Geosci. Model Dev. Discuss.,
doi:10.5194/gmd-2016-99-RC2, 2016
© Author(s) 2016. CC-BY 3.0 License.

Interactive comment on "An Approach to Computing Discrete Adjoints for MPI-Parallelized Models Applied to the Ice Sheet System Model" *by* Eric Larour et al.

Anonymous Referee #2

Received and published: 15 June 2016

The paper illustrates how Algorithmic Differentiation (AD) is implemented in the Ice Sheet System Model (ISSM) to compute discrete adjoints and sensitivities. The paper details the operator overloading approach used to achieve AD without disrupting the existing forward capability. Further it explains the parallelization is performed. Overhead factors for computing adjoints are reported for a couple of model runs.

Although the topic of the paper is surely interesting for this journal, I think the exposition is not always clear. In particular, in the introduction there are some typos, inaccuracies and undefined quantities. I would recommend the authors to revise the paper, particularly the introduction, in order to make it more understandable to the reader not used to forward and reverse AD. The paper focuses on several (important) technical details but I think it lacks a mathematical/algorithmic description of the overall framework.

I would recommend the authors to consider a typical sensitivity problem encountered in applications and show how it is solved in ISSM by reverse AD. An exemplar problem could be to find the sensitivity (derivative) of a functional $f = f(x, p)$ with respect to the parameter p, subject to the (nonlinear) constraint $c(x, p) = 0$ (here f could represent a misfit of some quantity with observational data, and c the ice sheet model). How this problem (or a similar one, maybe time dependent) is solved by ISSM using revers AD? What linear/nonlinear systems need to be solved?

Some ice-sheet solvers compute sensitivities by solving an additional adjoint equation: BISICLES (Cornford at al. JCP, 2013), Ymir(Isaac et al, SISC, 2015), Albany/FELIX (perego et.al, JGR, 2014 and Tezaur et al. GMD, 2015). Albany/FELIX computes the partial derivatives of the constraint and functional w.r.t. the parameters and the independent variables using forward AD (Pawlowski et al, Scientific Programming, 2012). A different AD approach is used in Goldberg et al., GMD, 2016. How does the ISSM approach compares to these?

0.0.1 Detailed Review

Line 45 How can automatic differentiation be possible when a manual differentiation is not possible? AD is a way to perform differentiation, so if a model is not differentiable then AD would give NaN or Inf.. or similar.

Line 63 If the expression to be differentiated is implicit than the accuracy of the solution would depend on the implicit solvers and would mostly likely be larger than machine precision.

Line 73 Variable $\mathbf{y}$ has not been defined. I assume is $\mathbf{y} = \mathbf{f}(\mathbf{x})$.

Eq. 3 I think there is a typo here. Should it be

$$\bar{a} = \bar{a} + \frac{\partial \phi}{\partial a}\bar{r}; \quad \bar{b} = \bar{b} + \frac{\partial \phi}{\partial b}\bar{r} \quad ?$$

And why $r = 0$? Please check also expressions at line 81. I think this part should be expanded and the reverse AD method should be presented more clearly.

Line 82 The number of sweeps depends only on m but the computational cost depends on n as well (as mentioned at line 91).

Line 86 In this instance what does f denote? The misfit in the surface elevation? And x? Please describe better in the paper the parallel between the physical problem and the equation.

Line 201 How's $\bar{b}$ is defined? Please expand a bit this part to help the reader better understand how an implicit equation as $As = r$ is handled by reverse AD. Also, in the introduction r was used to denote an elemental operation (or its output), whether here it is used to denote the residual of the system, which is an input of the "elemental" operation $s = A^{-1}r$, I think this can be confusing. Let's say only a few elements of A are active (say, the part related to the boundary, if A comes from discretizing a PDE), then do you need to store the entire matrix $\bar{A}$ as well? What happens when the operator A is non linear (it depends on s)?

Line 324 Again, I don't think that the computational cost do not depend on n, even theoretically.

Figure 5-6 What is the specific application solved to produce results showed in Figures 4 and 5? Are these nonlinear problems? How many processors are you using when solving the probem with MUMPs? Please add also the overloading overhead in Figure 6, as done in Figure 5.