Arjun Ray

Gear’s Backward Differentiation

Facility for Rare Isotope Beams, Michigan State University, East Lansing, MI 48824 USA

2025

This material is based upon work supported by the U.S. Department of Energy Office of Science under Cooperative Agreement DE-SC0000661, the State of Michigan and Michigan State University. Michigan State University operates FRIB as a DOE Office of Science National User Facility in support of the mission of the Office of Nuclear Physics.

Gear's BDF

Backwards Differentiation is an implicit numerical method for solving stiff ODE's, where the derivative at the next time step is used, instead of the current one(future value to calculate current step).

It provides improved stability over explicit methods, especially for rapidly changing or stiff systems

- BDF1:

$$y_{n+1} - y_n = hf(t_{n+1}, y_{n+1})$$

(this is the backward Euler method)

- BDF2:

$$y_{n+2} - \frac{4}{3}y_{n+1} + \frac{1}{3}y_n = \frac{2}{3}hf(t_{n+2}, y_{n+2})$$

-The two on the left(orders 1 and 2) are best for solving the stiffness problem in our integrator in ETACHA4.

Code Implementation

Ideally, the implementation in code would look something like this-

- Blend the most recent solutions to make an initial guess for the next step.
- Iteratively refine that guess by checking it against your ODE model until the change is tiny.
- Save the new result, confirm it meets your accuracy needs, tweak the step size, and repeat.

SUNDIALS(https://computing.llnl.gov/projects/sundials/cvode) already has a pre-existing library with an industry grade BDF solver.

Furthermore, python also has an inbuilt BDF solver in the scipy library.