Geosci. Model Dev., 7, 883–907, 2014
www.geosci-model-dev.net/7/883/2014/
doi:10.5194/gmd-7-883-2014

Geoscientific Model Development
Open Access

A system of conservative regridding for ice–atmosphere coupling in a General Circulation Model (GCM)

E. Fischer[1,2], **S. Nowicki**[3], **M. Kelley**[2,4], **and G. A. Schmidt**[2]

[1]Center for Climate Systems Research, Columbia University, New York, NY, USA
[2]NASA Goddard Institute of Space Studies, New York, NY, USA
[3]NASA Goddard Space Flight Center, Greenbelt, MD, USA
[4]Trinnovim LLC, 2880 Broadway, New York, NY 10025, USA

Correspondence to: E. Fischer (elizabeth.fischer@columbia.edu)

Received: 12 November 2013 – Published in Geosci. Model Dev. Discuss.: 6 December 2013
Revised: 7 March 2014 – Accepted: 28 March 2014 – Published: 19 May 2014

Abstract. The method of elevation classes, in which the ice surface model is run at multiple elevations within each grid cell, has proven to be a useful way for a low-resolution atmosphere inside a general circulation model (GCM) to produce high-resolution downscaled surface mass balance fields for use in one-way studies coupling atmospheres and ice flow models. Past uses of elevation classes have failed to conserve mass and energy because the transformation used to regrid to the atmosphere was inconsistent with the transformation used to downscale to the ice model. This would cause problems for two-way coupling.

A strategy that resolves this conservation issue has been designed and is presented here. The approach identifies three grids between which data must be regridded and five transformations between those grids required by a typical coupled atmosphere–ice flow model. This paper develops a theoretical framework for the problem and shows how each of these transformations may be achieved in a consistent, conservative manner. These transformations are implemented in Glint2, a library used to couple atmosphere models with ice models. Source code and documentation are available for download. Confounding real-world issues are discussed, including the use of projections for ice modeling, how to handle dynamically changing ice geometry, and modifications required for finite element ice models.

1 Introduction

Many questions still surround the issues of how ice sheets respond to climate forcing and how those changes will affect sea level, regional and global climate. Recent observations have shown accelerating mass loss from the Greenland and Antarctic ice sheets (Vaughan et al., 2013), adding urgency to these questions. Although general circulation models (GCMs) are able to project changes in ice sheet surface mass balance, projection of changes in ice sheet mass due to ice dynamics with GCMs remains a challenge (Church et al., 2013). A number of climate modeling groups are addressing these deficiencies by adding dynamic ice flow effects to existing GCMs: GISS ModelE (Schmidt et al., 2006), CESM (Hurrell et al., 2013), HadGEM2 (Collins et al., 2011), etc. This is done by coupling the GCM with an existing *ice flow model*, such as Glimmer–CISM (Rutt et al., 2009), BISICLES (Cornford et al., 2013), ISSM (Larour et al., 2012), PISM (Bueler and Brown, 2009), etc. A full understanding of the long-term evolution of an ice sheet within a coupled climate system requires coupling with the ocean as well as atmosphere. Surface runoff, ocean cavity circulation and salinity gradient effects are all important. In this paper, we focus only on coupling with the atmosphere.

One can distinguish between *one-way* and *two-way* coupling. In one-way coupling, the GCM is used to develop surface mass balance (SMB) and temperature fields, which are then used to drive the ice flow model off-line. This process misses effects caused by feedbacks from the ice sheet to the rest of the Earth system: for example, decreased ice sheet

albedo (Qu and Hall, 2006) or lowered atmosphere orography (Ridley et al., 2005). Past studies with one-way coupling have yielded useful insight into the future of present-day ice sheets; examples include Huybrechts (1994), Greve (2000), Stone et al. (2010), Bindschadler et al. (2013), Lipscomb et al. (2013), Nowicki et al. (2013a), Nowicki et al. (2013b) and Goelzer et al. (2013). However, ice sheet feedbacks are expected to be increasingly important for simulations of the long-term evolution of ice sheets and the climate associated with them. This kind of feedback probably plays a significant role in many events in the paleorecord (Dansgaard–Oeschger events, Heinrich events, the Younger Dryas).

Two-way coupling strategies address these issues by allowing the atmosphere to be influenced by changes in the ice sheet elevation, extent and albedo over time. We distinguish between *loose* and *tight* two-way coupling. Loose two-way coupling involves running a series of GCM simulations with different ice sheet configurations, each based on the result of the previous. Each GCM run is a separate simulation, without continuity of mass or energy between runs. Studies with loose two-way coupling have yielded insight into future equilibrium states for ice sheets and climate (Ridley et al., 2005, 2010). However, ice sheets change configuration in these runs without accompanying mass and energy fluxes required to make those changes happen. This is equivalent to applying an unknown impulse forcing to the ice sheet each time it is changed. Although the coupled ice sheet might eventually reach the correct equilibrium state, the transients involved in achieving that equilibrium will be suspect. Unfortunately, results relevant to human society all require an understanding of the transients. For successful simulation of transients, we turn to tight two-way coupling. It involves running the ice flow model, step by step, along with the rest of the GCM – while conserving mass and energy along the way. Attention to conservation is required, since the GCM is simulating a more nearly closed system that could run for a long time.

When one couples dynamic ice flow models with GCM atmospheres, two models operating on different grids and timescales must communicate: ice flow models operate at low frequency on a high-resolution grid with local projection, while GCM atmospheres operate at high frequency on low-resolution global grids. A number of issues arise due to this mismatch, including how one creates high-resolution surface mass balance fields from low-resolution GCM input. Elevation classes address the latter issue. They were first introduced for precipitation downscaling in a GCM by Leung and Ghan (1998) and later applied to one-way coupling from GCM atmosphere to ice flow models by Lipscomb et al. (2013). The key insight is that mass and energy fluxes between the atmosphere and an ice sheet vary approximately by elevation within a local region.

When using elevation classes, a third grid is introduced, the *elevation grid*. This allows the GCM to compute surface fields at a variety of elevations within each atmosphere grid cell, not just the elevation seen by the atmosphere. A high-resolution surface mass balance is produced on the ice grid by first computing SMB on the elevation grid, and then using a vertical interpolation scheme to produce SMB on the ice grid. This method of interpolation produces surprisingly good results: although it cannot capture certain localized effects (e.g., wind direction), it has been shown to allow GCMs to produce surface mass balance fields approaching the quality of those produced by regional climate models (Vizcaino et al., 2013).

Inside the GCM, SMBs computed on the elevation grid must be regridded to the atmosphere grid as well as the ice grid. In order to maintain conservation in a tight two-way coupled system, it is essential that the set of regridding operation chosen is self-consistent: that is, if a flux field on the elevation grid is regridded simultaneously to the atmosphere and ice grid, then the total amount of flux represented by the resulting two fields should be the same. For conservation purposes, the specifics of these two transformations are not important, as long as they are consistent with each other. Past efforts at one-way coupling have defined these two transformations in ways that each make intuitive sense, but are not consistent with each other. This is not a problem for one-way coupling, but it would cause conservation problems for two-way coupling.

This paper develops the concept of the elevation grid on which the ice surface model runs, and then derives a set of conservative regridding transformations between the atmosphere, elevation and ice grids. The coupled processes under consideration along with the transformations required for tight two-way coupling are introduced in Sect. 2. Section 3 focuses on the elevation grid, while the grid fundamentals necessary for the two-way coupling are presented in Sect. 4. Section 5 deals with the use of projection and associated issues encountered when bridging between the spherical geometry of GCMs and Cartesian ice sheet models. We show how to choose and implement the transformations in Sects. 6 through 9 and work through realistic examples of these transformations in Sects. 10 through 12. We touch on a number of extra "wrinkles" in the real-world problem: procedures to use when the elevation grid is based on a horizontal grid other than the atmosphere grid in Sect. 13, and regridding procedures required when ice elevations, ice extent or elevation grid change in the simulation in Sect. 14. Finally, we present in Sect. 15 a library, Glint2, that can be used to tightly couple GCMs and ice sheet models.

2 Coupled processes

The coupled atmosphere–ice system involves three models interacting with each other: an atmosphere model, an ice flow model and an ice surface model situated between them (Fig. 1). The atmosphere model is coupled with the ice surface model, which tracks the top few meters of ice. Processes

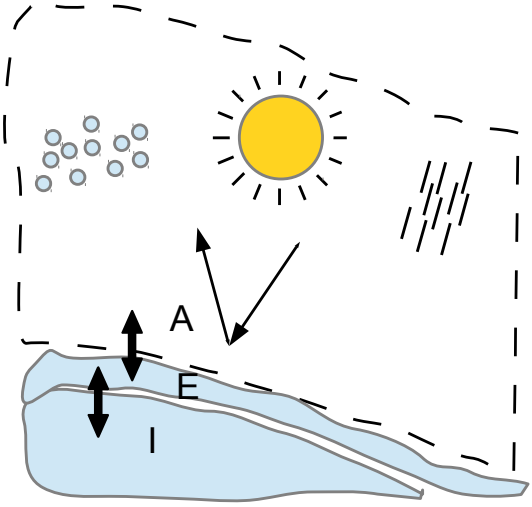

Fig. 1. Configuration of the three models, and the separate spatial domains they occupy. The atmosphere model (A) is coupled to the ice surface model (E), which tracks the top few meters of ice. The ice surface model (E) is then coupled to the dynamic ice flow model (I). The dynamic ice flow model, operating at long time steps, is insulated from high-frequency surface processes in the ice–atmosphere interaction.

modeled here include a full surface mass–energy balance computation, snow–firn compaction, albedo effects, meltwater percolation, runoff, refreezing, etc. The bottom of the ice surface model is coupled to the ice flow model.

2.1 Ice surface model

In order to couple an ice sheet to a GCM atmosphere, SMB on the ice sheet must be calculated from GCM outputs. Some one-way coupled studies have used temperature index schemes (Huybrechts, 1994; Greve, 2000; Stone et al., 2010; Bindschadler et al., 2013; Nowicki et al., 2013a, b; Goelzer et al., 2013): the mean surface temperature over each coupling time step (typically one month or year) is used to compute SMB, and both are passed to the ice flow model. It is hard to see how energy can be conserved with such a scheme. The problem is that the atmosphere must run for many time steps before atmosphere–ice sheet energy fluxes are computed on a coupling time step. The computed flux on the ice sheet will not be the same as that sum of fluxes seen by the atmosphere over the previous coupling time step – and it is "too late" to go back and change the atmosphere to match.

For this reason, a full energy balance scheme, computed each atmosphere time step (typically one hour) and integrated over the coupling time step, is considered essential in a GCM setting. Energy flux between atmosphere and ice sheet follows diurnal and seasonal cycles, making positive and negative contributions to the integrated flux. It is important that energy flux is computed at a small enough time step to capture these effects accurately; typically, the atmosphere time step is sufficient.

While ice sheets move slowly and can be modeled with long time steps, the modeling of the *surface* of the ice sheet requires short time steps. This is accomplished by introducing the ice surface model as a third model, sitting in between the ice flow model and atmosphere model. The top of the ice surface model couples with the atmosphere model every atmosphere time step, whereas the bottom couples with the ice sheet model every coupling time step. The ice surface model needs to be thick enough so there is little variation in temperature at the bottom: 15 m is sufficient. This ensures that the heat flux with the ice flow model will be small and low frequency.

The ice surface model can serve an additional purpose of modeling the great variety of surface processes that may be relevant to the long-term evolution of ice sheets: snow–firn compaction, surface runoff/drainage networks, water percolation, refreeze, albedo effects and wind-blown snow, to name a few.

The use of an ice surface model solves some important problems, but it also introduces nonphysical elements into the model. Ice contained in the ice surface model does not advect along with the ice flow model, nor does it contribute to the stress field of the ice below it in the ice flow model. In both cases, we expect the relative error to be small: the top 15 m of snow/firn contains less than 1 % of the mass of a 1000 m thick ice sheet. There may be ways to fix these problems – however, there is no need to make the ice surface model thicker than it needs to be. Numerous studies have shown that ice sheets below about 15 m are fully insulated from surface weather and seasonal cycles (Zagorodnov et al., 2006).

2.2 Three models, three grids

Each of the three models in the coupled system runs on its own grid. The atmosphere is run on the atmosphere grid (A) and the ice flow model is run on the ice grid (I). The ice surface model is run on the elevation grid (E), which is based on the atmosphere grid (see Sect. 3). All three grids are two-dimensional, in the sense that they are used to construct two-dimensional functions $f(x, y)$ over the domain. Regridding operations are needed to pass mass and energy fluxes between the models.

It is important to keep in mind the relative size of the three grids. We set up a test using the GISS $2^\circ \times 2\frac{1}{2}^\circ$ atmosphere grid (Schmidt et al., 2006), overlapping with the 5 km grid from SeaRISE (Sea-level Response to Ice Sheet Evolution; Bindschadler et al., 2013). We used 40 elevation points, spaced every 100 m from 0 m to 4000 m. In this case, A had 146 grid cells, I had 66 906 and E had 1829. These numbers only account for grid cells involved with the Greenland ice sheet. In general, the ice grid will be finest, the atmosphere grid coarsest and the elevation grid in the middle.

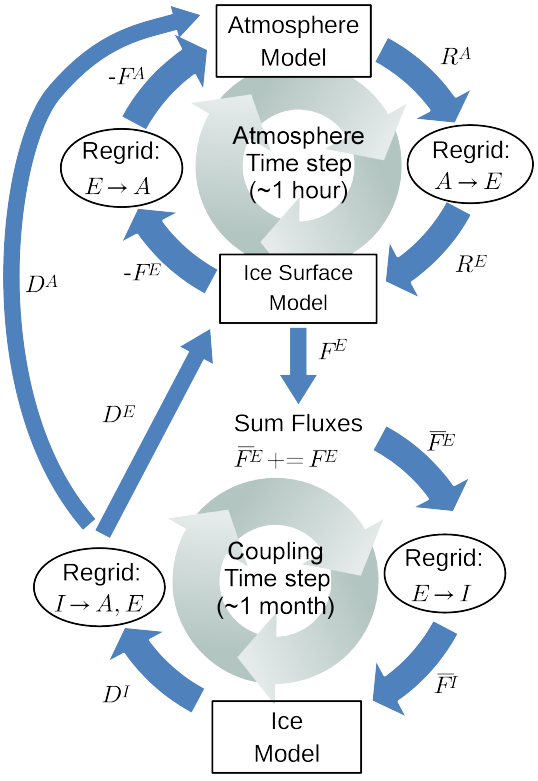

Fig. 2. Data flow (blue arrows) for the coupling between atmosphere, ice surface and dynamic ice flow models (boxes). Since the three models run on different grids, regridding operations (ovals) are required at each step. Figure 21 shows the inputs required to compute these regridding operations.

A	Atmosphere grid, projected to Cartesian plane
A'	Original atmosphere grid on the sphere
I	Ice grid
G	Interpolation grid (e.g., ice or exchange grid)
E	Elevation grid
$\mathbf{M}$	Interpolation matrix: $E \to G$
$\mathbf{R}$	Area-weighted remapping matrix: $G \to A$
$\mathbf{\Lambda}$	"Repeat" transformation matrix: $A \to E$
$\boldsymbol{X}$	Area-weighted remapping matrix: $G \to I$
$\boldsymbol{X}'$	Area-weighted remapping matrix: $I \to G$
$\boldsymbol{P}$	Diagonal scaling matrix: $A' \to A$
L0	Function defined to be constant within grid cells
L1	Function defined by piecewise linear interpolation between grid points

Fig. 3. Definition of symbols used throughout the paper. See Appendix A for notational conventions.

Time frequency mismatches are another issue. The atmosphere runs at high frequency, each time step being typically 1 h. On the other hand, the atmosphere and ice flow models are coupled at much lower frequency, typically one month or any other time period. We call these two time steps the *atmosphere time step* and *ice coupling time step*.

2.3 Fully coupled system

Figure 2 shows the data flow of the fully coupled system. Steps of the data flow are organized based on their frequency: the top circle of steps runs at the same frequency as the GCM atmosphere, while the bottom circle runs at the *ice coupling* frequency – typically one month or more. We describe the steps involved in coupling the GISS ModelE (Schmidt et al., 2006) with PISM (Bueler and Brown, 2009); however, these steps are general for any GCM or ice flow model. We now trace through the data flow on a typical coupled run.

2.4 Atmosphere time step

When the atmosphere runs, it produces a set of fields on the atmosphere grid that affect the processes in the ice surface model: for example, downwelling long-wave radiation, downwelling shortwave radiation, precipitation, etc. These fields must be regridded from A to E. We denote the set of fields to be regridded with a capital letter, $\mathbf{R}^A$; the result of the regridding is $\mathbf{R}^E$ (see Appendix A and Fig. 3 for notational conventions).

The ice surface model is run at the same frequency as the atmosphere. Among its outputs are mass and energy fluxes with the atmosphere: evaporation, sublimation, upwelling long-wave radiation, latent heat release, etc. These are represented by $-\mathbf{F}^E$ and are regridded to $-\mathbf{F}^A$ before being passed back to the atmosphere on the next time step.

The ice surface model also produces fluxes in the ice flow model; these are described immediately below.

2.5 Coupling time step

On each atmosphere time step, relevant flux outputs from the ice surface model are accumulated as $\bar{\mathbf{F}}^E$ for future coupling with the ice flow model. They are named $\bar{\mathbf{F}}^E$ because these fluxes are in general equal and opposite to fluxes sent to the atmosphere. Every coupling time step – about once a month – the accumulated $\bar{\mathbf{F}}^E$ is regridded to the ice grid ($\bar{\mathbf{F}}^I$) and passed to the ice flow model.

The ice flow model produces changes in ice surface topography and extent, as well as a small energy flux between the ice flow and ice surface models (together, we call these $\mathbf{D}^I$). Changes in ice topography and extent are regridded to the atmosphere grid ($\mathbf{D}^A$) and used to adjust the atmosphere's orography. The energy flux is regridded to the elevation grid ($\mathbf{D}^E$) and applied to the ice surface model.

2.6 Regridding requirements

We see that the fully coupled system requires the use of five regridding operations at various points in its run (Fig. 2). These regridding operations must be conservative, in the sense that *none* of them can change the integral of the field

on the domain as a whole. In fact, we would like to impose a stronger conservation condition so that values are conserved within each atmosphere grid cell.

We develop these five regridding operations in the sections below. We begin by discussing the method of elevation classes in a general manner (Sect. 3) and then move on to developing basics of conservative regridding (Sect. 4). Because conservation is defined in terms of integration over areas, part of our discussion involves a definition of how to integrate functions on the elevation grid E. Finally, we show how the required set of operations can be constructed in a fully self-consistent manner.

Throughout, we assume that the atmosphere grid A has L0 parameterization: functions on A are represented by their mean value within each grid cell. This becomes our basis for conservation: all regridding operations are conservative over every atmosphere grid cell. We do not require any specific parameterization for the ice grid, but we consider cases for L0 (piecewise constant) and L1 (piecewise linear) parameterizations, the latter which are commonly used in finite element ice models.

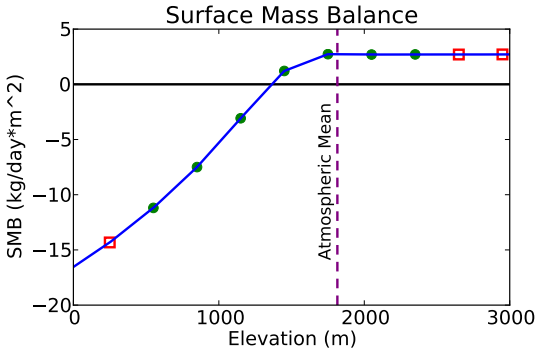

Fig. 4. Interpolated surface mass balance (SMB) function within one atmosphere grid cell. These values come from one month (July) of a run of GISS ModelE with elevation points. The vertical dashed line indicates the mean elevation of the grid cell, "seen" by the atmosphere. Dots and squares represent the results of extrapolated SMB computations at other elevations: dots represent elevations found within the grid cell, whereas squares represent elevations outside the cell's range. SMB values at intermediate elevations are interpolated.

3 Elevation points

The method of elevation classes, when applied to the atmosphere–ice coupling problem, involves running the ice surface model at one or more elevations in each atmosphere grid cell (Lipscomb et al., 2013). Temperature, pressure and precipitation are extrapolated to a set of elevations within the grid cell. They are then used in an ice surface model to compute a full surface mass and energy balance at each elevation. The result is a set of "what-if" scenarios, giving an estimate of what the fluxes *would have been*, had the ice surface of the grid cell been at the given elevation – rather than the elevation seen by the atmosphere model.

The modeler must choose which elevations to use for each atmosphere grid cell. The simplest approach is to use a fixed set of elevations across all grid cells – for example, every 100 m from 0 m to 4000 m.

However elevations are chosen, the result is a new "grid" – the *elevation grid* – on which the ice surface model is run and surface fluxes are generated. The elevation grid is derived from the atmosphere grid, in the sense that each elevation grid "cell" (or *elevation point*) is associated with one parent atmosphere grid cell. If elevation point E_j is associated with atmosphere grid cell A_i, we write $E_j \in A_i$.

Once the GCM has computed a conserved quantity on the elevation grid, those values can be used to develop a relation between elevation and SMB within each atmosphere grid cell. Suppose we have computed a flux field $\boldsymbol{f}^E$: SMB, for example. For an atmosphere grid cell A_i, consider the components of the flux field f_j^E that are related to the enclosing atmosphere grid cell A_i. Or more formally, consider f_j^E for all j such that $E_j \in A_i$. We can think of these component values as samples of a 1-D function relating elevation to flux (SMB), *within the localized region* of A_i. By interpolating between those points using standard methods, one can construct a continuous function relating flux to elevation within A_i. As long as enough points are used and the function being interpolated is smooth enough, this procedure will yield an arbitrarily accurate representation of the "true" function.

In fact, the functions we typically wish to interpolate – surface mass and energy balance averaged over about a month – are quite smooth as functions of elevation (Fig. 4). In the face of spatially invariant precipitation, one would expect SMB to be constant to first order above the equilibrium line altitude (ELA), and to decrease linearly with elevation below the ELA. Near the ELA, one would expect a smooth transition because the ELA goes up and down over a month of diurnal cycles. This is in agreement with experimental work, which has shown SMB below the ELA to vary linearly with temperature (Braithwaite, 1981; Box et al., 2013).

How many elevation points are required to properly resolve the elevation–flux relationship? This depends on the interpolation scheme: higher order schemes will require fewer points than piecewise linear interpolation. But the general shape of the function – two straight lines connected by a curve near the ELA – implies that not many points are needed, except for near the ELA.

If one knows where the ELA is on every atmosphere grid cell, then this is a useful guide in selecting elevation points. But if one is studying ice sheets in a changing climate, then the ELA will be expected to move over time. It is possible to adaptively move elevation points as the ELA moves. But it is simpler just to use many points everywhere. In our tests, we have used 40 points at 100 m spacing, which is probably more than sufficient for coupled ice sheet simulations.

We have not done a careful study of the optimal number of elevation points.

The method of elevation classes – or elevation points – provides a way to construct an interpolated relation between elevation and surface mass–energy flux within each atmosphere grid cell. Additional choices need to be made in order to produce fully downscaled flux fields on the ice grid (Sect. 6). We are now almost ready to define the transformations posited in Sect. 3. But first, we pause for some groundwork on grid fundamentals in Sect. 4 and projection issues in Sect. 5.

4 Grid fundamentals

Numerical models represent continuous fields with linear combinations of a finite set G of basis functions – which we call a *parameterization* (Appendix A describes our mathematical notation). For example, suppose that G uses the basis functions $\boldsymbol{g}(x,y) = [g_1(x,y),\ldots,g_n(x,y)]$, where $\boldsymbol{g}(x,y)$ is the vector of all the basis functions. Suppose we have an n-dimensional vector $\boldsymbol{f}^G$ with components f_i^G. That vector represents the function $f^G(x,y)$ formed by a weighted sum of the basis functions:

$$f^G(x,y) = \boldsymbol{f}^G \cdot \boldsymbol{g}(x,y). \tag{1}$$

A wide variety of basis function sets are used for different problems. Most commonly in climate modeling, f_i^G represents the mean value of $f^G(x,y)$ within some well-defined bounded region, which we call a *grid cell*. From a conservation point of view, a function $f^G(x,y)$ with L0 parameterization may be taken to be constant within each grid cell, with discrete "jumps" from one grid cell to the next. In this case, the basis function $g_i(x,y)$ for grid cell i is equal to 1 inside the cell and 0 outside.

L0 parameterizations are widely used in climate and finite difference ice flow models. Finite element ice flow models might use L1 or even higher order parameterizations (Zienkiewicz et al., 2013), and they use the term "mesh" to describe the geometry of their basis functions. In this paper, we use the term "grid" to refer to both the vector space in which $\boldsymbol{f}^G$ lives and its associated parameterization.

4.1 Integration on grids

Because we are working with conserved quantities, it is essential that we can integrate functions over any well-defined region B. Because a parameterization defines $f^G(x,y)$ on every point, this integral is well-defined. Substituting from Eq. (1) and using linearity, we get

$$\int_B f^G(x,y)\mathrm{d}A = \boldsymbol{f}^G \cdot \int_B \boldsymbol{g}(x,y)\mathrm{d}A. \tag{2}$$

If we wish to integrate over B repeatedly, we can precompute $\int_B \boldsymbol{g}(x,y)\mathrm{d}A$, and then take dot products with $\boldsymbol{f}^G$ as needed. How one integrates the basis functions depends on the nature of those basis functions. We give specifics for L0 and L1 grids in Appendices B and C, respectively.

4.2 Comparison across grids

In regridding applications, we need to compare fields across different grids. They cannot be tested for simple point-by-point equality, due to differences in grid structure. Instead, we compare by integrating two fields over the same region or set of regions.

If we have two fields $\boldsymbol{f}^G$ and $\boldsymbol{f}^H$ on two different grids, we say that $\boldsymbol{f}^G$ is equivalent to $\boldsymbol{f}^H$ on region B, or $\boldsymbol{f}^G \equiv^B \boldsymbol{f}^H$ if

$$\boldsymbol{f}^G \cdot \int_B \boldsymbol{g}(x,y)\mathrm{d}A = \boldsymbol{f}^H \cdot \int_B \mathbf{h}(x,y)\mathrm{d}A. \tag{3}$$

Suppose we wish to compare $\boldsymbol{f}^G$ and $\boldsymbol{f}^H$ over an entire domain? If we have a set of regions $A = \{A_1,\ldots,A_n\}$ tiling that domain, then we can say the two fields are *equivalent on* A if they are equivalent on all of $A_1,\ldots,A_n$. Note that A could be an L0 grid, or simply a set of regions on the domain. If we have a regridding transformation $\mathcal{F}()$ such that $\boldsymbol{f}^H = \mathcal{F}(\boldsymbol{f}^G)$, then we say that $\mathcal{F}()$ is *conservative on* A if $\boldsymbol{f}^G \equiv^A \boldsymbol{f}^H$.

4.3 Area-weighted remapping

Suppose we have two grids G and H, with H being L0 parameterized – for example, G might be an ice grid and H an atmosphere grid. Suppose we have a field $\boldsymbol{f}^G$, which we wish to regrid in a conservative manner to result in the field $\boldsymbol{f}^H$. One common way to do this is to compute each component f_i^H based on the value of $f^G(x,y)$ integrated over the area covered by the ith grid cell in the grid H. Or, more formally,

$$f_i^H = \frac{1}{|H_i|}\left(\boldsymbol{f}^G \cdot \int_{H_i} \boldsymbol{g}(x,y)\mathrm{d}A\right). \tag{4}$$

By construction, this transformation is conservative on H. Note that $\boldsymbol{f}^G$ and $\boldsymbol{f}^H$ will generally *not* be equivalent over other regions other than grid cells in H – for example, over grid cells in G (if G happens to be L0 parameterized).

Area-weighted remapping is closely related to our definition of equivalence above. If we have two fields on two different grids, equivalence on H can be tested by regridding both fields to H and comparing the resulting vectors. The transformation from G to H is linear and thus representable by a matrix. It is variously called *area-weighted remapping* or *conservative regridding* (Ramshaw, 1985). The matrix is computed by integrating the source basis functions $g_1(x,y),\ldots,g_n(x,y)$ over every grid cell in the destination grid.

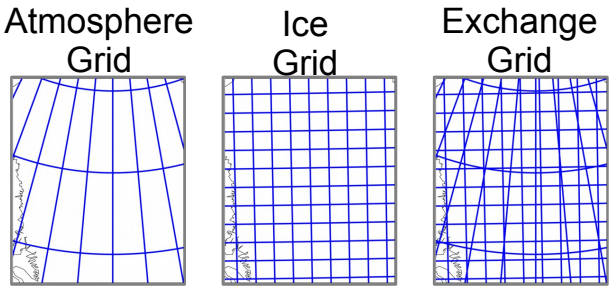

Fig. 5. An exchange grid, obtained by overlapping a sample atmosphere grid and ice grid (not to scale). Each resulting grid cell, irregularly shaped, overlaps with exactly one atmosphere grid cell and exactly one ice grid cell.

4.4 Interpolation grid

We are now almost ready to discuss our implementations of the transformations $E \to I$ and $E \to A$ (see Appendix A and Fig. 3 for mathematical notation). But first, we must introduce the *interpolation grid* G, which is used to rigorously define basis functions on the elevation grid E. The user has a choice of how G is to be chosen.

In general, G is chosen simply as the ice grid I. If I is L0-parameterized, we might instead choose G to be the *exchange grid* between A and I (Fig. 5): the L0 grid whose grid cell outlines are formed by the intersection of grid cells in A and I (Balaji et al., 2006). The exchange grid is a useful choice for G because every exchange grid cell overlaps at most one atmosphere grid cell. This choice has its pros and cons, which we explore in Sect. 9. Either way, the interpolation grid G is similar to the ice grid I and can be thought of as an ice grid proxy in most cases.

5 Projection issues

Whether the source grid is L0 or L1, area-weighted remapping algorithms need to find the intersection of grid cell outlines from two grids. Technically, this is only possible if the two grids exist on the same surface. In our case, atmosphere models exist on the surface of a sphere, whereas ice flow models work on a Cartesian plane. Unless an ice flow model formulated in spherical coordinates is used, the exchange grid between an atmosphere grid and an ice grid cannot be directly computed.

This problem is solved using a map projection (Snyder, 1987). We let A' be the original atmosphere grid and then let A be the projected atmosphere grid – projected to the Cartesian plane using the chosen projection. Regridding computations described in this paper are made to/from A, the projected grid.

In general, the area of a grid cell can change when it is projected. Changes in area due to the use of a projection that does not preserve areas are called *projection error* (Lauritzen and Nair, 2008). Even if an equal area projection is used, in practice grid cells still experience small changes in area. This is because projected grid cells have complex shapes, but the algorithms in Appendix D are only able to approximate these shapes with polygons. Changes in area due to polygonal approximation of curves, or numerical artifacts in the methods used, are called *geometric error* (Lauritzen and Nair, 2008).

Whether a change in grid cell area arises from projection or geometric error, data must be rescaled to maintain conservation when transforming a field $\boldsymbol{f}^{A'}$ to $\boldsymbol{f}^{A}$:

$$f_i^A = \frac{|A_i'|}{|A_i|} f_i^{A'}. \tag{5}$$

Note that this transformation is diagonal and invertible. Because this is simply a rescaling, the regridding schemes presented in this paper are not operationally affected by projection issues.

Although this rescaling scheme may be used to address any change in grid cell area, it is preferable to eliminate or minimize these changes to begin with. To this end, we consider projection and geometric error separately.

5.1 Projection error

The stereographic projection used in SeaRISE (Bindschadler et al., 2013) can change the area of grid cells by up to 10 %. If a projected grid cell is 10 % smaller then the original, then that means that 1 m of accumulation in the GCM will turn into 1.1 m in the ice flow model. This could cause significant discrepancies in dynamic ice flow. Because the projection in SeaRISE is constructed to minimize errors in a band along the 71° N parallel, we would not expect the magnitude of projection error to change significantly if an oblique stereographic projection were used instead.

One solution to the problem of projection error would be to use an equal area projection – the Lambert equal area projection, for example. However, equal area projections do not preserve angles, which distorts the ice dynamics at a local scale. We conclude that nonphysical distortions happen whether an area-preserving or angle-preserving projection is used. It is not yet clear which choice gives better results in the end.

One innovative approach to this problem is to use an angle-preserving projection, but to allow ice flow model grid cells to vary in size, based on the local distortion of space introduced by the projection. Parameters dx and dy are kept for each grid cell. These grid parameters are included in the ice flow model equations, thereby accounting for shape and area distortion caused by the projection in a physically meaningful way (Pollard and DeConto, 2012).

5.2 Geometric error

Geometric errors are typically three orders of magnitude smaller than projection errors. In theory, they may be

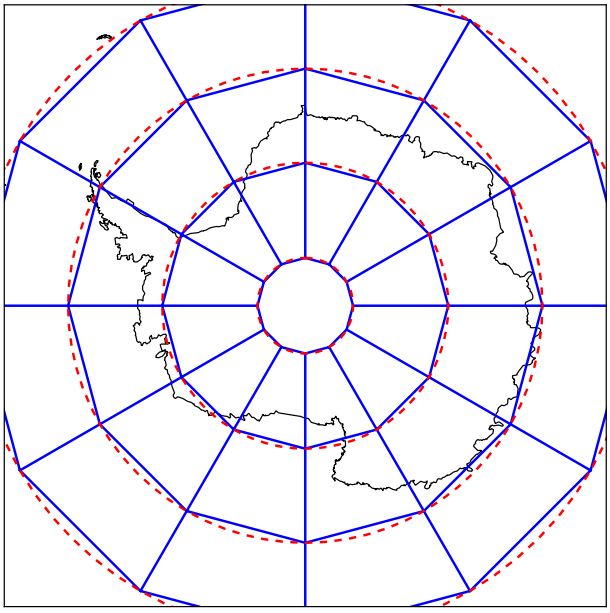

Fig. 6. An example of geometric error, in which grid cells change size due to polygonal approximation. This map, made using a Lambert equal area projection centered on the North Pole, shows a set of latitude–longitude grid cells – a kind commonly used in atmosphere models. Solid blue lines show polygonal approximations, while dotted red shows the actual grid cells on the sphere. Note that all grid cells (in this local map) shrink when approximated.

arbitrarily reduced further by increasing the number of sides used in the approximating polygons – the parameter n specifies the number of line segments used to approximate each side of the original grid cell in its projected state. But this method has its limit in practical terms.

Consider a typical latitude–longitude grid on the sphere with a Lambert equal area projection centered at the North Pole (Fig. 6). In this case, the area of a polygonal approximation will always be *smaller* than the area of the actual grid cell: we end up inscribing polygons inside of circles. The geometric error will depend on the number of sides of the inscribed polygon, which in this case depends on the number of grid cells in the circle, and on n.

This is the method used by Archimedes to approximate the value of π in ~ 250 BCE (Heath and Archimedes, 1897, p. 91). Unfortunately, it converges only quadratically, as $O(1/n^2)$. Meanwhile, memory use to store all those polygons goes up by $O(n)$. Memory and time requirements will therefore be exponential in terms of the number of digits of accuracy required: each additional digit of accuracy will require an increase in the number of sides by a factor of $\sqrt{10}$. It is therefore not practical to make geometric error arbitrarily small by increasing n. In our experience, a value of $n = 2$ yields geometric error of approximately 10^{-4}. We believe $n = 2$ offers an acceptable trade-off between efficiency and accuracy.

6 Transformation: elevation to ice/interpolation grid

We can now derive the first of the five transformations posited in Fig. 2. We begin with the transformation $E \rightarrow G$ from the elevation grid to interpolation grid, because this is the transformation that ice modelers are most concerned with when seeking high-quality downscaled flux fields. In various contexts, this transformation might be referred to as an *interpolation*, *downscaling* or *regridding* operation.

In Sect. 3, we showed how values $\boldsymbol{f}^E$ at elevation points may be used to construct a flux–elevation relationship $f_i(z)$ in each atmosphere grid cell A_i. These relationships may then be used to construct $\boldsymbol{f}^G$ on the interpolation grid. The resulting transformation for $E \rightarrow G$ potentially involves horizontal and vertical interpolation, of which the modeler has considerable choice.

We present here three methods of interpolation from E to G: z interpolation (Sect. 6.1), bilinear interpolation (Sect. 6.2) and elevation class interpolation (Sect. 6.3). All three methods assume that G is L0 parameterized and can be used for the basis of a conservative coupling. And since they are all linear, they can all be represented by a (sparse) matrix, which we will call $\mathbf{M}$. The methods are extended to the L1 case in Sect. 6.4

6.1 Z interpolation

Z interpolation constructs a flux field $\boldsymbol{f}^G$ on the interpolation grid through direct application of the flux–elevation relationship $f_i(z)$ derived in Sect. 3 for atmosphere grid cell A_i. Supposing G is L0 parameterized, consider an interpolation grid cell G_j, wholly contained in one atmosphere grid cell A_i, with mean elevation z_j. We would set the value of that grid cell to $f_j^G = f_i(z_j)$. If G_j intersects more than one atmosphere grid cell, the same procedure is followed for each atmosphere grid cell A_i it intersects – and the results are summed together, area-weighted by $|G_j \cap A_i|$.

We call this *z interpolation*. As long as standard interpolation methods are used to construct $f_i(z)$, z interpolation defines a linear function from $\boldsymbol{f}^E$ to $\boldsymbol{f}^G$ and can be represented by a matrix. An example of the result is shown in Fig. 7a. In general, z interpolation produces SMB fields that vary smoothly within each atmosphere grid cell but that contain discontinuities between grid cells.

6.2 Bilinear interpolation

There is some concern that discontinuities created by z interpolation could cause problems as an input to an ice flow model. In that case, a bilinear interpolation step may be used to create a smooth field, as in Lipscomb et al. (2013). This scheme sets the value of interpolation grid cell G_j equal to a linear combination of $f_i(z)$, $f_k(z)$, $f_l(z)$ and $f_m(z)$, where atmosphere grid cells A_i, A_k, A_l and A_m are the four grid

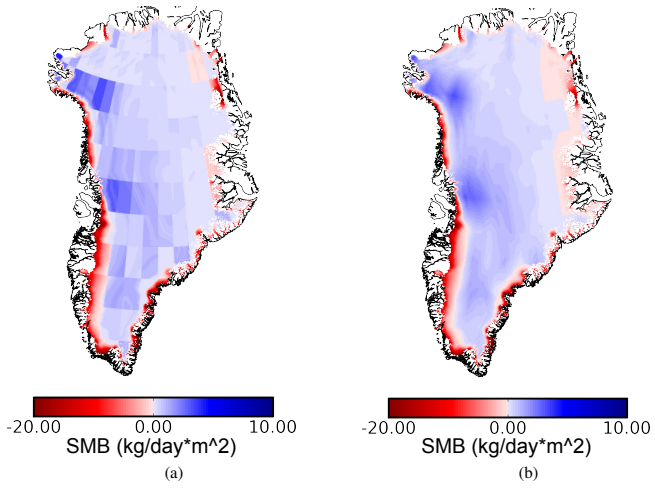

Fig. 7. Results of **(a)** z interpolation and **(b)** bilinear interpolation, to develop a downscaled field on the ice grid from an elevation point field. Bilinear interpolation eliminates the discontinuities present at atmosphere grid cell boundaries. However, total SMB is changed, particularly for localized events such as snowstorms, for example in the northwest. This could introduce biases in the long-term evolution of the ice sheet, even though both schemes can be made to conserve mass and energy. This figure is for demonstration purposes only. See Sects. 10 and 11 for thorough regridding examples.

cells with centers closest to the center of G_j (Fig. 8). Results are shown in Fig. 7b.

Bilinear interpolation has an advantage over z interpolation in that it produces smooth fields. However, bilinear interpolation presents a number of problems:

- The fields it produces will have a significantly different total mass than the fields produced by z interpolation. Our experience with monthly SMB fields over Greenland indicates that storms tend to leave large amounts of snow in a few localized areas. Bilinear interpolation tends to reduce the total amount of snowfall in these cases, causing potentially significant differences in the GCM model run.

  Variations in total mass caused by the choice of interpolation procedure will *not* cause conservation problems: we make this clear in Sect. 7. However, an interpolation scheme that produces a significantly different mass from the apparent "intent" of the GCM could cause biases. In particular, a scheme that makes snowstorms smaller than the GCM "intended" would produce a negative bias in the equilibrium extent of the ice sheet.

- It introduces significant numerical diffusion into the system.

- The $A \rightarrow E$ transformation derived with bilinear interpolation can introduce nonphysical artifacts (Sect. 12).

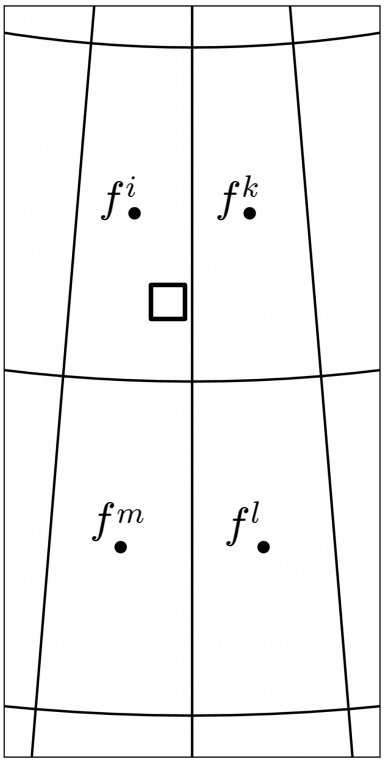

Fig. 8. Setup for bilinear interpolation. The value in an ice grid cell (square) will be the sum of the values at the four nearest atmosphere grid cell centers, weighted by Δ-longitude and Δ-latitude along the axes.

- It is also not always clear how to extend bilinear interpolation to the case of non-rectangular atmosphere grids. This problem can also affect non-regular points in mostly regular grids (e.g., at the poles of a latitude–longitude grid).

6.3 Elevation class interpolation

One final choice for regridding is to define elevation *classes* as they were originally formulated (Leung and Ghan, 1998). In this approach, each atmosphere grid cell is grouped into sub-regions based on elevation bands. For example, an atmosphere grid cell near the coast spanning elevations of 0 m–600 m might be grouped into sub-regions of 0 m–200 m, 200 m–400 m and 400 m–600 m. The grid E can then be thought of as having irregularly shaped grid cells formed by the intersection of atmosphere grid cell boundaries and elevation contours (Fig. 9). Area-weighted remapping is used to interpolate from the elevation grid to the atmosphere grid.

Elevation class interpolation is equivalent to a specialized form of z interpolation, in which the 1-D function $f_i(z)$ (for grid cell A_i) is interpolated as a discontinuous piecewise constant function (Fig. 10), rather than a piecewise polynomial (Fig. 4). Unless one expects significant discontinuities in topography or the elevation–flux relationship, such a

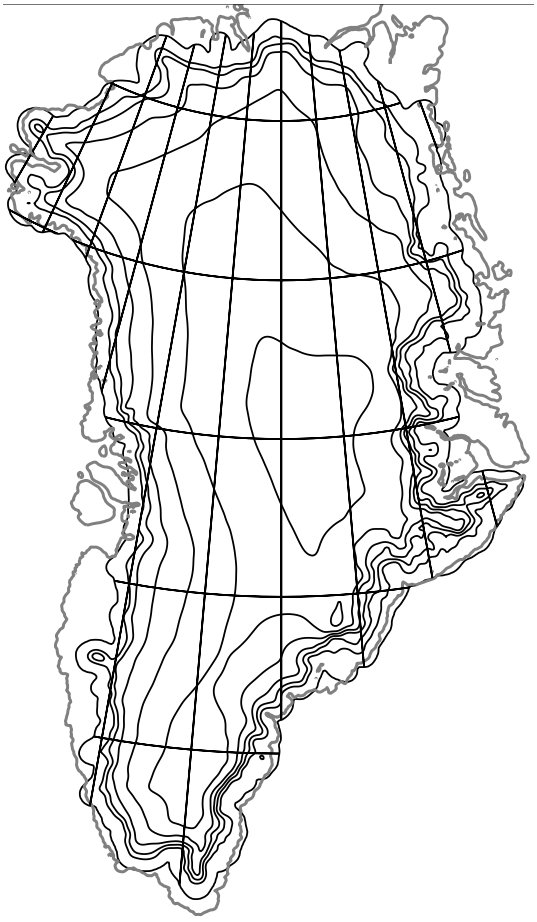

Fig. 9. Traditional elevation class schemes are equivalent to running the ice surface model on an L0 grid, where grid cell outlines are created by the intersection of the atmosphere grid and elevation contours. One such grid is shown in this figure. Note that grid cells extend only as far as the ice sheet; the grey line shows the Greenland coast.

zero-order interpolation scheme would be expected to give a less accurate version of $\boldsymbol{f}^G$ than a higher order scheme such as z interpolation. Since we expect the elevation–flux relationship to be smooth with elevation, we recommend the use of z interpolation over elevation class interpolation.

6.4 Interpolating to L1 grids

We have outlined methods for the interpolation $E \to G$, as long as G uses an L0 parameterization. The procedures need to be modified for interpolation or ice grids using L1 parameterization. In a finite element mesh ("grid"), field values are determined at mesh vertices and linearly interpolated within each triangular element (Zienkiewicz et al., 2013). Any of the interpolation methods above may be used to determine the value of $f^G(x, y)$ at each vertex. Once vertex values have been interpolated on the vertices of a finite element mesh, the value of $f^G(x, y)$ at all other points is fully determined.

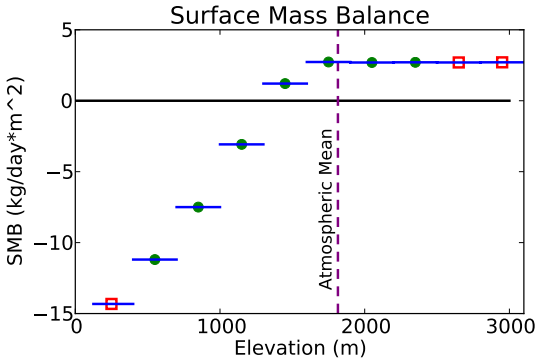

Fig. 10. Interpolated SMB function within one atmosphere grid cell when using elevation classes. The resulting piecewise constant interpolation is almost never preferable to the piecewise linear interpolation in Fig. 4. Traditional elevation class schemes are discouraged because they offer no benefit over first-order z interpolation.

7 Transformation: elevation to atmosphere grid

As shown in Fig. 2, a regrid step from the elevation to the atmosphere grid is required on every GCM time step. Once the modeler has chosen the transformation $E \to G$, we show here how to derive a transformation for $E \to A$ that is consistent with it.

To derive this transformation, consider one GCM time step, during which a flux field $\boldsymbol{f}^E$ between the atmosphere and ice surface is computed on the elevation grid. That field will be regridded both to the interpolation and atmosphere grids. Conservation requires that the resulting fields are equivalent on A, i.e,

$$\boldsymbol{f}^A \equiv^A \boldsymbol{f}^G. \tag{6}$$

If the transformation $E \to G$ derived in Sect. 6 is represented by the matrix $\mathbf{M}$, then we can write $\boldsymbol{f}^G = \mathbf{M}\boldsymbol{f}^E$. Substituting into Eq. (6), we get the requirement

$$\boldsymbol{f}^A \equiv^A \mathbf{M}\boldsymbol{f}^E. \tag{7}$$

Equivalence on A can be tested by conservatively remapping both sides to A and then comparing (in this case, the left-hand side is already on A, so no remapping is required there). Denoting $\mathbf{R}$ as the matrix that conservatively regrids from G to A, we have the requirement

$$\boldsymbol{f}^A = \mathbf{R}\mathbf{M}\boldsymbol{f}^E. \tag{8}$$

If we treat this as a definition for the transformation $E \to A$, we have derived a transformation for $E \to A$ that is consistent with the transformation we already chose for $E \to G$.

In other words (Fig. 11): we will regrid from E to A by first regridding from E to G (represented by the matrix $\mathbf{M}$) and then use area-weighted remapping from G to A (represented by the matrix $\mathbf{R}$). The GCM, which needs to use the transformation $E \to A$, does not need to know anything about the ice grid used to derive that transformation. It

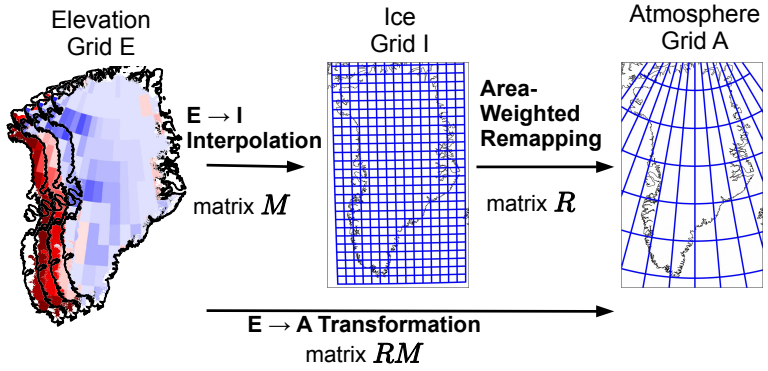

Fig. 11. The transformation from elevation grid to atmosphere grid is derived by first interpolating to the ice grid (represented by matrix **M**) and then using a conservative area-weighted remapping step to the atmosphere grid (matrix **R**). The two steps may be combined by computing the matrix product **RM**. This construction ensures that a quantity computed on the elevation grid will have the same total mass when regridded to the ice grid or atmosphere grid. The two transformations from the elevation grid to the ice and atmosphere grids are said to be *consistent*.

just needs to know the final matrix **RM**, which can be precomputed via matrix multiplication.

We have derived a transformation for $E \to A$ that *by definition* is consistent with a previously chosen transformation for $E \to G$. Our approach differs from previous efforts, which would start out with $E \to A$ and try to find a transformation for $E \to G$ that is consistent with it.

7.1 A basis for the elevation grid

Every vector space has basis functions, including the elevation grid E. Our choice for the transformation $E \to A$ constrains the basis functions we use on E. To find these basis functions, we expand on the central idea in the previous section, i.e., that we can determine properties of a field on the elevation grid by regridding to the ice grid. In this case, we define $f^E(x, y)$ to be equal to $f^G(x, y)$, where $\boldsymbol{f}^G = \mathbf{M}\boldsymbol{f}^E$. In other words, we can evaluate $f^E(x, y)$ at a point by interpolating $\boldsymbol{f}^E$ to G and then evaluating $f^G(x, y)$. This definition is consistent with our method for $\boldsymbol{f}^A$ in the previous section.

We now use this principle to obtain a formula for the basis functions of E. Substituting into Eq. (1), we get

$$f^E(x, y) = f^G(x, y) = \mathbf{M}\boldsymbol{f}^E \cdot \boldsymbol{g}(x, y). \tag{9}$$

Rewriting in indicial notation, using the commutative property of multiplication, and swapping index letters, we get

$$f^E(x, y) = f_i^E M_{ji} g_j(x, y). \tag{10}$$

We can use this, along with Eq. (1), to determine the basis functions for the elevation grid:

$$e_i(x, y) = M_{ji} g_j(x, y) \tag{11}$$

or, switching back to vector notation,

$$\boldsymbol{e}(x, y) = \mathbf{M}^T \boldsymbol{g}(x, y). \tag{12}$$

Equivalently, the basis function $e_i(x, y)$ is the function obtained if we set the $f_i^E = 1$ and all other components of $\boldsymbol{f}^E = 0$, regrid to the interpolation grid, and then examine the resulting function $f^G(x, y)$.

In general, these basis functions will be not orthogonal. Their exact form depends on choices made in choosing **M**, i.e., vertical and horizontal interpolation choices, as well as grid geometry issues. We have plotted some example basis functions in Fig. 12.

Note that if elevation class interpolation is used (Sect. 6.3) and G is the exchange grid, then our scheme reduces to a traditional elevation class scheme, and basis functions will represent constant-value sub-grid tiles, which are orthogonal.

7.2 Forward transformations: summary

We have now defined three of the five regridding transformations required by Fig. 2: $E \to G$, $G \to A$ and $E \to A$. We call these the *forward transformations* because they are linear and can be represented by matrices. See Fig. 13 for a diagram of how these transformations may be used to regrid fields.

We defined these three transformations in a consistent manner. We began by allowing the user choice in constructing the transformation from the elevation to interpolation grids, represented by matrix **M**: this makes sense because users desire choice for $E \to G$. We then noted that standard area-weighted remapping, represented by matrix **R**, may be used to regrid from the interpolation grid to the atmosphere grid. From these two choices, transitivity requires that the transformation from the elevation to atmosphere grid is represented by the matrix **RM**. Conservation is maintained because $E \to G$ and $E \to A$ are consistent with each other, producing $\boldsymbol{f}^G$ and $\boldsymbol{f}^A$ that are equivalent on every grid cell in A.

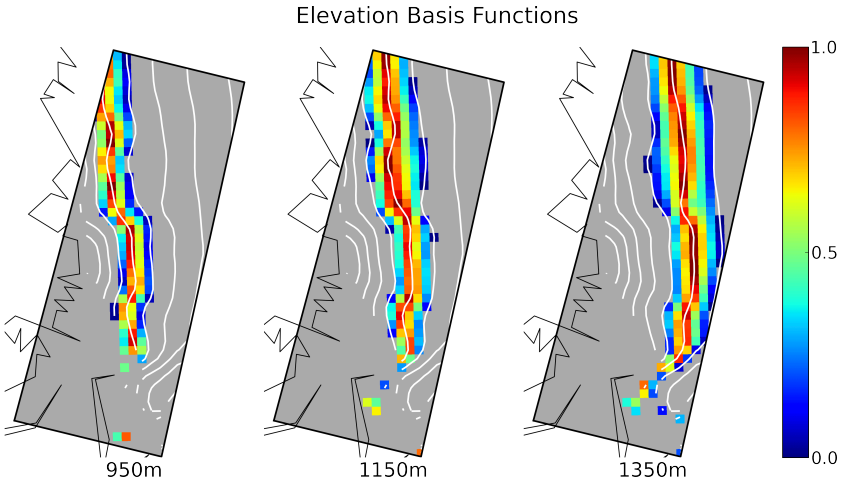

Fig. 12. Unitless basis functions for the elevation grid E, constructed using 20 elevation points and z interpolation (the exchange grid was used as the interpolation grid). The grey box represents one atmosphere grid cell on the west coast of Greenland, with the coastline shown as black lines. White lines in the atmosphere grid cell represent elevation contours corresponding to each elevation point. The basis functions corresponding to elevation points at 950 m, 1150 m and 1350 m are shown. Note that basis functions overlap and are not orthogonal. Because of the z interpolation, each basis function has maximum value at its corresponding elevation, but it has a non-zero support up to one elevation point away.

Our choices for these transformations lead directly to a well-defined set of basis functions for the grid E, examples of which we plotted. Fields on this grid will be represented in terms of these basis functions – for example, surface–atmosphere fluxes and ice surface model state.

8 Reversing the transformations

We have defined three transformations so far, but Fig. 2 indicates that five are necessary for full functioning of the coupled system. We still need to derive appropriate procedures for the "reverse transformations" $A \to E$ and $G \to E$, indicated in Fig. 13 as dotted lines. Because of the differences in dimensionality between the three grids (Sect. 2.3), **M** and **RM** are not invertible in a simple linear algebraic sense: $A \to E$ is underdetermined and $G \to E$ is overdetermined. We must still find ways to compute these transformations that retain conservation over atmosphere grid cells.

8.1 Transformation: atmosphere to elevation grid

Suppose we have a flux field $\boldsymbol{f}^A$ on the atmosphere grid A. We wish to regrid it to an "equivalent" flux field $\boldsymbol{f}^E$ on the elevation grid E. In this case, $\boldsymbol{f}$ might represent precipitation or downwelling radiation from the atmosphere.

The problem is underdetermined. Any solution for which $\boldsymbol{f}^A \equiv^A \boldsymbol{f}^E$ will be conservative. This is the same as requiring that

$$\mathbf{RM}\boldsymbol{f}^E = \boldsymbol{f}^A. \tag{13}$$

Because of the many-to-one relationship between elevation and atmosphere grid cells, our intuition tells us that $\boldsymbol{f}^E$ may be constructed simply by repeating values of $\boldsymbol{f}^A$ within each atmosphere grid cell. This is physically self-consistent. More precisely, if $E_j \in A_i$, then we would like to set $f_j^E = f_i^A$. We define a simple linear transformation Λ that does this: $\boldsymbol{f}^E = \Lambda \boldsymbol{f}^A$.

Surprisingly, this definition for $A \to E$ is only conservative in some cases. If **RM** computes each f_i^A as a weighted sum *only* of values f_j^E where $E_j \in A_i$, then we say that **RM** is a *local transformation* – it uses only data from "within" an atmosphere grid cell to compute a value on that grid cell. In that case, it is easy to prove conservation by showing that $\mathbf{RM}\Lambda\boldsymbol{f}^A = \boldsymbol{f}^A$ (see Eq. 13). The weights involved in **RM** do not matter, as long as they sum to 1 for each component of $\boldsymbol{f}^A$.

Therefore, if **RM** is local, we can simply use Λ for our transformation $A \to E$. Even better, it is easy to show that there will be no numerical dispersion in the round-trip transformation $E \to A \to E$ because $\Lambda\mathbf{RM}$ is the identity matrix. The lack of dispersion on this round-trip is important because it is computed every atmosphere time step.

8.2 Nonlocal RM

If **RM** is not local, then in general $\Lambda\boldsymbol{f}^A$ and $\boldsymbol{f}^A$ will not be equivalent on A or even the entire ice sheet. We are forced to trade off between the most physically self-consistent value for $\boldsymbol{f}^E = \Lambda\boldsymbol{f}^A$ vs. something that conserves. We can use Λ, along with quadratic optimization, to guide us to such a compromise.

We seek a vector $\boldsymbol{f}^E$ such that

$$\mathbf{RM}\boldsymbol{f}^E = \boldsymbol{f}^A. \tag{14}$$

We also wish that vector to be as close as possible to our intuition above. That is, we wish to minimize the quantity

$$||\boldsymbol{f}^E - \Lambda \boldsymbol{f}^A||^2, \tag{15}$$

where our L_2 norm is weighted by the weight of each basis function. That is, we use a weight vector **w** where

$$w_i = \int_{\infty} e_i(x, y)\mathrm{d}A. \tag{16}$$

This is a sparse quadratic optimization problem with equality constraints. A number of numerical packages can solve it; we used GALAHAD (Gould et al., 2003). In our tests, solution typically requires a fraction of a second on a single core. This is so fast that we have found no need to consider suboptimal solutions to this problem.

This procedure introduces some numerical dispersion into the fully coupled system from nonlocal regridding operations; by numerical dispersion, we mean movement of mass between adjacent atmosphere grid cells, thereby violating our desired property of conservation with each atmosphere grid cell. Considering the round-trip transformation $E \rightarrow A \rightarrow E$, it is clear that this transformation is not local, both because **RM** is not local, and because the quadratic program set up for $A \rightarrow E$ will be nonlocal.

8.3 Practical issues for nonlocal RM

GCMs are well equipped to deal with *sub-grid tiles*, each one occupying a fraction of the overall grid cell. The GCM will typically implement $E \rightarrow A$ by computing a weighted sum over sub-grid tiles in each grid cell, with weights based on each tile's fractional area. This capability is used to implement traditional elevation class schemes.

If the **RM** matrix is local, then by definition the value on each atmosphere grid cell is a weighted sum of the elevation points in that grid cell. This is compatible with existing GCM practice that assumes sub-grid tiles. To use the methods in this paper, the computation of $E \rightarrow A$ inside the GCM does not need to be replaced, it only needs to be fed a new set of weights. Even though elevation points do not have well-defined areas, the weight for elevation point j can be thought of as the "fractional area" that an elevation point contributes to its containing atmosphere grid cell. The GCM can be coded as if traditional elevation classes were being used, even if the user has chosen a more numerically accurate form of vertical interpolation in the construction of the **M** matrix.

Things get more complicated for the GCM if the **RM** matrix is *not* local. Instead of using a simple set of weights, the GCM will have to be multiplied by a general sparse matrix **RM** when computing $E \rightarrow A$. An MPI (Message Passing Interface) gather will be required to compute $A \rightarrow E$.

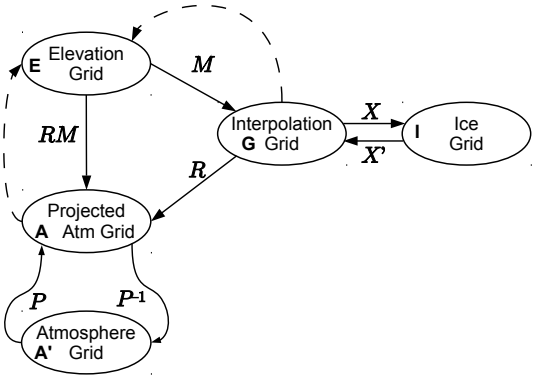

Fig. 13. The five grids used in the coupling problem, and transformations between them. The (linear) interpolation step **M**, from E to G, is chosen by the user. The (linear) transformation **R**, from G to A, is an area-weighted remapping step. These two fully constrain the transformation from E to A as **RM**. If I is L0-parameterized, then G is the exchange grid between A and I, and **X** and $\mathbf{X}'$ are area-weighted remapping transformations. If I is L1-parameterized, then G is equivalent to I, making **X** and $\mathbf{X}'$ the identity. P is a diagonal rescaling operation between A' and A. Reverse transformations required by the coupled system are shown as dotted lines.

Finally, some GCMs use implicit schemes for ice surface–atmosphere coupling (Best et al., 2004), requiring a matrix inversion on every time step. If **RM** is nonlocal, then a large, sparse, global matrix inversion will be required, rather than a number of small, local inversions.

Because of the significant practical complications that arise with the use of a nonlocal **RM** matrix, most users find it simplest to use a local **RM** matrix if at all possible.

8.4 When is RM not local?

The use of a nonlocal **RM** transformation can be an added burden to the GCM developer. It is therefore important to delineate cases in which **RM** is not local.

For L1 ice grids, as used with a finite element ice flow model, **RM** will be nonlocal: ice elements that straddle two atmosphere grid cells will by necessity involve elevation point values from two atmosphere grid cells. This will make **R** slightly nonlocal.

Even for L0 grids, **M** could be nonlocal, depending on the choice of interpolation schemes for $E \rightarrow G$. Bilinear interpolation is inherently nonlocal, whereas the other interpolation strategies mentioned above are local.

For L0 ice grids with a local interpolation scheme, **RM** can be made local, as long as G was chosen to be the exchange grid (see Sect. 4.4). In this case, **M** will be local – because each exchange grid cell attains its value from elevation points in just one atmosphere grid cell. **R** will be local as well, for the same reason.

8.5 Transformation: ice/interpolation to elevation grid

Suppose we have a flux field $\boldsymbol{f}^G$ on the interpolation grid G. We wish to regrid it to an "equivalent" flux field $\boldsymbol{f}^E$ on the elevation grid E. As before, we will set up a quadratic optimization problem.

It would be nice if we could find $\boldsymbol{f}^E$ such that $\mathbf{M}\boldsymbol{f}^E = \boldsymbol{f}^G$. This will not usually be possible, since G has far more degrees of freedom than E. Instead, we will minimize the quantity

$$||\mathbf{M}\boldsymbol{f}^E - \boldsymbol{f}^G||^2 \tag{17}$$

while we maintain conservation on each atmosphere grid cell (where $\mathbf{R}$ defines area-weighted remapping from G to A; see Fig. 13):

$$\mathbf{RM}\boldsymbol{f}^E = \mathbf{R}\boldsymbol{f}^G. \tag{18}$$

This quadratic optimization problem may be solved with the same methods as in Sect. 8.2. We weight components of our L_2 norm by the integral of each basis function in G.

9 Exchange grid or ice grid?

So far, we have defined our transformations in terms of the *interpolation grid* G – which could be either the ice grid or the exchange grid. But real ice flow models operate on the ice grid, and transformations must ultimately transform to/from that grid. If we have chosen G as the exchange grid, we need to extend our transformations above to regrid to/from the ice grid. We do this by using area-weighted remapping as necessary between G and I.

More formally, we construct transformations for $\mathbf{X} : G \to I$ and $\mathbf{X}' : I \to G$ using area-weighted remapping. We can then represent $E \to I$ as the matrix product $\mathbf{XM}$. Similarly, we can construct $I \to E$ by first computing $\boldsymbol{f}^G = \mathbf{X}'\boldsymbol{f}^I$ and then using the reverse transformation $G \to E$ to obtain $\boldsymbol{f}^E$ (see Fig. 13).

Note that the transformation $\mathbf{X} : G \to I$ is not conservative on A. For this reason, the transformation $E \to I$ represented by $\mathbf{XM}$ is not conservative on A either, although it is conservative overall: "mass" lost from one atmosphere grid cell will be gained by neighboring cells. This will cause numerical dispersion when these transformations are used in the fully coupled system (Fig. 2).

Assuming the ice grid is L0 parameterized, we will now address the practical issue faced by the user, namely whether to choose the exchange grid or ice grid as G. If one uses the exchange grid for G in conjunction with a local vertical interpolation scheme, then the $\mathbf{RM}$ matrix will be local. This has a number of advantages. It eases implementation (Sect. 8.3). There will be no numerical dispersion for the transformation $E \to A$, used every atmosphere time step. The reverse transformation $A \to E$, also used every atmosphere time step, will be trivial. However, there *will* be numerical dispersion for the transformation $E \to I$, which is used once every ice coupling time step.

If one directly uses the ice grid for G, then the $\mathbf{RM}$ matrix will *not* be local, producing numerical dispersion for the transformation $E \to A$ and complicating the reverse transformation $A \to E$. However, there will be no numerical dispersion for the transformation $E \to I$.

On balance, we recommend keeping the $\mathbf{RM}$ matrix local if possible. Not only does this simplify implementation, it moves unavoidable numerical dispersion away from the atmosphere time step to the less frequent ice coupling time step.

9.1 Exchange grids for finite elements

The discussion in Sect. 9 assumes an L0 ice grid. The results can be extended to L1 or higher order parameterizations: finite element ice models, for example. In this case, the concept of exchange grid is not meaningful, since exchange grids are by definition L0.

Instead of the exchange grid, one can choose G to be just about *any* L0 grid, of resolution at least as high as the ice mesh, where grid cells do not cross atmosphere cell boundaries: we will call this a *generalized exchange grid*. The user would then need to develop an appropriate conservative transformation for $G \to I$, whereas the transformation for $I \to G$ would remain an area-weighted remapping (see Appendix C). Details of this scheme for a particular ice mesh parameterization are left to the reader.

As in Sect. 9, the transformation $G \to I$ would cause some numerical dispersion, due to the geometry of the ice mesh. Additional dispersion would be introduced because of mismatch between the L0 grid G and the higher order mesh I – although this could be reduced by making G finer. The trade-offs of choosing G to be a generalized exchange grid vs. the ice grid are the same as in Sect. 9: numerical dispersion is introduced in the ice coupling time step, in exchange for simplifying implementation and eliminating dispersion in the atmosphere time step.

10 Regridding examples: local RM

Having shown how to compute all five required regridding transformations, we now demonstrate them working within a realistic GCM context. We set up a test using the GISS $2° \times 2\frac{1}{2}°$ atmosphere grid (Schmidt et al., 2006), overlapping with the SeaRISE 5 km L0 grid (Bindschadler et al., 2013) and 40 elevation points, spaced every 100 m from 0 m to 4000 m. We chose the exchange grid as G (see Appendix A and Fig. 3 for notation conventions). We ran GISS ModelE in this configuration, using fixed sea surface temperatures corresponding to the years 1996–2005. The ice surface model

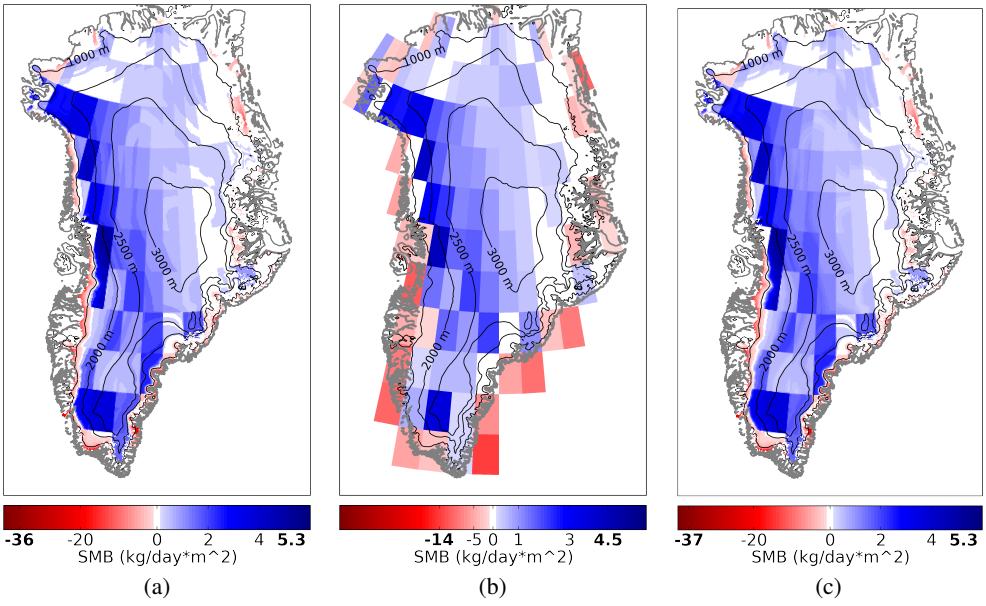

Fig. 14. July SMB computed by ModelE on the elevation grid E **(a)**, and regridded to the atmosphere grid A via the exchange grid **(b)** or to the ice grid I using z interpolation **(c)**. For plotting purposes, each elevation point in **(a)** is assigned to a nearby region of similar elevation; the value of the elevation point is then plotted in its corresponding region. Bold numbers on the color scale indicate the extreme values of the plot. Elevation contours are plotted.

was run on the elevation grid, producing monthly averages of SMB over Greenland, which we label $\boldsymbol{f}^E$ (Fig. 14a).

As expected, the broad pattern shows strong melting in narrow bands at low elevations, along with weak accumulation at high elevations. Atmosphere grid cell boundaries are prominent because precipitation – the primary source of positive SMB – is not downscaled to sub-grid resolution (Leung and Ghan, 1998). The small oscillations at high elevation are artifacts introduced by ModelE's ice surface model.

10.1 Example: elevation to atmosphere grid

Figure 14b shows the original field $\boldsymbol{f}^E$ computed on the elevation grid and regridded to the atmosphere grid A, via the transformation $\boldsymbol{f}^A = \mathbf{RM}\boldsymbol{f}^E$. Note how low-elevation ablation regions become broader and weaker (compared to Fig. 14a), whereas the interior of the ice sheet remains about the same. This demonstrates the benefits obtained through the use of elevation points, as compared with running the ice surface model on the atmosphere grid (van den Broeke et al., 2008).

By definition, the transformation **RM** is conservative on A: we can evaluate conservation properties of other transformations by comparing to Fig. 14b.

10.2 Example: elevation to ice grid

Figure 14c shows the original field $\boldsymbol{f}^E$ regridded to I, via $\boldsymbol{f}^I = \mathbf{XM}\boldsymbol{f}^E$ using z interpolation for **M**. This plot looks like a smoothed version of Fig. 14a. Atmosphere grid cell boundaries are still visible because z interpolation is local.

The transformation $\mathbf{X} : G \to I$ (Fig. 13) introduces a small amount of nonlocality. And although it is conservative over the ice sheet in general, it is not conservative over A. This can be seen (Fig. 15) by regridding $\boldsymbol{f}^I$ to A and comparing with the results of Fig. 14b. This plot quantifies the amount of numerical dispersion the simulation will encounter every month when preparing SMB input for the ice model. This dispersion is caused by ice grid cells that overlap more than one atmosphere grid cell.

In most areas, numerical dispersion is low, less than 1 %. That is because most atmosphere grid cells are overlapped by many ice grid cells, with only a few lying on an atmosphere grid cell boundary. However, numerical dispersion can be significant for atmosphere grid cells that just nick the edge of the ice sheet, where a high proportion of their overlapping ice grid cells also overlap with a neighboring atmosphere grid cell.

The numerical dispersion encountered here is inherent in the regridding problem itself, rather than our approach to that problem. Short of using an ice grid whose grid cells do not overlap atmosphere grid cells, we see no obvious way to eliminate this issue. However, we do not believe it to be a serious problem: the total area of ice sheet affected by it will be small.

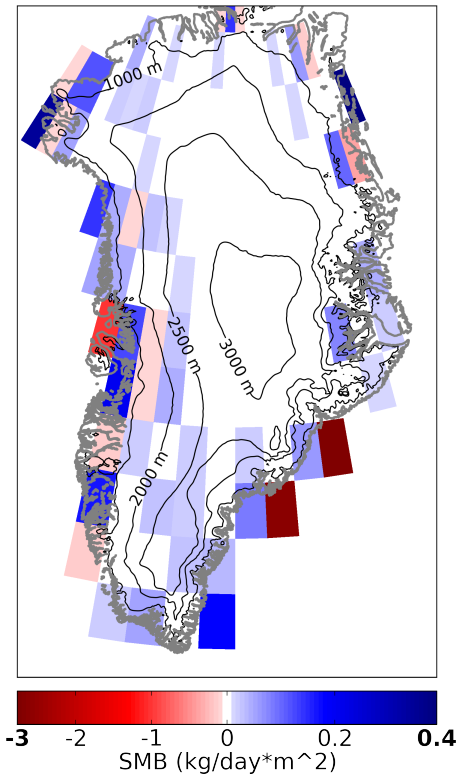

Fig. 15. Difference in July SMB obtained with $E \to A$, using two choices for the interpolation grid G: either the exchange grid or the ice grid. Differences (exchange minus ice) are due to ice grid cells that overlap more than one atmosphere grid cell.

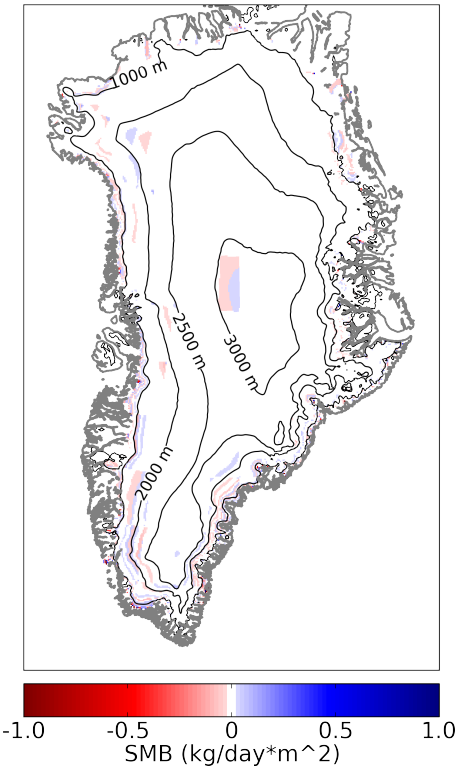

Fig. 16. Example of numerical dispersion caused by an ill-posed quadratic optimization problem. July SMB on elevation grid E was regridded to the exchange grid. The reverse transformation was then used to recover the original SMB on E. Plot shows the difference between the result and the original. In theory, the two should be exactly the same. Although mass is conserved, differences appear because the associated quadratic optimization problem is underdetermined in some areas.

It is important to point out that although $\mathbf{X}'$ is not (quite) conservative over A, it is still conservative over the ice sheet in general. We have verified this numerically in our examples.

10.3 Example: ice to elevation grid

Since we have not yet developed the surface boundary condition described in Sect. 2.1, we do not have realistic fields on I to try regridding to E. We will therefore test $I \to E$ at this point using $\boldsymbol{f}^I = \mathbf{XM}\boldsymbol{f}^E$ as input.

We test the transformation in two steps. First, we test $G \to E$ using $\boldsymbol{f}^G = \mathbf{M}\boldsymbol{f}^E$ as input. Then, we test $I \to E$ using $\boldsymbol{f}^I = \mathbf{XM}\boldsymbol{f}^E$ as input. This allows us to evaluate the $G \to E$ transformation separately from confounding dispersion factors involved in $G \to I$.

We computed $\boldsymbol{f}'^E = [G \to E](\boldsymbol{f}^G)$ where $\boldsymbol{f}^G = \mathbf{M}\boldsymbol{f}^E$. Theoretically, $\boldsymbol{f}'^E$ and $\boldsymbol{f}^E$ should be equal. The conservation property imposed by the quadratic program held: we found that $\boldsymbol{f}'^E \equiv^A \boldsymbol{f}^G \equiv^A \boldsymbol{f}^E$ to machine precision. However, we found measurable differences between $\boldsymbol{f}'^E$ and $\boldsymbol{f}^E$ (Fig. 16). Although differences are small in most cases, one area has a difference of more than 8 out of 37 mm day^{-1}, resulting in no more than one digit of precision. Differences tend to be greatest in sparsely populated atmosphere grid cells on the edge of the ice sheet. We conclude that the answer to the quadratic program posed in Sect. 8.5 is poorly constrained. A different numerical solver than the one we used might in theory result in a better match between $\boldsymbol{f}'^E$ and $\boldsymbol{f}^E$.

But this is not a problem in practice: the goal of $G \to E$ is to obtain a physically plausible field on E with the correct conservation properties. The preliminary tests presented in this section are consistent with that goal.

The transformation $I \to E$ is constructed by transforming $I \to G \to E$ (Sect. 9). We used this to compute $\boldsymbol{f}'^E = [I \to E](\boldsymbol{f}^I)$ where $\boldsymbol{f}^I = \mathbf{XM}\boldsymbol{f}^E$. The conservation property imposed by the quadratic program held: we found that $\boldsymbol{f}'^E$ was equivalent to $\boldsymbol{f}^I$ on A to machine precision.

However, differences between $\boldsymbol{f}'^E$ and $\boldsymbol{f}^E$ are even greater in this case: compare Fig. 17 to Fig. 16. This is because the transformation $\mathbf{X} : G \to I$ is not conservative on A. The end result of transforming $E \to G \to I \to G \to E$ will produce an $\boldsymbol{f}'^E$ that is significantly different from the

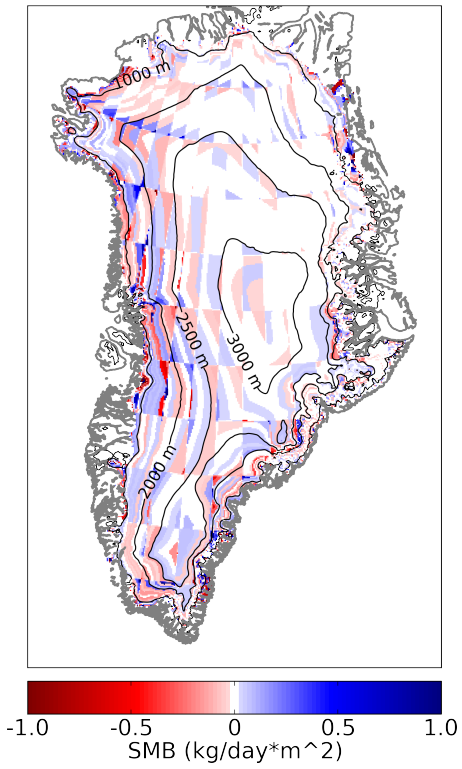

Fig. 17. Example of numerical dispersion in the transformation $I \to E$, caused by ice grid cells that overlap more than one atmosphere grid cell. July SMB on elevation grid E was regridded to the ice grid I. The reverse transformation was then used to recover the original SMB on E. Plot shows the difference between the result and the original. Although mass is conserved, differences are due mainly to ice grid cells that overlap more than one atmosphere grid cell, making it impossible to recover the exact original SMB. The same color scale is used as in Fig. 16.

original $\boldsymbol{f}^E$. Differences are greatest for elevation points near an atmosphere grid cell boundary.

11 Regridding examples: almost local RM matrix

In Sects. 8.3 and 9, we discussed the trade-offs between using the ice vs. exchange grid for the interpolation grid G. Having shown in Sect. 10 an example of our transformations using the exchange grid as G, we now show how things change if the ice grid is chosen for G by re-running the above examples. In this case, the **RM** matrix will be mostly local, except for nonlocality caused by ice grid cells that intersect more than one atmosphere grid cell. We say that **RM** is *almost local*.

In general, differences in results here vs. those in Sect. 10 are insignificant. However, the choice of I as interpolation grid *does* change the basis functions used for E, yielding a system in which the transformation $E \to I$ is now conservative over A.

With a change in $E \to I$, the transformation $E \to A$ is also changed. We calculated the difference between $\boldsymbol{f}^A$ computed using this method vs. $\boldsymbol{f}^A$ computed in Fig. 14b: this difference is exactly the same as Fig. 15. In practice, this difference does not present a real problem: it is impossible to say which version of $\boldsymbol{f}^A$ is more "correct." Both maintain conservation over the ice sheet. As predicted in Sect. 9, grid-related dispersion in $I \to E$ is eliminated. Thus, the grid-related "errors" demonstrated in Fig. 17 are eliminated.

Finally, the use of a nonlocal **RM** matrix requires use of the algorithm described in Sect. 8.1 for $A \to E$. Fig. 18a shows a July precipitation field $\boldsymbol{p}^A$ produced by ModelE, for the same month as the SMB fields above. ModelE currently does not downscale precipitation to sub-grid resolution (Leung and Ghan, 1998): it therefore yields a precipitation field on A.

We then computed $\boldsymbol{p}^E = [E \to A](\boldsymbol{p}^A)$, which could serve as input to the land surface model. Recall that $\Lambda \boldsymbol{p}^A$ is the most physically self-consistent value for $\boldsymbol{p}^E$, but that we choose a different value for $\boldsymbol{p}^E$ in order to maintain conservation.

Figure. 18b shows the difference between the two, $\boldsymbol{p}^E - \Lambda \boldsymbol{p}^A$. Differences are typically in the range $[-0.1, 0.1]$ (about 2 %). They tend to be relatively constant within each atmosphere grid cell and show no discernible pattern between grid cells. In particular, differences are not related to elevation. Any errors introduced by this scheme will be dwarfed by other precipitation errors in the model.

In this example, we used an L0 ice grid to generate a prototypical almost-local **RM** matrix. We expect similar results when using an L1 ice grid because the nonlocality in the L0 case was caused by a small number of ice grid cells that overlap more than one atmosphere grid cell. A similar situation exists with any type of higher order mesh.

We conclude by considering, in the case of an L0 ice grid, whether I or G is a better choice for the interpolation grid. In many cases, the use of a nonlocal **RM** matrix requires significantly more effort in GCM model development. However, our experience shows fewer "surprises" in the transformations when using I as the interpolation grid. In the end, we expect either choice to yield serviceable results that conserve mass and energy in the regridding.

12 Regridding examples: nonlocal RM matrix

In the above sections, we considered how the transformations in this paper work when the **RM** matrix is local or almost local. Here, we consider the case in which **RM** is significantly nonlocal – for example, if bilinear interpolation were used for the transformation $E \to I$. We would expect the "corrections" and dispersive properties that appear in the case of an almost-local **RM** matrix to be significantly larger.

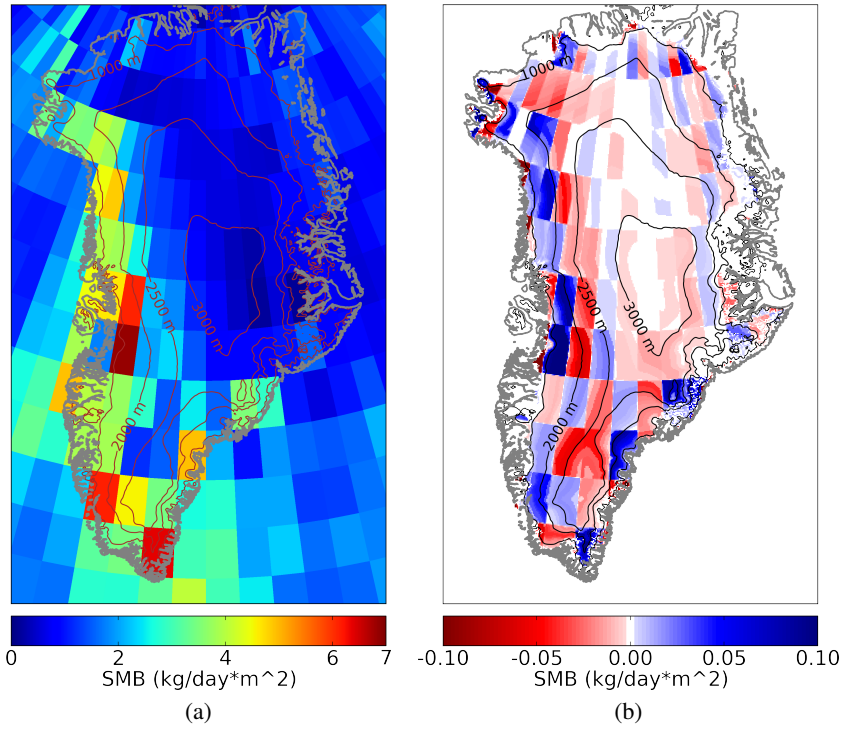

Fig. 18. Nonphysical changes in precipitation field introduced by $A \to E$ in order to ensure conservation. Panel **(a)** shows a July precipitation field computed by ModelE on the atmosphere grid. This field was regridded to the elevation grid E using an almost-local **RM** matrix for $E \to A$. It produced a slightly different result than the physically intuitive procedure of "repeating" precipitation values for elevation points in each atmosphere grid cell — which is not conservative. Those differences are plotted in panel **(b)**.

We encountered numerous problems when we attempted the use of a significantly nonlocal **RM** matrix, constructed using bilinear interpolation for $E \to I$. Most significantly, the $A \to E$ transformation produced elevation points of negative precipitation when regridding a typical precipitation field over Greenland (Fig. 19) – an artifact that we believe would be unacceptable to the majority of modelers. It might be possible to find a solution to this problems. In the meantime, it is simpler to avoid nonlocal interpolation schemes such as bilinear interpolation.

13 Independent elevation grid

The elevation grid E is constructed by adding elevation points to each grid cell of an underlying horizontal grid, which we will denote E^0. So far we have assumed that $E^0 = A$, meaning the elevation grid E is derived from the atmosphere grid A. Some GCMs use a horizontal layout for E that is not related to A. In this section, we extend our methods to address that case.

The first problem is a choice of the interpolation grid G. The idea of locality in **RM** no longer makes sense when $E^0 \neq A$. For that reason, $G = I$ is the right choice, $\mathbf{X}$ and $\mathbf{X}'$ will both become identity transformations. The transformation $E \to I$ does not need to change, nor does $I \to A$: none of these rely on any specific relationship between E^0 and A. Similarly, $E \to A = \mathbf{RM}$ can be computed as before.

The only other thing that must change is the construction of $A \to E$: the intuitive definition for Λ (Sect. 8.1) no longer makes sense. Instead, we construct an intuitive regridding operation $\bar{\mathbf{F}}$ as follows. Given $\boldsymbol{f}^A$, first regrid to $\boldsymbol{f}^{E^0}$ using area-weighted remapping. Then convert $\boldsymbol{f}^{E^0}$ to $\boldsymbol{f}^E$ using a "repeat" operator akin to Λ above. We can now apply the quadratic programming–based regrid operator developed in Sect. 8.2, using $\bar{\mathbf{F}}$ instead of Λ in Eq. (15).

The use of $E^0 \neq A$ introduces numerical dispersion into the system. This can be seen by evaluating the locality of the round-trip transformation $E \to A \to E$. This numerical dispersion is a fundamental consequence of the use of a different underlying grid for the ice surface and atmosphere models.

14 Regridding in elevation space

We have tacitly assumed so far that there is one single fixed elevation grid E with one fixed set of basis functions. This is not the case in a changing climate, since the basis functions used for E depend on ice elevation and extent (Sect. 7.1). The user might wish to explicitly move elevation points as well – for example, to track a mountain glacier as it moves up or down. When the vector space E changes, the ice surface

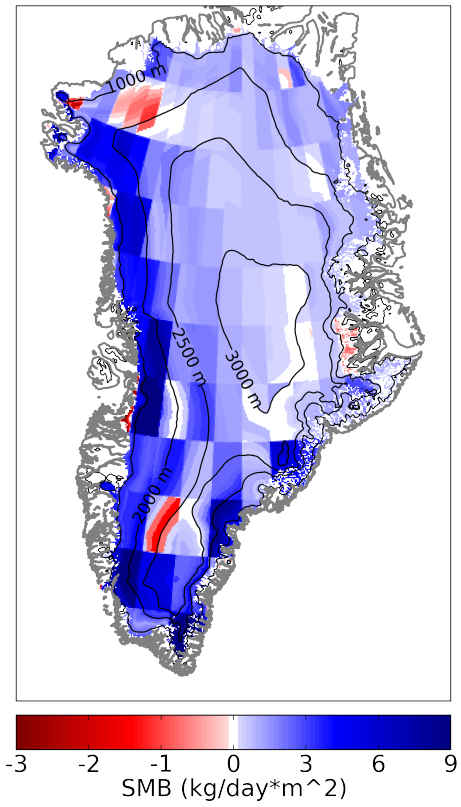

Fig. 19. Implausible precipitation field produced by $A \to E$ when bilinear interpolation is used. July precipitation (Fig. 18a) was regridded to the elevation grid E using the $A \to E$ reverse transformation, where the matrix for $E \to A$ is constructed using (nonlocal) bilinear interpolation. Note the unphysical artifacts (negative precipitation) for some elevation points.

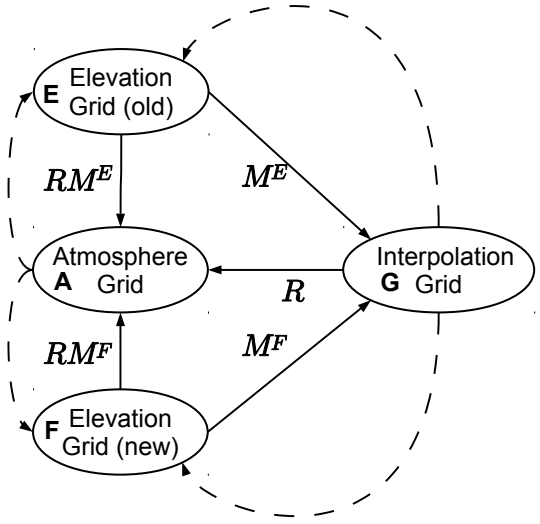

Fig. 20. When ice extent, ice topography or elevation points change, the basis functions for the elevation grid E change along with it. Ice surface model state, which exists on E, must be regridded to the new set of basis functions. Shown is a grid system that can serve as a map for this regridding: E is the old elevation grid, F is the new one and G is the interpolation grid (same for old and new). Ice surface model state may be regridded from E to F by first regridding $E \to G$, then $G \to F$. Note that this diagram is a simplified version of Fig. 13 in which two different elevation grids have been accounted for.

model state must be regridded from the old elevation grid to the new elevation grid. We address that issue in this section.

Assume two elevation grids, an "old" grid E and a "new" grid F. We wish to conservatively regrid a field $\boldsymbol{f}^E$ to $\boldsymbol{f}^F$. In this case, $\boldsymbol{f}$ will not be a flux variable, but rather a conserved state variable of the ice surface model: snow depth, water fraction, etc.

This problem can be approached by examining the system of grids and transformations available in Fig. 13 when one has multiple elevation grids (see Fig. 20). By "following the arrows," the most direct way to transform $\boldsymbol{f}^E$ to $\boldsymbol{f}^F$ is to first regrid to the interpolation grid, computing $\boldsymbol{f}^G = \mathbf{M}^E \boldsymbol{f}^E$. Then use the procedure in Sect. 8.5 to compute $\boldsymbol{f}^F$. Because all these transformations are conservative on the atmosphere grid A, the end result $\boldsymbol{f}^F$ will be equivalent to $\boldsymbol{f}^E$ on A.

14.1 Conserved model state

In previous sections, $\boldsymbol{f}$ represented fluxes between models, which are conserved. In this section, $\boldsymbol{f}$ is a *model state* variable of the ice surface model. In order for this regridding procedure to be physically meaningful, the model state must be expressed in terms of conserved quantities.

Not all ice surface models are formulated in terms of conserved quantities. In this case, the regridding procedures may still be used, as long as the ice flow model can be converted to/from a form that *is* expressed in conserved quantities. For example, an ice surface model might track the temperature, mass and water fraction of the top layer of ice. Temperature is not conserved, so this regridding procedure cannot be used directly. However, model state can be converted to enthalpy and mass alone (Aschwanden et al., 2012). These quantities are conserved and can be correctly regridded with conservative transformations. After regridding to the new elevation grid, model state can then be converted back to the original nonconserved parameterization.

14.2 When to regrid

Regridding in elevation space is not just required if one changes elevation points, but also *any time* that the transformation $\mathbf{M}$ for $E \to I$ changes. Any time that happens, the regridding described in Sect. 14 should be used. Relevant events that change $\mathbf{M}$ include

1. changes in ice topography: for example, as an ice sheet inflates or deflates due to changing climate; when an ice grid cell changes elevation, the weights in $\mathbf{M}$ used to compute it will change; this change will either be

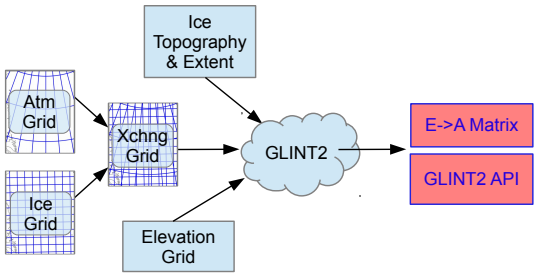

Fig. 21. The Glint2 workflow used to compute the regridding operations required by the fully coupled system (Fig. 2). Glint2 produces regridding operations based on a variety of factors: atmosphere and ice grid geometry, ice topography and extent, and levels chosen for the elevation grid. Glint2 must recompute the operations when any of these factors changes.

continuous (as in z interpolation) or will jump at certain thresholds (as in elevation class interpolation).

2. changes in ice sheet extent, as an ice sheet grows or melts; for example, if an ice sheet shrinks, then some ice grid cells will no longer participate in the regridding process, and their associated coefficients in **M** will turn to zero.

3. changes in the elevation or number of elevation points.

15 Glint2 coupling library

Here we describe Glint2, an open-source implementation of the transformations developed in this paper. Glint2 is a GCM–ice flow model coupling library whose core function is to compute the five transformations required for a coupled GCM–ice flow model system (Fig. 2). These transformations are computed based on a variety of factors: atmosphere and ice grid geometry, ice topography and extent, and elevation levels chosen for the elevation grid (Fig. 21).

Glint2 does not just compute these transformations and provide them as a library, it also provides an application programming interface (API) for GCMs to use to couple with ice flow models. As a realization of the mediator design pattern (Gamma et al., 1994), Glint2 is able to shield the GCM from a number of details of the coupling. Although it uses a different codebase, Glint2 has the same purpose as GLINT (Glimmer Interface; Rutt et al., 2009), namely to build a coupling library between GCM and ice models. The GCM programmer who wishes to couple their GCM with an ice model need only do the following:

1. implement an elevation points scheme, and move the ice surface model to the elevation grid.

2. request the **RM** matrix from Glint2, and then apply it as needed during normal model run; this might be more complex than it sounds, depending on the GCM's system of domain decomposition; storage and application of the **RM** matrix is simplified if the GCM author knows in advance that it is local.

3. ask Glint2 to regrid ice surface model inputs from $A \rightarrow E$ as needed; this step is only required if **RM** is nonlocal; otherwise, the transformation Λ is simple enough for the GCM to do on its own.

4. accumulate ice surface fluxes on the elevation grid, and pass them to Glint2 every coupling time step.

5. apply fields returned from the ice flow model via Glint2; these fields are returned on the elevation and atmosphere grids as appropriate, eliminating the need for the GCM to regrid them.

Note that the GCM does *not* need to know anything about the ice grid or ice flow model. Interface code is added to Glint2, not the GCM, for every ice flow model one wishes to support. Over time, we expect the number of supported GCMs and ice flow models to increase, according to researcher demand. This structure is useful because it will give practitioners a way to try different ice models with a GCM fairly easily.

15.1 Adoption issues

We expect that Glint2 could be useful to anyone with a GCM who wishes to couple it with an ice flow model. However, many GCMs have centralized regridding strategies that Glint2 does not really fit into. We do not believe this should be a significant barrier to adoption for two reasons:

- In most cases, the elevation grid will have unusual "customized" basis functions (Fig. 12). Centralized GCM regridding schemes are not typically equipped to regrid to/from the elevation grid. Nor are they equipped with the algorithms required for the reverse transformations.

- Glint2 hides all details of the ice grid from the GCM, communicating with the GCM with fields on the atmosphere and elevation grids. Because all ice grid-related issues are encapsulated in Glint2, it should not matter to the GCM *how* or even *whether* regridding to/from the ice grid is accomplished. As far as the GCM is concerned, the ice flow model might as well be running on the elevation grid.

Another barrier to adoption is the fact that Glint2 is packaged as a library. Many GCM projects are reluctant to add additional library dependencies, due to the complications such dependencies introduce in the build process. We do not believe this should be a serious problem because coupling with an ice model already involves the use of outside libraries. With or without Glint2, the GCM must still manage additional external dependencies.

15.2 Availability

Glint2 is written C++ and designed to couple with GCMs and ice flow models written in Fortran 90/2003, C or C++. It also comes with a Python interface, making it easy to test and plot sample regridding problems before incorporating a coupling strategy into a GCM. Glint2 source and documentation are available for download (Fischer, 2013).

16 Discussion and conclusions

This paper focuses on a system of conservative regridding strategies needed to support tight two-way coupling between a GCM and an ice flow model in the context of elevation-based downscaling in the GCM. Past efforts at one-way coupling have produced downscaling methods that provide SMB fields that match remarkably well with observations and with regional models. However, these efforts used an inconsistent set of transformations, which would result in nonconservation of mass and energy in a two-way coupled system.

In order to achieve consistency, we began by recognizing the elevation grid as an integral part of the coupling problem, along with the atmosphere and ice grids that had previously been considered. We observed that the ice surface model runs on the elevation grid and that the downscaling step (from elevation to ice grid) is actually another form of regridding. We have therefore transformed a coupling problem involving two models and two grids into one with three models and three grids.

We analyzed the regridding transformations needed in a typical two-way coupling and determined that five such transformations are required: three "forward" transformations and two "reverse" transformations. We then set out to develop a consistent implementation for these five transformations, starting with the forward transformations.

We observe that conservation of mass and energy requires consistency between the transformations from the elevation to ice grid ($E \rightarrow I$) and elevation to atmosphere grid ($E \rightarrow A$). We achieve consistency by allowing the user to choose $E \rightarrow I$ and then constructing a transformation for $E \rightarrow A$ that is consistent with the user's choice. This is a good approach because it allows the user freedom in choosing a downscaling transformation for $E \rightarrow I$. Our transformations imply a set of basis functions for the elevation grid, which we demonstrated in plots.

We then addressed the reverse transformations, using the notions of consistency developed in the previous sections. Problems of underdeterminism and overdeterminism, caused by mismatches in dimensionality between grids, are addressed by using quadratic optimization for these transformations.

The result is a set of five conservative transformations needed to support a two-way coupling of GCMs and ice models. We defined a property of our $E \rightarrow A$ transformation called *locality*. We showed a number of theoretical and practical benefits if that transformation is local. We implemented all five transformations and demonstrated that they work in practice on realistic input fields.

Although three of those five transformations are well-known in the literature (Ramshaw, 1985; Lipscomb et al., 2013), the reverse transformations are new. Most importantly, we showed how to choose a set of grids, basis functions and regridding schemes so that *all five* regridding operations are conservative. Note that the elevation grid uses a "custom" set of basis functions. This is a significant step beyond past efforts that have focused on conservative regridding to L0 pre-chosen grids (Ramshaw, 1985).

Our downscaling transformation from the elevation to ice grids is taken from previous one-way studies (Lipscomb et al., 2013). Our results "look" almost identical, except for subtle differences in the other related transformations needed to ensure conservation. We therefore provide practitioners with a way to bring already proven downscaling techniques into a conservative two-way coupled setting.

We went on to package these transformations in Glint2, a coupling library. As a piece of software, Glint2 serves as a mediator (Gamma et al., 1994) between the GCM and ice flow model, insulating each from the specific details of the other. This will simplify the coupling of GCMs with multiple ice models, as needed to support future research.

Appendix A

Mathematical conventions

We use the following mathematical conventions in this paper:

1. A "grid" or "parameterization" is a set of basis functions that may be linearly combined to produce functions over a 2-D domain. Grids are denoted by non-bold capital letters: A for the atmosphere grid, I for the ice grid, E for the elevation grid and G for an interpolation grid. Basis functions are denoted by non-bold lower case letters: $a_i(x, y)$ is a basis function for A. At times, the set of basis functions may be represented as a bold-face vector: $\boldsymbol{a}(x, y) = [a_1(x, y), \ldots, a_n(x, y)]$.
2. Subscripts are used to indicate elements of a vector. If $\boldsymbol{f}$ is a vector, then f_i is the value of the ith index of $\boldsymbol{f}$.
3. If A is an L0-parameterized grid and A_i one of the grid cells, then $|A_i|$ is the area of that grid cell.
4. Vector values are indicated in boldface, scalars in non-bold.
5. Vectors are used to construct a 2-D function ("field") within the context of a particular grid. For example, suppose that the atmosphere grid A uses the basis functions $\boldsymbol{a}(x, y) = [a_1(x, y), \ldots, a_n(x, y)]$, and we have an n-dimensional vector $\boldsymbol{f}^A$ with components f_i^A. That vector represents the function:
$$f^A(x, y) = \boldsymbol{f}^A \cdot \boldsymbol{a}(x, y). \tag{A1}$$
6. Since this paper is about regridding, we need to talk about the "same" field in different grids. Superscripts are used for this: if the vector $\boldsymbol{f}^I$ denotes the vector representing a field on the ice grid, then $\boldsymbol{f}^A$ denotes a vector representing the "same" field on the atmosphere grid.
7. Similarly, $f^A(x, y)$ denotes the function implied by the vector $\boldsymbol{f}^A$ and the vector of basis functions $\boldsymbol{a}(x, y)$.
8. We use arrows to talk about transformations between grids, which transform fields on one grid to fields on another. Since fields on a grid are represented by vectors, the transformations are functions from one vector space to another. For example, $E \to A$ can be read as "the transformation from the elevation grid to the atmosphere grid." Many but not all of these transformations are linear and can be represented by matrices.
9. If a transformation is nonlinear, we use functional notation with square brackets to denote an application of that transformation: $\boldsymbol{f}^E = [A \to E](\boldsymbol{f}^A)$.
10. We use indicial notation in one place.
11. See Fig. 3 for definition of the symbols used throughout the text.

Appendix B

Integration of L0 grid

Suppose we have a region B and an L0-parameterized grid G with grid cells $G_1, \ldots, G_n$. The basis function $g_i(x, y)$ is equal to 1 inside of grid cell G_i and zero elsewhere. We wish to compute

$$\int_B g_i(x, y)\mathrm{d}A = |B \cap G_i|. \tag{B1}$$

This is simply equal to the area of intersection of B and G_i.

If we wish to regrid to an L0 grid A, we must do the above computation multiple times, setting B to every grid cell $A_j \in A$. This creates an *overlap matrix* $\mathbf{L}$ where L_{ij} is equal to the overlap between source grid cell G_i and destination grid cell A_j.

The overlap matrix is directly related to the exchange grid between G and A: every non-zero element of L_{ij} is equal to the area of one grid cell in the exchange grid. The exchange grid may be computed as described in Appendix D. The Surveyor's Formula (Braden, 1986), a special case of Green's Theorem, can then be used to compute the overlap matrix from the exchange grid.

Appendix C

Integration of L1 grid

Appendix B shows how to integrate over areas on an L0 ice grid. But if an L1 finite element ice flow model is used, one will need to do this over an L1 ice grid. Here, we show how to integrate L1 finite element basis functions over an arbitrary area B.

An L1 finite element mesh is made up of triangular elements, where the value of a function is defined at the vertices of the triangles. Values inside each element are interpolated based on the values at the vertices.

Each basis function $N_i(x, y)$ in a finite element mesh corresponds to a vertex (Zienkiewicz et al., 2013). This function is non-zero only in the triangular elements in which the vertex participates. We define a sub-basis function $N_{ij}(x, y)$ to be equal to $N_i(x, y)$ within triangular element j, and zero elsewhere. Thus we define

$$N_i(x, y) = \sum_j N_{ij}(x, y). \tag{C1}$$

Within its element, a sub-basis function takes the shape of a plane, having the functional form

$$N_{ij}(x, y) = a + bx + cy. \tag{C2}$$

The coefficients a, b and c are functions of the locations of the vertices, not of the values assigned to those vertices.

Using the techniques from Appendix D, one can compute the polygon β_j as the intersection between element j and region B. Green's Theorem may then be used to compute

$$\int_{\beta_j} N_{ij}(x, y)\mathrm{d}A. \tag{C3}$$

We can then apply our definition of sub-basis functions to achieve our goal:

$$\int_{B} N_i(x, y)\mathrm{d}A = \sum_j \int_{\beta_j} N_{ij}(x, y)\mathrm{d}A. \tag{C4}$$

This section has provided just an outline of the process. The algebra can become complex at times, and a symbolic computation system such as Maxima can be useful. But in the end, integration of a vector $\boldsymbol{f}^I$ over an area B is computed as a linear combination of the elements of $\boldsymbol{f}^I$, just as with L0 grids.

Appendix D

Computing the exchange grid

Appendices B and C can be applied repeatedly to compute conservative regridding matrices from an L0 or L1 grid to an L0 destination grid. Both procedures assume a way to compute polygonal intersections between two sets of polygons – also known as an *exchange grid* (Balaji et al., 2006).

This task is simple in principle, using modern computational geometry packages such as CGAL (2013). If A and G are two sets of polygons, we explicitly construct each polygon in A and G and then compute pairwise intersections between the two sets (Chin and Wang, 1983). This algorithm provides not just integration formulas, but also the actual polygonal outlines of the exchange grid.

If an appropriately robust polygon intersection algorithm is used, our procedure can deal with nonconvex polygons and other possible irregularities. This is not just a theoretical issue: latitude–longitude grid cells commonly used in GCMs are not convex in spherical or planar geometry. In other cases, practitioners might wish to use grid cells consisting of multiple disjoint polygons.

With issues of polygonal intersection taken care of by prepackaged algorithms, the main challenge here is to find those intersections in a scalable manner. The naive algorithm is to write a nested loop, requiring $|A| \times |G|$ iterations. This algorithm, with $O(n^2)$ complexity, takes too long even on grids commonly used by GCMs and ice flow models today. Most of the "intersections" of A_i and G_j will result in nothing, because A_i and G_j are far from each other and obviously do not intersect.

This problem is solved by using an R-tree (Guttman, 1984) to avoid having to consider intersections of grid cells that are far away from each other. The procedure works as follows: first load all the grid cell outlines of A into the R-tree, indexed by their bounding rectangles. Then loop through each grid cell in G, checking the R-tree for any grid cells in A that it might intersect with. The polygon intersection algorithm is run on each of those grid cell pairs to determine the exact outlines of the exchange grid cells. Running time is cut down to a more reasonable $O(n \log n)$.

Past algorithms exist to compute regridding matrices by integrating functions around each cell in an exchange grid (Ramshaw, 1985; Dukowicz and Kodis, 1987; Jones, 1999). These algorithms were originally presented in terms of quadrilateral meshes, but they can also be applied to arbitrary meshes. However, they never explicitly compute the exchange grid polygonal outlines. By explicitly computing an exchange grid and then using that to produce the regridding matrix, we have presented here a procedure that is conceptually simpler, possibly more flexible, but almost certainly not faster to run.

Acknowledgements. Thanks to Ed Bueler, Jeremy Fycke, William Lipscomb, Christian Rodehacke and Ryan Walker for reading drafts. Thanks also to Roger Braithwaite, Nicholas Gould, Regine Hock and David Rind. This research is sponsored by the NASA MAP program.

Edited by: D. Roche

References

Aschwanden, A., Bueler, E., Khroulev, C., and Blatter, H.: An enthalpy formulation for glaciers and ice sheets, J. Glaciol., 58, 441–457, doi:10.3189/2012JoG11J088, 2012.

Balaji, V., Anderson, J., Held, I., Winton, M., Durachta, J., Malyshev, S., and Stouffer, R. J.: The Exchange Grid: a mechanism for data exchange between Earth System components on independent grids, in: Parallel Computational Fluid Dynamics: Theory and Applications, Proceedings of the 2005 International Conference on Parallel Computational Fluid Dynamics, College Park, MD, USA, 24–27 May, edited by: Deane, A., Brenner, G., Ecer, A., Emerson, D., McDonough, J., Periaux, J., Satofuka, N., and Tromeur-Dervout, D., Elsevier, 2006.

Best, M. J., Baljaars, A., Polcher, J., and Viterbo, P.: A proposed structure for coupling tiled surfaces with the planetary boundary layer, J. Hydrometerol., 5, 1271–1278, doi:10.1175/JHM-382.1, 2004.

Bindschadler, R., Nowicki, S., Abe-Ouchi, A., Aschwanden, A., Choi, H., Fastook, J., Granzow, G., Greve, R., Gutowski, G., Herzfeld, U., Jackson, C., Johnson, J., Khroulev, C., Levermann, A., Lipscomb, W., Martin, M., Morlighem, M., Parizek, B., Pollard, D., Price, S., Ren, D., Saito, F., Sato, T., Seddik, H., Seroussi, H., Takahashi, K., Walker, R., and Wang, W.: Ice-sheet model sensitivities to environmental forcing and their use in projecting future sea level (the SeaRISE project), J. Glaciol., 59, 195–224, doi:10.3189/2013JoG12J125, 2013.

Box, J. E., Cappelen, J., Chen, C., Decker, D., Fettweis, X., Mote, T., Tedesco, M., van de Wal, R. S. W., and Wahr, J.: Greenland Ice Sheet [in Arctic Report Card 2012], available at: http://www.arctic.noaa.gov/reportcard/greenland_ice_sheet.html (last access: 3 December 2013), 2013.

Braden, B.: The surveyor's area formula, Coll. Math. J., 17, 326–337, doi:10.2307/2686282, 1986.

Braithwaite, R.: On glacier energy balance, ablation, and air temperature, J. Glaciol., 27, 381–391, 1981.

Bueler, E. and Brown, J.: Shallow shelf approximation as a sliding law in a thermomechanically coupled ice sheet model, J. Geophys. Res., 114, F03008, doi:10.1029/2008JF001179, 2009.

Cgal: Computational Geometry Algorithms Library, available at: http://www.cgal.org, last access: 3 December 2013.

Chin, F. and Wang, C. A.: Optimal algorithms for the intersection and the minimum distance problems between planar polygons, IEEE T. Comput., 32, 1203, doi:10.1109/TC.1983.1676186, 1983.

Church, J., Clark, P., Cazenave, A., Gregory, J, Jevrejeva, S., Levermann, A., Merrifield, M., Milne, G, Nerem, R., Nunn, P., Payne, A., Pfeffer, W., Stammer, D., and Unnikrishnan, A. :Sea Level Change. In Climate Change 2013: The Physical Sciences Basis, Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, Cambridge University Press, 2013.

Collins, W. J., Bellouin, N., Doutriaux-Boucher, M., Gedney, N., Halloran, P., Hinton, T., Hughes, J., Jones, C. D., Joshi, M., Liddicoat, S., Martin, G., O'Connor, F., Rae, J., Senior, C., Sitch, S., Totterdell, I., Wiltshire, A., and Woodward, S.: Development and evaluation of an Earth-System model – HadGEM2, Geosci. Model Dev., 4, 1051–1075, doi:10.5194/gmd-4-1051-2011, 2011.

Cornford, S. L., Martin, D. F., Graves, D. T., Ranken, D. F., Le Brocq, A. M., Gladstone, R. M., Payne, A. J., Ng, E. G., and Lipscomb, W. H.: Adaptive mesh, finite volume modeling of marine ice sheets, J. Comput. Phys., 232, 529–549, doi:10.1016/j.jcp.2012.08.037, 2013.

Dukowicz, J. K. and Kodis, J. W.: Accurate conservative remapping (rezoning) for arbitrary Lagrangian–Eulerian computations, SIAM J. Sci. Stat. Comp., 8, 305–321, doi:10.1137/0908037, 1987.

Fischer, R.: Glint2, available at: http://citibob.github.io/glint2 (last access: 3 December 2013), 2013.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software, 1st Edn., Addison-Wesley Professional, 1994.

Goelzer, H., Huybrechts, P., Fürst, J. J., Nick, F. M., Andersen, M. L., Edwards, T. L., Fettweis, X., Payne, A., J., and Shannon, S.: Sensitivity of Greenland ice sheet projections to model formulations, J. Glaciol., 59, 733–749, doi:10.3189/2013JoG12J182, 2013.

Gould, N., Orban, D., and Toint, P.: GALAHAD, a library of thread-safe Fortran 90 packages for large-scale nonlinear optimization, ACM T. Math. Software, 29, 353–372, doi:10.1145/962437.962438, 2003.

Greve, R.: On the response of the Greenland Ice Sheet to greenhouse climate change, Climatic Change, 46, 289–303, doi:10.1023/A:1005647226590, 2000.

Guttman, A.: R-trees – a dynamic index structure for spatial searching, in: SIGMOD '84, Proceedings of the 1984 ACM SIGMOD international conference on management of data, 14, 47–57, doi:10.1145/971697.602266, 1984.

Heath, T. and Archimedes: The Works of Archimedes, C. J. Clay and Sons, Cambridge University Press Warehouse, available at: https://archive.org/details/worksofarchimede029517mbp (last access: 3 December 2013), 1897.

Hurrell, J. W., Holland, M. M., Gent, P. R., Ghan, S., Kay, J. E., Kushner, P. J., Lamarque, J. F., Large, W. G., Lawrence, D., Lindsay, K., Lipscomb, W. H., Long, M. C., Mahowald, N., Marsh, D. R., Neale, R. B., Rasch, P., Vavrus, S., Vertenstein, M., Bader, D., Collins, W. D., Hack, J. J., Kiehl, J., and Marshall, S.: The Community Earth System Model: a framework for collaborative research, B. Am. Meteorol. Soc., 94, 1339–1360, doi:10.1175/BAMS-D-12-00121.1, 2013.

Huybrechts, P.: The present evolution of the Greenland ice sheet: an assessment by modelling, Global Planet. Change, 9, 39–51, doi:10.1016/0921-8181(94)90006-X, 1994.

Jones, P. W.: First- and second-order conservative remapping schemes for grids in spherical coordinates, Mon. Weather Rev., 127, 2204–2210, doi:10.1175/1520-0493(1999)127<2204:FASOCR>2.0.CO;2, 1999.

Larour, E., Seroussi, H., Morlighem, M., and Rignot, E.: Continental scale, high order, high spatial resolution, ice sheet modeling using the Ice Sheet System Model (ISSM), J. Geophys. Res., 117, F01022, doi:10.1029/2011JF002140, 2012.

Lauritzen, P. H. and Nair, R. D.: Monotone and Conservative Cascade Remapping between Spherical Grids (CaRS): Regular Latitude–Longitude and Cubed-Sphere Grids, Mon. Weather Rev., 136, 1416, doi:10.1175/2007MWR2181.1, 2008.

Leung, L. R. and Ghan, S. J.: Parameterizing subgrid orographic precipitation and surface cover in climate models, Mon. Weather Rev., 126, 3271–3291, doi:10.1175/1520-0493(1998)126<3271:PSOPAS>2.0.CO;2, 1998.

Lipscomb, W. H., Fyke, J. G., Vizcaíno, M., Sacks, W. J., Wolfe, J., Vertenstein, M., Craig, A., Kluzek, E., and Lawrence, D. M.: Implementation and initial evaluation of the Glimmer Community Ice Sheet Model in the Community Earth System Model, J. Climate, 26, 7352–7371, doi:10.1175/JCLI-D-12-00557.1, 2013.

Nowicki, S., Bindschadler, R., Abe-Ouchi, A., Aschwanden, A., Bueler, E., Choi, H., Fastook, J., Granzow, G., Greve, R., Gutowski, G., Herzfeld, U., Jackson, C., Johnson, J., Kroulev, C., Larour, E., Levermann, A., Lipsomb, W., Martin, M., Morlighem, M., Parizek, B., Pollard, D., Price, S., Rignot, E., Ren, D., Saito, F., Sato, T., Seddik, H., Seroussi, H., Takahashi, K., Walker, R., and Wang, W.: Insights into spatial sensitivities of ice mass response to environmental change from the SeaRISE ice sheet modeling project I: Antarctica, J. Geophys. Res., 118, 1002–1024, doi:10.1002/jgrf.20081, 2013a.

Nowicki, S., Bindschadler, R., Abe-Ouchi, A., Aschwanden, A., Bueler, E., Choi, H., Fastook, J., Granzow, G., Greve, R., Gutowski, G., Herzfeld, U., Jackson, C., Johnson, J., Kroulev, C., Larour, E., Levermann, A., Lipsomb, W., Martin, M., Morlighem, M., Parizek, B., Pollard, D., Price, S., Rignot, E., Ren, D., Saito, F., Sato, T., Seddik, H., Seroussi, H., Takahashi, K., Walker, R., and Wang, W.: Insights into spatial sensitivities of ice mass response to environmental change from the SeaRISE ice sheet modeling project II: Greenland, J. Geophys. Res.-Earth, 118, 1025–1044, doi:10.1002/jgrf.20076, 2013b.

Pollard, D. and DeConto, R. M.: Description of a hybrid ice sheet–shelf model, and application to Antarctica, Geosci. Model Dev., 5, 1273–1295, doi:10.5194/gmd-5-1273-2012, 2012.

Qu, X. and Hall, A.: Assessing snow albedo feedback in simulated climate change, J. Climate, 19, 2617–2630, doi:10.1175/JCLI3750.1, 2006.

Ramshaw, J. D.: Conservative rezoning algorithm for generalized two-dimensional meshes, J. Comput. Phys., 59, 193–199, doi:10.1016/0021-9991(85)90141-X, 1985.

Ridley, J. K., Huybrechts, P., Gregory, J. M., and Lowe, J. A.: Elimination of the Greenland Ice Sheet in a high CO_2 climate, J. Climate, 18, 3409–3427, doi:10.1175/JCLI3482.1, 2005.

Ridley, J., Gregory, J. M., Huybrechts, P., and Lowe, J.: Thresholds for irreversible decline of the Greenland ice sheet, Clim. Dynam., 35, 1059–1057, doi:10.1007/s00382-009-0646-0, 2010.

Rutt, I. C., Hagdorn, M., Hulton, N. R. J., and Payne, A. J.: The Glimmer community ice sheet model, J. Geophys. Res., 114, F02004, doi:10.1029/2008JF001015, 2009.

Schmidt, G. A., Bitz, C. M., Mikolajewicz, U., and Tremblay, L. B.: Ice–ocean boundary conditions for coupled models, Ocean Model., 7, 59–74, doi:10.1016/S1463-5003(03)00030-1, 2004.

Schmidt, G. A., Ruedy, R., Hansen, J. E., Aleinov, I., Bell, N., Bauer, M., Bauer, S., Cairns, B., Canuto, V., Cheng, Y., Del Genio, A., Faluvegi, G., Friend, A. D., Hall, T. M., Hu, Y., Kelley, M., Kiang, N. Y., Koch, D., Lacis, A. A., Lerner, J., Lo, K. K., Miller, R. L., Nazarenko, L., Oinas, V., Perlwitz, J. P., Perlwitz, J., Rind, D., Romanou, A., Russell, G. L., Sato, M., Shindell, D. T., Stone, P. H., Sun, S., Tausnev, N., Thresher, D., and Yao, M.-S.: Present day atmospheric simulations using GISS ModelE: comparison to in-situ, satellite and reanalysis data, J. Climate, 19, 153–192, doi:10.1175/JCLI3612.1, 2006.

Snyder, J. P.: Map projections – a working manual, available at: http://pubs.er.usgs.gov/publication/pp1395 (last access: 3 December 2013), 1987.

Stone, E. J., Lunt, D. J., Rutt, I. C., and Hanna, E.: Investigating the sensitivity of numerical model simulations of the modern state of the Greenland ice-sheet and its future response to climate change, The Cryosphere, 4, 397–417, doi:10.5194/tc-4-397-2010, 2010.

van den Broeke, M., Smeets, P., Ettema, J., and Munneke, P. K.: Surface radiation balance in the ablation zone of the west Greenland ice sheet, J. Geophys. Res., 113, D13105, doi:10.1029/2007JD009283, 2008.

Vaughan, D., Comiso, J., Allison, I., Carrasco, J., Kaser, G., Kwok, R., Mote, P., Murray, T., Paul, F., Ren, J., Rignot, E., Solomina, O., Steffen, K. and Zhang, T.: Observations: Cryosphere. In Climate Change 2013: The Physical Sciences Basis, Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, Cambridge University Press, 2013.

Vizcaino, M., Lipscomb, W., Sacks, W., van Angelen, J. H., Wouters, B., and van den Broeke, M. R.: Greenland surface mass balance as simulated by the Community Earth System Model. Part I: model validation and 1850–2005 results, J. Climate, 26, 7793–7812, doi:10.1175/JCLI-D-12-00615.1, 2013.

Zagorodnov, V., Nagornov, O., and Thompson, L. G.: Influence of air temperature on a glacier's active-layer temperature, Ann. Glaciol., 43, 285–291, doi:10.3189/172756406781812203, 2006.

Zienkiewicz, O. C., Taylor, R. L., and Nithiarasu, P.: The Finite Element Method for Fluid Dynamics, 7th Edn., Butterworth–Heinemann, 2013.