The CESM Land Ice Model (CISM): Documentation and User's Guide

William Lipscomb (Los Alamos National Laboratory)
William Sacks (National Center for Atmospheric Research)

November 2012

Table of Contents

1 Introduction 3
1.1 Scientific background 3
1.2 Ice sheets in CESM 4
1.3 Limitations 5
1.4 What's new in CESM1.1 with respect to ice sheet modeling? 6
1.5 Known problems in CESM1.1 7

2 Quick-start guide 8
2.1 Overview 8
2.2 Downloading the code 9
2.3 Creating a case 10
2.4 Setting up a case 11
2.5 Building a case 12
2.6 Running a case 13
2.7 Modifying namelist settings 16
2.8 Modifying source code 17
2.9 Branch and hybrid runs 17
2.10 Interpinic 19

3 Running the standalone ice sheet model within CESM: TG compsets 20
3.1 Background 20
3.2 Running with existing forcing data 21
3.3 Creating and running with your own forcing data 21
3.4 Changes to some CESM defaults for TG compsets 23

4 The dynamic ice sheet model 24
4.1 Equations of ice flow 24
4.2 Glimmer, the Community Ice Sheet Model 27
4.3 Directory structure 27
4.4 Coupling to CLM 28
4.5 Configuring and running the model 30

5 Ice sheets in the Community Land Model 33
5.1 CLM and the surface mass balance of ice sheets 33
5.2 Details of the surface-mass-balance and coupling schemes 35
5.3 Model controls 36
5.4 CLM surface datasets used for runs with CISM 38

6 Frequently Asked Questions 39
6.1 How do I run an ensemble of ice sheets, which differ in their parameters, initial conditions, and/or forcing data? 39
6.2 How do I run CLM with something other than 10 elevation classes? 42
6.3 How do I add new settings in CISM's namelist or config file? 42
6.4 How do I add a CLM history field that provides averages only over ice? 43

7 Acknowledgments 43

8 References 44

1 Introduction

This document accompanies the Community Earth System Model (CESM) User's Guide and is intended for users who would like to run CESM with dynamic ice sheets and/or an improved surface mass balance scheme for glaciated regions. For more information, please see the CESM User's Guide: http://www.cesm.ucar.edu/models/cesm1.1/cesm/.

The introduction provides some scientific background, along with a brief history of land-ice model development within CESM. Section 2 is a quick-start guide for new users. Section 3 describes how to run the standalone ice sheet model within CESM, forced by output from a previous, coupled run. Section 4 describes Glimmer, the Community Ice Sheet Model (Glimmer-CISM), the dynamic ice sheet model in CESM. Section 5 gives a detailed description of the surface-mass-balance scheme for ice sheets in the Community Land Model (CLM). Section 6 answers some common questions about model usage. Section 7 lists some anticipated model improvements.

It should be emphasized that this is an initial implementation with a number of scientific limitations that are detailed below. Model developers are keenly aware of these limitations and are actively addressing them. Several major improvements are planned for the next one to two years and will be released as they become available.

This documentation is itself in progress. If you find errors, or if you would like to have some additional information included, please contact the author at lipscomb@lanl.gov.

1.1 Scientific background

Historically, ice sheet models were not included in global climate models (GCMs), because they were thought to be too sluggish to respond to climate change on decade-to-century time scales. In the Community Climate System Model (CCSM), as in many other global climate models, the extent and elevation of the Greenland and Antarctic ice sheets were assumed to be fixed in time. Interactions between ice sheets and other parts of the climate system were largely ignored.

Recent observations, however, have established that the Greenland and Antarctic ice sheets can respond to atmospheric and ocean warming on time scales of a decade or less. Satellite gravity measurements show that both ice sheets are losing mass at a rate of more than 200 Gt/yr, roughly double the values from earlier this decade (Velicogna 2009). (A mass loss of 360 Gt corresponds to global sea-level rise of 1 mm.) Greenland mass loss is caused by increased surface melting and the acceleration of large outlet glaciers (van den Broeke et al. 2009). In Antarctica, mass is being lost primarily because of the acceleration of outlet glaciers, especially in the Amundsen Sea Embayment of West Antarctica (Rignot et al. 2008).

Small glaciers and ice caps (GIC) also have retreated in recent years. Although the total volume of GIC (~0.6 m sea-level equivalent; Radić and Hock 2010) is much less than that of the Greenland ice sheet (~7 m) and the Antarctic ice sheet (~60 m), glaciers and ice caps can respond quickly to climate change. Mass loss from GIC has grown during the past decade and is now about 400 Gt/yr (Meier et al. 2007). GCMs generally assume that the mass of glaciers and ice caps, like that of ice sheets, is fixed.

Global sea level is rising at a rate of about 30 cm/century, with primary contributions from land ice retreat and ocean thermal expansion. A recent study (Cazenave et al. 2008) suggests that land ice has accounted for up to 80% of recent sea-level rise. Estimates of 21st century ice-sheet mass loss and sea-level rise are highly uncertain. The IPCC Fourth Assessment Report (Meehl et al. 2007) projected 18 to 59 cm of sea-level rise by 2100 but specifically excluded ice-sheet dynamical feedbacks, in part because existing ice sheet models were deemed inadequate. A widely cited semi-empirical study (Rahmstorf 2007) estimated 40 to 150 cm of 21st century sea-level rise, based on the assumption that the rate of rise is linearly proportional to the increase in global mean temperatures from preindustrial values. This assumption may not be valid as additional land-ice processes come into play.

Modeling of land ice has therefore taken on increased urgency. Many recent workshops (e.g., Little et al. 2007; Lipscomb et al. 2009) have called for developing improved ice sheet models. There is general agreement on the need for (1) "higher-order" flow models with a unified treatment of vertical shear stresses and horizontal-plane stresses, (2) finer grid resolution (~5 km or less) for ice streams, outlet glaciers, and other regions where the flow varies rapidly on small scales, and (3) improved treatments of key physical processes such as basal sliding, subglacial water transport, iceberg calving, and grounding-line migration. These improvements are beginning to be incorporated in numerical ice sheet models. One such model is Glimmer, the Community Ice Sheet Model (Glimmer-CISM), which has been coupled to CESM and is described below.

Although much can be learned from ice sheet models in standalone mode, coupled models are required to capture important feedbacks. For example, surface ablation may be underestimated if an ice sheet model is forced by an atmospheric model that does not respond to changes in surface albedo and elevation (Pritchard et al. 2008). At ice sheet margins, floating ice shelves are closely coupled to the ocean in ways that are just beginning to be understood and modeled (Holland et al. 2008a, 2008b). Also, changes in ice sheet elevation and surface runoff could have significant effects on the regional and global circulation of the atmosphere and ocean. The inclusion of dynamic ice sheets in CESM and other GCMs will likely lead to scientific insights that cannot be obtained from standalone ice sheet models.

1.2 Ice sheets in CESM

Since 2006, researchers in the Climate, Ocean and Sea Ice Modeling (COSIM) group at Los Alamos National Laboratory (LANL) have worked with scientists at the National Center for Atmospheric Research (NCAR) to incorporate an ice sheet model in the CCSM/CESM framework. This work has been funded primarily by the DOE Scientific

Discovery through Advanced Computing (SciDAC) program, with additional support from NSF. The Glimmer ice sheet model (Rutt et al. 2009), developed by Tony Payne and colleagues at the University of Bristol, was chosen for coupling. Although Glimmer's dynamical core was relatively basic, a higher-order dynamics scheme was under development. In addition, the model was well structured and well documented, with an interface (GLINT) to enable coupling to GCMs.

Glimmer was initially coupled to CCSM version 3.5. The surface mass balance (SMB; the difference between annual accumulation and ablation) was computed using Glimmer's positive-degree-scheme, which uses semi-empirical formulas to relate surface temperatures to summer melting. It was decided that the PDD scheme was not appropriate for climate change modeling, because empirical relationships that are valid for present-day climate may not hold in the future. Instead, a surface-mass-balance scheme for ice sheets was developed for the Community Land Model (CLM). This scheme computes the SMB in each of ~10 elevation classes per grid cell in glaciated regions. The SMB is passed via the coupler to the ice sheet component, where it is averaged, downscaled, and used to force the dynamic ice sheet model at the upper surface. (See Section 4 for details.) When the CCSM4 coupling framework became available, the coupling was redone for the new framework.

In 2009, the U.K. researchers who designed Glimmer joined efforts with U.S. scientists who were developing a Community Ice Sheet Model (CISM), and the model was renamed Glimmer-CISM. Model development is overseen by a six-member steering committee including Magnus Hagdorn (U. Edinburgh), Jesse Johnson (U. Montana), William Lipscomb (LANL), Tony Payne (U. Bristol), Stephen Price (LANL), and Ian Rutt (U. Swansea). The model resides on the BerliOS repository (http://glimmer-cism.berlios.de/). It is an open-source code governed by the GNU General Public License and is freely available to all. The version included in the initial CESM release is a close approximation of Glimmer-CISM version 1.6.

1.3 Limitations

There are a number of significant limitations of the ice sheet model within CESM. Most of these are under active development by members of the Land Ice Working Group.

1.3.1 Limitations of the ice sheet model

- The model is technically supported but is still undergoing scientific testing and validation. We cannot guarantee that the default values of model parameters will yield an optimal simulation.

- The dynamical core is similar to that in the original Glimmer code and is based on the shallow-ice approximation (SIA). The SIA is valid in the interior of ice sheets, but not in fast-flowing regions such as ice shelves, ice streams, and outlet glaciers. A higher-order scheme that is valid in all parts of the ice sheet is being tested and will become part of CESM in 2013 with the release of Glimmer-CISM version 2.0.

- The current Glimmer-CISM code is serial. This is not a limitation for the SIA model, which is computationally fast, but will be an issue for the higher-order model. A parallel version of the code is under development and is expected to be available by 2013.

- Glimmer-CISM simulates only the large ice sheets (Greenland and Antarctica). There is currently no ability in CESM to simulate evolution of smaller glaciers. A separate model for simulating smaller glaciers is under development and is expected to be available in 2013.

1.3.2 Limitations of other components of the CESM modeling system

- Glimmer-CISM has been coupled to CLM, but the current coupling is one-way. That is, the surface mass balance computed by CLM is passed to Glimmer-CISM and used to drive ice sheet evolution, but the resulting ice sheet topography is not used to update the surface elevation or landunit types in CLM. Two-way coupling is under development and should be ready in 2013.

- Topography in the atmosphere model (CAM) does not respond to changes in ice sheet geometry. A scheme for dynamic topography is under development.

- The ice sheet model has not been coupled to the ocean model; that coupling is under development. For this reason the initial implementation is for the Greenland ice sheet only. Since ice-ocean coupling is critical for the dynamics of the Antarctic ice sheet, it was decided that Antarctic simulations without ocean coupling would be of limited scientific value.

- The division of glaciers into elevation classes in CLM is fairly coarse and static in space. The current scheme is reasonable for Greenland, but not ideal for mountain glaciers. An improved scheme is under development, and should be ready by early 2013.

- When downscaling to elevation class, the partitioning between rain and snow in CLM is simplistic. There are plans to make this partitioning dependent on surface temperature.

1.4 What's new in CESM1.1 with respect to ice sheet modeling?

Compared to the last public release (CESM1.0.4), there have been a number of improvements in CESM that are relevant for ice sheet modeling:

- A new compset type, *TG*, has been added. This allows running the standalone ice sheet model forced by output from a previous, coupled CESM simulation. We provide a variety of out-of-the-box forcing data, or you can generate your own forcing data. See Section 3 of this document for more details.

- Support for longer time steps in CISM and in CESM scripts – e.g., 1-year time step, useful for TG runs

- Changed default GLC grid to 5 km (previously was 20 km)

- Changed a number of other default CISM configuration settings to produce a more robust ice sheet evolution, especially at 5 km resolution

- Ensemble capability for all CESM components, including CISM (see Section 6.1 for details)

- More robust namelist generation facility, standardized across CESM components (see Section 2.7 for details)

- Enabled ESMF interface for CISM

- Fixed memory leak in CISM

- Bug fix for glacier virtual columns in CLM

- New high-resolution *pct_glacier* input file for CLM, based on the Randolph Glacier Inventory, and new CLM surface datasets based on this (see Section 5.4 for details)

- New diagnostic capabilities in CLM, including ability to output fields averaged only over the glacier portion of each grid cell (see Section 6.4 for details)

- New IG4804 compset

- Improved testing capability for TG compsets in the CESM test framework

1.5 Known problems in CESM1.1

The following are known problems in CESM1.1 that are particularly relevant for ice sheet modeling:

- CISM restarts can only happen on year boundaries

- CLM's interpinic tool does not work properly for input files with multiple glacier elevation classes

- BG1850CN at f09 resolution currently has initial conditions that are incompatible with the run configuration, for both CLM and CISM. Thus, this compset-resolution combination should only be used if you provide your own initial conditions for CLM and CISM.

- Geothermal heat flux doesn't work – the code dies if you use the [GTHF] section in cism.config (enabled by setting do_gthf to .true. in *user_nl_cism*)
- CLM's code for multiple elevation classes (and thus coupling to CISM) does not work correctly for GLC_NEC=1 (i.e., a single elevation class, but using the glc_mec code)
- There are a number of bugs with the use of a calendar that includes leap years; for now we recommend only using a no-leap calendar

2 Quick-start guide

This section provides a recipe for checking out code from the CESM repository, building and running a case, and modifying parameters related to land ice. It is assumed that the reader is already somewhat familiar with CESM and is working on a machine that supports CESM out of the box. For more details on building and running CESM, please see the CESM User's Guide, from which this section has been adapted. See also the CESM website, http://www.cesm.ucar.edu/models/cesm1.1/.

2.1 Overview

CESM consists of six physical models: atmosphere (atm), ocean (ocn), land (lnd), sea ice (ice), land ice (glc), and river runoff (rof). A central coupler coordinates the models and passes information between them. Each model can have active, data, dead, and stub components. An active component is a prognostic physical model. Data components provide scientifically valid input data for cases where an active model is not needed, whereas dead components generate invalid data that is used only for system testing. Stub components are present only to meet interface requirements when a component is not needed.

The active components of CESM are CAM (the Community Atmosphere Model), POP (the Parallel Ocean Program), CLM (the Community Land Model), CICE (the Community Ice CodE), CISM (the Community Ice Sheet Model), and RTM (the River Transport Model). Currently, CISM is synonymous with Glimmer-CISM, the dynamic ice sheet model described in Section 4. The ice-sheet surface mass balance is computed in CLM and passed to CISM via the coupler.

These components can be run in many different configurations. A particular mix of components, along with component-specific configuration and namelist settings, is called a component set, or "compset". Among the common compsets are B (fully coupled, with all active components), F (active CAM, CLM, and CICE, with a data ocean), and I (active CLM with data atmosphere and stub ice and ocean). These configurations all use a stub land-ice component. The corresponding configurations with an active land-ice component are denoted BG, FG, and IG. In addition, starting with CESM1.1, a TG compset allows running the ice sheet model in standalone mode, forced by output from a previous BG, FG or IG run. TG compsets are described in detail in Section 3.

The CESM components can be run on a variety of grids. The atmosphere and land are often run on the same grid, but CESM supports running them on different grids. The ocean and sea ice components must always run on the same grid. The ice-sheet component has a grid for each active ice sheet. Unlike the other component grids, which are global, the ice-sheet grids have limited domains. The current grids are polar stereographic projections with rectangular grid cells.

Three Greenland grids are supported, with resolutions of 20 km, 10 km, and 5 km, respectively. Antarctic grids at comparable resolutions will be supported in a future release. Each Greenland grid can be run with one of three land/atmosphere grids: T31 (a coarse-resolution spectral grid with dimensions 48 x 96), FV2 (a 1.9^{o}x2.5^{o} finite volume grid), and FV1 (a 0.9^{o}x1.25^{o} finite-volume grid). The land resolution is important for ice sheets because a higher-resolution land grid can be expected to provide a more realistic surface mass balance.

These definitions will be used in the rest of the section:

- $COMPSET is the component set.
- $RES is the grid resolution.
- $MACH is the machine name.
- $CCSMROOT is the CESM root directory.
- $CASE is the case name.
- $CASEROOT is the pathname of the directory where the case is created.
- $EXEROOT is the pathname of the executable directory, which is usually different from $CASEROOT.
- $RUNDIR is the directory where CESM is run
- $DOUT_S_ROOT is the root directory for local short-term archiving.

2.2 Downloading the code

CESM code is publicly available through a Subversion repository. You will need Subversion client software, version 1.4.2 or later. For more information on Subversion, see http://subversion.tigris.org.

To check out a release version of the code, you should register as a CESM user here:

http://www.cesm.ucar.edu/models/cesm1.0/register/register_cesm1.0.cgi

You will be sent a username and password you can use to access the repository.

Now log onto the machine where you plan to build and run CESM. For a list of available releases, type this command:

➢ svn list https://svn-ccsm-release.cgd.ucar.edu/model_versions

The first time you access the repository from a given machine, you will need to enter your username and password. This information should be cached automatically so that you do not need to enter it repeatedly.

To check out a particular release (say, model version cesm1_1), type this:

➢ svn co https://svn-ccsm-release.cgd.ucar.edu/model_versions/cesm1_1 cesm1_1

This command puts a copy of CESM in a directory called cesm1_1. This is the $CCSMROOT directory. If successful, you will see a message of the form "Checked out revision xx."

Other useful Subversion commands include *svn info* (for various information about the release version), *svn status* (for a list of files that have changed since checkout), and *svn diff* (to see differences between the release version and your working copy).

You may want to take a quick tour of the model. Type this:

➢ cd cesm1_1
➢ ls

You will see directories called *models*, *scripts* and *mapping*. The *scripts* directory contains useful scripts for creating new cases and tests. The *models* directory contains subdirectories called *atm, ocn, lnd, ice*, *glc*, and *rof,* corresponding to the physical models of CESM. There are also directories called *drv* (the driver), *utils* (various utilities), *csm_share* (code shared among different models), and *dead_share* (code shared among the dead models). The *mapping* directory contains utilities for creating the mapping files that are needed by the CESM coupler when you define a new model grid.

2.3 Creating a case

The next step is to create a case. From $CCSMROOT, go to the scripts directory:

➢ cd scripts

Here there is a script called *create_newcase*. The basic form of the *create_newcase* command is

```
➢ create_newcase  -case $CASEROOT  \
                  -mach $MACH  \
                  -compset $COMPSET  \
```

```
-res $RES
```

For more information about this command, type

➢ `create_newcase -h`

For a list of supported machines, compsets, and resolutions, type

➢ `create_newcase -l`

Suppose you are running on bluefire, an IBM machine at NCAR, and you want to create an IG case (active land and land-ice components, data atmosphere, stub ocean and sea ice) using the 1.9°x2.5° finite-volume grid for the land and atmosphere and a 1° grid for the ocean. You want to call this case "testIGfv2". You would type

➢ `create_newcase -case testIGfv2 -mach bluefire -compset IG -res f19_g16`

If successful, you will see a message like this:

```
Locking file /ptmp/lipscomb/cesm1_1/scripts/testIGfv2/env_case.xml
Successfully created the case for bluefire
```

It is possible to configure cases with many different combinations of compsets and grids, but not all these cases have been validated scientifically. For a list of validated configurations, see the README file in the *scripts* directory.

In $CCSMROOT/scripts there is now a subdirectory called *testIGfv2*. This is the $CASEROOT directory. Go to $CASEROOT:

➢ `cd testIGfv2`

This directory contains a number of scripts and xml files. Take a look at the file *env_case.xml*. This file specifies $CASE, $CASEROOT, $CCSMROOT, and other settings that cannot be changed once the case is created. For an explanation of these and other settings in the various xml files, see the online documentation for CESM (http://www.cesm.ucar.edu/models/cesm1.1/cesm/doc/modelnl/index.html).

2.4 Setting up a case

To configure the case, go to $CASEROOT and type

➢ `cesm_setup`

This command creates a machine-specific Macros file and run script in $CASEROOT. It also creates the files, *user_nl_*xxx (where *xxx* denotes the set of components targeted for this case); *these files are where all user component namelist modifications are now made.*

Once you setup the case, the file *env_mach_pes.xml* (which determines the processor layout) is locked. It cannot be changed unless you undo and redo the setup, as follows:

- ➢ cesm_setup -clean
- ➢ # Make changes to env_mach_pes.xml
- ➢ cesm_setup

Although xml files can be edited manually, it is safer to use the *xmlchange* command. For example, to change $NTASKS_ATM before setting up a case, you would type

- ➢ xmlchange -file env_mach_pes.xml -id NTASKS_ATM -val 128

Or simply:

- ➢ xmlchange NTASKS_ATM=128

Another setting relevant to runs with active land ice is $CLM_FORCE_COLDSTART. Currently, this is automatically set to "on" for compsets with an active land-ice component (e.g., IG, FG, and BG). This means that CLM will be initialized with arbitrary initial conditions rather than a restart file. If this option is set to "off", the model will fail at initialization because the current CLM restart files are not compatible with active land ice.

2.5 Building a case

The $CASEROOT directory should now contain a script called $CASE.build (*testIGfv2.build* in our case). To build the model, run this script:

- ➢ testIGfv2.build

The build script does the following:

- creates component namelists in $RUNDIR.
- checks for required data sets and (hopefully) downloads missing data automatically.
- creates the required utility and component libraries.
- creates the model executable in $EXEROOT.

If the build is successful, you will see this:

CCSM BUILDEXE SCRIPT HAS FINISHED SUCCESSFULLY

Otherwise, you will get a message that the build failed, referring you to a file in $RUNDIR for information about why it failed. Each component has its own build-log file; for land ice the filename is of the form *glc.bldlog.yymmdd-hhmmss.*

Look at the file *env_build.xml*, which contains settings such as $DEBUG and $EXEROOT. The $DEBUG option is false by default; set it to true to turn on compile-time and run-time debugging. When running on bluefire, $EXEROOT is /ptmp/$CCSMUSER/$CASE/bld by default. A related setting, $RUNDIR, specifies the location of the directory in which the model is run. $RUNDIR is set in *env_run.xml*, and on bluefire is /ptmp/$CCSMUSER/$CASE/run by default.

The *env_build.nml* file is locked after the code is built. To change settings after building, you must clean and redo the build:

- testIGfv2.clean_build
- # Make changes to env_build.xml
- testIGfv2.build

2.6 Running a case

Once the code has built successfully, it can be run by submitting the run script, $CASE.$MACH.run. On bluefire, which manages jobs using LSF, the first few lines of the run script look like this:

```
#BSUB -n 64
#BSUB -R "span[ptile=64]"
#BSUB -q regular
#BSUB -N
#BSUB -x
#BSUB -a poe
#BSUB -o poe.stdout.%J
#BSUB -e poe.stderr.%J
#BSUB -J testIGfv2
#BSUB -W 0:50
#BSUB -P 12345678
```

Here, -n specifies the number of processors that will be requested; -q is the queue; -o and -e give the names of standard output and error files that will be written to $CASEROOT; -W is the wall-clock time requested (50 minutes in the above example); and -P is the project code. The queue can be changed to raise or lower the run priority. The wall-clock time should be set to a value sufficient to complete the run (or at least to write restart files) without exceeding the machine limit. If a machine is busy, then reducing the time requested will usually allow the run to start earlier.

Next, look at *env_run.xml*. Unlike the other xml files, this file can be modified at any point during a sequence of runs. (As mentioned above, it is safest to use the *xmlchange* command to edit xml files.) Among the key settings are $RUN_TYPE, $RUN_STARTDATE, and (for cases with an active land-ice component) $GLC_GRID. The $RUN_TYPE is *startup* by default. Other options are *hybrid* and *branch*; these are explained in section 2.8. The default start date is compset-dependent, but is often simply 0001-01-01, which denotes January 1 of year 1. The default value of $GLC_GRID is *gland5UM*, which refers to the 5-km Greenland grid, using an updated dataset from the University of Montana. To run with a coarser grid, change this to *gland10* or *gland20 (or gland5 to use an alternative 5-km dataset)*.

Another important setting is $CONTINUE_RUN, which is false by default. This means that the run will start from the beginning (usually from the date 0001-01-01). If $CONTINUE_RUN is true, then the run will start from a restart file generated by a previous run. Restart files reside in $RUNDIR. $RUNDIR also contains several one-line pointer files (e.g, *rpointer.glc* for land ice) that specify the name of the current restart file for each component.

The next setting is $RESUBMIT, an integer (set to 0 by default) that specifies the number of times the run script will be resubmitted automatically after a run completes. This option is useful for longer production runs when you are sure the code is working. If $CONTINUE_RUN is initially false and $RESUBMIT > 0, then $CONTINUE_RUN is automatically set to true when the run script is resubmitted. If you do not use the $RESUBMIT option, $CONTINUE_RUN will remain false unless you change it manually.

To determine the length of the run, you set $STOP_OPTION and $STOP_N. By default, $STOP_OPTION is *ndays*, and $STOP_N = 5, which means the run will complete after 5 model days. Just after building a case, it is a good idea to do a 5-day trial run before trying a longer run. The allowed units of $STOP_OPTION are seconds, minutes, hours, days, months, and years. The value of $STOP_N can be any integer.

The settings $RESTART_OPTION and $RESTART_N determine how often restart files are written. By default, these are set to $STOP_OPTION and $STOP_N, respectively, but they can be modified to write restart files more often.

For runs with an active land-ice component, $HIST_OPTION and $HIST_N are also important. These generally specify the output frequency for coupler history files, but CISM's history frequency is currently hard-wired to be the same as the coupler history frequency. For compsets with an active land-ice component, these are set to give annual output by default.

The *env_run.xml* file also contains settings for archiving. If $DOUT_S is TRUE (the default), the model output will be written to $DOUT_S_ROOT for short-term archiving. On bluefire, $DOUT_S_ROOT is set to ptmp/$CCSMUSER/archive/$CASE by default.

Once you have modified the run script and *env_run.xml* as needed, you are ready to submit the run script. The following commands are for a machine that manages batch jobs using LSF (e.g., bluefire). The commands will be different for other load-sharing systems, such as PBS. To submit the run, type

- bsub < testIGfv2.bluefire.run

If successful, you will get a message that the job is submitted. To check the job status, type

➢ bjobs

You will see something like this:

```
 JOBID  USER    STAT  QUEUE   FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME
123456 myname PEND  regular  be1005en              testIGfv2    Dec 31   23:59
```

If the status is PEND, your job is waiting in the queue; if the status is RUN, then the job is running. If you type *bjobs* after the job has finished, you will see something like "No unfinished job found". In your $CASEROOT directory there will be two files with names of the form *poe.stderr.123456* and *poe.stdout.123456*. Look at the latter file. If all has gone well, you will see lines like this near the end of the file:

```
Sat Jan 01 01:23:45 MDT 2011 -- CSM EXECUTION HAS FINISHED
(seq_mct_drv): =========== SUCCESSFUL TERMINATION OF CPL7-CCSM ===============
```

If the job starts but fails to complete, the last line may look like this:

```
Sat Jan 01 00:12:34 MDT 2011 -- CSM EXECUTION BEGINS HERE
```

In this case you will have to figure out why the job failed before completion. A common cause of failure is not allowing enough time to finish the run. Otherwise, one of the components probably has failed during execution. You can investigate by looking at the various component log files in $RUNDIR. Log files are moved from $RUNDIR to the local archive only if the run completes successfully.

If you are running on a machine that uses PBS instead of LSF (e.g., titan, a Cray XT6 machine at Oak Ridge), then to submit the run script you would type

➢ qsub testIGfv2.bluefire.run

To check the job status, type

➢ qstat

When a run finishes successfully, you will find various log files (e.g., *glc.log.yymmdd-hhmmss* for the land-ice model) in $CASEROOT/logs. These log files are also written to the local archive, $DOUT_S_ROOT. The land-ice log files are in $DOUT_S_ROOT/glc, and similarly for the other models.

History files are also written to the archive, usually in netCDF format. The history files for land ice are in $DOUT_S_ROOT/glc/hist, and those for the land model are in $DOUT_S_ROOT/lnd/hist. These files contain gridded output written at regular intervals, usually once per model year for land ice and monthly for other active components. These files can be viewed and post-processed using a netCDF viewer such as ncview (for a quick look), Ferret, MATLAB, or IDL.

2.7 Modifying namelist settings

Once the code is running, you may want to change namelist or configuration variables. Variables related to land ice are set in the files *cism_in*, *cism.config* and *lnd_in*. These files appear in $RUNDIR, and also in $CASEROOT/CaseDocs. User modifications can be made to these files by adding lines to *user_nl_cism* (for variables in *cism_in* or *cism.config*) or *user_nl_clm* (for variables in *lnd_in*); this is described in more detail below. The various *user_nl_xxx* files are created when you first run *cesm_setup* for your case. They can be modified any time between running *cesm_setup* and the start of the run: the model does NOT need to be rebuilt after making namelist changes in these files.

Most parameters directly relevant to ice sheet modeling are set in *cism.config*. This config file contains settings used by Glimmer-CISM—for example, grid information (which is set automatically based on the value of GLC_GRID in *env_run.xml*), physics parameter settings, and the names and frequency of input and output files. See Section 4 and the Glimmer-CISM documentation for more information about these settings. The *cism_in* file contains some additional parameters controlling the CISM run.

The file *lnd_in* provides settings for CLM. An important variable related to land ice is *create_glacier_mec_landunit*, which is true by default for any case with active land ice. When this setting is true, CLM creates special glacier landunits with multiple elevation classes as described in Section 5. Another relevant setting is *glc_smb*, which also is true by default. If *glc_smb* is true, then CLM sends the surface mass balance to CISM via the coupler. If *glc_smb* is false, CLM instead sends input for Glimmer-CISM's positive-degree-day scheme. (The PDD option is not currently enabled in Glimmer-CISM, but will be supported in a future release.) Finally, *albice* sets the albedo of bare glacier ice in the visible and near IR bands of the spectrum.

Changes to both *cism.config* and *cism_in* can be made by adding lines with the following format to the *user_nl_cism* file in your case directory:

```
namelist_variable = value
```

Note that there is no distinction in *user_nl_cism* between variables that will appear in *cism_in* vs. those that will appear in *cism.config*: CISM's build-namelist utility knows where each variable belongs. For example, to set the value of *cism_debug* to .true. and *basal_tract* to the array (/1,2,3,4,5/), include the following in *user_nl_cism*:

```
cism_debug = .true.
basal_tract = 1 2 3 4 5
```

After running *preview_namelists*, the following will appear in *cism_in*:

```
&cism_params
  ...
  cism_debug = .true.
  ...
/
```

and the following will appear in *cism.config*:

```
[parameters]
basal_tract = 1 2 3 4 5
...
```

Changes to lnd_in can be made by adding similar lines to user_nl_clm. For example, to change the ice albedo (values give albedo in the visible and near-infrared), add the following line to *user_nl_clm*:

```
albice = 0.55,0.45
```

After changing any of the *user_nl_xxx* files, you can preview the generated namelists by running the *preview_namelists* utility in the case directory. Generated namelists will then appear in the $CASEROOT/CaseDocs, as well as in $RUNDIR.

All of these namelist parameters are documented online, on CESM's namelist documentation page.

2.8 Modifying source code

Advanced users may want to modify source code in the model directories. CISM source code is located in $CCSMROOT/models/glc/cism, and CLM source code is in $CCSMROOT/models/lnd/clm/src. Although it is possible to change the files in these directories, it is safer to copy files to the appropriate SourceMods directories and edit them there. Modified CISM files are placed in $CASEROOT/SourceMods/src.cism, and modified CLM files should go in $CASEROOT/SourceMods/src.clm. Once the modified files are in place, the code can be rebuilt. The files in the SourceMods directories automatically replace the files of the same name in the model directories.

2.9 Branch and hybrid runs

As mentioned above, there are three kinds of runs: startup, branch, and hybrid. In a startup run (the default), all components are initialized using baseline states that are determined independently by each component. The start date is determined by the setting $RUN_STARTDATE and is sent from the coupler to the components at initialization.

In a branch run, the components are initialized using a consistent set of restart files from an earlier run. Usually the case name of the branch ran is different from that of the previous run. Branch runs are often used for sensitivity studies. For example, suppose you want to study the effect of changes in the ice albedo. You could spin up the model for 100 years, then launch a series of branch runs, each starting from the same point but proceeding with a different albedo value. A branch run is bit-for-bit identical to a continuation of the original run provided that no source code or namelist values are changed. The start date for the branch run is determined by the restart files from the original run.

Suppose you want to set up a branch run on bluefire, starting from the beginning of year 2 of reference case *testIGfv2*. First go to $CCSMROOT/scripts and create a new case:

- create_newcase -case testIGfv2_br2 -mach bluefire -compset IG -res f19_g16
- cd testIGfv2_br2

Before configuring the case, modify *env_conf.xml* as follows:

- xmlchange -file env_conf.xml -id RUN_TYPE -val branch
- xmlchange -file env_conf.xml -id RUN_REFCASE -val testIGfv2
- xmlchange -file env_conf.xml -id RUN_REFDATE -val 0002-01-01

Then setup and build:

- cesm_setup
- testIFfv2_br2.build

Then modify these settings, which appear in *env_run.xml:*

- xmlchange RUN_TYPE=branch
- xmlchange RUN_REFCASE=testIGfv2
- xmlchange RUN_REFDATE=0002-01-01

Next, copy the restart files from the reference case into $RUNDIR for the branch case. If the reference restart files are in /ptmp/$USER/archive/testIGfv2/rest, you would go to the new $RUNDIR and type

- cp /ptmp/$USER/archive/testIGfv2/rest/0002-01-01-00000/* .

This command will copy the required restart and pointer files to $RUNDIR. You can then return to the branch $CASEROOT, edit *env_run.xml* as needed, and submit the run script.

A hybrid run is similar, except that the starting date (specified by $RUN_STARTDATE) can be changed relative to the reference case. Bit-for-bit reproducibility is generally not possible for hybrid runs, but the overall climate is continuous provided that no source code or namelist values are changed. For more details, see the CESM User’s Guide.

2.10 Interpinic

CAUTION: The interpinic tool currently does not work correctly if the input file (the file you are interpolating from) has glacier_mec landunits. However, it should work correctly for the first case described below.

When CESM is run in a G configuration (i.e., IG, FG, or BG), the land model is initialized with a surface data set different from the standard CLM surface data sets. Glacier landunits (with one column each) are replaced by glacier_mec landunits (with ~10 columns each). This means that a G simulation cannot be initialized directly with the output from a non-G run. Output from the non-G run must first be interpolated into the CLM data structure appropriate for a G run. This can be done using the CLM *interpinic* tool.

For example, suppose we have a CLM restart file called "*bcase.1deg.clm2.r.1000-01-01-00000.nc*". This file contains CLM output from year 1000 of a fully coupled (B) case. We want to use this output to initialize CLM when running a BG case (to save the expense of spinning up the BG case for ~1000 years from a cold start). We will use the B file to initialize all landunits other than glacier_mec. For glacier_mec landunits, we will use the output from an IG case, e.g., "*igcase.1deg.clm2.r.0100-01-01-00000.nc*". The B file will be the input file, and the IG file will be the output template file (since it has the required data structures, including glacier_mec landunits and columns, for initializing a BG run).

The interpinic tool is located here: *$CCSMROOT/models/lnd/clm/tools/interpinic*. The first step is to build the interpinic executable. Go to the *src* subdirectory and edit *interpinic.F90* as follows: search for the variable "*override_missing*" and change its value from "true" to "false". (By setting this variable to false, we stipulate that glacier_mec values, which are missing in the input B file, are *not* overwritten in the output template IG file.) Save the modified version, and build the executable:

- gmake interpinic

Copy the executable to the directory containing the input B file and the output template IG file. Save a copy of the output template file under a different name, because this file will be rewritten during the interpinic. Type the following:

- ./interpinic -i *input_filename* -o *output_filename*

Screen output will let you know how the interpinic process is going. If the process completes successfully, the template version of *output_filename* will be converted to a new file of the same name, containing a mixture of B data (for landunits other than glacier_mec) and IG data (for glacier_mec), and it will be in the correct format for initializing a BG run.

[Add a few words here about how to stage the run? E.g., rename the output file and copy it to the BG run directory.]

Note that the input file and output file do not have to reside on the same grid. If the grids are different, the input file values will be written to the nearest neighbor point containing an equivalent landunit in the output file.

Another use of interpinic: **CAUTION: The interpinic tool currently does not work correctly for this case:** Suppose that the CLM surface data set has changed, such that the number of glacier_mec landunits per gridcell is different. We want to use a restart file from a G run with the old data set to initialize a G run with the new data set. In this case the input file is the restart file from the old run, and the output template file is a restart file from a short (e.g., 5-day) new run. Go to the interpinic directory and make the executable as above, but do *not* change the value of *override_missing*. (In this case we are not trying to prevent any information in the output file from being overwritten.) Again, save a copy of the output file and type the following:

- ./interpinic –i *input_filename* –o *output_filename*

The new version of *output_filename* will consist of output from the old run, converted to the data structure required for initializing the new run.

3 Running the standalone ice sheet model within CESM: TG compsets

3.1 Background

The ice sheet model requires much less computational time than other CESM components, but often needs to be run for a much greater length of simulation time (e.g., tens of thousands of years rather than a century). Thus, it can be desirable to run CISM in standalone mode, forced by output from a previous CESM run. In this way, you can cycle many times through the forcing data from a previous run (e.g., to spin up the ice sheet model), or run parameter sensitivity analyses much faster than you could within the coupled system.

A run with standalone CISM in the CESM context is known as a TG compset. This compset uses the active ice sheet model forced by a data land model; all other components are stubs. Before running a TG compset, you must have coupler history files from a previous run that included CLM (see Section 3.2). Alternatively, you can run with existing forcing data (see Section 3.3).

3.2 Running with existing forcing data

There are currently four out-of-the-box TG compsets, using different periods of forcing data, as shown in the following table:

Compset short name	Compset long name	Forcing data
TG	T_PRESENT_DAY_GLC	From a BG20TRCN run (20th century transient, fully coupled), years 1976 – 2005
TG1850	T_1850_GLC	From a BG1850CN run (preindustrial spinup, fully coupled)
TG20TR	T_1850-2000_GLC	From a BG20TRCN run (20th century transient, fully coupled), years 1850 – 2005
TGRCP85	T_RCP8.5_GLC	From a BGRCP85 run (21st century, RCP 8.5 scenario, fully coupled), years 2006 – 2100

The fully-coupled runs that provided the forcing data for these compsets are described in a file in the CESM inputdata repository, alongside the forcing data (https://svn-ccsm-inputdata.cgd.ucar.edu/trunk/inputdata/lnd/dlnd7/CPLHIST_SNO/run_documentation.txt_c120808.txt).

To run with one of these existing sets of forcing data, simply create a case with one of the above compsets. **The resolution for this run should be f09_g16, since that is the resolution at which the forcing data were created.** (In theory, you can run using a different resolution, but that will involve a spatial interpolation of the forcing data.) You *can* run with a different GLC_GRID than the one used to create the forcing data (gland5UM).

3.3 Creating and running with your own forcing data

Currently, TG compsets are only able to handle forcing data from a previous CESM run. (Although, in theory, it should be possible to "fake" CESM output by creating files with the same format as existing TG forcing files.) Thus, performing a TG run with your own forcing data is a two-step process: (1) Perform a CESM run that includes an active land model (CLM), saving the necessary forcing files, and (2) perform a TG run using these new forcing data.

3.3.1 Performing a run to create forcing data

To create the necessary forcing data (surface mass balance and surface temperature), you need to perform a CESM run using a compset that includes an active land model (CLM) and an active ice sheet model (CISM). This can be done with IG, FG or BG compsets, or their variations.

In order to save the necessary forcing data, you need to set the driver namelist variable, histaux_s2x1yr. This can be set by adding the following line to user_nl_cpl in your case directory:

```
histaux_s2x1yr = .true.
```

This will cause the coupler to write out annual averages of the forcing fields sent from CLM to CISM. The files containing these averages will appear in the cpl/hist directory within your archive space, with names like:

```
$CASE.cpl.hs2x.0001-01-01.nc
```

A TG run that later uses these coupler history files as forcing should give *nearly* identical CISM results as the original run. Small differences arise because these forcing files are only written with single precision, leading to roundoff error on the order of 10^{-7}. If you need exact reproducibility, you can make a small source code modification prior to performing this run: In the routine *seq_hist_writeaux* in models/drv/driver/seq_hist_mod.F90, change the four instances of `use_float=.true.` to `use_float=.false.`

3.3.2 Performing a TG run using your own forcing data

To perform a standalone CISM run forced by your newly-created forcing data, first create a new case using one of the existing TG compsets (see above table). It can be easiest to use the compset whose forcing data most closely matches yours (e.g., if you are using forcing data from a future RCP simulation, then use TGRCP85), but the choice of TG compset does not matter as long as you take care in setting the necessary xml variables appropriately, as described below. The resolution of the TG run (as specified by the -res flag to create_newcase) should match the resolution of the run used to create the forcing data. You *can* run with a different GLC_GRID than the one used to create the forcing data.

The following variables in env_run.xml should be modified appropriately for your forcing data:

- `DLND_CPLHIST_DIR`: Directory in which your cpl.hs2x files can be found
- `DLND_CPLHIST_CASE`: Case name of the case used to create the cpl.hs2x files (files are assumed to be named `$DLND_CPLHIST_CASE.cpl.hs2x.yyyy-01-01.nc`)
- `DLND_CPLHIST_YR_START`: First year of forcing data (can be set later than the first existing year of data if you want to use a subset of the available years)
- `DLND_CPLHIST_YR_END`: Last year of forcing data (can be set earlier than the last existing year of data if you want to use a subset of the available years)

- RUN_STARTDATE: Determines the model year in which the run starts. This can be set to anything you want, but a good convention is:
  - For transient TG runs forced by output from a transient CESM run, set to the first year of forcing data (this corresponds to the real-world year, in some sense)
  - For non-transient TG runs (forced either by output from a non-transient run, or by cycling through the available forcing data multiple times), set to 0001-01-01 (in this case, there is no real-world meaning to the start year)
- DLND_CPLHIST_YR_ALIGN: The simulation year corresponding to DLND_CPLHIST_YR_START. This will usually be the same as the year in RUN_STARTDATE, but it can be set to a different year to start the simulation with a different year of forcing data.

The CESM scripts currently assume the forcing data are from a run that was done at f09_g16 resolution. If the resolution of your forcing data differs from this, you will need to make one additional change: After you have made all necessary modifications to the *DLND_** variables in *env_run.xml*, do the following from your case directory:

- preview_namelists
- cp CaseDocs/dlnd.streams.txt.sno.cplhist ./user_dlnd.streams.txt.sno.cplhist
- chmod u+w user_dlnd.streams.txt.sno.cplhist

Then change the domain file in *user_dlnd.streams.txt.sno.cplhist* to correspond to the domain file that was used for the run that created the forcing data. This file is listed in the *fileNames* subsection of the *domainInfo* section in that file, and is currently hard-coded to *domain.lnd.fv0.9x1.25_gx1v6.090309.nc*.

Important note: Once you have created a *user_dlnd.streams.txt.sno.cplhist* file, further changes to the *DLND_** variables in *env_run.xml* will not be picked up correctly by the scripts. Thus, if you need to change any of these variables, first remove the *user_dlnd.streams.txt.sno.cplhist* file, then make necessary modifications to these xml variables. Finally, repeat the above procedure for modifying the domain file.

3.4 Changes to some CESM defaults for TG compsets

TG compsets have much lower computational expense and much higher typical run lengths compared to most CESM configurations. Thus, a number of settings are changed automatically when running with a TG compset. These include:

- Default run length: 10 years (rather than 5 days)
- Default coupling frequency: annual (rather than daily or more frequent)
- Default PE layout: single processor

4 The dynamic ice sheet model

This section gives a brief overview of ice flow modeling and of Glimmer-CISM, the dynamic ice sheet model in CESM. For more details, including a technical description of the model, please see Rutt et al. (2009) and the Glimmer-CISM documentation. The documentation is slightly out of date but still provides a useful description of Glimmer-CISM's dynamical core (GLIDE) and climate model interface (GLINT). Updated documentation will be provided with the release of Glimmer-CISM 2.0 later in 2012.

4.1 *Equations of ice flow*

Here we give a short overview of the equations of ice flow. For more details, see, e.g., Greve and Blatter (2009) and the Glimmer-CISM documentation.

An ice sheet is typically modeled as an incompressible, heat-conducting, viscous, non-Newtonian fluid. The basic field equations can be written as (e.g., Pattyn 2003)

$$\nabla \bullet \mathbf{u} = 0\,, \tag{1}$$

$$\rho \frac{d\mathbf{u}}{dt} = \nabla \bullet \sigma + \rho \mathbf{g}\,, \tag{2}$$

$$\rho \frac{d(c_p T)}{dt} = \nabla(k \nabla T) + \Phi\,, \tag{3}$$

where d/dt is the material derivative, ρ is the ice density, $\mathbf{u}$ is the 3D velocity, T is the temperature in degrees Celsius, $\mathbf{g}$ is the gravitational acceleration, σ is the stress tensor, c_p is the specific heat of ice, k is the thermal conductivity, and Φ is the deformational heat source. These three equations express conservation of mass, linear momentum, and energy, respectively. The continuity equation (1) implies that glacier ice is incompressible.

In Cartesian coordinates (x, y, z) with $\mathbf{g} = (0, 0, -g)$, the continuity equation becomes

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0\,, \tag{4}$$

where $\mathbf{u} = (u, v, w)$. The momentum equations are

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} = 0\,, \tag{5}$$

$$\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} = 0, \tag{6}$$

$$\frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} = \rho g, \tag{7}$$

where σ_{ij} is the force per unit area in the j direction on the plane normal to the i direction, and acceleration terms (which are small for ice sheets) have been neglected. Equations (5)–(7) are known as the full-Stokes equations. Since the stress tensor is symmetric (ensuring conservation of angular momentum), only six of the nine components are independent. It is convenient to write σ as

$$\sigma_{ij} = \tau_{ij} - p\delta_{ij}, \tag{8}$$

where τ_{ij} is the stress deviator tensor, δ_{ij} is the Kronecker delta, and p is the static pressure, defined as

$$p = -\frac{1}{3}(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}). \tag{9}$$

This definition implies that $p > 0$, since the three normal stresses are negative for an ice sheet at rest. The stress deviator tensor is traceless: $\tau_{xx} + \tau_{yy} + \tau_{zz} = 0$.

The components of τ are related to the strain rate by means of a constitutive law. The standard constitutive law is Glen's flow law:

$$\dot{\varepsilon}_{ij} = A(T)\tau_e^{n-1}\tau_{ij}, \tag{10}$$

where the strain rate tensor $\dot{\varepsilon}_{ij}$ is the symmetric part of the tensor $\nabla \mathbf{u}$:

$$\dot{\varepsilon}_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{11}$$

Equation (11) implies that $\dot{\varepsilon}_{ij}$, like τ_{ij}, is traceless. The effective stress τ_e is a function of the second invariant of the stress deviator tensor and may be written as

$$\tau_e^2 = \frac{1}{2}\tau_{ij}\tau_{ij} = \frac{1}{2}(\tau_{xx}^2 + \tau_{yy}^2 + \tau_{zz}^2) + \tau_{xy}^2 + \tau_{yz}^2 + \tau_{xz}^2. \tag{12}$$

The exponent of (10) is usually chosen as $n = 3$. The rate factor $A(T)$ is typically computed using an Arrhenius relation (Payne et al. 2000):

$$A(T^*) = a\exp\left(-\frac{Q}{RT^*}\right), \tag{13}$$

where a is a proportionality constant, Q is the activation energy for creep, R is the universal gas constant, and T^* is the absolute temperature corrected for the dependence of the melting point on pressure. Often it is desirable to express the deviatoric stress in terms of the strain rate. Using the relation $\dot{\varepsilon}_e = A\tau_e^n$, equation (10) can be inverted to give

$$\tau_{ij} = B(T)\varepsilon_e^{(1/n-1)}\dot{\varepsilon}_{ij}\,, \tag{14}$$

where $B = A^{-1/n}$. This expression is of the standard form for a viscous fluid,

$$\tau_{ij} = 2\mu\dot{\varepsilon}_{ij}\,, \tag{15}$$

where $\mu = \frac{1}{2}B(T)\dot{\varepsilon}_e^{(1/n-1)}$ is the effective viscosity.

Using (8)–(14), the full-Stokes equations (5)–(7) together with the continuity equation (4) can be written as a system of four coupled equations with four unknowns: *u*, *v*, *w*, and *p*. Since these equations are hard to solve, most numerical ice sheet models solve the momentum equation in approximate form. For example, Pattyn (2003) neglects the first two terms on the LHS of (7) and uses a hydrostatic approximation,

$$\frac{\partial \sigma_{zz}}{\partial z} \cong \rho g\,, \tag{16}$$

to eliminate σ_{zz}. After some algebraic manipulation, the resulting momentum equations are

$$\frac{\partial}{\partial x}(2\tau_{xx} + \tau_{yy}) + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} = \rho g \frac{\partial s}{\partial x}\,, \tag{17}$$

$$\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial}{\partial y}(\tau_{xx} + 2\tau_{yy}) + \frac{\partial \tau_{yz}}{\partial z} = \rho g \frac{\partial s}{\partial y}\,, \tag{18}$$

where s is the surface elevation. This is a set of two coupled equations in two unknowns, *u* and *v*, which are easier to solve than the full-Stokes equations. Once *u* and *v* are determined, *w* and *p* are found using the continuity equation (given that $w = 0$ at the lower boundary) and the hydrostatic relation.

Equations (17)–(18) are often referred to as a "higher-order" approximation of the full-Stokes equations. Other higher-order approximations exist; for example, Schoof and Hindmarsh (2010) used an additional simplification to obtain a vertically averaged higher-order model. In this model, *u* and *v* are solved in a single layer (rather than three dimensions as in the Pattyn model), and the velocities at other elevations are found by vertical integration.

Two lower-order approximations are widely used. The most common is the shallow-ice approximation (SIA), in which vertical shear stresses are assumed to be dominant, and lateral and longitudinal stresses are neglected. In other words, the third term on the LHS of (17) and (18) is assumed to be much larger than the first two terms. The SIA is valid in the slow-moving interior of ice sheets, where basal sliding is small and the motion is dominated by vertical shear. Another common approximation is the shallow-shelf approximation (SSA), in which lateral and longitudinal stresses are assumed to dominate, and vertical shear stresses are ignored. The SSA is valid for floating ice shelves, where the basal shear stress is negligible and there is little or no vertical shear. The SSA is sometimes used in modified form to model flow in regions of rapid sliding, such as ice streams, where the basal shear stress is small but nonzero (e.g., MacAyeal 1989).

4.2 Glimmer, the Community Ice Sheet Model

Glimmer-CISM is a thermomechanical ice sheet model that solves the equations of ice flow, given suitable approximations and boundary conditions. The source code is written primarily in Fortran 90 and 95. The model resides on the BerliOS repository (http://glimmer-cism.berlios.de/), where it is under active development. Glimmer-CISM is an open-source code governed by the GNU General Public License and is freely available to all.

The initial release of CESM contains source code from Glimmer-CISM version 1.6. The main differences from version 1.0 are that (1) the directory structure has been reorganized, and (2) the GLINT climate model interface has been significantly changed to support coupling to CESM and other GCMs.

The dynamical core of the model, known as GLIDE, solves equations (1)–(3) for the conservation of mass, momentum, and internal energy. The version of GLIDE currently in CESM uses the shallow-ice approximation. However, a higher-order model is under development and will be included in future releases.

The surface boundary conditions (e.g., the surface temperature and surface mass balance) are supplied by a climate driver. When Glimmer-CISM is run in CESM, the climate driver is GLINT, which receives the temperature and SMB from the coupler and downscales them to the ice-sheet grid. The lower boundary conditions are given by an isostasy model, which computes the elevation of the lower surface, and by a geothermal model, which supplies heat fluxes at the lower boundary.

The model currently has simple treatments of basal hydrology and sliding. More complex schemes for subglacial water hydrology and evolution of basal till strength are being developed. Glimmer-CISM also provides several simple schemes for calving at the margins; these will be replaced by more realistic lateral boundary conditions in the future.

For a detailed description of Glimmer-CISM's dynamical core and software design, please see Rutt et al. (2009) and the latest model documentation.

4.3 Directory structure

In the CESM directory structure, each model component sits under a directory with a three-letter acronym: e.g., *atm* for the atmosphere model, *lnd* for the land surface, *ocn* for the ocean, and *ice* for sea ice. The ice sheet component resides in a directory called *glc*. Within the *glc* directory are three subdirectories: *sglc* (a stub model), *xglc* (a "dead" model), and *cism* (the physical model).

Inside the *cism* subdirectory are several more subdirectories:

- *source_glimmer_cism*, which contains source code from Glimmer-CISM. Most modules begin with the prefix "glide" (for GLIDE modules), "glint" (for GLINT modules), or "glimmer" (for general-purpose modules).
- *source_glc*, which contains wrapper modules that link Glimmer-CISM to the CESM coupler.
- *source_slap,* which has source code for the SLAP (Sparse Linear Algebra Package) solvers used by Glimmer-CISM to solve implicit equations.
- *drivers*, which contains two versions of the *glc* driver: one for use in the MCT coupling framework and the other for the ESMF framework.
- *bld,* which contains files required to build the code and create namelist files. The sub-directory *namelist_files* contains xml files that describe all possible namelist / configuration settings, and their default values (see Section 2.7 for the preferred way to modify these settings).
- *tools*, which contains tools for generating land/ice-sheet grid overlap files.
- *test*, which contains code to test parts of the source code
- *doc,* which contains model documentation.
- *mpi* and *serial*, which have appropriate versions of source code that can be used for parallel and serial runs, respectively. The *serial* directory is obsolete; now the *mpi* directory is used even when running on a single processor.

The files of most interest to users are in the *source_glimmer_cism* and *source_glc* directories. The safest way to change source code in these directories is to copy the file to the *SourceMods/src.cism* subdirectory within the case directory and edit the file there. When the code is built, the contents of *src.cism* will automatically overwrite any files of the same name in the model source code directories.

4.4 Coupling to CLM

GLINT, the climate model interface of Glimmer-CISM, is designed to accumulate, average, and downscale fields received from other climate model components. These fields are interpolated from a global grid to the individual ice sheet grid(s). In general there can be multiple non-overlapping ice sheet grids, but only Greenland is currently enabled. The global grid must be a regular lat-lon grid, but the latitudes need not be equally spaced. For CESM the global grid is assumed to be the same as the CLM grid.

GLINT needs to know (1) one or more 2D fields necessary for computing the surface mass balance, (2) an upper boundary condition, usually surface temperature, and (3) the latitudes and longitudes of the grid cells where these fields are defined. There are two general ways of computing the surface mass balance:

1. a positive-degree-day (PDD) scheme, either annual or daily, for which the required inputs to GLINT are the 2-m air temperature and the precipitation. This is the default scheme for Glimmer-CISM, but it may not be appropriate for climate change studies. The PDD option is not currently enabled for CESM runs, but will soon be added as an option for comparison with the surface-mass-balance scheme.

2. a surface-mass-balance (SMB) scheme for land ice embedded in CLM. In this case the required input to GLINT is the SMB itself. This is the preferred approach for climate-change experiments. The mass balance is computed for a specified number of elevation classes for each grid cell on the coarser land grid (~100 km). This is much less computationally expensive than computing the SMB for each cell on the finer ice sheet grid (~10 km). Values of 1, 3, 5, and 10 elevation classes are currently supported, with 10 being the default.

For the SMB scheme, the fields passed to GLINT are (1) the surface mass balance, *qsmb* (kg/m^2/s, positive for ice growing, negative for ice melting), (2) the surface temperature, *Tsfc* (deg C), and (3) the surface elevation, *topo* (m) for each elevation class. These fields are received from the coupler once per simulation day, accumulated and averaged over the course of a mass balance accumulation time step (typically one year) and then downscaled to the ice sheet grid. The downscaling occurs in two phases. First, the values on the global grid are interpolated in the horizontal to the local ice sheet grid. Next, for each local grid cell, values are linearly interpolated between adjacent elevation classes. For example, suppose that at a given location the coupler supplies a surface mass balance at elevations of 300 and 500 m, whereas the local grid cell has an elevation of 400 m. Then the local SMB is equal to the average of the values at 300 and 500 m.

In some parts of the ice sheet grid the fields supplied by CLM are not valid, simply because there are no land-covered global grid cells in the vicinity. For this reason, GLINT computes a mask on the global grid at initialization. The mask has a value of 1 for global grid cells that have a nonzero land fraction (and hence supply valid data) and is zero otherwise. GLINT then computes a local mask for each grid cell on the ice sheet grid. The local mask has a value of 1 if one or more of the four nearest global neighbors supplies valid data (i.e., has a global mask value of 1). Otherwise, the local mask has a value of zero. In this case ice sheets are not allowed to exist, and in output files, the SMB and temperature fields are given arbitrary values, typically zero. This masking has not been a restriction in practice, since the Greenland ice sheet does not extend far from the land margin. Alternatives may need to be considered for modeling the Antarctic ice sheet.

After downscaling the surface mass balance to the ice sheet grid, GLINT calls the ice sheet dynamics model, which returns a new profile of ice sheet area and extent. The following fields can be upscaled to the global grid and returned from GLINT to the coupler: (1) the ice area fraction, *gfrac*, (2) the ice sheet elevation, *gtopo* (m), (3), the frozen portion of the freshwater runoff, *grofi*, (4) the liquid portion of the runoff, *grofl*, and (5) the heat flux from the ice sheet interior to the surface, *ghflx*. These fields are computed for each elevation class of each grid cell. The frozen runoff corresponds to iceberg calving and the liquid runoff to basal meltwater. Surface runoff is not supplied by GLINT because it has already been computed in CLM. Upscaling is not enabled in the current release but will be included in the near future.

There are two modes of coupling Glimmer-CISM to CLM: one-way and two-way. For one-way coupling, Glimmer-CISM receives the surface mass balance from CLM via the coupler, and the ice sheet extent and thickness evolve accordingly. However, the land surface topography is fixed, and the fields received by CLM from the ice sheet model are ignored. In this case CLM computes surface runoff as in earlier versions of CCSM; excess snow is assumed to run off, and melted ice stays put at the surface. (See Section 4 for more details.) For two-way coupling, the CLM surface topography is modified based on input from the ice sheet model. In this case, surface runoff is computed in a more realistic way; excess snow remains in place and is converted to ice, and melted ice runs off. In either case, CLM computes the surface runoff, which is directed toward the ocean by the river routing scheme. Only one-way coupling is currently enabled, but two-way coupling is under development and will be added in 2013.

4.5 Configuring and running the model

Timesteps: There are several kinds of timesteps in Glimmer-CISM.

1. The *forcing timestep* is the interval in hours between calls to GLINT. Currently, the forcing timestep is the same as the *coupling interval* at which information is passed from the coupler to GLC. The forcing timestep is determined by the CISM namelist variables *dt_option* and *dt_count*. It is 24 hours by default for most compsets, but 1 year for TG compsets. Note that these are the only values that have been tested extensively; results should be checked carefully if the forcing timestep is changed from these defaults.
2. The *mass balance timestep* is the interval over which accumulation/ablation forcing data is summed and averaged. This timestep is set in subroutine *glint_mbal_init* in module *glint_mbal.F90*. The current default is one year. With the default settings of the forcing timestep and mass balance timestep, GLINT will accumulate forcing data from the coupler over 365 daily forcing timesteps and average the data before downscaling it to the local ice sheet grid. The mass balance timestep must be an integer multiple of the forcing timestep.
3. The *ice sheet timestep* is the interval in years between calls to the dynamic ice sheet model, GLIDE. The ice sheet timestep should divide evenly into the mass balance timestep. The current default is 0.05 year for 5-km, and 0.1 year for 10-km and 20-km.

Two optional runtime parameters can be used to make the time-stepping more intricate:

1. The mass balance accumulation time, *mbal_accum_time* (in years), is the period over which mass balance information is accumulated before calling GLIDE. By default, the mass balance accumulation time is equal to either the ice sheet timestep or the mass balance timestep, whichever is larger (for current defaults, this means that *mbal_accum_time* is set equal to the mass balance timestep: 1 year). But suppose, for example, that the ice sheet timestep is 5 years. If we set mbal_accum_time = 1.0, we accumulate mass balance information for 1 year and use this mass balance to force the ice sheet model (thus avoiding 4 additional years of accumulating mass balance data). **Note that this parameter cannot currently be modified via *user_nl_cism*, because it is not recommended that users change it.**
2. The timestep multiplier*, ice_tstep_multiply*, is equal to the number of ice sheet timesteps executed for each accumulated mass balance field. Suppose that the mass balance timestep is 1 year, the ice sheet timestep is 1 year, and ice_tstep_multiply = 10. GLINT will accumulate and average mass balance information for 1 year, then execute 10 ice sheet model timesteps of 1 year each. In other words, the ice sheet dynamics is accelerated relative to the land and atmosphere. This option may be useful in CESM for multi-millennial ice-sheet simulations where it is impractical to run the atmosphere and ocean models for more than a few centuries.

These time options (apart from the forcing timestep) are set in *cism.config*. This file contains (or may contain) the following timestep information:

1. The ice sheet timestep *dt* (in years) is set in the section [*time*] in the ice config file.
2. The mass balance time step is not set directly in the config file, but is related to the accumulation/ablation mode, *acabmode*, which is set in the section [*GLINT climate*]. If *acabmode = 0* (the default value for CESM runs), then the mass balance time step is set to the number of hours in a year (i.e., 8760 hours for a 365-day year).
3. The values of *ice_tstep_multiply* and *mbal_accum_time*, if present, are listed in the section [*GLINT climate*].

See the Glimmer-CISM documentation for more details.

Note that the total length of the simulation is not determined by Glimmer-CISM, but is set in the file *env_run.xml* in the case directory.

Input/output: All model I/O is in netCDF format. The *cism.config* file controls input. Near the end of this file, there is a section labeled [*CF input*]. This section contains the name of the ice sheet grid file used for initialization. This file typically includes the ice thickness and surface elevation, or equivalent information, in each grid cell. Other information (e.g., internal ice temperature) may be included; if not, then these fields are set internally by Glimmer-CISM.

Model history frequency is controlled by *HIST_OPTION* and *HIST_N* in *env_run.xml*; by default, history files are written once a year. Among the standard fields written to the history file are the ice thickness (*thk*), upper surface elevation (*usurf*), temperature (*temp*), and velocity (*uvel*, *vvel*) fields, along with the surface mass balance (*acab*) and surface air temperature (*artm*) downscaled to the ice sheet grid; these fields are set by the variable *cesm_history_vars* in *cism_in*.

Model restart frequency is coordinated by the CESM coupler. The restart or hotstart file contains all the fields required for exact restart. However, the restart will be exact only if the file is written immediately after an ice dynamics time step. This will normally be the case for restart files written at the end of any model year.

Many other fields can be written out if desired, simply by adding them to the variable list, *cesm_history_vars*. The source files with names "*_io.F90" specify the fields than can be written out. The easiest way to write out new variables is to add them to a file ending in "vars.def" and then rebuild the "*_io.F90" files using a python script. The necessary script can be found in $CASEROOT/Buildconf/cismIOconf. See the README.cismIO file in that directory for details. However, please note that this script has not been tested extensively in the CESM context.

Grids: GLINT can downscale fields from any global lat/lon grid. The latitudes need not be equally spaced. Three global grid resolutions are currently supported: T31 (spectral), FV2 (~2^o finite-volume), and FV1 (~1^o finite-volume). The global resolution (i.e., the resolution of the land and atmosphere) is set when a case is created.

Local ice sheet grids must be rectangular; typically they are polar stereographic projections. For Greenland, three grids are currently supported, with resolutions of 20 km, 10 km, and 5 km, respectively. Each local grid is compatible with any of the three supported global resolutions. There are two versions of the 5 km grid – *gland5* and *gland5UM* – which provide different data for initializing the model. The current default is the *gland5UM* 5 km grid. This can be changed by modifying GLC_GRID to the desired value (gland5, gland10 or gland20) in env_run.xml. A number of configuration defaults depend on the grid. You can see the effect of changing *GLC_GRID* by running *preview_namelists* before and after making this change. You can also see the rules that determine default values in *models/glc/cism/bld/namelist_files/namelist_defaults_cism.xml*.

Simulating the Greenland Ice Sheet: A primary motivation for having a CESM ice sheet model is to do climate change experiments with a dynamic Greenland Ice Sheet (GrIS). The first step is to simulate a present-day (or preindustrial) ice sheet that is in steady-state with the CESM climate and is not too different in thickness, extent, and velocity from the real GrIS. If we cannot do this, then either we will start climate change simulations with an unrealistic GrIS, or we will start with a realistic GrIS that is far from steady state, making it difficult to distinguish the climate-change signal from model transients.

It may be challenging to generate a realistic ice sheet, for several reasons: (1) The surface mass balance computed in CESM could be inaccurate; (2) Glimmer-CISM currently uses the shallow-ice approximation, which is not accurate for fast outlet glaciers; and (3) the present-day GrIS may not be in steady-state with the present-day (or preindustrial) climate. Our working hypotheses are that (1) If the SMB is reasonably accurate, we can obtain a reasonable large-scale thickness and extent for the GIS; (2) With a higher-order dynamics scheme and some judicious tuning, we can generate ice streams and outlet glaciers in the right locations with realistic velocities; and (3) The present-day GrIS is not far from steady-state with the preindustrial climate. These hypotheses are now being tested; results will be reported in an upcoming special issue of the *Journal of Climate*.

Obtaining an accurate surface mass balance may require some tuning in CLM; see Section 4 for details. We are also experimenting with different dynamics settings in the ice config file. The current default settings may not be optimal. The config files will be updated when we have more experience in running the model.

5 Ice sheets in the Community Land Model

This section describes changes made in the Community Land Model to accommodate ice sheets. For more information, see the CLM4 documentation.

5.1 CLM and the surface mass balance of ice sheets

The surface mass balance of a glacier or ice sheet is the net annual accumulation/ablation of mass at the upper surface. Ablation is defined as the mass of water that runs off to the ocean. Not all the surface meltwater runs off; some of the melt percolates into the snow and refreezes. Accumulation is primarily by snowfall and deposition, and ablation is primarily by melting and evaporation/sublimation.

Two kinds of surface mass balance schemes are widely used in ice sheet models:

- positive-degree-day (PDD) schemes, in which the melting is parameterized as a linear function of the number of degree-days above the freezing temperature. The proportionality factor is empirical and is larger for bare ice than for snow.
- surface-energy-balance (SEB) schemes, in which the melting depends on the sum of the radiative, turbulent, and conductive fluxes reaching the surface.

The current version of Glimmer-CISM has only a PDD scheme. It is generally believed that PDD schemes are not appropriate for climate change studies, because empirical degree-day factors could change in a warming climate. Comparisons of PDD and energy-balance schemes (e.g., van de Wal 1996; Bougamont et al. 2007) suggest that PDD schemes may be overly sensitive to warming temperatures. Bougamont et al. (2007) found that a PDD scheme generates runoff rates nearly twice as large as those computed by an SEB scheme.

In CESM, the ice-sheet surface mass balance is computed using an SEB scheme in CLM. Before discussing the scheme, it is useful to describe CLM's hierarchical data structure. Each grid cell is divided into one or more landunits; landunits can be further divided into columns; and columns can be subdivided into plant functional types, or PFTs. Each column within a landunit is characterized by a distinct snow/soil or snow/ice temperature and water profile. PFTs within a column have the same vertical snow/soil profiles but can have different surface fluxes. In the current version, landunit areas in each grid cell are fixed at initialization, but PFT and column areas can evolve during the simulation.

Previously, CLM supported up to five landunits per grid cell: soil, urban, wetland, lake, and glacier. Each of these landunits generally contains a single column, and soil columns (but not urban, wetland, lake, and glacier columns) consist of multiple PFTs. CLM now supports a sixth landunit, *glacier_mec*, where "mec" denotes "multiple elevation classes". Each glacier_mec landunit is divided into a user-defined set of columns based on surface elevation. The default is 10 elevation classes whose lower boundaries are 0, 200, 400, 700, 1000, 1300, 1600, 2000, 2500, and 3000 m. Each column is characterized by a fractional area and surface elevation that are read in during model initialization. The fractional area and elevation of each column are allowed to evolve during the run. Each glacier_mec column within a grid cell has distinct ice and snow temperatures, snow water content, surface fluxes, and surface mass balance.

These elevation classes provide a mechanism for downscaling the surface mass balance from the relatively coarse (~100 km) land grid to the finer (~5 km) ice sheet grid. The SMB is computed for each elevation class in each grid cell and is accumulated, averaged, and passed to the GLC (dynamic ice-sheet) component via the coupler once per day. The mass balance is downscaled by GLINT to the ice-sheet grid as described in Section 3.

There are several reasons for computing the surface mass balance in CLM rather than in Glimmer-CISM:

1. It is much cheaper to compute the SMB in CLM for ~10 elevation classes than in Glimmer-CISM. For example, suppose we are running CLM at a resolution of ~50 km and Glimmer at ~5 km. Greenland has dimensions of about 1000 x 2000 km. For CLM we would have 20 x 40 x 10 = 8,000 columns, whereas for GLIMMER we would have 200 x 400 = 80,000 columns.
2. We can use the sophisticated snow physics parameterization already in CLM instead of implementing a separate scheme for Glimmer-CISM. Any improvements to the CLM will be applied to ice sheets automatically.
3. The atmosphere model can respond during runtime to ice-sheet surface changes. As shown by Pritchard et al. (2008), runtime albedo feedback from the ice sheet is critical for simulating ice-sheet retreat on paleoclimate time scales. Without this feedback the atmosphere warms much less, and the retreat is delayed.
4. Mass is conserved, in that the rate of surface ice growth or melting computed in CLM is equal to the rate seen by the dynamic ice sheet model.

5. The improved surface mass balance is available in CLM for all glaciated grid cells (e.g., in the Alps, Rockies, Andes, and Himalayas), not just those which are part of ice sheets.

5.2 Details of the surface-mass-balance and coupling schemes

When the model is initialized, CLM reads a high-resolution data file classifying each point as soil, urban, lake, wetland, glacier, or glacier_mec. For runs with dynamic ice sheets, the default is to classify all glaciated regions as glacier_mec. If there are no dynamic ice sheets, then these regions are normally classified as glacier landunits with a single column per landunit. Glacier_mec columns, like glacier columns, are initialized with a temperature of 250 K. While glacier columns are initialized with a snow liquid water equivalent (LWE) equal to the maximum allowed value of 1 m, glacier_mec columns begin with a snow LWE of 0.5 m so that they will reach their equilibrium mean snow depth sooner.

Surface fluxes and the vertical temperature profile are computed independently for each glacier_mec column. Each column consists of 15 ice layers and up to 5 snow layers, depending on snow thickness. As for other landunits with a snow cover, surface albedos are computed based on snow fraction, snow depth, snow age, and solar zenith angle. By default, the bare ice albedo is prescribed to be 0.60 for visible radiation and 0.40 for near IR; this is lower than the values assumed by CLM for glacier landunits (0.80 for visible radiation and 0.55 for near IR). The latter values are higher than those usually assumed by glaciologists.

The atmospheric surface temperature, potential temperature, specific humidity, density, and pressure are downscaled from the mean gridcell elevation to the glacier_mec column elevation using a specified lapse rate (typically 6.0 deg/km) and an assumption of uniform relative humidity. At a given time, lower-elevation columns can undergo surface melting while columns at higher elevations remain frozen. This gives a more accurate simulation of summer melting, which is a highly nonlinear function of air temperature. The precipitation rate and radiative fluxes are not currently downscaled, but could be in the future if care were taken to preserve the cell-integrated values.

CLM has a somewhat unrealistic treatment of accumulation and melting for glacier landunits. The snow depth is limited to a prescribed depth of 1 m liquid water equivalent, with any additional snow assumed to run off to the ocean. (This amounts to a crude parameterization of iceberg calving.) Snow melting is treated in a realistic fashion, with meltwater percolating downward through snow layers as long as the snow is unsaturated. Once the underlying snow is saturated, any additional meltwater runs off. When glacier ice melts, however, the meltwater is assumed to remain in place until it refreezes. In warm parts of the ice sheet, the meltwater does not refreeze, but stays in place indefinitely.

In the modified CLM with glacier_mec columns, snow in excess of the prescribed maximum depth is assumed to turn into ice, contributing a positive surface mass balance to the ice sheet model. Melting ice is assumed to run off to the ocean, giving a negative surface mass balance. The net SMB associated with ice formation (by conversion from snow) and melting/runoff is computed for each column, averaged over the coupling interval, and sent to the coupler. This quantity, denoted *qice*, is then passed via the coupler to GLINT, along with the surface elevation *topo* in each column. GLINT downscales the SMB (renamed as *qsmb*) to the local elevation on the ice sheet grid, interpolating between values in adjacent elevation classes. The units of *qice* are mm/s, or equivalently km/m^2/s. If desired, the downscaled quantities can be multiplied by a normalization factor to conserve mass exactly. (This normalization is not yet implemented.)

Note that the surface mass balance typically is defined as the total accumulation of ice and snow, minus the total ablation. The *qice* flux passed to GLINT is the mass balance for ice alone, not snow. We can think of CLM as owning the snow, whereas Glimmer owns the underlying ice. Fluctuations in snow depth between 0 and 1 m LWE are not reflected in the SMB passed to GLINT.

In addition to *qice* and *topo*, the ground surface temperature *tsfc* is passed from CLM to GLINT via the coupler. This temperature serves as the upper boundary condition for Glimmer-CISM's temperature calculation.

Given the SMB from the land model, Glimmer-CISM executes one or more dynamic time steps and then has the option to upscale the new ice sheet geometry to the global grid and return it to CLM via the coupler. The fields passed to the coupler for each elevation class are the ice sheet fractional area (*gfrac*), surface elevation (*gtopo*), liquid (basal meltwater) runoff *grofl*, frozen (calving) runoff *grofi*, and surface conductive heat flux *ghflx*.

The current coupling is one-way only. That is, CLM sends the SMB and surface temperature to GLINT but does not do anything with the fields that are returned. The CLM surface topography is therefore fixed in time. One-way coupling is permissible for runs of ~100 years or less, in which ice-sheet elevation changes are modest. For longer runs with larger elevation changes, two-way coupling is highly desirable. A two-way coupling scheme is under development.

5.3 Model controls

The number of elevation classes is determined by the variable *GLC_NEC* in the file *env_run.xml* in the case directory. Values of 1, 3, 5, and 10 elevation classes are currently supported by the code, with 10 classes being the default. **However, running with anything other than 10 elevation classes will require that you create your own surface dataset – see the relevant question in the Frequently Asked Questions section, below. Furthermore, there is currently a bug in CLM that prevents**

running the glacier_mec code with 1 elevation class. (This bug is in the setting of the elevation of virtual columns in subgridMod.F90.)

The array *glc_topomax*, which is read from CLM's surface dataset (set by models/lnd/clm/tools/mksurfdata_map/src/mkglcmecMod.F90), defines the maximum elevation (in meters) in each class. For 10 elevation classes, glc_topomax is set to (0, 200, 400, 700, 1000, 1300, 1600, 2000, 2500, 3000, 10000). Note that this array must also agree with the *topomax* array set in CISM, in *glint_type.F90.*

At initialization, CLM's surface dataset specifies the areal percentage of each grid cell classified as wetland, vegetation, lake, urban, glacier, or glacier_mec. For glacier_mec cells, the area and surface elevation are specified in each elevation classes. The area and surface elevation may change during the course of the run, but the total glacier_mec area in a given grid cell is fixed; glacier_mec landunits cannot change to vegetated landunits or vice versa. This restriction will be relaxed in future model releases.

The fundamental control variable is *create_glacier_mec_landunit*, a logical variable declared in *clm_varctl.F90*. It is false by default, but is automatically set to true when we create a case that includes a dynamic ice sheet component (e.g., IG, FG, or BG). If *create_glacier_mec_landunit* = T, the following occurs:

- Memory is allocated for the areal percentage (*pct_glcmec*) and surface elevation (*topo_glcmec*) in each elevation class, and these values are read in from the surface dataset. The sum of *pct_glcmec* in each grid cell is checked to make sure it agrees with *pctgla*, the total glaciated fraction in each grid cell.
- Glacier_mec landunits and columns are defined for all grid cells where either (1) the fractional glacier area is greater than zero or (2) the dynamic ice sheet model may require a surface mass balance, even if CLM does not have glacier landunits in that location. To allow for case (2), grid overlap files have been precomputed. For given resolutions of CLM and Glimmer-CISM, these files identify all land-covered grid cells that overlap any part of the ice sheet grid. In these overlapping cells, glacier_mec columns are defined in all elevation classes. Some columns may have zero area and are called "virtual" columns. These columns do not affect energy exchange between the land and the atmosphere, but are included for potential forcing of Glimmer-CISM.

The logical variable *glc_smb* determines what kind of information is passed from CLM to the ice sheet model via the coupler. If *glc_smb* is true, then the surface mass balance is passed. Specifically, *qice* is interpreted by the ice sheet model as a flux ($kg/m^2/s$) of ice freezing/melting. If *glc_smb* is false, then the ice sheet model should compute the surface mass balance using a positive-degree-day scheme, with *qice* interpreted as the precipitation and *tsfc* as the 2-m air temperature. (The PDD option is not currently supported, but will be included in a future release.) In either case, *tsfc* is downscaled and applied as the upper boundary condition for the dynamic ice sheet.

The logical variable *glc_dyntopo* controls whether CLM surface topography changes dynamically as the ice sheet evolves (i.e., whether the coupling is one-way or two-way). The default (and the only option currently supported) is *glc_dyntopo* = F, in which case the land topography is fixed. In this case the surface runoff for glacier_mec landunits is computed as for glacier landunits: (1) Any snow in excess of 1 m LWE runs off to the ocean, and (2) Melted ice remains in place until it refreezes. Excess snow and melted ice still contribute to positive and negative values, respectively, of *qice*, but only for the purpose of forcing Glimmer-CISM.

If *glc_dyntopo* = T, then CLM receives updated topographic information from the ice sheet model. In this case the CLM surface runoff is computed in a more realistic way: (1) Any snow in excess of 1 m LWE is assumed to turn to ice and does not run off. (2) Melted ice runs off.

Two physical parameters may be useful for tuning the surface mass balance: (1) the surface bare ice albedo, *albice*, which is set in SurfaceAlbedoMod.F90, and (2) the surface air temperature lapse rate, *lapse_glcmec*, which is used for downscaling temperature and is set in *clm_varcon.F90*. By default, the bare ice albedo is 0.80 for visible wavelengths and 0.55 for near IR, but for glacier_mec columns the bare ice albedo is automatically changed to 0.60 / 0.40 (for the two wavelengths) in the namelist. The default lapse rate is 6.0 deg/km.

The snow albedo is not easily tunable. It is computed in a complicated way based on snow fraction, snow depth, snow age, and solar zenith angle. Snow albedo in glacier_mec columns is treated identically to snow in other landunits.

Another possible tuning mechanism is to convert rain to snow and vice versa as a function of surface temperature. This conversion would violate energy conservation, but might give more realistic precipitation fields in columns with elevations much higher or lower than the gridcell mean.

The default values of *albice*, *create_glacier_mec_landunit*, *glc_smb*, and *glc_dyntopo* may each be overwritten by specifying the desired values in the namelist. This is done automatically for *albice* and *create_glacier_mec_landunit* when a case is created with dynamic ice sheets.

5.4 CLM surface datasets used for runs with CISM

When running a compset with CISM present (i.e., an IG, FG or BG compset, for which CLM will use multiple elevation classes), the default CLM surface dataset uses a newer dataset to specify glacier coverage, compared with CLM runs in the absence of CISM. In most places, this uses the Randolph Glacier Inventory (Arendt et al., 2012). However, over Greenland, the surface dataset uses data from CISM's *gland5UM* initialization file, so that CLM and CISM give consistent glacier coverage at initialization. These new glacier coverage fields are described in more detail in the CESM1.1 addendum to the CLM4 tech note.

6 Frequently Asked Questions

This section answers some miscellaneous questions about ice sheet modeling within CESM.

6.1 How do I run an ensemble of ice sheets, which differ in their parameters, initial conditions, and/or forcing data?

Starting with CESM1.1, it is possible to run an ensemble of any model component(s) within a single CESM executable. Thus, you have a single case directory and a single run directory, but with separate input and output files for each ensemble member. In much of the CESM documentation, this capability is referred to as "multi-instance". But here we refer to it as "multiple ensemble members" to avoid confusion with CISM's capability to run multiple ice sheets (e.g., Greenland and Antarctica), which is also referred to as multiple instances.

An ensemble can be used in a TG compset to run multiple versions of CISM, either forced by the same data or each forced by different data. An ensemble can also be used in a coupled configuration (IG, FG or BG compsets). In this case, a requirement is that if there are N ensemble members of any one active component, then there must be N ensemble members of ALL active components. For example, in an IG compset, if you want to run with 32 CISM ensemble members, then you must also have 32 CLM ensemble members, where CLM #1 will be coupled to CISM #1, etc. In this case, you could choose to have either a single data atmosphere, or 32 data atmospheres, each providing their own forcing to one of the 32 CLMs.

Note that different ensemble members can ONLY differ by differences in namelist / configuration settings, and initial conditions / restart files – they are all using the same model executable.

The use of the ensemble capability is as follows; this assumes you are running a TG compset:

1. Create the case, as normal
2. Modify *env_mach_pes.xml*:
    a. Specify the desired number of CISM ensemble members by modifying *NINST_GLC*
    b. Also change *NTASKS_GLC* to this same value: this specifies the total number of GLC tasks, which are divided among the N ensemble members
    c. If you want to use separate forcing data for each ensemble member, then also modify *NINST_LND* and *NTASKS_LND* similarly, in order to run with a

separate data land model for each CISM ensemble member (see below for more details on how to use different forcing data for each ensemble member)

   d. Probably also change *NTASKS_CPL* to be equal to *NTASKS_GLC*, although for large ensembles you may want to experiment with a few different values of this variable to obtain the best performance

3. Run *cesm_setup*. This will create a separate *user_nl_cism_NNNN* file for each ensemble member (where *NNNN* is the number of the ensemble member). If you are running with multiple data land ensemble members, then there will also be separate *user_nl_dlnd_NNNN* and *user_nl_dsno_NNNN* files.
4. Put any namelist / configuration modifications specific to a single ensemble member in that member's *user_nl_cism_NNNN* file. Modifications that apply to all ensemble members must be put in ALL *user_nl_cism_NNNN* files.
5. Build and run the model as normal. Output files will be similar to those in a standard run, except there will now be N instances of each output file (history, restart and log files), which are differentiated by an occurrence of *_NNNN* in the file name.

6.1.1 How do I initialize each ensemble member with a different restart file?

In each *user_nl_cism_NNNN* file, set the value of the *cisminputfile* parameter to point to the desired restart file for that ensemble member. For example:

- echo "cisminputfile = '/path/to/file/for/member1.nc' " >> user_nl_cism_0001
- echo "cisminputfile = '/path/to/file/for/member2.nc' " >> user_nl_cism_0002

6.1.2 How do I use different forcing data for each CISM ensemble member in a TG compset?

To run with different forcing data for each CISM ensemble member in a TG compset, you need to use N datm instances, as described above. datm #1 will force CISM #1, etc. Note that, even if the different CISM ensemble members are identical in everything except for their forcings, you still need to use N CISM ensemble members in this case.

After running *cesm_setup*, you will see multiple instances of each of the dlnd input files in CaseDocs: *dlnd_in_NNNN, dsno_in_NNNN,* and *dlnd.streams.txt.sno.cplhist_NNNN.*

You can then modify each *dlnd.streams.txt.sno.cplhist_NNNN* file so that each dlnd ensemble member uses forcing data from a different set of files. To do this: After you have made all necessary modifications to the *DLND_** variables in *env_run.xml*, do the following from your case directory; where there is an *NNNN* in the following commands, you should run the command once for each ensemble member (i.e., replace *NNNN* with *0001, 0002*, etc.):

- preview_namelists
- cp CaseDocs/dlnd.streams.txt.sno.cplhist_NNNN ./user_dlnd.streams.txt.sno.cplhist_NNNN
- chmod u+w user_dlnd.streams.txt.sno.cplhist_NNNN

Then change each *user_dlnd.streams.txt.sno.cplhist_NNNN* file as desired. Usually, you will change the variables listed under *filePath* and/or *fileNames* in the *fieldInfo* section of this file.

Important note: Once you have created a *user_dlnd.streams.txt.sno.cplhist* file, further changes to most of the *DLND_** variables in *env_run.xml* will have no effect. Thus, if you need to change any of these variables, first remove the *user_dlnd.streams.txt.sno.cplhist_NNNN* files, then make necessary modifications to these xml variables. Finally, repeat the above procedure for modifying the streams files.

6.1.3 How do I force each ensemble member with a different set of years?

If you want to use a different set of years for each CISM ensemble member in a TG compset, you can do this by setting the *streams* variable in each of the *user_nl_dsno_NNNN* files. This technique can be used separately from or together with the above technique for using a different set of forcing files for each ensemble member.

For example, if you have forcing data from 1850 – 2005, and you want to force CISM #1 by looping through the years 1850 – 1899, and CISM #2 by looping through the years 1900 – 2005, you would do the following:

1. In *env_run.xml,* set *DLND_CPLHIST_YR_START* to 1850 and *DLND_CPLHIST_YR_END* to 2005. These variables determine the full set of files available to dlnd, by creating the file names in the *dlnd.streams.txt.sno.cplhist_NNNN* files. It is fine for all available years to be listed in each of these streams files. So if all you want to do is use a different subset of years for each ensemble member (drawing from the same full set of files), you will not need to modify these streams files.

2. Add the following line to *user_nl_dsno_0001* (note that the three values are: *yr_align, yr_start, yr_end*. This example assumes that you want to use *yr_align*=1, which would be appropriate if *RUN_STARTDATE* is year 1; otherwise, you likely will want to set *yr_align* to match the start year of the run):

   streams = "dlnd.streams.txt.sno.cplhist_0001 1 1850 1899"

3. Add the following line to *user_nl_dsno_0002:*

   streams = "dlnd.streams.txt.sno.cplhist_0002 1 1900 2005"

6.2 How do I run CLM with something other than 10 elevation classes?

For compsets that use CLM's *glacier_mec* code (IG, FG and BG compsets), the default is to use 10 elevation classes. As discussed in Section 5.3, the code currently supports running with 1, 3, 5 or 10 elevation classes. **However, there is currently a bug in CLM that prevents running the glacier_mec code with 1 elevation class.** (This bug is in the setting of the elevation of virtual columns in subgridMod.F90.)

Each number of elevation classes requires a different surface dataset for CLM. Currently, CLM surface datasets have only been created with 10 elevation classes. Thus, if you want to run with a different number of elevation classes, you must first create a new surface dataset, along with a transient pft dataset if you will be doing a transient run. The process for creating these datasets is described in the CLM User's Guide. Briefly, you will use CLM's *mksurfdata_map* tool, contained in *models/lnd/clm/tools/mksurfdata_map.* It is easiest to use the *mksurfdata.pl* wrapper script contained in that directory, providing the argument *-glc_nec.* For example, to create a surface dataset with five elevation classes at 0.9x1.25 degree resolution, for the year 1850, along with a transient pft dataset spanning the late 19^{th} and 20^{th} centuries, you would run:

➢ `mksurfdata.pl -res 0.9x1.25 -y 1850-2000 -glc_nec 5`

You can then create a case using an IG, FG or BG compset. Before running, you will need to change three settings:

- *GLC_NEC* in *env_run.xml* (e.g., *xmlchange GLC_NEC=5*)
- Point CLM to the new surface dataset you created (specify *fsurdat* in *user_nl_clm*)
- If relevant to your case, point CLM to the new transient pft dataset you created (specify *fpftdyn* in *user_nl_clm*)

6.3 How do I add new settings in CISM's namelist or config file?

If your code development requires the addition of a runtime setting, set either in *cism_in* or *cism.config*, you will need to add information about the new variable in the xml file that is used to generate these input files. See the documentation in *models/glc/cism/bld/README.build-namelist*, and particularly the section "CISM Use Cases".

6.4 How do I add a CLM history field that provides averages only over ice?

In general, CLM history fields give weighted averages over the entire grid cell. If you are interested in diagnostics just over ice landunits for certain history fields, you can make a source code modification for each field of interest. This is done in *models/lnd/clm/src/main/histFldsMod.F90*: Find the history field(s) of interest in this file, and add the following optional argument to the *hist_addfld1d* or *hist_addfld2d* call for that history field: *l2g_scale_type='ice'*. You may want to copy and paste the call in order to maintain the original history field and add a new field that applies just over ice (being sure to change *fname*). For example, examine the difference between the fields *TSOI* and *TSOI_ICE*:

```
call hist_addfld2d (fname='TSOI',  units='K', type2d='levgrnd', &
   avgflag='A', long_name='soil temperature (vegetated landunits only)', &
   ptr_col=clm3%g%l%c%ces%t_soisno, l2g_scale_type='veg')

call hist_addfld2d (fname='TSOI_ICE',  units='K', type2d='levgrnd', &
   avgflag='A', long_name='soil temperature (ice landunits only)', &
   ptr_col=clm3%g%l%c%ces%t_soisno, l2g_scale_type='ice')
```

7 Acknowledgments

Many researchers in the glaciology and climate modeling communities have contributed to this work. Miren Vizcaíno (UC Berkeley) has worked closely on the development of the surface-mass-balance scheme and analysis of model output. Jon Wolfe (NCAR) was the lead software engineer for the initial implementation of ice sheets in CESM. Tony Craig, Erik Kluzek, David Lawrence, and Mariana Vertenstein (NCAR) have supported overall software development, including the incorporation of ice sheets in CLM.

Development of Glimmer-CISM is overseen by a steering committee consisting of Magnus Hagdorn (U. Edinburgh), Jesse Johnson (U. Montana), William Lipscomb (LANL), Tony Payne (U. Bristol), Stephen Price (LANL), and Ian Rutt (U. Swansea). They and many others have contributed to the dynamic ice sheet model.

Anjuli Bamzai (formerly DOE, now NSF), Jay Fein (NSF), Renu Joseph (DOE), and Julie Palais (NSF) have provided essential funding for ice sheet model development and integration into CESM. Phil Jones (LANL) has supported this work as manager of LANL's COSIM project and as co-PI, with John Drake (ORNL), of the DOE SciDAC project that has funded much of this effort.

Lali Chatterjee, William Spotz, and Randall Laviolette (DOE) have directed the DOE Ice Sheet Initiative for Climate ExtremeS (ISICLES) project, which is contributing to numerical advances that are now being incorporated in Glimmer-CISM. William Collins

(LBNL) has led the Investigation of the Magnitudes and Probabilities of Abrupt Climate TransitionS (IMPACTS) project, which is developing new ice-ocean coupling methods that will be incorporated in future versions of CESM.

William Collins, Peter Gent and James Hurrell have supported this project while managing overall CCSM/CESM development at NCAR. David Holland (NYU), Jeff Ridley (U.K. Met Office), Slawek Tulaczyk (UC Santa Cruz), and many others have given helpful advice and encouragement along the way.

8 References

Arendt, A., Bolch, T., Cogley, J. G., Gardner, A., Hagen, J.-O., Hock, R., Kaser, G., Pfeffer, W. T., Moholdt, G., Paul, F., Radić, V., Andreassen, L., Bajracharya, S., Beedle, M., Berthier, E., Bhambri, R., Bliss, A., Brown, I., Burgess, E., Burgess, D., Cawkwell, F., Chinn, T., Copland, L., Davies, B., Dolgova, E., Filbert, K., Forester, R., Fountain, A., Frey, H., Giffen, B., Glasser, N., Gurney, S., Hagg, W., Hall, D., Haritashya, U. K., Hartmann, G., Helm, C., Herreid, S., Howat, I., Kapustin, G., Khromova, T., Kienholz, Koenig, M., Kohler, J., Kriegel, D., Kutuzov, S., Lavrentiev, I., LeBris, R., Lund, J., Manley, W., Mayer, C., Li, X., Menounos, B., Mercer, A., Moelg, N., Mool, P., Nosenko, G., Negrete, A., Nuth, C., Pettersson, R., Racoviteanu, A., Ranzi, R., Rastner, P., Rau, F., Rich, J., Rott, H., Schneider, C., Seliverstov, Y., Sharp, M., Sigurðsson, O., Stokes, C., Wheate, R., Winsvold, S., Wolken, G., Wyatt, F., and Zheltyhina, N., 2012: Randolph Glacier Inventory: A Dataset of Global Glacier Outlines Version: 1.0, Global Land Ice Measurements from Space, Boulder Colorado, USA. Digital Media.

Bougamont, M., J.L. Bamber, J.K. Ridley, R.M. Gladstone, W. Greuell, E. Hanna, A.J. Payne, and I. Rutt, 2007: Impact of model physics on estimating the surface mass balance of the Greenland ice sheet, *Geophys. Res. Lett., 34*, 10.1029/2007GL030700.

Cazenave, A., K. Dominh, S. Guinehut, E. Berthier, W. Llovel, G. Ramilien, M. Ablain, and G. Larnicol, 2008: Sea level budget over 2003-2008: A reevaluation from GRACE space gravimetry, satellite altimetry, and Argo, *Global Planet. Change*, doi:10.1016/j.gloplacha.2008.10.004.

Greve, R., and H. Blatter, 2009: Dynamics of Ice Sheets and Glaciers. Springer, Berlin, 287 pp.

Holland, D.M., R.H. Thomas, B. De Young, M.H. Ribergaard, B. Lyberth, 2008a: Acceleration of Jakobshavn Isbrae triggered by warm subsurface ocean waters, *Nature Geo., 1*, 659-664.

Holland, P.R., A. Jenkins, and D.M. Holland, 2008b, The response of ice shelf basal melting to variations in ocean temperature, *J. Climate, 21*, 2558-2572.

Lipscomb, W., R. Bindschadler, E. Bueler, D. Holland, J. Johnson, and S. Price, 2009: A community ice sheet model for sea level prediction, *Eos Trans. AGU*, *90*, 23.

Little, C.M., M. Oppenheimer, R.B. Alley, V. Balaji, G.K.C. Clarke, T.L. Delworth, R. Hallberg, D.M. Holland, C.L. Hulbe, S. Jacobs, J.V. Johnson, H. Levy, W.H. Lipscomb, S.J. Marshall, B.R. Parizek, A.J. Payne, G.A. Schmidt, R.J. Stouffer, D.G. Vaughan, and M. Winton, 2007: Toward a new generation of ice sheet models, *Eos Trans. AGU*, *88*, 578-579, doi:10.1029/2007EO520002.

MacAyeal, D.R., 1989: Large-scale ice flow over a viscous basal sediment: Theory and application to Ice Stream B, Antarctica, *J. Geophys. Res., 94 (B4),* 4071-4087.

Meehl, G.A., et al., 2007: Global climate projections, in Climate Change 2007: Contribution of Working Group I to the Fourth Assessment Report of the IPCC, edited by S. Solomon et al., pp. 747–846, Cambridge Univ. Press, New York.

Meier, M.F., M.B. Dyurgerov, U.K. Rick, S. O'Neel, W.T. Pfeffer, R.S. Anderson, S.P. Anderson, and A.F. Glazovsky, 2007: Glaciers dominate eustatic sea-level rise in the 21^{st} century, *Science*, *317*, 1064-1067.

Pattyn, F., 2003: A new three-dimensional higher-order thermomechanical ice sheet model: Basic sensitivity, ice stream development, and ice flow across subglacial lakes, *J. Geophys. Res., 108(B8),* 2382, doi:10.1029/2002JB002329.

Payne, A. J., P. Huybrechts, A. Abe-Ouchi, R. Calov, J.L. Fastook, R. Greve, S.J. Marsahll, I. Marsiat, C. Ritz, L. Tarasov, and M.P.A. Thomassen, 2000: Results from the EISMINT model intercomparison: the effects of thermomechanical coupling, *J. Glaciol., 46(153)*, 227-238.

Pritchard, M.S., A.B.G. Bush, and S.J. Marshall, 2008: Neglecting ice-atmosphere interactions underestimates ice sheet melt in millennial-scale deglaciation simulations, *Geophys. Res. Lett. 35*, L01503, doi:10.1029/2007GL031738.

Radić, V., and R. Hock, 2010: Regional and global volumes of glaciers derived from statistical upscaling of glacier inventory data, *J. Geophys. Res., 2215*, F01010.

Rahmstorf, S., 2007: A semi-empirical approach to projecting future sea-level rise, *Science, 315*, 368-370.

Rignot, E., J.L. Bamber, M.R. van den Broeke, C. Davis, Y. Li, W.J. van de Berg, and E. van Meijgaard, 2008: Recent Antarctic ice mass loss from radar interferometry and regional climate modeling, *Nature Geo., 1*, 106-110.

Rutt, I., M. Hagdorn, N.J.R. Hulton, and A.J. Payne, 2009: The Glimmer community ice sheet model, *J. Geophys. Res., 114*, F02004.

Schoof, C. and R.C.A. Hindmarsh, 2010: Thin-film flows with wall slip: An asymptotic analysis of higher order glacier flow models, *Quart. J. Mech. Appl. Math., 63, 73-114,* doi: 10.1093/qjmam/hbp025.

van de Wal, R.S.W., 1996: Mass-balance modeling of the Greenland ice sheet: A comparison of an energy-balance and a degree-day model, *Ann. Glaciol., 23*, 36-45.

van den Broeke, M., J. Bamber, J. Ettema, E. Rignot, E. Schrama, W.J. van der Berg, E. van Meijgaard, I. Velicogna, and B. Wouters, 2009: Partitioning recent Greenland mass loss, *Science, 326*, 984-986.

Velicogna, I., 2009: Increasing rates of ice mass loss from the Greenland and Antarctic ice sheets revealed by GRACE, *Geophys. Res. Lett., 36*, L19503.