

Lead contact

Further requests for resources and questions should be directed to and will be fulfilled by the lead contact, Prof. Zijian Hong (hongzijian100@zju.edu.cn).

Materials availability

This study did not generate new unique reagents.

Data and code availability

The data/code supporting the current study are available from the lead contact upon reasonable requests.

ACKNOWLEDGMENTS

Z.H. would like to acknowledge a startup grant from Zhejiang University.

AUTHOR CONTRIBUTIONS

Z.H. conceived the idea. S.J. conducted the experiments, performed data extraction, and visualization. S.J., Y.W., and Z.H. prepared the initial draft. Y.W., Y.H., H.Y., and Z.H. reviewed and edited the manuscript. Z.H., Y.W., Y.H., H.Y. supervised the works. All authors commented on the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

Kerman, K., Luntz, A., Viswanathan, V., Chiang, Y.M., and Chen, Z. (2017). Review-practical challenges hindering the development of solid state Li ion batteries. J. Electrochem. Soc. *164*, A1731–A1744. https://doi.org/10.1149/2.1571707jes.

Lin, D., Liu, Y., and Cui, Y. (2017). Reviving the lithium metal anode for high-energy batteries. Nat. Nanotechnol. *12*, 194–206. https://doi.org/10.1038/nnano.2017.16.

Ahmad, Z., Hong, Z., and Viswanathan, V. (2020). Design rules for liquid crystalline electrolytes for enabling dendrite-free lithium metal batteries. Proc. Natl. Acad. Sci. USA *117*, 26672–26680. https://doi.org/10.1073/pnas.2008841117.

Cheng, X.-B., Zhang, R., Zhao, C.Z., and Zhang, Q. (2017). Toward safe lithium metal anode in rechargeable batteries: a review. Chem. Rev. *117*, 10403–10473. https://doi.org/10.1021/acs.chemrev.7b00115.

Gaston, D., Newman, C., Hansen, G., and Lebrun-Grandié, D. (2009). MOOSE: a parallel computational framework for coupled systems of nonlinear equations. Nucl. Eng. Des. *239*, 1768–1778. https://doi.org/10.1016/j.nucengdes.2009.05.021.

Hong, Z., Ahmad, Z., and Viswanathan, V. (2020). Design principles for dendrite suppression with porous polymer/aqueous solution hybrid electrolyte for Zn metal anodes. ACS Energy Lett. *5*, 2466–2474. https://doi.org/10.1021/acsenergylett.0c01235.

Hong, Z., and Viswanathan, V. (2018). Phase-field simulations of lithium dendrite growth with open-source software. ACS Energy Lett. *3*, 1737–1743. https://doi.org/10.1021/acsenergylett.8b01009.

Hong, Z., and Viswanathan, V. (2019). Prospect of thermal shock induced healing of lithium dendrite. ACS Energy Lett. *4*, 1012–1019. https://doi.org/10.1021/acsenergylett.9b00433.

Monroe, C., and Newman, J. (2003). Dendrite growth in lithium/polymer systems-A propagation model for liquid electrolytes under galvanostatic conditions. J. Electrochem. Soc. *150*, A1377–A1384. https://doi.org/10.1149/1.1606686.

Monroe, C., and Newman, J. (2004). The effect of interfacial deformation on electrodeposition kinetics. J. Electrochem. Soc. *151*, A880. https://doi.org/10.1149/1.1710893.

Monroe, C., and Newman, J. (2005). The impact of elastic deformation on deposition kinetics at lithium/polymer interfaces. J. Electrochem. Soc. *152*, A396. https://doi.org/10.1149/1.1850854.

Permann, C.J., Gaston, D.R., Andrš, D., Carlsen, R.W., Kong, F., Lindsay, A.D., Miller, J.M., Peterson, J.W., Slaughter, A.E., Stogner, R.H., and Martineau, R.C. (2020). MOOSE: enabling massively parallel multiphysics simulation. SoftwareX *11*, 100430. https://doi.org/10.1016/j.softx.2020.100430.

Tonks, M.R., Gaston, D., Millett, P.C., Andrs, D., and Talbot, P. (2012). An object-oriented finite element framework for multiphysics phase field simulations. Comput. Mater. Sci. *51*, 20–29. https://doi.org/10.1016/j.commatsci.2011.07.028.

Xu, W., Janocha, A.J., Leahy, R.A., Klatte, R., Dudzinski, D., Mavrakis, L.A., Comhair, S.A.A., Lauer, M.E., Cotton, C.U., and Erzurum, S.C. (2014). Lithium metal anodes for rechargeable batteries. Redox Biol. *2*, 513–519. https://doi.org/10.1039/c3ee40795k.

Yurkiv, V., Foroozan, T., Ramasubramanian, A., Shahbazian-Yassar, R., and Mashayek, F. (2018). Phase-field modeling of solid electrolyte interface (SEI) influence on Li dendritic behavior. Electrochim. Acta *265*, 609–619. https://doi.org/10.1016/j.electacta.2018.01.212.

Yurkiv, V., Foroozan, T., Ramasubramanian, A., Ragone, M., Shahbazian-Yassar, R., and Mashayek, F. (2020). Understanding Zn electrodeposits Morphology in secondary batteries using phase-field model. J. Electrochem. Soc. *167*, 060503. https://doi.org/10.1149/1945-7111/ab7e91.