# Overview of SUNDIALS

Consortium for the Advancement of Scientific Software (CASS)
BoF Days, 2026

Feb. 12, 2026

Carol S. Woodward (LLNL)

In collaboration with the SUNDIALS team: Cody J. Balos (LLNL), David J. Gardner (LLNL), Dan Reynolds (UMBC), Steven B. Roberts (LLNL)

Prepared by LLNL under Contract DE-AC52-07NA27344.

BOF Days

February 10 - 12, 2026

https://cass.community

# The Consortium for the Advancement of Scientific Software (CASS)

CASS

BOF DAYS

https://cass.community/news/2026-02-10-cass-bof-days.html

# CASS: Stewardship and Advancement of the Scientific Software Ecosystem

- **Inward-facing activities:** Strengthening software products
  - Improve development practices, sustainability, quality, and trustworthiness
  - Enhance user experience and integration within the broader ecosystem
- **Outward-facing activities:** Community engagement and discovery
  - Curate and evolve the software portfolio
  - Help teams connect with and grow their user communities
  - Enable the broader community to discover and adopt useful software

CASS Members

**CORSA**
Partnering with foundations to provide sustainable pathways for scientific software

**FASTMATH**
Stewardship, advancement, and integration for math and ML/AI packages

**PESO**
Stewarding, evolving and integrating a cohesive ecosystem for DOE software

**RAPIDS**
Stewardship, advancement, and integration for data, visualization and ML/AI packages

**S4PST**
Stewardship, advancement and engagement for programming systems

**STEP**
Stewardship, advancement of software tools for understanding performance and behavior

Sponsored by the Department of Energy, Office of Advanced Scientific Computing Research

# Engage with CASS!

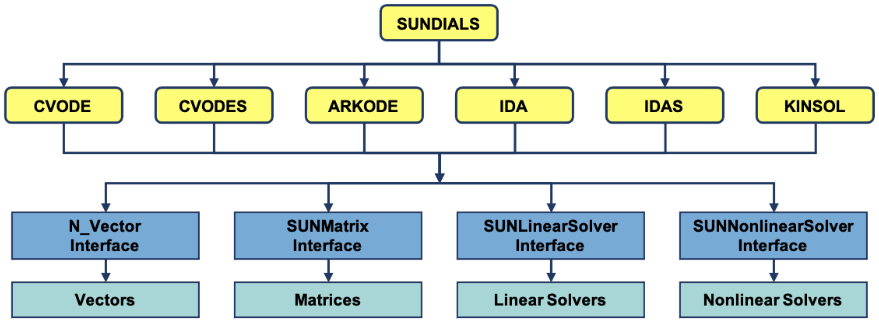
- Learn about CASS:
  - https://cass.community/about/
- Join the CASS Announcement list (low-volume):
  - http://eepurl.com/iRiSnY
- Find out more about our **software products**
  - Catalog: https://cass.community/software/
  - Collected as part of the Extreme-Scale Scientific Software Stack (E4S)
- Participate in **CASS Working Groups**
  - Impact Framework, Integration, Metrics, Software Ecosystem, User-Developer Experience, Workforce
  - https://cass.community/working-groups/

# BoF Session Outline

- “Overview of SUNDIALS,” Carol Woodward (LLNL)
- “New Features in SUNDIALS,” Daniel Reynolds (UMBC)
- “Python Interfaces in SUNDIALS,” David Gardner (LLNL)
- “Using SUNDIALS to Efficiently Drive Time-Integration in AMReX-Based PDE Solvers,” Andy Nonaka (LBNL)
- “Dynamics of atomic nuclei with SUNDIALS,” Francesca Bonaiti (MSU)
- “Towards near-real-time discontinuous Galerkin gyrokinetic simulations of fusion plasmas: collisions via super time stepping,” Manaure Francisquez (PPPL)
- “Integrating SUNDIALS into MATLAB,” Brett Buckner (Mathworks)
- Discussion
  - New or interesting uses of SUNDIALS
  - Identify features needed for further SUNDIALS development
  - Share successes and failures of SUNDIALS uses among applications

# SUite of Nonlinear and DIfferential-ALgebraic Solvers

- SUNDIALS is a software library consisting of ODE and DAE integrators and nonlinear solvers
- Written in C with interfaces to Fortran and Python
- New interfaces in Matlab starting with release R2024a
- Designed to be incorporated into existing codes
- Includes a rich infrastructure of support on exascale systems and applications
- Freely available; released under the BSD 3-Clause license ( >100,000 clones/downloads per year)
- Active user community supported by sundials-users email list
- Detailed user manuals included with each package and online at https://sundials.readthedocs.io



- Nonlinear and linear solvers and all data use is fully encapsulated from the integrators and can be user-supplied
- All parallelism is encapsulated in vector and solver modules and user-supplied functions

**https://computing.llnl.gov/casc/sundials**

# SUNDIALS brings methods to nonlinear time dependent systems

- CVODE, IDA, and sensitivity analysis variants (forward and adjoint), CVODES and IDAS, use linear multistep methods
  - CVODE solves ODEs, $\dot{y} = f(t, y)$
  - IDA solves DAEs, $F(t, y, \dot{y}) = 0$
  - Adaptive in both order and step sizes; include event detection
  - Both packages include stiff BDF methods; CVODE includes nonstiff Adams-Moulton methods
- ARKODE provides adaptive one-step, multistage time integration methods
  - Solves $M\dot{y} = f_I(t, y) + f_E(t, y), \quad y(t_0) = y_0 \quad M(t)$: any nonsingular (optionally time-dependent) mass matrix
  - Supports adaptive time steps through multistage embedded methods
  - Six steppers:
    - ARKStep: explicit, implicit, and additive ImEx Runge-Kutta methods; with adaptivity, event detection, etc.
    - ERKStep: streamlined explicit RK implementation
    - LSRKStep: Low storage explicit RK methods
    - MRIStep: multirate infinitesimal step methods
    - SplittingStep: Operator splitting methods (low and high order)
    - SPRKStep: symplectic explicit RK for partitioned Hamiltonian systems (conserves energy or similar quantities)
  - Parallel-in-Time support: XBraid wrappers for SUNDIALS vectors and explicit, implicit, and ImEx methods in ARKStep
- KINSOL solves nonlinear algebraic systems with Newton or accelerated fixed point methods

# What are some of the newer features in SUNDIALS?

- **Command line control over SUNDIALS options** - enables the use of AI/ML tools to tailor SUNDIALS options to a specific application and improve performance
- **Low storage RK methods** along with dominant eigenvalue estimation including both Power iteration and Arnoldi dominant eigenvalue estimation options provided
- **An interface to batched, iterative, linear solvers from the Ginkgo library** - allows any batched solver in Ginkgo to be utilized
- **Discrete adjoints for explicit methods in ARKStep and ERKStep**
- **Python interfaces** to enable greater SUNDIALS use in the Python and ML/AI communities, with zero copy data exchange
- **Operator splitting** module with arbitrary order methods

# What are we working on now?

- Discrete adjoint capability for explicit multirate RK methods in ARKODE (with constant time steps)
- Complex vector and data structure support
- Embedded SSP ImEx methods
- Automatic switching between Newton and fixed point solvers for problems that change stiffness

# Where to learn more and get the software

Visit the SUNDIALS website: https://computing.llnl.gov/sundials

See presentations list at the SUNDIALS publications page: https://computing.llnl.gov/sundials/publications

Visit the SUNDIALS GitHub page: https://github.com/LLNL/sundials

Clone the repository or download the tarball from the SUNDIALS GitHub page and build with CMake: https://github.com/LLNL/sundials/releases

Detailed user manuals online: https://sundials.readthedocs.io

Install SUNDIALS using Spack: “spack install sundials”

**https://computing.llnl.gov/sundials**

# SUNDIALS Team

Current Team:

Cody Balos

David Gardner

Dan Reynolds

Steven Roberts

Carol Woodward

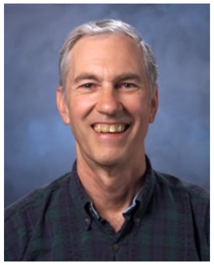
PostDocs (UMBC):

Mustafa Ağgül

Sylvia Amihere

Yifan Hu

Alumni:


Alan Hindmarsh

Radu Serban

Scott D. Cohen, Peter N. Brown, George Byrne, Allan G. Taylor, Steven L. Lee, Keith E. Grant, Aaron Collier, Lawrence E. Banks, Steve G. Smith, Cosmin Petra, Slaven Peles, John Loffeld, Dan Shumaker, Ulrike M. Yang, James Almgren-Bell, Shelby L. Lockhart, Rujeko Chinomona, Daniel McGreer, Hunter Schwartz, Hilari C. Tiedeman, Ting Yan, Jean M. Sexton, and Chris White

# Acknowledgements

*This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientic Discovery through Advanced Computing (SciDAC) Program through the FASTMath Institute*

**computing.llnl.gov/sundials**

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR) via the Frameworks, Algorithms, and Scalable Technologies for Mathematics Institute Scientific Discovery through Advanced Computing (SciDAC) program.