LSODA Stiff Solver

Arjun Ray

Undergraduate Research Assistant I
rayarjun@frib.msu.edu

14 August 2025

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics and used resources of the Facility for Rare Isotope Beams (FRIB) Operations, which is a DOE Office of Science User Facility under Award Number DE-SC0023633, and by the US National Science Foundation under Grants No. PHY-20-12040 and 23-10078 “Windows on the Universe: Open Quantum Systems in Atomic Nuclei at FRIB”.

LSODA.cpp

LSODA stands for *Livermore Solver for Ordinary Differential equations with Automatic method switching written by Alan C. Hindmarsh and Linda R. Petzold.*

Paper – **https://computing.llnl.gov/sites/default/files/ODEPACK_pub1_u88007.pdf**

LSODA Features

Hybrid Solver – Combines two methods: fast Adams predictor–corrector for nonstiff regions and stable implicit Backwards Differentiation (BDF) for stiff regions, switching automatically when stiffness is detected.

Stiffness Detection – Monitors error growth vs. step size; switches to BDF before instability occurs, so it never wastes time taking thousands of tiny steps like ode.cpp.

Adaptive Step Size & Order – Continuously tunes both step length and polynomial order for efficiency and accuracy, ensuring minimal computation while meeting tolerance requirements.

Banded Jacobian Support – Can exploit narrow bandwidth in ETACHA's Jacobian, cutting stiff-solve cost and giving massive speed-ups for large systems.

Implementation

Why LSODA Needs Banded Jacobians

- Banded LU slashes cost and memory(~60%).
- Potentially **1000× faster** in stiff phases.

Why LSODA is Faster When Stiff

- Detects stiffness early, switches to large stable implicit steps.
- ode.cpp shrinks steps until max-step error; LSODA leaps over stiff bottleneck.

LSODA Advantages

- **Euler, Adams PECE, RK45** all work well for non-stiff problems but fail or become extremely slow when stiffness appears.
- **LSODA** automatically switches between non-stiff (Adams) and stiff (BDF) methods, so it can handle both without user intervention.

LSODA.cpp is around 60% faster than the original Fortran version EQDIF.for.

LISE package