The Journal of Open Source Software

Diff Studio: Ecosystem for Interactive Modeling by Ordinary Differential Equations

Viktor Makarichev[1¶], **Larisa Bankurova**[1], **Gennadii Zakharov**[1,3], **Leonid Stolbov**[1], **Steven Mehrman**[2], **Dan Skatov**[2], **Jeffrey Cohen**[2], **Paul Sass**[2], **Davit Rizhinashvili**[1], **and Andrew Skalkin**[1]

1 Datagrok Inc, USA **2** Johnson & Johnson Inc, USA **3** Wellcome Sanger Institute, UK ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Owen Lockwood

Reviewers:

- @idoby
- @josemanuel22

Submitted: 02 September 2025
Published: unpublished

Summary

Ordinary differential equations (ODEs) are crucial in modeling complex systems and phenomena. Their applications range from pharmacology and drug manufacturing to financial modeling and environmental studies.

Diff Studio is a high-performance TypeScript application for solving initial value problems (IVPs) for ODEs directly within web browsers. It consists of two components. The **Diff Grok library** implements numerical methods and formula parsing tools. The **Diff Studio application** integrates Diff Grok tools with **Datagrok**, a scientific computing platform free for personal and academic use.

Diff Studio provides an ecosystem for rapid development of ODE-based applications with reproducible and accessible models.

Statement of need

Scientific modeling of complex processes and phenomena uses ODEs. They are widely applied in diverse fields, including physical processes (Chicone, 2006), biochemical kinetics (Ingalls, 2013), drug delivery systems (Mircioiu et al., 2019), cloud computing (Ghomi et al., 2019), and population dynamics (Hastings, 2013).

Analytic methods providing exact solutions can be applied only to a limited class of ODEs. The use of analytic solutions often proves impractical due to their complexity (Hairer et al., 2008). Numerical methods computing approximate solutions are often preferred.

Many methods for solving ODEs have been recently developed (Hairer et al., 2008; Hairer & Wanner, 2002). These methods have been implemented in various software tools, including libraries and packages for programming languages and scientific computing environments. Notable examples include SUNDIALS (Gardner et al., 2022; Hindmarsh et al., 2005), Julia Differential Equations package (Rackauckas & Nie, 2017), SciPy (Virtanen et al., 2020), Maple (Maplesoft, 2025), Mathematica (Wolfram Research Inc., 2024), Matlab (The MathWorks Inc., 2022), and deSolve (Soetaert et al., 2010).

Most tools require expertise, shifting focus from research to development. The goal of this project is to develop an ecosystem providing a combination of a "no-code" approach with comprehensive capabilities for in-browser modeling and analysis.

The solution: Diff Studio

Diff Grok library

This library provides numerical methods and automatic generation of JavaScript code from a declarative problem specification. It includes:

- **Solving tools:** numerical methods for solving IVPs;
- **Scripting tools:** methods for automatic generation of JavaScript code that solves problems specified in the declarative form.

Solving tools implement:

- `mrt` - Modified Rosenbrock triple (MRT) (Shampine & Reichelt, 1997)
- `ros3prw` - the ROS3PRw method (Jax et al., 2021)
- `ros34prw` - the ROS34PRw method (Rang, 2015)

To solve

$$\begin{aligned} dy/dt &= f(t, y), \\ y(t_0) &= y_0 \end{aligned} \tag{1}$$

on the interval $[t_0, t_1]$, define an ODEs object. This object specifies the independent variable (t), its range ($[t_0, t_1]$), solution grid step size (h), initial conditions (y_0), right-hand side of the ODEs, tolerance, and names of dependent variables. Next, apply a selected method (`mrt`, `ros3prw` or `ros34prw`) to this object. The output consists of a list of `float64` arrays containing the values of the independent variable and the corresponding approximate solutions.

For example, consider

$$\begin{aligned} dx/dt &= x + y - t, \\ dy/dt &= xy + t \\ x(0) = 1&, y(0) = -1 \\ t \in [0, 2]&, h = 0.001 \end{aligned} \tag{2}$$

In this case, the ODEs object is defined as follows:

```
const task: ODEs = {
    name: 'Example',
    arg: {
        name: 't',
        start: 0,
        finish: 2,
        step: 0.001,
    },
    initial: [1, -1],
    func: (t: number, y: Float64Array, output: Float64Array) => {
        output[0] = y[0] + y[1] - t;
        output[1] = y[0] * y[1] + t;
    },
    tolerance: 1e-7,
    solutionColNames: ['x', 'y'],
};
```

The following code solves the given problem:

```
const solution = mrt(task);
```

The solution contains three items:

- `solution[0]` - values of `t`, i.e., the range `0..2` with the step `0.001`;
- `solution[1]` - values of `x(t)` at the points of this range;
- `solution[2]` - values of `y(t)` at the same points.

Diff Grok delivers outstanding computational performance (Datagrok Inc, 2025b), benchmarked on **Robertson** (Robertson, 1966), **HIRES** (Schäfer, 1975), **VDPOL** (Pol (van der), 1926), **OREGO** (Hairer & Wanner, 2002), **E5** (Hairer & Wanner, 2002), and **Pollution** (Verwer, 1994). It allows users to obtain the modeling results in near-real time.

Scripting tools enable specification of IVPs declaratively using an intuitive syntax (see Figure 1).

```
#name: Example

#equations:
  dx/dt = x + y - t
  dy/dt = x * y + t

#argument: t
  initial = 0
  final = 2
  step = 0.001

#inits:
  x = 1
  y = -1

#tolerance: 1e-7
```

Figure 1: Diff Studio model corresponding to the Equation 2.

The method `getIVP()` parses a model and produces the IVP object specifying the problem. The method `getJScode()` generates JavaScript code, involving an appropriate ODEs object, that can be applied for solving equations.

Diff Studio package

The Diff Studio package integrates Diff Grok with Datagrok (Datagrok Inc, 2025a). It implements an application with a model editor (Figure 2).

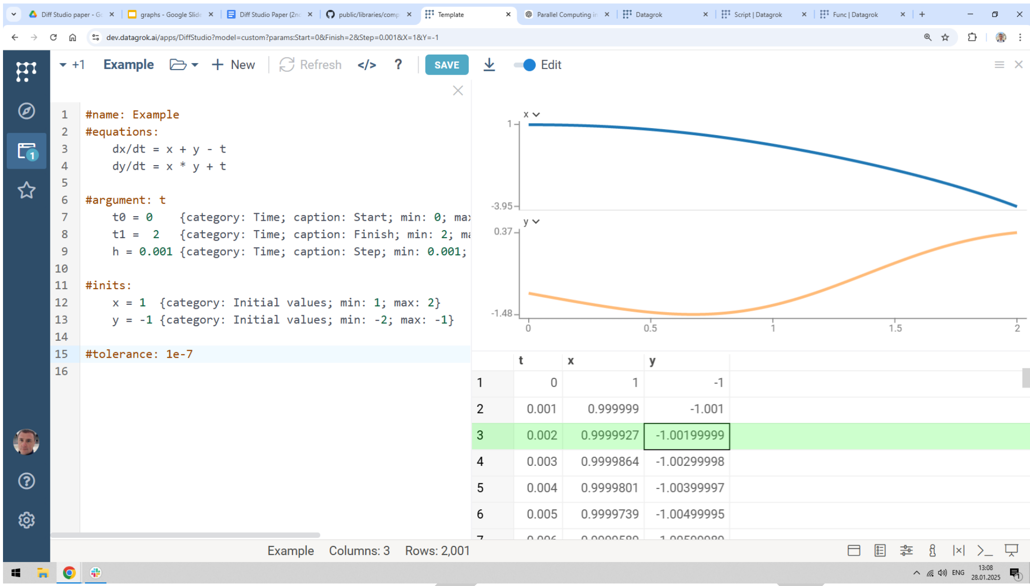

Figure 2: The Diff Studio application: the equation editor, numerical solution of the problem Equation 2, and its visualization.

Diff Studio automatically generates the user interface (Figure 3). Each model input can be annotated using self-explanatory options (Figure 4).

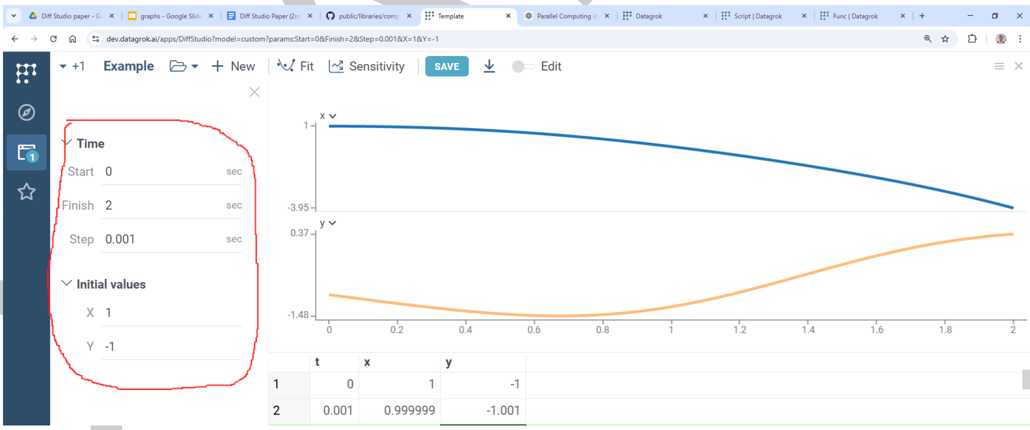

Figure 3: The Diff Studio application, Autogenerated UI: Diffstudio creates input entries (highlighted) for all variables listed in the equation editor. Each time model inputs are changed, a solution is computed and displayed.

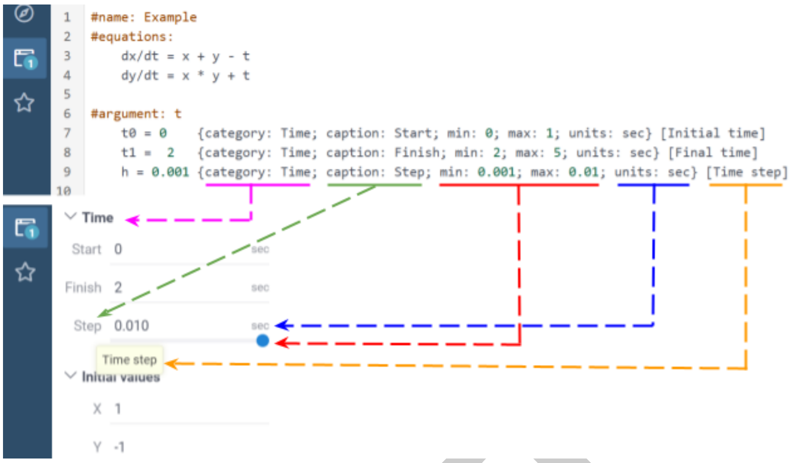

Figure 4: The correspondence of input annotation from Figure 2 and UI elements from Figure 3

Datagrok provides in-browser computations and visualizations. Other features include:

- Sensitivity analysis and parameter optimization;
- Storing and sharing computations.

Thus, Diff Studio serves as a comprehensive modeling environment.

Availability

Diff Grok is available on GitHub.

The Diff Studio package is accessible on GitHub, while its documentation can be found at Datagrok Help pages.

Run Diff Studio online here, or complete an interactive tutorial.

Acknowledgements

The authors are grateful to the entire **Datagrok Inc** team and to the **JnJ ModelHub** project team for their contributions and feedback, which significantly improved the project.

Conflicts of interest

The authors declare no conflict of interest.

References

Chicone, C. (2006). *Ordinary differential equations with applications* (2nd ed., Vol. 34). Springer. https://doi.org/10.1007/0-387-35794-7

Datagrok Inc. (2025a). *Datagrok: Swiss army knife for data.* https://datagrok.ai/

Datagrok Inc. (2025b). *Diff-grok performance.* https://github.com/datagrok-ai/diff-grok?tab=readme-ov-file#performance

Gardner, D. J., Reynolds, D. R., Woodward, C. S., & Balos, C. J. (2022). Enabling new flexibility in the SUNDIALS suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS).* https://doi.org/10.1145/3539801

Ghomi, J. E., Rahmani, A. M., & Qader, N. N. (2019). Applying queue theory for modeling of cloud computing: A systematic review. *Concurrency and Computation: Practice and Experience*, *31*(17), e5186. https://doi.org/10.1002/cpe.5186

Hairer, E., Nørsett, S. P., & Wanner, G. (2008). *Solving ordinary differential equations I: Nonstiff problems* (2nd ed., Vol. 8). Springer. https://doi.org/10.1007/978-3-540-78862-1

Hairer, E., & Wanner, G. (2002). *Solving ordinary differential equations II: Stiff and differential-algebraic problems* (2nd ed., Vol. 14). Springer. https://doi.org/10.1007/978-3-642-05221-7

Hastings, A. (2013). Population dynamics. In S. A. Levin (Ed.), *Encyclopedia of biodiversity* (2nd ed., pp. 175–181). Academic Press. https://doi.org/10.1016/B978-0-12-384719-5.00115-5

Hindmarsh, A. C., Brown, P. N., Grant, K. E., Lee, S. L., Serban, R., Shumaker, D. E., & Woodward, C. S. (2005). SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, *31*(3), 363–396. https://doi.org/10.1145/1089014.1089020

Ingalls, B. P. (2013). *Mathematical modeling in systems biology: An introduction*. MIT Press. ISBN: 9780262018883

Jax, T., Bartel, A., Ehrhardt, M., Günther, M., & Steinebach, G. (2021). *Rosenbrock-Wanner-Type methods*. Springer. https://doi.org/10.1007/978-3-030-76810-2

Maplesoft. (2025). *Maple*. https://www.maplesoft.com/products/maple/

Mircioiu, C., Voicu, V., Anuta, V., Tudose, A., Celia, C., Paolino, D., Fresta, M., Sandulovici, R., & Mircioiu, I. (2019). Mathematical modeling of release kinetics from supramolecular drug delivery systems. *Pharmaceutics*, *11*(3), 140. https://doi.org/10.3390/pharmaceutics11030140

Pol (van der), B. (1926). On 'relaxation-oscillations'. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 978–992. https://doi.org/10.1080/14786442608564127

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), 15. https://doi.org/10.5334/jors.151

Rang, J. (2015). Improved traditional Rosenbrock-Wanner methods for stiff ODEs and DAEs. *Journal of Computational and Applied Mathematics*, *286*, 128–144. https://doi.org/10.1016/j.cam.2015.03.010

Robertson, H. H. (1966). The solution of a set of reaction rate equations. In J. Walsh (Ed.), *Numerical analysis: An introduction* (pp. 178–182). Academic Press.

Schäfer, E. (1975). A new approach to explain the 'high irradiance responses' of photomorphogenesis on the basis of phytochrome. *Journal of Mathematical Biology*, *2*, 41–56. https://doi.org/10.1007/BF00276015

Shampine, L. F., & Reichelt, M. W. (1997). The MATLAB ODE suite. *SIAM Journal on Scientific Computing*, *18*(1), 1–22. https://doi.org/10.1137/S1064827594276424

Soetaert, K., Petzoldt, T., & Setzer, R. W. (2010). Solving differential equations in r: Package deSolve. *Journal of Statistical Software*, *33*(9), 1–25. https://doi.org/10.18637/jss.v033.i09

The MathWorks Inc. (2022). *MATLAB version: 9.13.0 (R2022b)*. The MathWorks Inc. https://www.mathworks.com/products/matlab.html

Verwer, J. (1994). Gauss-seidel iteration for stiff ODEs from chemical kinetics. *SIAM Journal* *on Scientific Computing*, *15*(5), 1243–1250. https://doi.org/10.1137/0915076

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Wolfram Research Inc. (2024). *Mathematica version 14.1*. https://www.wolfram.com/mathematica