The Journal of Open Source Software

PathView: A Graphical User Interface for System Simulation

Remi Delaporte-Mathurin [1¶], **Tasnim Zulfiqar** [1], **and James Dark** [1]

1 Plasma Science and Fusion Center, Massachusetts Institute of Technology, Cambridge, 02139, USA
¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 13 August 2025
Published: unpublished

Summary

`PathView` is an interactive, browser-based graphical interface for the system simulation framework `PathSim` (Rother, 2025). It enables users to build models quickly in an intuitive, visual environment. Built with `ReactFlow` (Moritz Klack, 2025), `PathView` allows users to:

- Drag and drop simulation nodes onto a canvas.
- Connect nodes to define system structure.
- Create and manage subsystems for hierarchical modelling.
- Configure event detection and advanced simulation options.
- Configure solver parameters
- Easily extend functionality through a modular architecture.

In addition to mirroring nearly all of `PathSim`'s capabilities, `PathView` uses `Jinja2` (*Pallets/Jinja*, 2025) templates to generate fully executable Python scripts from graphical models, enabling seamless transition between GUI-based and code-based workflows.

Statement of need

`PathSim` is a powerful and flexible simulation framework for modelling complex systems. However, building large-scale or intricate models solely through Python scripting can be cumbersome and error-prone, particularly for new users or for projects that benefit from visual inspection of system layout. This is for example the case for nuclear fusion fuel cycle applications (Meschini et al., 2023, 2025). Many established simulation platforms, such as MathWorks Simulink (*Simulink - Simulation and Model-Based Design*, n.d.) or Aspen Plus (*Aspen Plus | Leading Process Simulation Software | AspenTech*, n.d.), provide graphical user interfaces to enhance usability, model comprehension, and collaboration. Until now, such a visual modelling environment was missing for `PathSim`. `PathView` fills this gap by providing a modern, interactive, and extensible GUI, reducing the barrier to entry for new users and improving productivity for experienced modellers.

Examples

`PathView` includes several pre-built example graphs in the example_graphs directory that demonstrate different functionalities:

- `harmonic_oscillator.json` - Simple oscillator simulation (see Figure 1 and Figure 2)
- `pid.json` - PID controller example
- `linear_feedback.json` - Linear feedback system
- `pendulum.json` - Pendulum example
- `spectrum.json` - Spectral analysis example

- `bouncing_ball.json` - Example showcasing event detection
- `thermostat.json` - Thermostat demo with event detection
- `stick_slip.json` - Example showcasing the use of the `Switch` block

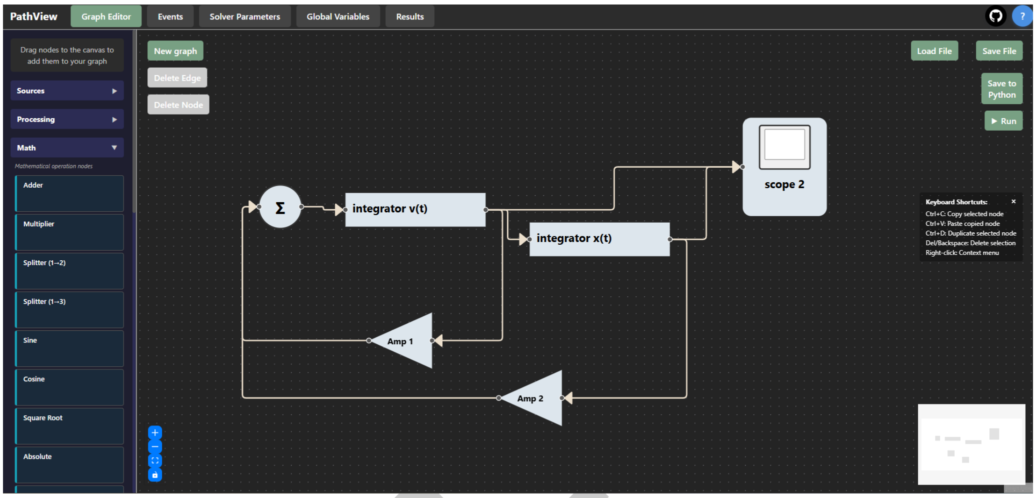

Figure 1: Graph editor tab (harmonic oscillator demo).

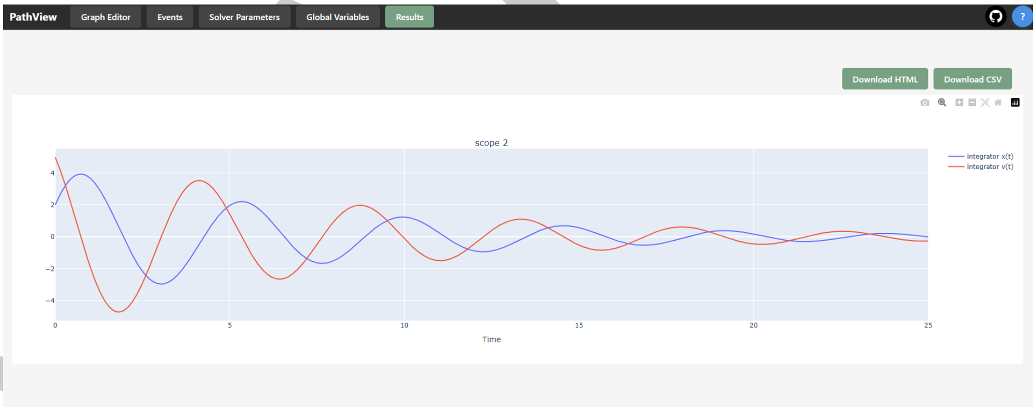

Figure 2: Results tab with interactive graph (harmonic oscillator demo).

We also provide an example demonstrating the integration of external tools in `festim_two_walls.json`: a two-wall hydrogen diffusion model integrating the FESTIM hydrogen tranport code (Delaporte-Mathurin et al., 2024).

Features

- **Node creation and connection**: Choose from 60+ different simulation node types, configure parameters, and connect them visually.
- **Integrated simulation**: Run `PathSim` simulations directly from the GUI.
- **Interactive visualisation**: Embedded `plotly` plots for interactive data exploration. Export results data to CSV or export graph to HTML.
- **Advanced global variables**: Define global variables, including via an integrated Python editor for complex expressions.
- **Flexible I/O**: Save and load models in JSON format; export to Python scripts for advanced or automated use.

- **Modular and extensible**: Designed for easy integration of new node types and custom functionality.
- **Custom styling**: Change node colours for improved readability.

References

Aspen Plus | Leading Process Simulation Software | AspenTech. (n.d.). Retrieved August 13, 2025, from https://www.aspentech.com/en/products/engineering/aspen-plus

Delaporte-Mathurin, R., Dark, J., Ferrero, G., Hodille, E. A., Kulagin, V., & Meschini, S. (2024). FESTIM: An open-source code for hydrogen transport simulations. *International Journal of Hydrogen Energy*, *63*, 786–802. https://doi.org/10.1016/j.ijhydene.2024.03.184

Meschini, S., Delaporte-Mathurin, R., Tynan, G. R., & Ferry, S. E. (2025). Impact of trapping on tritium self-sufficiency and tritium inventories in fusion power plant fuel cycles. *Nuclear Fusion*, *65*(3), 036010. https://doi.org/10.1088/1741-4326/adacfa

Meschini, S., Ferry, S. E., Delaporte-Mathurin, R., & Whyte, D. G. (2023). Modeling and analysis of the tritium fuel cycle for ARC- and STEP-class D-T fusion power plants. *Nuclear Fusion*, *63*(12), 126005. https://doi.org/10.1088/1741-4326/acf3fc

Moritz Klack, et al., Peter Kogo. (2025). React flow - powerful open source libraries for building node-based UIs with react. In *GitHub repository*. GitHub. https://github.com/xyflow/xyflow

Pallets/jinja. (2025). Pallets. https://github.com/pallets/jinja

Rother, M. (2025). PathSim - a system simulation framework. *Journal of Open Source Software*, *10*(109), 8158. https://doi.org/10.21105/joss.08158

Simulink - Simulation and Model-Based Design. (n.d.). Retrieved August 13, 2025, from https://www.mathworks.com/products/simulink.html