The Ice Sheet System Model (ISSM, [1]) is an ice flow modeling software developed by NASA/JPL. It has a C++ code base of about 95,000 lines of code with data pre-processed by been handled in Matlab. The computation of large scale gradients for, e.g., sensitivity studies is achieved by employing the operator overloading AD-tool ADOL-C [2]. Notable features of ISSM:

- Separation between active and passive data types using alias definitions
- Thin wrapper layer for MPI to allow for serial or parallel model computation
- Treatment of calls to (direct) linear solvers (e.g., Mumps) with special wrappers

Thus, ISSM provides a validated AD implementation for studying other AD tools. We explore the AD tool CoDiPack [3] w.r.t. (i) performance improvements and (ii) feature completeness.

CoDiPack includes the following main features:

- ADOL-C (AD Core)
- Generates derivatives after core call by calling the respective driver functions.
- Memory Consumption
- Future Work

Collaboration with NASA/JPL ISSM developers on (i) additional tests for verification and (ii) setup of large, real world models.

Test Setup

- Hardware: 2×Intel Xeon Processor E5-2680 v3 at 2.5 GHz with 64 GB RAM
- The benchmark is based on a square domain mesh
- Complexity is controlled by generating a finer mesh.
- Reverse mode is employed, i.e., CoDiPack’s RealReverse and ADOL-C’s driver fos_reverse.
- Custom ADOL-C trace parameters to avoid file IO.

Aspects of a Type Change

Trivial changes to ISSM include (i) Autotools build system modifications (ii) alias declaration of active type (iii) additional overloads of mathematical functions added to CoDiPack as C++ style overloads, e.g., fmin.

Declaration of (In)Dependents

- CoDiPack uses the tape API, i.e., registerInput and registerOutput instead of &c= and &g= operators.
- Added a global structure to store indices of the active tape: ISSM/ADOL-C paradigm is to fetch indices provided by the user to set the weight vector before the AD driver call.

MPI

- No major changes needed: API of AdjIndexableMPI (6) ADOL-C and AMPI are almost identical.
- External functions: Linear Solver

- Taping of linear solver requires too much memory: Call special implementation during reverse and ADOL’s driver call.
- Usually the same code as ADOL-C: Instead of unpacking the values in one big vector, several vectors can be used (also available in the generalized external function interface).
- Less management for reverse functions. They are directly written to the tape.

References


Scientific Computing, TU Kaiserslautern

‡Paderborn University

Study of ADOL-C and CoDiPack Applied to the Ice Sheet System Model

Figure 2: Primal and AD timings for 16 MPI Processes w.r.t. to different mesh resolutions. Labels are omitted for values less than 2%. (AD Core) rate does not include the solve. Mumps_F is the solver call during the core computation. Mumps is the solve call during the backward solve. Each Mumps call takes ∼ 2.6 s.

The AD Core is the dominating factor w.r.t. total runtime, except for ADOL-C.