Impacts of improved time evolution in BISICLES using SUNDIALS

Daniel F. Martin (DFMartin@lbl.gov)[1], Steven B. Roberts[2], Hans Johansen[1], David J. Gardner[2], Carol S. Woodward[2]

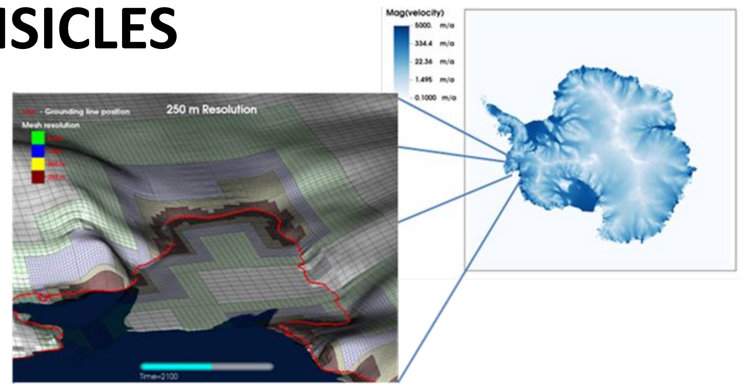
1. Lawrence Berkeley National Laboratory, Berkeley, CA, USA **2**. Lawrence Livermore National Laboratory, Livermore, CA, USA

The Big Picture (TLDR…)

The necessity of high spatial accuracy for accurate simulation of ice sheet dynamics has been well-demonstrated. However, many models use low-accuracy and explicit temporal integration schemes. **Our goal:** Using a modern time-integrator library (SUNDIALS), we investigate the impacts of more-sophisticated time integration schemes in the BISICLES model.

BISICLES

We use the BISICLES ice sheet model [1], an adaptive mesh refinement (AMR) ice sheet model which dynamically adds refined meshes where needed to assure solution accuracy.



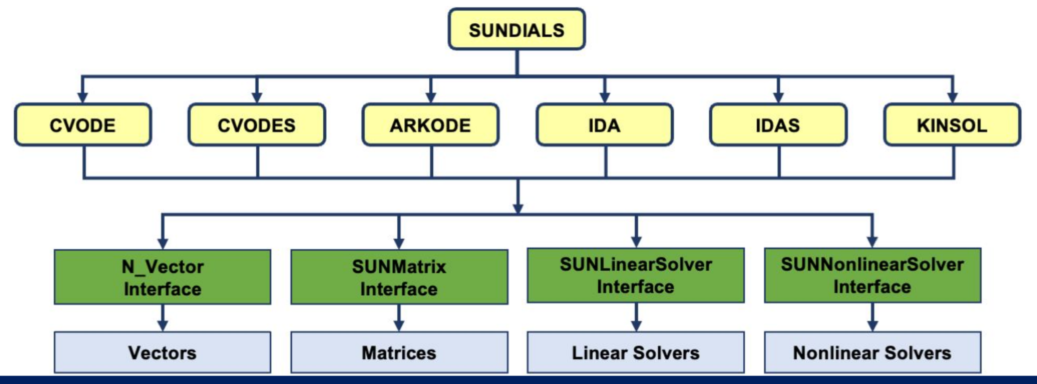
BISICLES-computed Antarctic ice velocity field. Inset shows adaptive meshing near the Pine Island Glacier grounding line.

However, while BISICLES delivers high spatial accuracy, the baseline version employs an *explicit* temporal-integration scheme which is only first-order in time. While the ice-thickness evolution equation looks like a hyperbolic advection equation, it has a diffusive character, meaning that explicit schemes will also suffer from severe time step restrictions due to numerical stability constraints.

SUNDIALS

The **SU**ite of **N**onlinear and **DI**fferential/**AL**gebraic equation **S**olvers (SUNDIALS) [3,4] provides robust and efficient time integration methods for ODE and DAE systems and iterative solvers for nonlinear algebraic systems.

In this work, we developed an `N_Vector` wrapper that enables SUNDIALS to operate on the AMR data structures used by BISICLES. Then we leveraged adaptive step size explicit Runge-Kutta methods implemented in the ARKODE package from SUNDIALS.



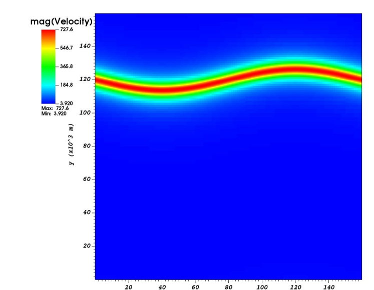
Support for this work was provided through the Scientific Discovery through Advanced Computing (SciDAC) program funded by the U.S. Department of Energy (DOE), Office of Science, Biological and Environmental Research and Advanced Scientific Computing Research programs, as a part of the FASTMath Institute and the ProSPect SciDAC Partnership.
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344 and at the Lawrence Berkeley National Laboratory under contract No. DE-AC02-05CH11231.

Improved Accuracy

- Mathematically speaking, if a method is *consistent* and *stable,* then it is also *convergent* – the error converges to zero at a rate which is proportional to the timestep to a power *p*:

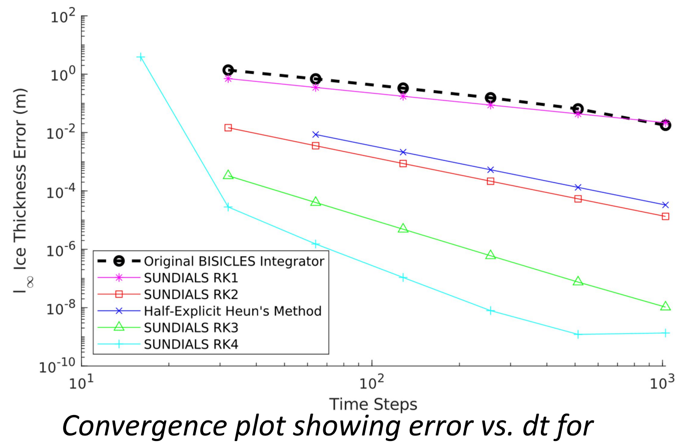
  Error $\propto (\Delta t)^p$

  where *p* is known as the *order* of the method



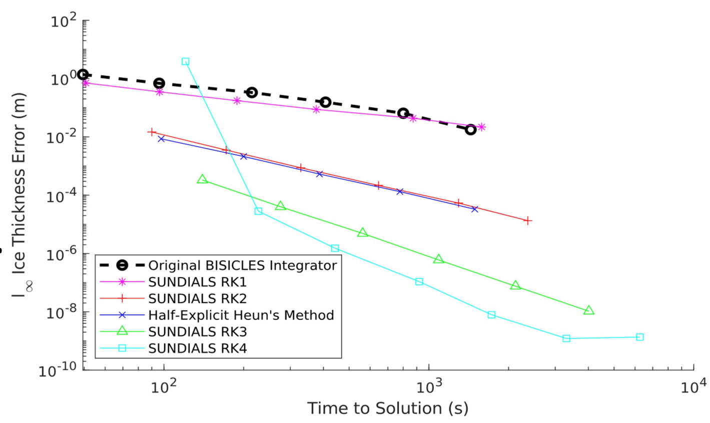
Ice-stream benchmark problem [2] – ice velocity field

- The BISICLES time integration scheme for ice thickness evolution is based on the native advection schemes in the Chombo library, which are based on an upwinded Piecewise Parabolic Method; use of a lagged velocity for computing thickness fluxes means that the scheme is only **1st order in time**.



Convergence plot showing error vs. dt for BISICLES-SUNDIALS for ice-stream benchmark.

- Coupling BISICLES to SUNDIALS allows easy experimentation with a range of time integrators; we tried 1st, 2nd, and 4th-order explicit Runge-Kutta and a half-explicit variant of Heun's method.
- Like many ice sheet models, BISICLES execution time is dominated by the nonlinear solve for the ice velocity field.



Higher-order methods – more solves! Is it worth it?

- Plotting solution error vs. execution time *(above)* shows the true value of higher-order time integration – better accuracy for less cost!

References

[1] `https://bisicles.lbl.gov`
[2] Cornford, S.L. et al: "Adaptive mesh, finite-volume modeling of marine ice sheets", *Journal of Computational Physics*, 232, 529-549, `https://doi.org/10.1016/j.jcp.2012.08.037`, 2013.
[3] `https://computing.llnl.gov/projects/sundials`
[4] Hindmarsh, Alan C., et al. "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers." *ACM TOMS*, 31.3, 363-396, `https://doi.org/10.1145/1089014.1089020`, 2005.

Improved Stability via Adaptive Timestepping

While it's tempting to think of ice thickness evolution as a simple transport (advection) problem, mathematically it behaves more as an advection-diffusion equation for the ice thickness.

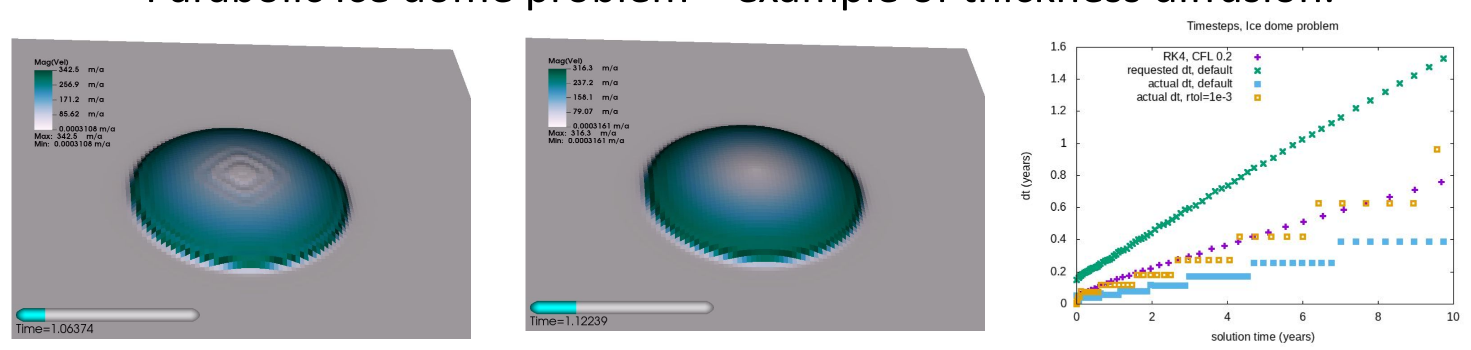
What this means:

- **Explicit** methods are the simplest to implement but suffer from an oppressive stability constraint: $\Delta t \propto (\Delta x)^2$ for stability!
  (as we move to higher resolution simulations, this starts to be painful!)
- **Implicit** methods are generally better-suited for diffusion problems, but the nonlinear nature of the ice velocity field makes this hard.

However, SUNDIALS has the option for automatic timestep control based on accuracy considerations. If instability manifests as a blowup in the solution error, can the automatic error control ensure solution stability even for an explicit method?

Test Case: Parabolic Ice Dome

Parabolic Ice dome problem – example of thickness diffusion.



Impact of automatic timestep control: dome problem after 1 year of simulation time.
(left) Original BISICLES integrator – note ridges near center which indicate numerical instability.
(center) Using SUNDIALS automatic timestep control – note smoothness of upper surface.
(right) Requested timesteps (green) vs. actual limited timestep (blue), and actual timestep with a tighter solution error tolerance (which results in more-conservative timesteps).

Conclusions

- Coupling BISICLES with the SUNDIALS time integrator library enables easy experimentation with time integration approaches.
- While they require more expensive nonlinear velocity solves per timestep, higher-order methods like RK4 still result in much smaller errors per unit of computational work (execution time).
- Automatic timestep control looks promising as a way to prevent numerical instabilities without the use of implicit methods.