Abstract

Computing sensitivities of climate-related quantities of interest (QoI) to very high-dimensional state or parameter spaces in an efficient manner is of considerable interest in climate science. Applications in which sensitivities are propagated in time by means of adjoint models include quantifying controlling factors of the QoI’s variations [24, 3], optimal state and parameter estimation (“data assimilation”) [25, 26], computing nonnormal transient amplification patterns [27], and formal uncertainty quantification [10, 11]. In an environment of rapid model development algorithmic differentiation (AD) plays a crucial role in supporting adjoint model generation for diverse applications based on up-to-date source codes [9]. With applications ranging from oceanography [19], atmospheric sciences [28], coupled carbon cycle modeling [12] and cryospheric sciences [5, 2], the need for sustained AD tool maintenance and improvements becomes a high priority.

Background/Research to Date

Climate models depend on a large (because spatially varying) parameter set that represents subgrid-scale processes and whose values are inherently uncertain. Similarly, initial and boundary conditions (at the surface or base) are uncertain and will dictate the solution. To understand model sensitivities derivatives are needed for relating the outputs of a model back to its inputs, i.e. control variables. Model calibration, in particular, which seeks to determine optimal estimates of control variables informed by available observations, plays an increasingly important role in attempts to initialize models for prediction. The conventional approach of finite-difference approximation becomes intractable for high-dimensional control spaces and simulations which require days for single-model execution on today’s supercomputers. Furthermore, this approach can be inaccurate because of accumulating round-off errors.

An alternative method is the use of adjoint models. For scalar-valued QoI’s such as least-squares misfit functions used in model calibration or state estimation, or for sensitivity analysis of climate metrics the adjoint provides an extremely efficient means for computing linear sensitivities. In climate modeling, the challenge, then, is to maintain up-to-date versions of the adjoint model given the evolving nature of the climate model at hand. Writing the adjoint by hand has frequently shown to be cumbersome, and producing code that is increasingly out of date with respect to its parent code. It also is prone to additional coding error.

AD provides a powerful alternative [7]. It exploits the code’s composition of elementary arithmetic operations and elementary function evaluations whose analytical derivatives are known. Applying the chain rule of differentiation, derivatives can be propagated from one model variable to another. Various AD tools have been developed based on techniques from source transformation [21, 8, 4, 16] or operator-overloading [22, 17]. Current AD research is directed in part at increasing the efficiency of derivative code for components of large-scale climate models. Reverse-mode AD necessarily uses checkpointing where the application state is determined by source analysis. Many checkpointing schemes are being examined to efficiently...
use the available memory and disk while computing adjoints. Preliminary research allows the 
suspension and restart of the forward and the adjoint model. Adjoinable MPI seeks to dif-
ferentiate through the MPI calls in the model code. AD advances are exploiting the inherent 
mathematical properties of the model to allow larger adjoint models to be run. For example,
through the special treatment of fixed-point iterations, which differs from a straightforward 
application of AD software, both computational and memory requirements of the adjoint model 
are reduced.

Proposed Direction of work

We propose to extend the use and improve the efficiency of adjoint codes of climate model 
components. At the center of the effort is the open-source AD tool OpenAD [21].

- **New Languages** Many existing models are written in Fortran. As models migrate to C, 
  C++ [14, 20], R, and Python, we propose to apply existing source transformation tools such as 
  ADIC [16] for models written in C. Source transformation AD is not possible for C++ because 
  of features such as templates and overloading. Instead, we propose to apply a mixed method [13] 
  that uses operator overloading AD for most of the application and source transformation AD 
  for the computation-intensive portions that are often written in a C-like fashion.

- **Scaling** An important bottleneck for scaling of adjoint computations is the memory re-
  quirement for storing partials and checkpointing of data at the granularity of timesteps. We 
  will apply optimal multilevel checkpointing involving disk storage and memory [1, 18].

- **Robustness** The mean time between failure (MTBF) of leadership-class machines is on the 
  order of hours. Because adjoint computations will often exceed MTBF, we propose to support 
  the restart of the adjoint climate application using existing optimal checkpointing techniques in 
  AD [6]. We will study the implementation of the restart mechanism in the context of multilevel 
  checkpointing.

- **Chaotic Systems** Because of the increasingly nonlinear dynamics with time encountered 
  in climate simulations, the adjoint results can diverge exponentially from the “macroscopic 
  climate sensitivity” [15]. By replacing the initial value problem in these models with the well-
conditioned “least squares shadowing (LSS) problem”, it may be possible to compute well-
behaved derivatives of climate quantities. [23]. We propose to study the non-straightforward 
  application of AD tools for the LSS approach.

- **Usability and Applicability** Many climate modelers have been reluctant to employ AD. 
  We will therefore strive to make source transformation tools easily applicable to climate codes. 
  Moreover, we will work to demonstrate to the climate community the benefits of AD.

Connections to Math, Comp Sci & and Climate Science

The proposed work covers various fields, including, the solution of (discretized) PDEs, high-
performance computing (techniques such as checkpointing and resilience) and different compo-
nents of climate models (e.g. ice sheets, oceans, atmosphere, carbon cycle).

Potential Impact on the field

Climate modelers who already use AD will be able to run adjoint models at larger scales 
than they can currently run. Adjoint models that previously could not be executed because 
of machine failures can be executed with checkpointing support. Climate modelers who are 
hesitant to migrate to new languages will be able to do so more easily. Success stories may 
encourage other climate researchers to begin to use AD.

2
References


Acknowledgement

This work was supported in part by a grant from the U.S. Department of Energy, Office of Science, under contract DE-AC02-06CH11357.