

HallThruster.jl: a Julia package for 1D Hall thruster discharge simulation

Thomas Marks¹, Paul Schedler², and Benjamin Jorns¹

¹ Plasmadynamics and Electric Propulsion Laboratory, University of Michigan, Ann Arbor, USA ² ETH Zurich, Zurich, Switzerland

DOI: [10.21105/joss.04672](https://doi.org/10.21105/joss.04672)

Software

- [Review](#)
- [Repository](#)
- [Archive](#)

Editor: [Lucy Whalley](#)

Reviewers:

- [@Rupali-Sahu](#)
- [@TomGoffrey](#)
- [@lucydot](#)

Submitted: 17 May 2022

Published: 28 June 2023

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

Although the Hall effect thruster is one of the most widely-used forms of in-space electric propulsion, there are aspects of these devices—particularly the electron dynamics—that remain poorly understood. This knowledge gap has precluded the development of fully predictive engineering models for Hall thruster operation. Instead, it is necessary to calibrate models against experimental data in order to achieve a high degree of agreement with data. This process is time intensive for state of the art multi-dimensional Hall thruster models and can be subject to a high degree of variability. HallThruster.jl is a one-dimensional fluid Hall thruster code which has sufficient fidelity to be partially predictive, while running quickly enough to be used as a surrogate for model calibration and discovery. It is written in Julia for its combination of speed, extensibility and ease-of-use, and it allows users to customize many aspects of the simulation including the critical model for the electron dynamics.

Statement of need

The Hall effect thruster is an annular plasma device characterized by an electric field applied across a radial magnetic field. [Figure 1](#) depicts the operation of a typical Hall thruster.

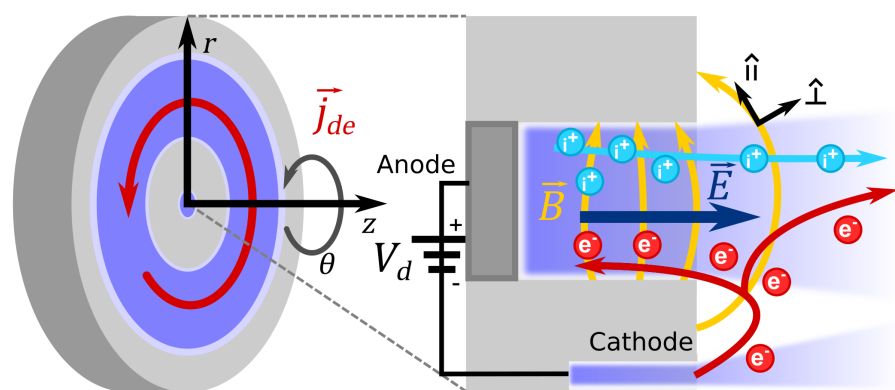


Figure 1: Hall thruster operation

The heavier, unmagnetized ions are accelerated by this electric field downstream at high speed while the electrons remain trapped in a strong, $E \times B$ azimuthal “Hall” current. In an ideal, collisionless environment, these trapped electrons are not able to cross the confining magnetic field. In practice, however, there is a high degree of electron flux across the magnetic field

lines, perpendicular to the applied magnetic field. The process, which is a critical driver for the overall dynamics of the thruster, is not well understood. `HallThruster.jl` was designed as a tool to more rapidly assess different models for this electron transport through parameter estimation, uncertainty quantification, and surrogate optimization. It allows users to alter many parts of the physics model for electron transport (and related processes) without modifying the source code of the package, and the one-dimensional domain only requires a few seconds to a few minutes to run a simulation, depending on the grid resolution. With that said, a central premise of `Hallthruster.jl` is that the dynamics in the Hall thruster are suitably one-dimensional that the 1D code is usually sufficient for an accurate approximation of a 2D or 3D code when run on the scale of the entire thruster. But, we note that with appropriate calibration, errors may be corrected by training a surrogate model correction for higher dimensional codes. Indeed, this is one of the anticipated use cases.

While one-dimensional codes are commonly used in Hall thruster research (Ahedo & Martinez-Sanchez, 2001; Boeuf & Garrigues, 1998; Hara, 2018; Mikellides et al., 2001; Sahu et al., 2020), `HallThruster.jl` is unique in that, in addition to its primary goal of informing closure models, it is written from the ground up with a strong emphasis on code verification, collaboration, and extension. As an open source code, our goal is that it will allow the community to reproduce computational results obtained and to compare and contrast with parallel studies.

Physics model

Figure 2 depicts the one-dimensional simulation domain for `HallThruster.jl`. In this model, we assume quasineutrality in the entire domain and treat all species—neutrals, ions, and electrons—as fluids. Neutrals are assumed to have constant temperature and velocity and are solved using the continuity equation, while ions are assumed isothermal and are modeled using the isothermal Euler equations. Electron inertia is neglected, so the electron momentum equation reduces to a generalized Ohm’s law. As discussed in the preceding section, electrons drift primarily in the $\hat{\theta} = \hat{z} \times \hat{r}$ direction but have a small axial drift enabled by collisions. This cross-field electron current observed in experiment is much higher than that which would be predicted from classical collisions alone, so we apply an additional “anomalous” collision frequency in the Ohm’s law to enable the user to better tune the simulation to match experiment. Different forms for this collision frequency can be specified to explore different closure models. We combine this Ohm’s law with current conservation to obtain a second-order partial differential equation for the electrostatic potential. Lastly, we solve a partial differential equation for the transport of electron internal energy, taking into account losses due to ionization, excitation, and wall losses. A detailed listing of the equations employed in the physics model is available in the code documentation.

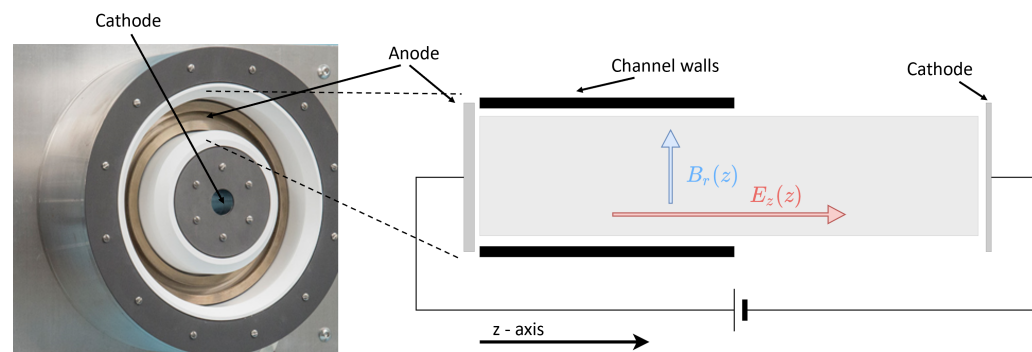


Figure 2: 1D simulation domain of `HallThruster.jl`

While many aspects of Hall thruster physics are well-represented by a one-dimensional model,

certain important features of real Hall thruster discharges will not be captured in such a simplified code. Among other effects, the most important of these missing features are plume divergence, impacts of magnetic field curvature, and effects of radial plasma non-uniformity on wall losses.

Numerics

The finite volume method is applied to discretize the ion and neutral equations, transforming them into a system of ordinary differential equations. These are integrated in time using `DifferentialEquations.jl` (Rackauckas & Nie, 2017), and all explicit integrators that work with `DifferentialEquations.jl` should work in `HallThruster.jl`. Several choice of numerical flux are available, including the Rusanov, global Lax-Friedrichs, and Hartford-Lax-van-Leer-Einfeld (HLLF) fluxes. The elliptic equation for the potential is transformed by finite differences into a tridiagonal linear system and solved using Thomas' algorithm (Thomas, 1949). The electron energy equation is solved semi-implicitly to ease timestep restrictions and is discretized in space using first-order upwind differences. We compute reaction and collision rate coefficients using look-up tables. The method of manufactured solutions is used to verify that we have correctly discretized the PDEs and obtain the design order of accuracy. We are aided in this by the `Symbolics.jl` (Gowda et al., 2022), which makes computation of the needed source terms simple.

Functionality

`HallThruster.jl` provides extensive options to allow the user to customize their simulation. These include

- Custom (one-dimensional) thruster geometry
- Anode and wall sheath model options
- Multiple propellants—the code has built-in capabilities for krypton and xenon. Additional propellants can be added by introducing the appropriate reaction rates and mass
- Restarts
- Custom anomalous transport models
- User-provided source terms
- Custom additional collisions/reactions

Detailed documentation about all of these features is available at the Github repository (<https://github.com/UM-PEPL/HallThruster.jl>). More features will be added as the code continues to develop, and the authors welcome contributions and ideas from users.

Example simulations

We show comparisons of our results to case 3 of the LANDMARK benchmark ([Landmark 1D Fluid Test Case, n.d.](#)), which provides reference 1D Hall thruster simulations to which code authors can compare their own simulations. This benchmark has three cases, distinguished by the amount of electron energy lost to the walls. We show all results with 1024 cells, time averaged after 2 ms of simulation time. We compare three included fluxes: HLLF, local Lax-Friedrichs / Rusanov (LLF), and global Lax-Friedrichs (GLF) to the three sub-cases of the LANDMARK benchmark - two fluid models with adjustable viscosity parameter δ and a hybrid-particle-in-cell case. Note that the benchmark does not include plots of the ion and electron velocities, nor does it include details about the numerics of the example case. This makes a quantitative comparison difficult. Nevertheless, we see good qualitative agreement. Differences between our model and the LANDMARK fluid model may be related to differences in boundary condition implementation, order of accuracy of method, or choice of numerical flux.

LANDMARK case 1:

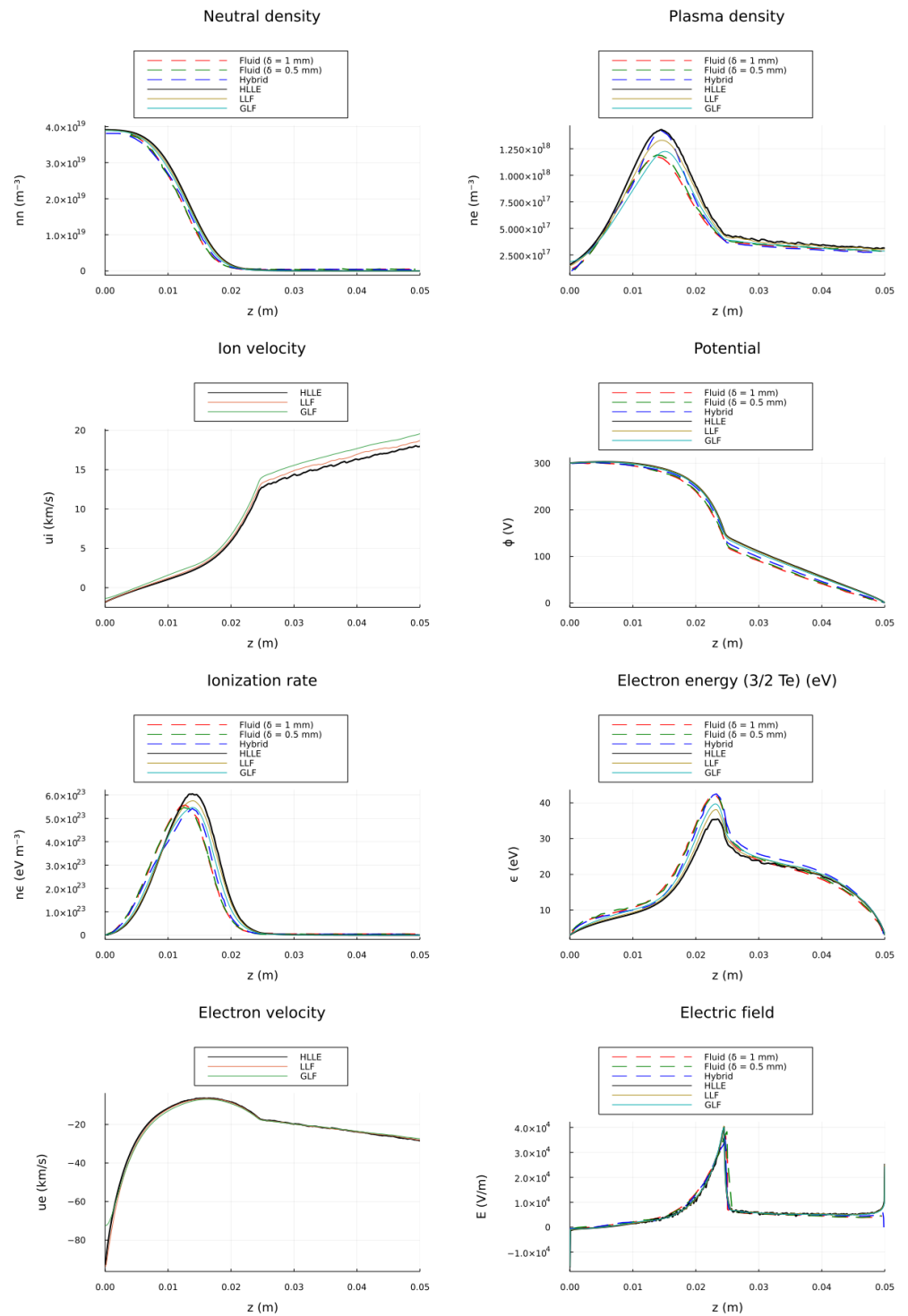


Figure 3: LANDMARK case 1, 1024 cells.

LANDMARK case 2:

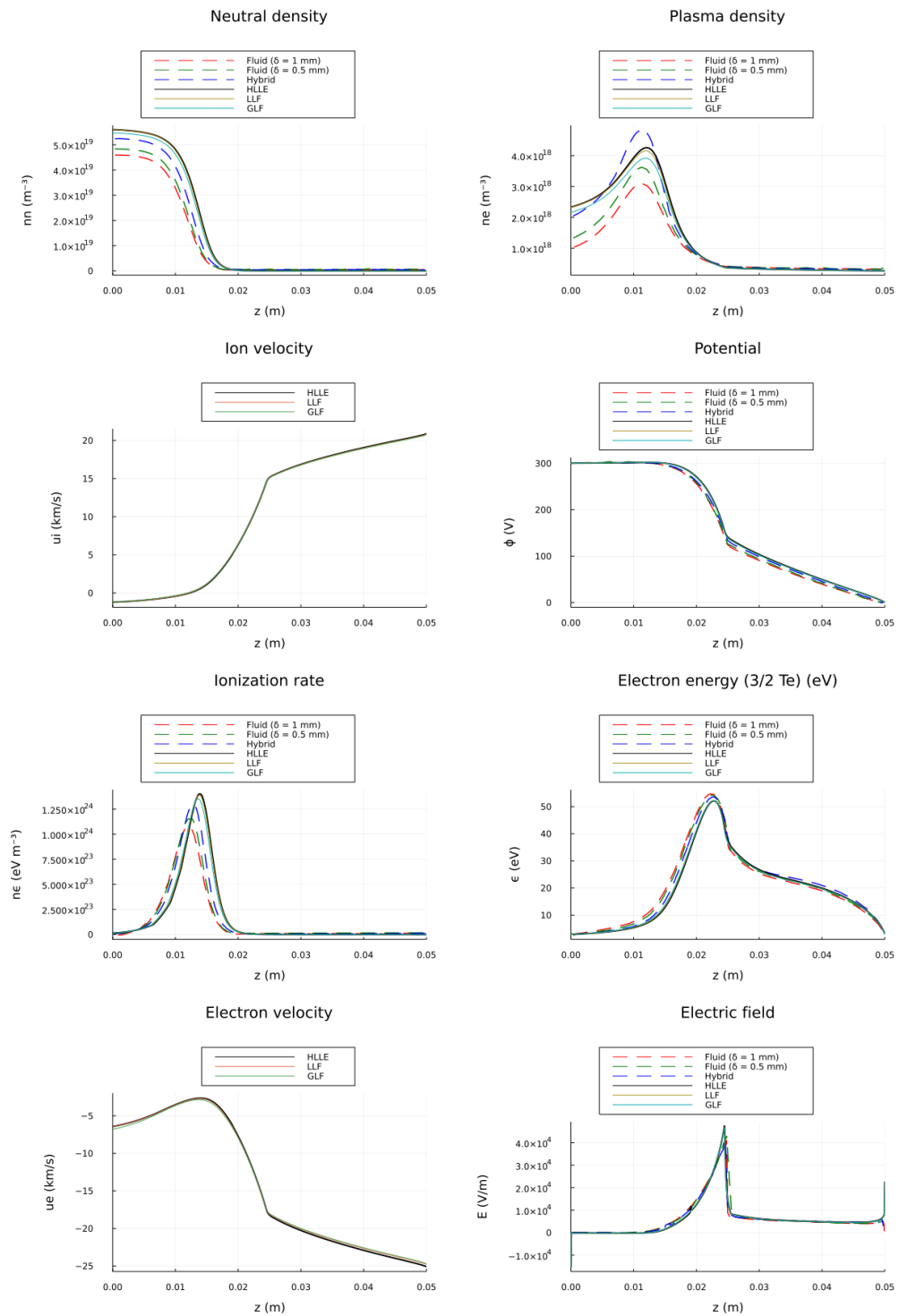


Figure 4: LANDMARK case 2, 1024 cells

LANDMARK case 3:

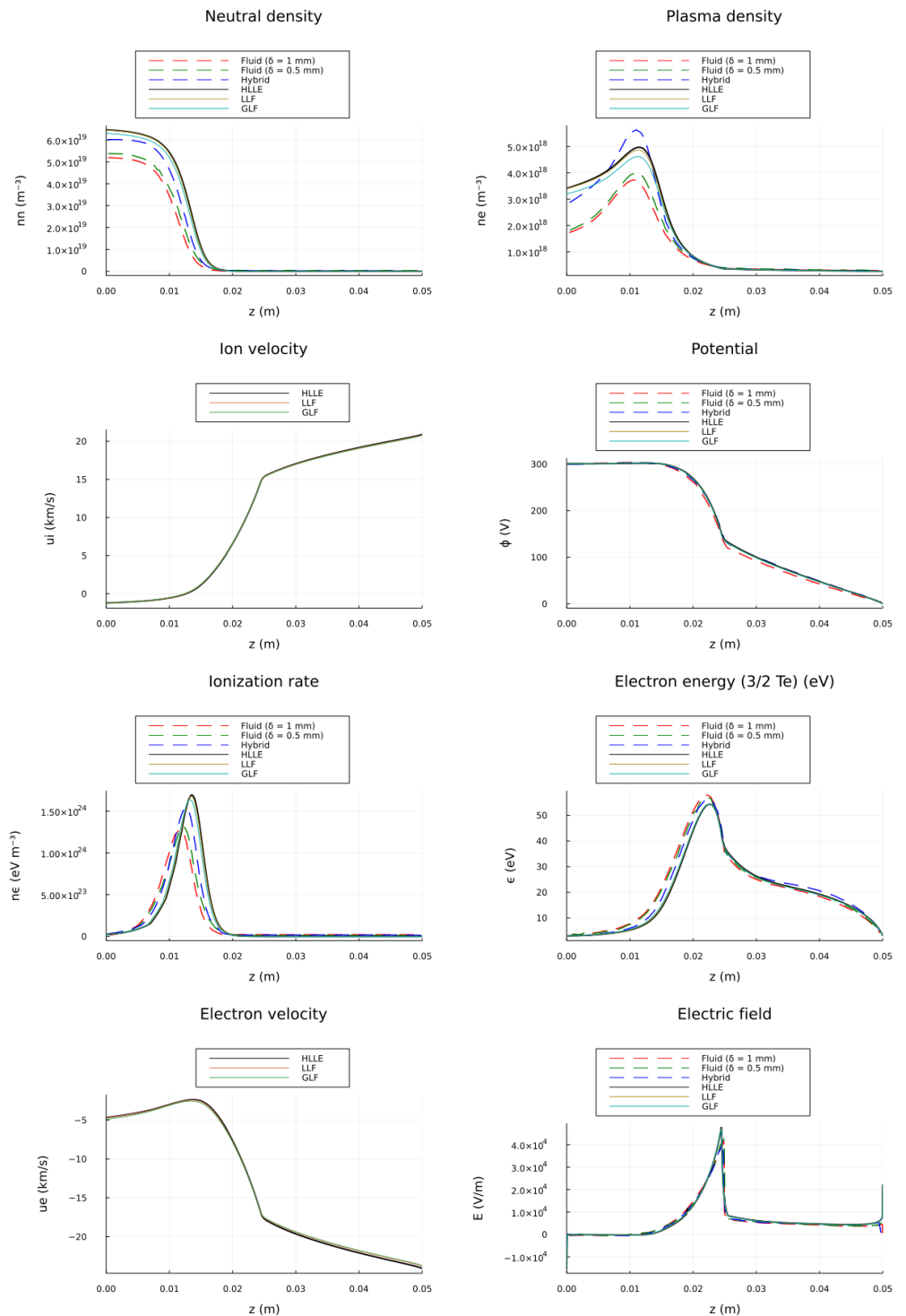


Figure 5: LANDMARK case 3, 1024 cells

Acknowledgements

T. Marks and B. Jorns were supported by AFOSR grant FA9550-19-1-0022 under the Space Propulsion and Power portfolio. P. Schedler was supported under a research grant from ETH Zurich, Zurich, Switzerland.

References

- Ahedo, E., & Martinez-Sanchez, M. (2001). One dimensional model of plasma flow in a Hall thruster. *Physics of Plasmas*, 8(6), 3058–3068. <https://doi.org/10.1063/1.1371519>
- Boeuf, J. P., & Garrigues, L. (1998). Low frequency oscillations in a stationary plasma thruster. *Journal of Applied Physics*, 84(7), 3541–3554. <https://doi.org/10.1063/1.368529>
- Gowda, S., Ma, Y., Cheli, A., Gwóźdz, M., Shah, V. B., Edelman, A., & Rackauckas, C. (2022). High-performance symbolic-numeric via multiple dispatch. *ACM Commun. Comput. Algebra*, 55(3), 92–96. <https://doi.org/10.1145/3511528.3511535>
- Hara, K. (2018). Non-oscillatory quasineutral fluid model of cross-field discharge plasmas. *Physics of Plasmas*, 25(12), 123508. <https://doi.org/10.1063/1.5055750>
- Landmark 1D fluid test case. (n.d.). <https://jpb911.wixsite.com/landmark/test-case-3>.
- Mikellides, I., Katz, I., Mandell, M., & Snyder, J. (2001). A 1-d model of the hall-effect thruster with an exhaust region. In *37th joint propulsion conference and exhibit*. <https://doi.org/10.2514/6.2001-3505>
- Rackauckas, C., & Nie, Q. (2017). Differentialequations.jl—a performant and feature-rich ecosystem for solving differential equations in julia. *Journal of Open Research Software*, 5(1).
- Sahu, R., Mansour, A. R., & Hara, K. (2020). Full fluid moment model for low temperature magnetized plasmas. *Physics of Plasmas*, 27(11), 113505. <https://doi.org/10.1063/5.0021474>
- Thomas, L. H. (1949). Elliptic problems in linear difference equations over a network. *Watson Sci. Comput. Lab. Rept., Columbia University, New York*, 1, 71.