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Published: unpublished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), typically formulated as initial value problems (IVPs), are widely used to model the dynamics of complex systems. They are utilized in many domains, 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).

While a broad ecosystem of mature numerical solvers exists, effective **interactive exploration**, **reproducible computation**, and **collaborative model development** remain difficult to achieve - particularly in browser-based environments. Most existing tools are designed around desktop or scripting workflows and provide limited support for real-time interaction, model sharing, or collaborative analysis.

Diff Studio is a **browser-native environment for solving and exploring IVPs**, designed to support interactive, collaborative, and reproducible ODE modeling without custom software development or local environment setup. It enables users to define models declaratively, explore their behavior interactively, and share fully reproducible simulations through the browser.

Diff Studio consists of two components: 1. **Diff Grok**: general-purpose high-performance TypeScript library for solving ODE systems defined in a declarative form 2. **Diff Studio**: web application that integrates Diff Grok into the **Datagrok** scientific computing platform. It provides autogenerated user interface, rich interactive visualizations, and the ability to manage ODE models and simulations.

Statement of need

ODEs can be solved either analytically or numerically. Analytic methods provide exact solutions but apply only to limited classes of problems and are often impractical due to their complexity (Hairer et al., 2008). Numerical methods, which compute approximate solutions, are therefore the dominant approach. A wide range of such methods has been developed (Hairer et al., 2008; Hairer & Wanner, 2002) and is available in established scientific computing tools, including SUNDIALS (Gardner et al., 2022; Hindmarsh et al., 2005), Julia DifferentialEquations (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).

These tools provide robust and efficient solvers but are primarily designed for **desktop or scripting-centric workflows**. In practice, scientific modeling is often exploratory: researchers iteratively adjust parameters, inspect transient behavior, compare scenarios, and refine model structure. Collaboration across teams and institutions is also common. Existing solutions typically require local installation, environment configuration, and custom scripting, which complicates sharing models and reproducing results. Building interactive analysis tools on top of these systems often requires substantial additional programming effort, shifting focus away from scientific inquiry toward software engineering.

Diff Studio addresses these limitations by providing a **browser-native**, **interactive modeling environment** that combines numerical performance with ease of use. It supports low-code model definition, immediate visual feedback, and seamless sharing of fully reproducible simulations via the web.

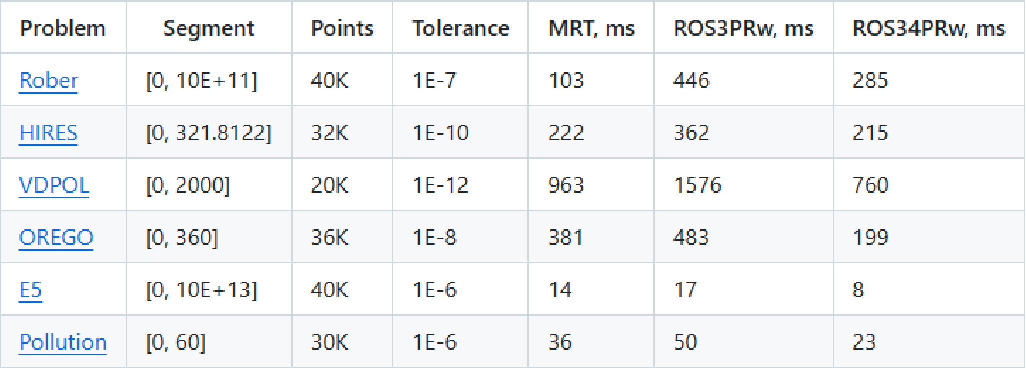
The solution: Diff Studio

Delivering high-performance ODE modeling in the browser introduces additional technical challenges. Approaches based on WebAssembly (*WebAssembly Specification*, 2025) or Pyodide (*Pyodide*, 2025) enable reuse of existing numerical libraries but impose important trade-offs. WebAssembly offers near-native performance for code written in C/C++ or Rust, but typically requires recompilation when equations or model structure change, limiting flexibility during iterative model design. Pyodide provides a WebAssembly-based Python environment that supports NumPy and SciPy but incurs large download sizes, potential performance overhead, and less seamless integration with browser APIs and reactive user interfaces.

In contrast, pure JavaScript and TypeScript solutions integrate naturally with the browser execution model and user interface frameworks. However, existing libraries such as Math.js (*MathJS*, 2025) and odex-js (*Odex-Js* , 2025) expose low-level APIs and generally require programming expertise from end users.

Diff Studio adopts a TypeScript-native approach combined with declarative modeling to eliminate the need for programming expertise while preserving performance and flexibility. Its core numerical engine, Diff Grok, provides:

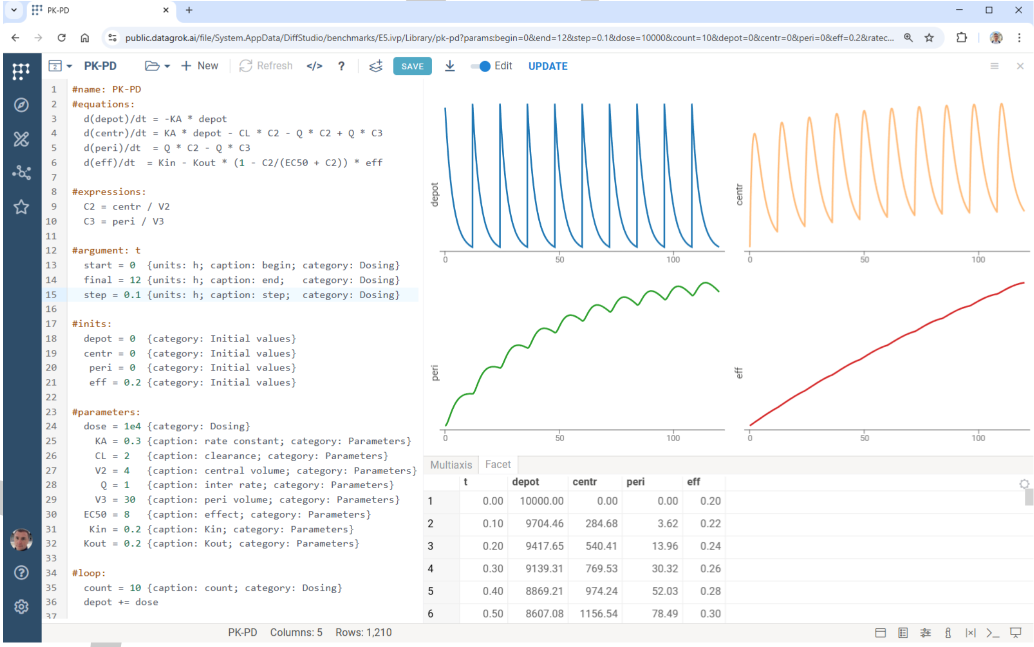
- **Solving tools**: A collection of numerical methods. Diff Grok implements the modified Rosenbrock triple (MRT) (Shampine & Reichelt, 1997), ROS3PRw (Jax et al., 2021), and ROS34PRw (Rang, 2015) methods, supporting both stiff and non-stiff systems. Performance was benchmarked on standard test problems including Robertson (Robertson, 1966), HIRES (Schäfer, 1975), VDPOL (Pol (van der), 1926), OREGO (Hairer & Wanner, 2002), E5 (Hairer & Wanner, 2002), and Pollution (Verwer, 1994). The results demonstrate near-real-time performance suitable for interactive exploration (see Figure 1).
- **Computational pipelines**: Support for multi-stage modeling and solving workflows executed in web workers, enabling parallel computation. These pipelines are used for parameter optimization and sensitivity analysis directly in the browser.
- **Declarative modeling language**: A domain-specific language for specifying IVPs as text, including equations and annotated model inputs. This representation supports automatic interface generation.



Problem	Segment	Points	Tolerance	MRT, ms	ROS3PRw, ms	ROS34PRw, ms
Rober	[0, 10E+11]	40K	1E-7	103	446	285
HIRES	[0, 321.8122]	32K	1E-10	222	362	215
VDPOL	[0, 2000]	20K	1E-12	963	1576	760
OREGO	[0, 360]	36K	1E-8	381	483	199
E5	[0, 10E+13]	40K	1E-6	14	17	8
Pollution	[0, 60]	30K	1E-6	36	50	23

Figure 1: Diff Grok performance: computational time comparison.

Diff Studio integrates Diff Grok into the Datagrok platform, providing a complete web application for ODE modeling. It includes an equations editor (Figure 2), an autogenerated interactive user interface (Figure 3), and rich visualization capabilities.


PK-PD
public.datagrok.ai/file/System.AppData/DiffStudio/benchmarks/E5.ivp/Library/pk-pd?params:begin=0&end=12&step=0.1&dose=10000&count=10&depot=0&centr=0&peri=0&eff=0.2&ratec...
PK-PD
New
Refresh
</>
?
SAVE
Edit
UPDATE
depot
centr
peri
eff
0
50
100
Multiaxis
Facet

```
#name: PK-PD
#equations:
  d(depot)/dt = -KA * depot
  d(centr)/dt = KA * depot - CL * C2 - Q * C2 + Q * C3
  d(peri)/dt  = Q * C2 - Q * C3
  d(eff)/dt  = Kin - Kout * (1 - C2/(EC50 + C2)) * eff

#expressions:
  C2 = centr / V2
  C3 = peri / V3

#argument: t
  start = 0  {units: h; caption: begin; category: Dosing}
  final = 12 {units: h; caption: end;   category: Dosing}
  step = 0.1 {units: h; caption: step;  category: Dosing}

#inits:
  depot = 0  {category: Initial values}
  centr = 0  {category: Initial values}
   peri = 0  {category: Initial values}
   eff = 0.2 {category: Initial values}

#parameters:
  dose = 1e4 {category: Dosing}
    KA = 0.3 {caption: rate constant; category: Parameters}
    CL = 2   {caption: clearance; category: Parameters}
    V2 = 4   {caption: central volume; category: Parameters}
     Q = 1   {caption: inter rate; category: Parameters}
    V3 = 30  {caption: peri volume; category: Parameters}
  EC50 = 8   {caption: effect; category: Parameters}
   Kin = 0.2 {caption: Kin; category: Parameters}
  Kout = 0.2 {caption: Kout; category: Parameters}

#loop:
  count = 10 {caption: count; category: Dosing}
  depot += dose
```

	t	depot	centr	peri	eff
1	0.00	10000.00	0.00	0.00	0.20
2	0.10	9704.46	284.68	3.62	0.22
3	0.20	9417.65	540.41	13.96	0.24
4	0.30	9139.31	769.53	30.32	0.26
5	0.40	8869.21	974.24	52.03	0.28
6	0.50	8607.08	1156.54	78.49	0.30

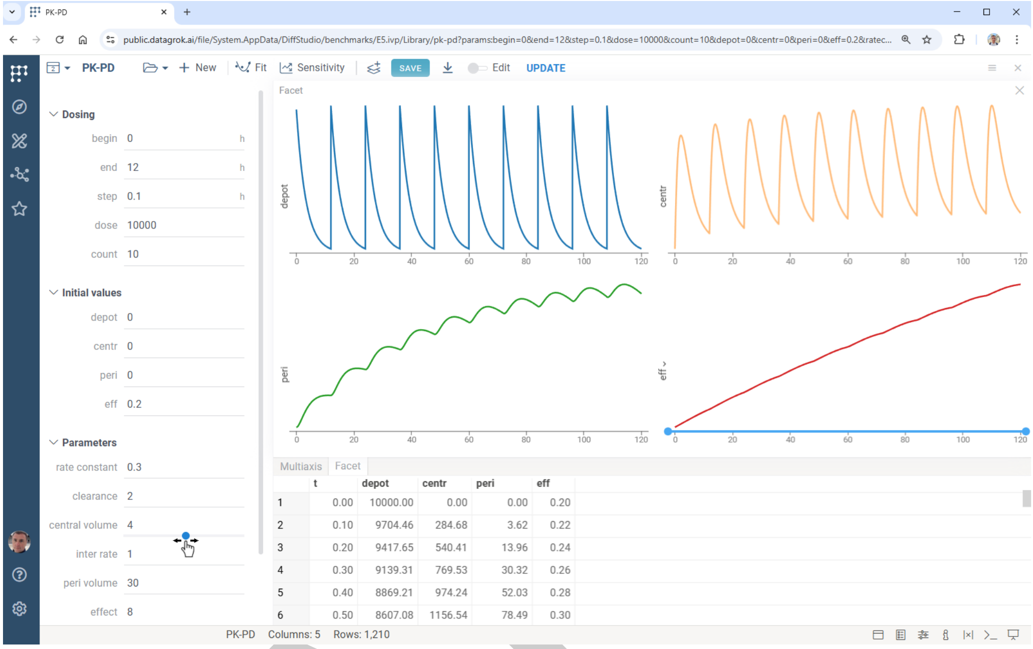
PK-PD Columns: 5 Rows: 1,210

Figure 2: Pharmacokinetic–pharmacodynamic simulation in Diff Studio, showing the declarative model specification, numerical solution, and interactive visualization.

A defining feature of Diff Studio is its real-time interactivity. Whenever a user adjusts a parameter or moves a slider, the system automatically recomputes the solution and updates all visualizations. Diff Grok enables these updates to occur almost instantaneously, supporting rapid hypothesis testing and intuitive exploration of model behavior.

Diff Studio also emphasizes reproducibility and collaboration. Models, parameter settings, and results can be shared via URLs, allowing collaborators to reproduce simulations without installing software or configuring environments. Shared models remain fully interactive, enabling continued exploration and comparison.

Beyond basic simulation, Datagrok provides built-in sensitivity analysis and parameter optimization, allowing users to study parameter influence and fit models to data. Together, these capabilities position Diff Studio as a centralized, browser-based hub for ODE-driven modeling, exploration, and collaboration.


PK-PD
public.datagrok.ai/file/System.AppData/DiffStudio/benchmarks/E5.ivp/Library/pk-pd?params:begin=0&end=12&step=0.1&dose=10000&count=10&depot=0&centr=0&peri=0&eff=0.2&ratec...
PK-PD
New
Fit
Sensitivity
SAVE
Edit
UPDATE
Facet
depot
centr
peri
eff
0
20
40
60
80
100
120
Multiaxis
Facet

Dosing

begin 0 h

end 12 h

step 0.1 h

dose 10000

count 10

Initial values

depot 0

centr 0

peri 0

eff 0.2

Parameters

rate constant 0.3

clearance 2

central volume 4

inter rate 1

peri volume 30

effect 8

	t	depot	centr	peri	eff
1	0.00	10000.00	0.00	0.00	0.20
2	0.10	9704.46	284.68	3.62	0.22
3	0.20	9417.65	540.41	13.96	0.24
4	0.30	9139.31	769.53	30.32	0.26
5	0.40	8869.21	974.24	52.03	0.28
6	0.50	8607.08	1156.54	78.49	0.30

PK-PD Columns: 5 Rows: 1,210

Figure 3: Diff Studio in model exploration mode, showing the autogenerated user interface with live parameter controls.

Availability

- Diff Grok: https://github.com/datagrok-ai/diff-grok
- Diff Studio: https://github.com/datagrok-ai/public/tree/master/packages/DiffStudio
- Run Diff Studio online: https://public.datagrok.ai/apps/DiffStudio
- Interactive tutorial: https://public.datagrok.ai/apps/tutorials/Tutorials/Scientificcomputing/Differentialequations

Acknowledgements

The authors thank the entire **Datagrok** team and the **JnJ ModelHub** project team for their contributions and feedback, which significantly improved this work.

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

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/

MathJS. (2025). https://mathjs.org/

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

Odex-js : ODEX in JavaScript. (2025). https://github.com/littleredcomputer/odex-js

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

Pyodide. (2025). https://pyodide.org/

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

WebAssembly specification. (2025). https://webassembly.github.io/spec/core/

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