Handling partial overlap fractions between CISM and CLM, and related conservation issues

1. CISM grid cells partially or entirely outside the CLM domain

1.1. CISM grid cell totally outside the CLM domain

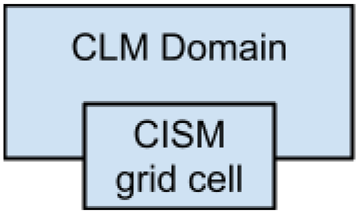
SMB -> CISM: 0

Surface temperature -> CISM: Assume some fixed value? I'm not sure the best way to implement this. It would be easy enough to hard-code a fixed temperature value in the merge routine in prep_glc_mod. However, you would need to add code to specially handle every state variable, if other state variables were added later.

Special handling in CISM: Also, eventually want to send a mask to CISM saying there's no CLM domain there. CISM will calve any ice formed here. This is desired (according to Jeremy Fyke and Bill Lipscomb) in order to avoid unchecked ice growth on the edge of the CISM domain, outside the CLM domain. (An alternative would be to send negative SMB in those regions, but I think that would break conservation.) I think we can create this mask in the coupler – i.e., I don't think CLM actually needs to send an additional field to the coupler, but the coupler will compute this mask and send it to CISM.

CISM -> CLM: Irrelevant for now. This will be relevant for the basal heat flux, but we'll come back to that later.

1.2. CISM grid cell partially outside the CLM domain



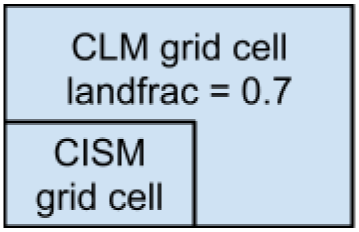
SMB -> CISM: Fractionally decrease the SMB: Implicitly send 0 SMB over the non-CLM domain.

Surface temperature -> CISM: Probably just use value from CLM. (Since this isn't a flux, we don't need to fractionally include the assumed ocean value, and it will be more accurate to just use the value from CLM rather than including the assumed ocean value.)

Special handling in CISM: Apply calving as in the case where the CISM cell is entirely outside the CLM domain? Ideally, we would calve (e.g.) ½ of the *new* ice sent in to this cell. But for simplicity, maybe just calve all ice in this cell? Bill Lipscomb agrees that we could just calve the whole grid cell for simplicity. However, he thinks that a better compromise between ease of implementation and scientific goodness would be: If the mapped CLM domain fraction is <= 0.5, then calve all ice; if it's > 0.5, then do not do any calving. (In the latter case, we would live with having the wrong SMB, knowing that it is at least correct within a factor of 2.) Note that (I think) we will want to apply the SMB *before* doing the calving.

Ice area -> CLM: Nothing special needs to be done, I think

1.3. CISM grid cell inside the CLM domain, but CLM's landfrac is less than 1



This is similar to the case of CISM partially outside the CLM domain.

One way to handle this would be to map the ocean grid to CISM, so we can figure out which CISM cells are truly inside / outside the CLM domain. But for now, it's probably easier to not try to use that extra information, and instead treat all CISM cells that are contained in this land cell as being 0.7 inside the land domain and 0.3 outside the CLM domain. This will be treated the same way as in the case of a CISM cell partially outside the CLM domain, laid out above. e.g., SMB will be decreased fractionally – so if CLM sends SMB = 1 m, the CISM grid cell in the above picture would receive an SMB of 0.7 m.

Bill Lipscomb feels that it would be ideal to do the more rigorous thing of figuring out which CISM cells are inside / outside the domain. But he is comfortable with the rougher approach of weighting SMB by the landfrac across the whole area of this CLM grid cell, for simplicity.

Update: See more detailed notes and different methods in section 3.2.2. Among other things, that shows that it is even more difficult than I originally thought to do the more rigorous approach, since it actually requires considering the overlap between three grids: CLM, CISM and Ocean.

(Note, though, that by not doing the rigorous treatment, we are essentially assuming that CISM cells are randomly distributed with respect to the land/ocean mask. In particular, CISM cells inside the icemask are just as likely to appear in ocean portions of the CLM cell as are CISM cells outside the icemask – whereas, in reality, it's likely that the CISM icemask is highly correlated with CESM's land/ocean mask.)

Special handling in CISM: If we use the simpler approach of weighting SMB across the whole area of this CLM grid cell, then we should also apply a calving rule similar to the one described in section 1.2. From the point of view of CISM, it will receive a CLM overlap fraction from the coupler, which will have the same effect in this case as in section 1.2.

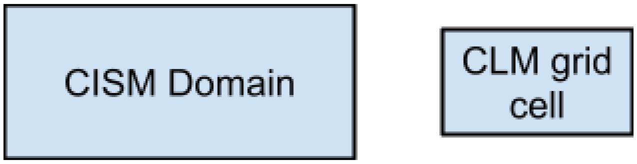
2. CLM grid cells partially or entirely outside the CISM domain

2.1. General notes about the definition of the CISM domain

We will define the CISM domain such that it excludes both (a) points totally outside the CISM grid, and (b) points inside the CISM grid, but which CISM currently classifies as ocean. Note that (a) is static, whereas (b) is potentially dynamic in time. (The dynamic nature of (b) arises both because of ice shelves that extend over the ocean – and so such points are no longer classified as ocean by CISM – and also because CISM allows bed topography to change in time, due to isostatic rebound, etc., and so the land-ocean boundary can shift over time.) We further assume that CISM classifies as ocean any land points that are outside its modeled domain. For example, on the Greenland grid, we assume that CISM classifies Iceland and Ellesmere Island as ocean (otherwise glaciers could form on these land cells due to positive surface mass balance – which we do not want).

Alternative: An alternative would be to define the CISM domain in the coupler such that it excludes points totally outside the CISM grid, but includes points classified as ocean on the CISM grid. This might be slightly simpler in terms of the handling within the coupler, because now the CISM domain would be static, and knowledge of this domain could be included in the static mapping files. However, this would place an extra burden on CISM: Any SMB sent to an ocean cell in CISM would need to be routed immediately to the calving flux. (But what would we do about negative SMB?) Ultimately, an ocean cell within CISM should be treated the same as a cell outside the CISM grid from the standpoint of the SMB fate. Thus, I feel it is best to handle these two classes of points the same way inside the coupler – i.e., by treating both of them as being outside of CISM's (dynamic) domain. However, if this proves to be difficult, then I think it would be reasonable to fall back on this alternative approach.

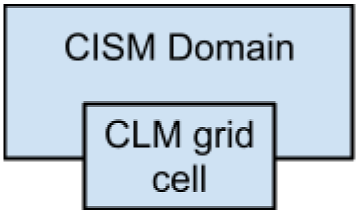
2.2. CLM grid cell totally outside the CISM domain



CISM -> CLM: All fractions and masks are 0. The 0 mask means that CLM ignores the ice fraction from CISM, and keeps its glacier cover matching whatever was on the surface dataset.

SMB: Rather than sending SMB to CISM, CLM instead sends the snow capping flux immediately to a calving flux. This logic is already done in CLM

2.3. CLM grid cell partially outside the CISM domain



SMB -> CISM: Currently in CLM, excess snow (due to snow capping) is routed either entirely to CISM via SMB, or entirely to runoff. The latter is done if the CLM grid cell is entirely outside the CISM domain (i.e., icemask_coupled_fluxes = 0). To handle this edge case, however, we need to rework this so snow capping is proportionally sent to SMB and runoff, with the proportion given by the remapped value of icemask_coupled_fluxes. In the above picture, this would mean that ½ of this grid cell's snow capping is sent to SMB, and ½ to runoff, for all elevation classes (and bare land). Negative SMB should similarly be handled with a proportion like this.

To accomplish this, we want to:

- Maintain the full, unscaled value of SMB *everywhere.* I'm pretty sure that is what is already done: Even outside the CISM domain, I think that CLM still computes and sends SMB for diagnostic purposes. (A portion of the integrated SMB quantity will be lost, as desired, due to the fact that there is no CISM cell to map it onto.)
- Change glc_dyn_runoff_routing to be a continuous quantity rather than a logical
- For code that depends on glc_dyn_runoff_routing: Change it to use the continuous quantity

    - When glc_dyn_runoff_routing is 0, this should give the same result as having glc_dyn_runoff_routing false in the old code
    - When glc_dyn_runoff_routing is 1, this should give the same result as having glc_dyn_runoff_routing true in the old code
    - When glc_dyn_runoff_routing is intermediate between 0 and 1, the results should be intermediate accordingly

Then I don't think anything else needs to be done within the coupler. (e.g., I don't think we need any normalization to handle this.)

However, for safety in this case (and in 2.2), we may want the coupler to explicitly zero out any SMB flux for CISM grid cells that are outside the mask specifying the current CISM domain. Currently, we rely on CISM to throw away any such fluxes, but it might be safer not to rely on that behavior, and instead for the coupler to enforce it.

Example: If CLM computes a SMB of 1 m for the CLM grid cell, the coupler (with area-conservative remapping) will end up throwing away 0.5 m in the above picture, because there will be no destination cell to map it onto. The extra 0.5 m will be conserved due to the routing of 0.5 m of snow capping to runoff in CLM, as laid out above. (NOTE, by tony: this is not correct. Unless we create masked mapping files, which I don't recommend, the mapping will not throw away the excess. What will happen is that the excess will be mapped onto a CISM gridcell where CISM will ignore it. This is partly why I'm concerned about the conservation in the case where CLM gives some of SMB to runoff and then we assume that CISM will ignore the equivalent amount internally after going through the coupler and mapping. But it theory, it should work.)

Example: If CLM computes a SMB of -1 m for the CLM grid cell: In the case where CLM was entirely in the CISM domain, this would be associated with -1 m of runoff, due to this code in HydrologyDrainageMod:

```
          if (glc_dyn_runoff_routing(g) .and. lun%itype(l)==istice_mec) then
             ! If glc_dyn_runoff_routing=T, add meltwater from istice_mec ice columns to the runoff.
             !    Note: The meltwater contribution is computed in PhaseChanges (part of
Biogeophysics2)
             qflx_qrgwl(c) = qflx_qrgwl(c) + qflx_glcice_melt(c)
```

But in the above case, only ½ of this negative SMB will be applied to CISM. Accordingly, with glc_dyn_runoff_routing = 0.5, only ½ of this negative SMB will have been routed to runoff in CLM. We can conserve by changing the above conditional code to instead read:

```
             qflx_qrgwl(c) = qflx_qrgwl(c) + qflx_glcice_melt(c) * glc_dyn_runoff_routing(g)
```

Example: Consider a case where both (a) the CLM grid cell is partially outside the CISM domain, and (b) the landfrac is less than 1 (section 1.3). This could happen at the land-ocean boundary, where part of the CLM grid cell is outside of the active CISM domain as well. In particular, assume: CLM landfrac = 0.7; active CISM domain fraction = 0.8. (Conceptually, the entire land portion is covered by active CISM cells, and there is another 0.1 grid cell's worth of active CISM cells.) Assume CLM generates an SMB of +1 m. Because the landfrac is 0.7, we want this to result in 0.7 m^3 / m^2 of ice sent to other components.

As laid out above in this section, (1 - 0.8) = 0.2 m of the SMB would be routed directly to calving. The coupler will multiply this by the landfrac, resulting in 0.14 m^3 / m^2.

As laid out in section 1.3, CISM will receive an SMB of 0.7 m. This gets applied over a fractional area of 0.8. This results in ice growth of 0.7*0.8 = 0.56 m^3 / m^2.

0.14 m^3/m^2 + 0.56 m^3/m^2 = 0.70 m^3/m^2, as desired.

Thus, it appears that the schemes laid out in this section and section 1.3 conserve in this double-edge case as well.

Alternative: An alternative is that CLM could send a single flux to the coupler, representing positive or negative SMB. There would be no distinction in CLM regarding icemask_coupled_fluxes / glc_dyn_runoff_routing. The coupler would then figure out what to route to CISM's SMB vs. what to route directly to runoff. This alternative is more appealing in that it simplifies some logic in CLM that is currently very hard to understand, and (I think) removes the need for CLM to know about the icemask_coupled_fluxes mask. However, the former implementation will be easier to accomplish in the current structure of the code, and avoids changing some CLM behavior (specifically, the meaning of the qflx_qrgwl variable outside of the CISM domain) – so I plan to try that first, falling back on this alternative if the first implementation proves difficult for some reason.

Alternative: It turns out that the exact method we use here ties in with the method we use for mapping ice sheet area -> CLM. See below for a discussion of an alternative method that could be used here if wanted to map ice sheet area -> CLM without normalization.

Ice sheet area -> CLM: We have two choices in sending ice sheet area to CLM:

1. Sum the glacier area from the CISM domain; assume the glacier area is 0 outside the CISM domain. This would be done via mapping without normalization.
2. Treat the glacier area in the CISM domain as a representative sample of the glacier area in the whole grid cell. This would be done via mapping with normalization.

Consider the case of a CLM grid cell at the land-ocean boundary. (This is probably the most important case to consider here. The other option is a CISM boundary in the middle of the land domain, but in that case, we would expect the CISM grid to be set up such that we expect 0 ice at the edge of the domain). The correct thing to do here, scientifically, seems to depend on what value of landfrac that CLM grid cell has. If it has a landfrac of 1, then it seems more correct to use method 1, hoping that CLM will call the remainder of the grid cell “wetland”. However, if the CISM domain lines up with the ocean model domain, then the landfrac would be less than 1. e.g., in the above picture, landfrac would be 0.5. In this case, we want the land cover in the CLM grid cell to represent the land cover of the non-ocean portion of the grid cell. In this case, that would mean using method 2. **I feel we should aim for a scheme that works correctly (scientifically) when the various models agree about the land-ocean boundary; so this argues for using method 2.**

But let’s see how the two methods fare in terms of conservation. The first examples implicitly assume a landfrac of 1. Then I’ll present an example with a landfrac less than 1.

Example: Method 1: Assume the CLM grid cell is ½ inside and ½ outside the CISM domain. Assume the CISM grid cell(s) that overlap this CLM cell dictate 0.1 EC 1, 0.2 EC 2, 0.7 bare. Then this CLM cell will have: 0.05 EC 1, 0.1 EC 2, 0.85 natural veg.

Does this conserve, in terms of the SMB flux? Assume that the snow capping flux is: 1 m in EC 1, 5 m in EC 2, 10 m in natural veg. For the sake of having actual quantities, assume that the CLM grid cell is 1000 m^2.

So the total snow capping quantity is: 0.05 * 1 * 1000 + 0.1 * 5 * 1000 + 0.85 * 10 * 1000 = 9050 m^3.

The SMB received by CISM is 0.1 * 1 * 500 + 0.2 * 5 * 500 + 0.7 * 10 * 500 = 4050 m^3.

The runoff flux is ½ of the snow capping quantity, so 4525 m^3.

So the total (SMB + runoff) flux is 8575 m^3.

This does NOT conserve!

Alternative: I believe it would be possible to achieve conservation in this case if we changed the logic determining how much SMB should be routed to the runoff flux. In particular: NONE of the SMB from glacier landunits should be routed to the runoff flux, but proportionally more of the bare land SMB should be routed to the runoff flux. In the above example, that means that the runoff flux would be 0.5 * 10 * 1000 = 5000 m^3 (where 0.5 is the fraction of the grid cell that is outside the CISM domain – which is now assumed to be entirely bare land, i.e., non-glacier). So we could still get conservation in this case, but I believe that it would add even more complexity to the equations routing SMB to runoff – and would especially add more complexity in the case where we potentially have more than one “bare land” column computing SMB. I don’t think the value (if any) is worth this extra complexity.

Example: Method 2: Assume the CLM grid cell is ½ inside and ½ outside the CISM domain. Assume the CISM grid cell(s) that overlap this CLM cell dictate 0.1 EC 1, 0.2 EC 2, 0.7 bare. Then this CLM cell will have: 0.1 EC 1, 0.2 EC 2, 0.7 natural veg.

Does this conserve, in terms of the SMB flux? Assume that the snow capping flux is: 1 m in EC 1, 5 m in EC 2, 10 m in natural veg. For the sake of having actual quantities, assume that the CLM grid cell is

1000 m^2.

So the total snow capping quantity is: 0.1 * 1 * 1000 + 0.2 * 5 * 1000 + 0.7 * 10 * 1000 = 8100 m^3.

The SMB received by CISM is 0.1 * 1 * 500 + 0.2 * 5 * 500 + 0.7 * 10 * 500 = 4050 m^3.

The runoff flux is ½ of the snow capping quantity, so 4050 m^3.

So the total (SMB + runoff) flux is 8100 m^3.

This DOES conserve!

Thus, in order for (SMB + snow capping runoff) to conserve, we need to use method 2, which involves mapping ice areas from CISM to CLM with normalization by the CISM domain.

Example: Method 2, with landfrac < 1: Let’s combine some of the above examples. Assume CLM landfrac = 0.7, active CISM domain fraction = 0.8. Assume the CISM grid cells that overlap this CLM cell dictate 0.1 EC 1, 0.2 EC2, 0.7 bare. Then this CLM cell will have: 0.1 EC 1, 0.2 EC 2, 0.7 natural veg. Assume that the snow capping flux is: 1 m in EC 1, 5 m in EC 2, 10 m in natural veg. For the sake of having actual quantities, assume that the CLM grid cell is 1000 m^2.

So the total snow capping quantity is: 0.1 * 1 * 1000 + 0.2 * 5 * 1000 + 0.7 * 10 * 1000 = 8100 m^3. Because the landfrac is 0.7, we want this to result in 5670 m^3 of ice sent to other components.

The SMB received by CISM is: 0.1 * 800 * 1 * 0.7 + 0.2 * 800 * 5 * 0.7 + 0.7 * 800 * 10 * 0.7 = 4536 m^3 (where the multiplications by 800 are because there is 800 m^2 of CISM area here, and the final multiplication by 0.7 of each term in the sum accounts for landfrac = 0.7, as laid out in section 1.3).

The SMB routed directly to calving is 0.2 times the total snow capping: 1620 m^3. The coupler will multiply this by the landfrac (0.7), resulting in 1134 m^3.

So the total (SMB + runoff) is 5670 m^3. **This does conserve!**

3. What normalizations do we need to accomplish the above, and to achieve scientific accuracy?

3.1. In general, do we want to weight by the ice fraction?

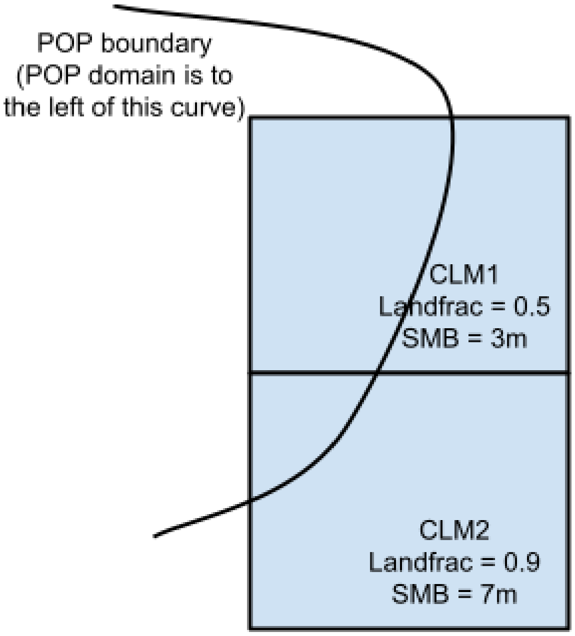
As shown here <https://drive.google.com/open?id=0B4hbp0iL0Is1flZDSlhRUkN5RTgtcUY3MzAtb3NUTURfLUNsV1pCMjZnNGVvaF9heVdMMGM&authuser=0> (update: this is now https://drive.google.com/drive/u/0/folders/1OLNVJ3GYyaRbCLB2rR8jsUoydiLu22oY), we want to weight by ice fraction when mapping fields from the ice grid to the land grid, but NOT when mapping fields from the land grid to the ice grid.

3.2. SMB from CLM -> CISM

For landfrac = 1, I think that we do *not* need any normalizations to handle the mapping of SMB from CLM -> CISM. However, we need to weight SMB by landfrac (i.e., the inverse of the ocean grid mapped to the land grid).

3.2.1. Should we use landfrac on the CLM grid or the CAM grid for doing this weighting?

Consider the following case:



Example: Case 1: Higher-resolution CAM grid than CLM grid (less likely in practice): For simplicity, consider the case where the CAM grid matches the CISM grid exactly. Assume that CLM1 is divided into 10 CAM/CISM grid cells, 5 of which are entirely inside the POP domain (landfrac=0), and 5 of which are entirely outside the POP domain (landfrac=1). Then this becomes the same as the landfrac case in section 1.3, in which the question was: Do we map the POP grid to CISM, or just apply CLM's landfrac uniformly? As with that question, I think it would be ideal to use CAM's landfrac here

– but then I think we'd need a mapping file from the ATM grid to the GLC grid. I'm pretty sure that we conserve in both cases, **so I'd be inclined to just go with CLM's landfrac for simplicity.**

Example: Case 2: Lower-resolution CAM grid than CLM grid (more likely in practice): Consider a CAM grid cell that exactly contains the above two CLM grid cells (which have equal area). From the above picture, we can see that its landfrac is 0.7. Let's say that the above SMB values were generated from 7 m of snow in the atmosphere, of which CLM1 generated 4 m runoff; there were no other water fluxes. So from the atmosphere's perspective:

- 7 * (0.5*0.5 + 0.5*0.1) = 2.1 m fell on the ocean
- 4 * (0.5*0.5) = 1 m ran off
- So CISM needs to absorb 7 - 3.1 = 3.9 m

For simplicity, assume that we have 10 CISM cells in each CLM cell: CISM 1-10 in CLM1, and CISM 11-20 in CLM2. (And the CLM grid cells are entirely within the CISM domain – i.e., there are active CISM cells even in the ocean portion of the above picture.) So the options are:

a. Use CLM's landfrac:
    i. CISM 1-10 get SMB = 3 * 0.5 = 1.5 m
    ii. CISM 11-20 get SMB = 7 * 0.9 = 6.3 m
    iii. Average is 3.9 m, as desired
b. Use CAM's landfrac:
    i. CISM 1-10 get SMB = 3 * 0.7 = 2.1 m
    ii. CISM 11-20 get SMB = 7 * 0.7 = 4.9 m
    iii. Average is 3.5 m – wrong!

CONCLUSIONS: We want to use landfrac on the land grid, *not* on the atmosphere grid

So, to restate the conclusions of these examples: We want to use landfrac on the LAND grid (*not* on the atmosphere grid).

Update 5-13-15: I realized a mistake in the above example: The amount of runoff was computed as 4 * (0.5*0.5) = 1 m. But in fact, LND -> ROF is merged using lfrac, which is landfrac on the atmosphere grid mapped to the land grid. So in fact, runoff in this case would be 4 * (0.5 * 0.7) = 1.4 m. So CISM needs to absorb 7 **-** 3.5 = 3.5 m. **So in fact, it is correct in this case to use CAM's landfrac! This means that we should be using lfrac rather than lfrin, which is consistent with section 3.2.3.**

3.2.2. Do we want to apply the landfrac values on the CLM grid or the CISM grid?

I cannot convince myself that either method is correct (see notes in this google drive folder – which does not have anything to do with CISM-specific mappings).

So for now I'm happy to go with Tony's suggestion, which follows:

I am going to argue that we want to use the landfrac conservatively mapped to the CISM grid for the "merge". We should call this a merge because what we're doing is

SMB_cism = mapl2g(SMB_lnd)*mapl2g(frac_lnd) +
mapl2g(SMB_non_lnd)*(1.-mapl2g(frac_lnd))

where

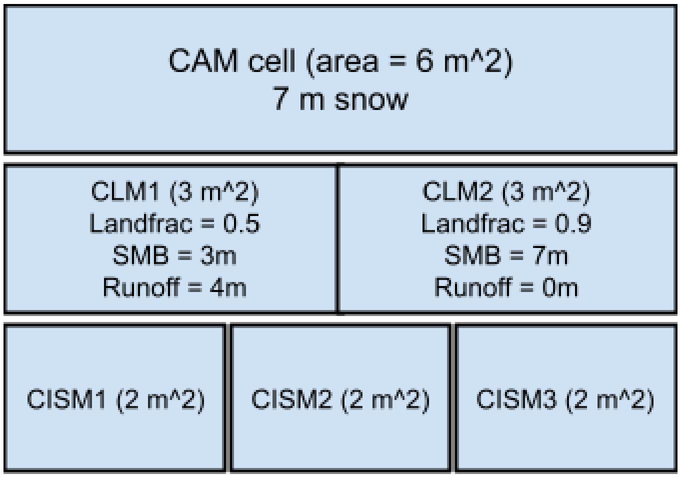
SMB_non_lnd = 0.

and I would say we want to do this operation on the CISM grid which is consistent with all other merges. It's done on the destination grid of the merge.

Update: Tony says: I'm pretty sure we're OK with conservation. Diffusion in mapping means the mapping is not reversible. In other words, you cannot map atm to ocean conservatively and then the resulting field back to atm and recover the initial field. But everything is locally and globally conservative to the extent the gridcells overlap. The other thing is that the "merge" equation and the "map" implementation are intricately tied together. The fractions used in the merge have to be mapped with the same weights at the fields being conserved otherwise conservation breaks down. Part of what's happening in the map/merge is there is some cancellation of terms if you look at the data movement from start to end.

So we’re probably fine to go with Tony’s suggestion, above.

Update (4-30-15): Consider something like “Example: Case 2” in section 3.2.1, but with a CISM cell that overlaps both land cells:



Assume that the above grid cells encompass the entire world (yes, there is one atmosphere cell that covers the globe, and yes, the entire globe is 6 m^2) – except there is an implicit ocean grid cell not pictured here, whose area is (3 m^2 * 0.5 + 3 m^2 * 0.1) = 1.8 m^2. There are no other water fluxes besides the ones shown above. For simplicity, assume that the runoff grid is exactly the same as the CLM grid (so landfrac on the runoff grid matches landfrac on the CLM grid).

Then, as in the above “Example: Case 2”:

- There is a total of 6*7 = 42 m^3 snow
- 7 * 1.8 = 12.6 m^3 fell on the ocean
- 4 * (3 * 0.5) = 6 m^2 ran off from CLM1
- So CISM needs to absorb 42 - 12.6 - 6 = 23.4 m^3

Method 1: Apply landfrac mapped from the land grid to the CISM grid – i.e., this involves applying landfrac on the destination grid only:

CISM1 (landfrac = 0.5): SMB = (3 m) * 0.5 = 1.5 m => 3 m^3
CISM2 (landfrac = 0.7): SMB = (0.5 * 3m + 0.5 * 7m) * 0.7 = 3.5 m => 7 m^3
CISM3 (landfrac = 0.9): SMB = (7 m) * 0.9 = 6.3 m => 12.6 m^3

Total SMB: 22.6 m^3. **This does NOT conserve**

Method 2: Apply landfrac on the land grid – i.e., apply landfrac on the source grid, without normalization. (Note: I believe that seq_map_map currently does not allow you to do this: if you want a weighting on the source grid, then you also need to use normalization. But let’s ignore that limitation for now.)

CISM 1 & CISM3 are as in Method 1:

- CISM1: (3 m) * 0.5 = 1.5 m => 3 m^3
- CISM3: (7 m) * 0.9 = 6.3 m => 12.6 m^3

CISM2: SMB = (0.5 * 0.5 * 3m + 0.5 * 0.9 * 7m) = 3.9 m => 7.8 m^3

Total SMB: 23.4 m^3. **This DOES conserve.**

Method 3: Apply landfrac on the land (source) grid with normalization, and then apply landfrac on the CISM (destination) grid using landfrac values mapped from the land grid to the CISM grid:

CISM 1 & CISM3 are as in Method 1:

- CISM1: (3 m) * 0.5 = 1.5 m => 3 m^3
- CISM3: (7 m) * 0.9 = 6.3 m => 12.6 m^3

CISM2:

- SMB before application of landfrac on the destination grid: (0.5 * 0.5 * 3m + 0.5 * 0.9 * 7m) / (0.5 * 0.5 + 0.5 * 0.9) = 5.571428571428572 m
- Landfrac mapped to the CISM grid: 0.7
- SMB after application of landfrac on the destination grid: 5.57… * 0.7 = 3.9 m => 7.8 m^3

Total SMB: 23.4 m^3. **This DOES conserve.**

It is not accidental that this gives the same result as Method 2: The application of normalization in the mapping means we divide by landfrac on the destination grid. So, by later multiplying by landfrac on the destination grid, we are simply undoing this normalization.

Note: I'm pretty sure that this method also works in the case laid out in section 1.2 (CISM grid cell partially outside the CLM domain), as long as landfrac on the CISM grid was mapped from LND -> CISM without normalization. Then, if a CISM grid cell is ½ outside the CLM domain, the flux would be multiplied by ½ (even if landfrac on the CISM grid cell is 1).

Method 4: Apply landfrac on the CISM (destination) grid, using landfrac values determined by directly mapping the ocean grid to the CISM grid.

Example 1: Assume that mapping the ocean grid to the CISM grid results in the following landfrac values on the CISM grid (note that these are consistent with the landfrac values on the CLM grid in the above picture; in this example I'm assuming that the landfrac in the CISM2-CLM1 area is the same as the landrac in the CISM2-CLM2 area):

- CISM1: 0.345
- CISM2: 0.81
- CISM3: 0.945

CISM1: SMB = (3 m) * 0.345 = 1.035 m => 2.07 m^3
CISM2: SMB = (0.5 * 3m + 0.5 * 7m) * 0.81 = 4.05 m => 8.1 m^3
CISM3: SMB = (7 m) * 0.945 = 6.615 m => 13.23 m^3

Total SMB: 23.4 m^3. **This DOES conserve.**

Example 2: Assume that mapping the ocean grid to the CISM grid results in the following landfrac values on the CISM grid (note that these are consistent with the landfrac values on the CLM grid in the above picture; in this example, I'm allowing for differences in the landfrac in the CISM2-CLM1 area vs the CISM2-CLM2 area):

- CISM1: 0.425
- CISM2: 0.8
    - area that overlaps with CLM1: 0.65
    - area that overlaps with CLM2: 0.95
    - overall landfrac: 0.8
- CISM3: 0.875

CISM1: SMB = (3 m) * 0.425 = 1.275 m => 2.55 m^3
CISM2: SMB = (0.5 * 3m + 0.5 * 7m) * 0.8 = 4.0 m => 8.0 m^3
CISM3: SMB = (7 m) * 0.875 = 6.125 m => 12.25 m^3

Total SMB: 22.8 m^3. **This does NOT conserve.**

Alternative: Rather than simply using the landfrac determined by mapping the ocean to the CISM grid, instead (somehow) account for the different landfrac values that exist for each overlap region between the CLM grid and the CISM grid. (I imagine this would involve some complex mappings….)

In this case, CISM1 and CISM3 cells have the same SMB as above.

CISM2: SMB = (0.5 * 3m * 0.65 + 0.5 * 7m * 0.95) = 4.3 m => 8.6 m^3

Total SMB: 2.55 m^3 + 8.6 m^3 + 12.25 m^3 = 23.4 m^3. **This DOES conserve.**

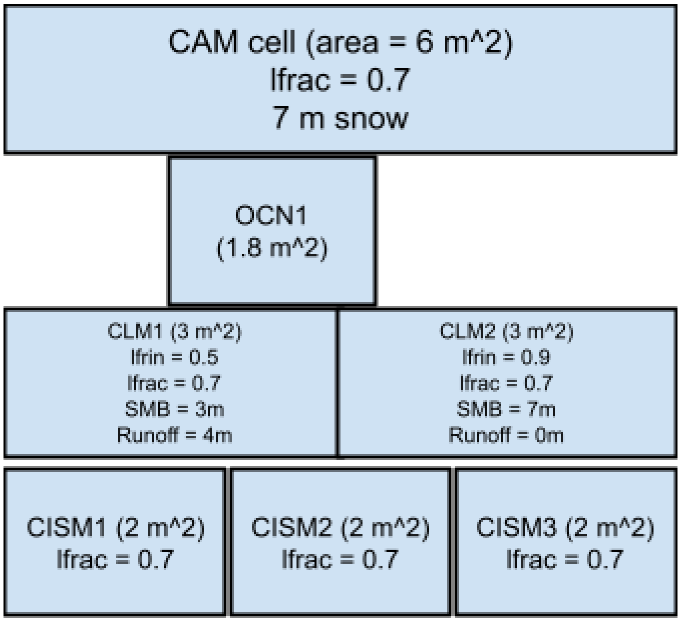
Conclusions from this method: In order to conserve when using Landfrac determined directly on the CISM grid, you actually need to use the Landfrac values on the intersection of the CISM and CLM grids. Although doing this would give you a more accurate division of SMB into the individual CISM grid cells, it seems hard to justify the extra complexity that this would entail.

Conclusions: We could use methods 2 or 3, both of which involve multiplying by landfrac on the land (source) grid. (We could also use method 4, if applied correctly, but it seems much too complex for now.) I slightly prefer method #3: (a) multiplying by landfrac on the source grid, with normalization, then (b) – for fluxes only – applying landfrac on the destination grid, using landfrac values that have been mapped [without normalization] from the land grid to the CISM grid. I slightly prefer this over #2 because we'll want to do mapping with normalization for states, so it makes the mapping of states and fluxes more similar – the only difference between the two is that fluxes will have an extra step at the end.

3.2.3. Should we use lfrac or lfrin in the normalization?

The example in 3.2.2 glossed over the difference between lfrac and lfrin. Here we extend that example to consider this difference. This is considering something similar to section 3.2.1, but being more rigorous about it.

Here I am assuming that we're using Method 3 from 3.2.2: Apply landfrac on the land (source) grid with normalization, and then apply landfrac on the CISM (destination) grid using landfrac values mapped from the land grid to the CISM grid.



Note that CAM's lfrac is 1 - ofrac. CLM's lfrac is determined by mapping CAM's lfrac to the land grid, so lfrac is the same in both CLM cells. CLM's lfrin comes from its domain file, so lfrin differs in the two CLM cells. CISM's lfrac is determined by mapping CLM's lfrac to the CISM grid, so it is 0.7 in all cells. (I am not 100% positive that I have this correct, but this is my understanding of how it works.)

As above, assume that the above grid cells encompass the entire world. The runoff grid is exactly the same as the CLM grid, so lfrac on the runoff grid matches lfrac on the CLM grid. (The system does not compute lfrin on the runoff grid, so that is not important here.)

Since the runoff grid matches the CLM grid, the normalization term for fluxes sent from CLM -> ROF is not important. The CLM -> ROF merge uses lfrac (as is currently done in the code).

So:

- There is a total of 6*7 = 42 m^3 snow
- 7 * 1.8 = 12.6 m^3 fell on the ocean
- ROF1 gets:
  - Before merge: 4 m
  - After merge: 4 * 0.7 = 2.8 m => 8.4 m^3
- ROF2 gets 0 m^3 runoff
- So CISM needs to absorb 42 - 12.6 - 8.4 = 21.0 m^3

Example 1, Method 1: Use lfrac as a normalization term in CLM -> CISM mapping.

CISM1: (3 m) * 0.7 = 2.1 m => 4.2 m^3

CISM3: (7 m) * 0.7 = 4.9 m => 9.8 m^3

CISM2:

- SMB before application of lfrac on the destination grid: (0.5 * 0.7 * 3m + 0.5 * 0.7 * 7m) / (0.5 * 0.7 + 0.5 * 0.7) = 5.0 m
- SMB after application of landfrac on the destination grid: 5.0 * 0.7 = 3.5 m => 7.0 m^3

Total SMB: 21.0 m^3. **This DOES conserve**

Example 1, Method 2: Use lfrin as a normalization term in CLM -> CISM mapping.

CISM1 & CISM3 are the same as in Method 1.

CISM2:

- SMB before application of lfrac on the destination grid: (0.5 * 0.5 * 3m + 0.5 * 0.9 * 7m) / (0.5 * 0.5 + 0.5 * 0.9) = 5.571428571428572 m
- SMB after application of landfrac on the destination grid: 5.57… * 0.7 = 3.9 m => 7.8 m^3

Total SMB: 21.8 m^3. **This does NOT conserve.**

Based on the above examples, we should use lfrac as a normalization term, *not* lfrin.

Now consider a slightly different picture: Instead of having 4 m of runoff, CLM1 has 4 m of evaporation and 0 m of runoff. Again, there are no other water fluxes in the system besides the ones shown.

We have two choices for how evaporation is mapped to the atmosphere: normalizing by lfrin (as is currently done) or normalizing by lfrac.

Normalizing by lfrin:

- Before merge (in the mapping) atm gets:
  (0.5 * 0.5 * 4m + 0.5 * 0.9 * 0m) / (0.5 * 0.5 + 0.5 * 0.9) = 1.4285714285714286 m
- After merge using lfrac, atm gets: 1.42… * 0.7 = 1.0 m => 6.0 m^3
- So CISM needs to absorb 42 - 12.6 - 6.0 = 23.4 m^3
- As shown above, neither method results in this SMB value
- **So it appears that a normalization by lfrin in the lnd -> atm mapping does NOT conserve.**

Normalizing by lfrac:

- Before merge (in the mapping), atm gets:
  (0.5 * 0.7 * 4m + 0.5 * 0.7 * 0m) / (0.5 * 0.7 + 0.5 * 0.7) = 2.0 m
- After merge using lfrac, atm gets: 2.0 m * 0.7 = 1.4 m => 8.4 m^3
- So CISM needs to absorb 42 - 12.6 - 8.4 = 21.0 m^3

- As shown above, this SMB is achieved if we use lfrac as a normalization in the CLM -> CISM mapping.
- **So normalization by lfrac in the lnd -> atm mapping conserves if we also normalize by lfrac (rather than lfrin) in the lnd -> glc mapping.**

So it appears that the only way we can get conservation is if we use lfrac as the normalization term in all mappings: CLM -> CISM as well as CLM -> CAM, etc.

3.2.4. Do we want to normalize by cismfrac?

I think this relates to: If there are two CLM cells, one entirely in the domain, one partially outside, and a cism cell that overlaps both: do we want to give more weight to the CLM cell that's entirely in the domain?

I believe the answer is: NO, we do not want to normalize by cismfrac. Similarly to other arguments (laid out above) for not weighting CLM grid cells based on ice area: The accuracy of the SMB flux from a grid cell does not depend on cismfrac. However, this statement depends in part on what we do for SMB in the case of a CLM grid cell partially outside the CISM domain. Currently, the plan laid out above (in section 2.3) is to maintain the full SMB in this case. If we do that, then the accuracy of SMB is independent of cismfrac. If, however, we changed the scheme so that it decreased the SMB flux in this case (which I think would be accompanied by a later normalization), then we may want to revisit this point.

3.3. Ice areas from CISM -> CLM

I think that we want to normalize by the CISM domain mask. As laid out above, this domain mask will be dynamic in time (sent from CISM to CPL).

3.4. Do we need to do area-corrections for anything other than fluxes?

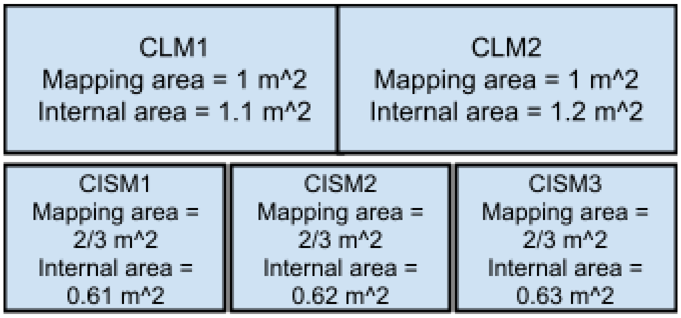
Each CESM component has two representations of the area of each grid cell: The areas on the mapping file (which are the grid cell areas from the coupler's point of view), and the areas internal to that component. In order to achieve conservation, fluxes (but not states) are multiplied by area correction factors when being sent to and from components.

Tony gives the example: take a single gridcell, that covers the same lon/lat region in both the atm and ocean model. maybe the area of the atm is 2.5, the mapping area is 2.45, and the ocean is 2.55. we wouldn't generally expect the areas to be that different but maybe they are.

anyway, if the flux out of the atm model is 1.0, then we correct that by multiplying by 2.5/2.45 on the way out of the atm model, we map (assume the weight is 1.0), and then we multiply the result by 2.45/2.55. that means the flux of 1 becomes 0.9804 on the ocean side. in that way 1.0*2.5 = 0.9804*2.55 and we conserve quantity.

We should do this same area-correction for fluxes as is done elsewhere in the system (I believe that is done automatically, with no other work needed on our part). **The question is: should we also do some sort of area-correction for ice fractions and/or topographic height, since these also need to be mapped conservatively? Or is the area correction for fluxes sufficient to get conservation, even if the total ice areas disagree between CISM and CLM due to differences in the models' internal representations of areas?**

Consider the following picture:



CLM1
Mapping area = 1 m^2
Internal area = 1.1 m^2

CLM2
Mapping area = 1 m^2
Internal area = 1.2 m^2

CISM1
Mapping area = 2/3 m^2
Internal area = 0.61 m^2

CISM2
Mapping area = 2/3 m^2
Internal area = 0.62 m^2

CISM3
Mapping area = 2/3 m^2
Internal area = 0.63 m^2

Example 1: Mapping SMB to CISM, with part bare ground, part ice-covered.

Assume that:
- CISM1: bare ground
- CISM2: ice-covered
- CISM3: bare ground

And the following fluxes:
- CLM1: 1 m snow (1.1 m^3)
- CLM2: 1.5 m snow (1.8 m^3)
- (So a total of 2.9 m^3)
- Bare land portions all evaporate
- ice portions all accumulate SMB (no evaporation, runoff, etc.)

So evaporation from CLM1 is ⅔ m (0.733333 m^3, after multiplying by 1.1 m^2), from CLM2 is 1.0 m (1.2 m^3)

SMB to the ice-covered portion from CLM1 is 1 m, from CLM2 is 1.5 m

SMB sent to CISM:
- SMB to cpl in ice-covered EC:
    - CLM1: 1 m * 1.1 / 1.0 = 1.1 m

    - CLM2: 1.5m * 1.2 / 1.0 = 1.8 m
    - Average SMB in cpl: 1.45 m
- SMB to cism: 1.45 m * (⅔) / 0.62 = 1.55913978…

So ice volume added in CISM = 1.55913978… * 0.62 m^2 = 0.966666 m^3

Total volume: evaporation of 0.73333 m^3 + 1.2 m^3; ice volume in CISM = 0.966666 m^3; total = 2.9 m^3

So this conserves, without needing to do any extra area corrections beyond those done for the fluxes

Example 2: Mapping SMB to CISM, with all ice-covered, with different topographic heights in the same elevation class.

This checks whether we need to do any extra area corrections in order for the vertical interpolation to conserve.

Assume that:
- CISM1: 50 m
- CISM2: 60 m
- CISM3: 80 m

Then we have:
- CLM1: 100% ice, with topo = ⅔ * 50 + ⅓ * 60 = 53.3333
- CLM2: 100% ice, with topo = ⅓ * 60 + ⅔ * 80 = 73.3333

Assume fluxes as in Example 1, so the only fluxes are ice accumulation (SMB): 1 m from CLM1, 1.5 m from CLM2 (total 2.9 m^3 as in Example 1).

Assume CLM1 has beta (SMB gradient with height) = 0.03 m SMB / m; CLM2 has beta = 0.04 m SMB / m

As in Example 1, SMB in the coupler is 1.1 m from CLM1, 1.8 m from CLM2.

We have the following SMB values sent to CISM, before the area correction:
- CISM1: 1.1m + 0.03 * (50 - 53.3333) = 1.0 m
- CISM3: 1.8m + 0.04 * (80 - 73.3333) = 2.06666 m
- CISM2
    - From CLM1: 1.1m + 0.03 * (60 - 53.3333) = 1.3 m
    - From CLM2: 1.8m + 0.04 * (60 - 73.3333) = 1.26666 m
    - Average: 1.283333 m

The area correction multiplies these fluxes by (⅔) / 0.61, (⅔) / 0.62, and (⅔) / 0.63, for CISM1, CISM2 and CISM3, respectively. Then multiply by 0.61, 0.62 and 0.63 to get the area-integrated ice volume in each CISM cell. This results in a sum of 2.9 m^3.

So again, we conserve without doing any area corrections beyond the normal ones done for fluxes.

Example 3: Basal heat flux sent from CISM to CLM

Assume that CISM produces the following basal heat fluxes:

- CISM1: 0.1 => area-integrated = 0.061
- CISM2: 0.2 => area-integrated = 0.124
- CISM3: 0.3 => area-integrated = 0.189
- So area-integrated sum = 0.374

Assume CISM1 and CISM3 are bare ground, CISM2 is ice-covered.

The area corrections when these are sent to cpl result in: 0.1 * 0.61/(⅔), 0.2 * 0.62/(⅔), 0.3 * 0.63/(⅔) = 0.0915, 0.186, 0.2835

To CLM, we get:

- CLM1, bare: 0.0915 * 1.0/1.1 = 0.083181818…
- CLM1, ice: 0.186 * 1.0/1.1 = 0.169090909…
- CLM2, ice: 0.186 * 1.0/1.2 = 0.155
- CLM2, bare: 0.2835 * 1.0/1.2 = 0.23625

CLM1 grid cell average = ⅔ * 0.083181818 + ⅓ * 0.169090909 = 0.11181818… => area-integrated (multiply by 1.1) = 0.123

CLM2 grid cell average = ⅓ * 0.155 + ⅔ * 0.23625 = 0.20916666 => area-integrated (multiply by 1.2) = 0.251

Total area-integrated = 0.374

So once again, we conserve without doing any area corrections beyond the normal ones done for fluxes

Conclusions: It appears that we do NOT need any area corrections beyond the normal ones used for fluxes. This means that CLM and CISM will disagree about the total glacier area, due to differences in their internal representations of grid cell areas. But fluxes should still be conserved.

Notes on CLM/CISM/POP overlap, 12/1/16

(The context here is that we're thinking about how to compute global sums on the land and CISM grid in order to apply a conservation correction after doing bilinear mapping of SMB from land to glc.)

There are three grids of interest: CLM, CISM and POP. Generically, these are lnd, glc and ocn grids, but here I'll use the component names.

CLM cells are rectangular and do not conform to the land-ocean boundary. Some part of each CLM cell can overlap the POP grid, meaning that part of the cell is not really land. The model defines lfrac as the fraction of a CLM cell that does *not* overlap the POP grid.

- lfrac is important for water conservation. For example, suppose we have an SMB of 1 m/yr in a CLM cell with lfrac = 0.2. Then 80% of the snow that falls within the area occupied by the land cell is routed by the coupler to POP. Only 20% falls on land and has the potential to add to the SMB in CISM. The fraction of the CLM-computed SMB sent to CISM should not exceed lfrac. That is, the area-averaged SMB sent to CISM from this cell should not exceed 0.2 m/yr.

The CISM domain can be divided into two parts:

- Ice-covered and/or land-covered. The SMB is applied only in these regions. A positive SMB leads to accumulation, and a negative SMB (if the ice has nonzero thickness) leads to ablation.
- Ocean-covered. The SMB is not applied over ocean-covered cells.

These two parts are identified by Sg_icemask_g. Where Sg_icemask_g = 1 (ice- and/or land-covered), SMB can be applied; where Sg_icemask_g = 0 (ice-free ocean), the SMB is ignored.

Each CLM cell has a certain fractional overlap with the ice/land-covered part of the CISM domain. This fraction is denoted by Sg_icemask_l, which is the mapping of Sg_icemask_g from CISM to CLM. Note that while Sg_icemask_g is binary (0 or 1), Sg_icemask_l can have intermediate values.

- Sg_icemask_l is also important for water conservation. If a given CLM cell is in Antarctica (and the CISM domain is Greenland), then Sg_icemask_l = 0, and the entire SMB should be routed to the ocean. If the cell straddles Ellesmere and Greenland (assuming lfrac = 1) with Sg_icemask_l = 0.5, then 50% of the SMB should be sent to CISM, and the remaining 50% should be routed to the ocean. The fraction of the SMB sent to CISM should not exceed Sg_icemask_l.

SMB budgeting is straightforward when lfrac and Sg_icemask_l are independent. If a CLM cell has Sg_icemask = 1, then the fraction of its SMB sent to CISM should be proportional to lfrac. Conversely, if a CLM cell has lfrac =1, then the fraction of its SMB sent to CISM should be proportional to Sg_icemask_l.

But in fact, these fractions are not independent. There are many land cells with 0 < lfrac < 1 and 0 < Sg_icemask < 1, because these cells (1) partly overlap the POP grid, and (2) partly overlap the ocean-covered part of the CISM grid. The two overlap regions are positively correlated. That is, if a portion of a CLM cell is ocean-covered (as defined by the POP grid), it is more likely than not to coincide with an ocean-covered part of the CISM grid. But the overlap regions are not perfectly correlated. It is possible to have a CISM cell that is ice/land-covered but overlaps the POP grid, or a CISM cell that is ocean-covered but does not overlap the POP grid.

If we assume that the SMB fraction sent from CLM to CISM is proportional to the product lfrac*Sg_icemask_l, then we are effectively assuming zero correlation. As a result, we will underestimate the SMB fraction that should be sent to CISM.

If, on the other hand, we assume a perfect correlation (lfrac = Sg_icemask_l), the SMB fraction sent from CLM to CISM will be proportional to lfrac (or equivalently to Sg_icemask_l). But this assumption is clearly wrong for glaciated regions outside the CISM domain (Antarctica, the Himalayas, etc.), where lfrac = 1 and Sg_icemask_l = 0. For regions within the CISM domain, the assumption is at least slightly wrong, because CISM and POP do not exactly agree on where the land-ocean boundary lies. In general, we will overestimate the SMB fraction sent to CISM if we assume a perfect correlation.

To summarize: We want the SMB fraction sent to CISM not to exceed lfrac and also not to exceed Sg_icemask_l. But where lfrac and Sg_icemask_l are exactly or approximately equal (as will be the case over much of the CISM domain), we want the SMB fraction not to be significantly less than lfrac or Sg_icemask_l.

The easiest way to satisfy these constraints is to specify that the SMB fraction sent to CISM is **min(lfrac, Sg_icemask_l)**. In other words, when summing the SMB over CLM grid cells, the contribution from each cell should be weighted by min(lfrac, Sg_icemask_l) * area_l, where area_l is the grid cell area.

Within CLM, the SMB can be divided into 3 fractions that sum to 1:

- min(lfrac,Sg_icemask_l); this fraction is sent to CISM via the coupler
- lfrac - min(lfrac,Sg_icemask_l); this fraction is sent to the runoff model via the coupler
- (1 - lfrac); this fraction is ignored by the coupler, because any precipitation that could contribute to the mass balance has already fallen into the ocean

To make this more concrete, consider the total CLM area obtained by different summation methods, on a 1-degree (f09) land grid that receives Sg_icemask_l from a 4-km Greenland grid:

1. The sum over cells of lfrac * area_l is 3.6662. This is the total land area. (The area units are such that the Earth's radius = 1, so the total area of land plus ocean is 4*pi.)
2. The sum over cells of Sg_icefrac * area_l is **5.3861e-2**.
3. The sum over cells of lfrac * Sg_icemask_l * area_l is **5.1279e-2**.
4. The sum over cells of min(lfrac,Sg_icemask_l) * area_l is **5.2956e-2**.

On the CISM side, the sum over cells of Sg_icemask_g * area_g is **5.4009e-2** if we use the native CISM areas (which are distorted by the stereographic projection), and **5.3856e-2** if we use the coupler areas.

Summation method 2 gives the closest agreement between SMB-eligible CLM and CISM area. For some grid cells, however, this method violates the requirement that the SMB fraction sent to

CISM should not exceed lfrac. (The local violations generally are small for the f09 grid, but can be larger on coarser grids.) Method 4 is slightly less accurate, but it satisfies the constraints not to exceed lfrac or Sg_icemask_l. Method 4 is more accurate than method 3, which is the only other method satisfying both constraints.

Suppose the SMB is a uniform 1 m/yr on the CLM grid. With bilinear mapping, this becomes a uniform SMB of 1 m/yr on the CISM domain. Then, if we use the native areas for the CISM summation, the renormalization factor will be 5.2956/5.4009 = 0.9803, giving an adjusted SMB of 0.9803 m/yr on the CISM domain, or a difference of ~2%. This is the error associated with grid masking, prior to any conservation errors associated with bilinear horizontal mapping and linear vertical mapping.

The combined errors can be a bit larger. Over the course of a 100-year TG run, the differences in summed annual SMB on the two grids are almost always <5% (but usually are <2%).

So I think the renormalization factor will be acceptably close to 1. The SMB in a given CLM grid cell will map to a similar SMB (within ~2%) in CISM. Local differences would likely be larger in a purely conservative method, which would need to adjust SMBs by as much as 5-10% to compensate for CISM area differences between the coupler and native grids.

Implementation of smooth, conservative SMB remapping in the coupler

The problem is to remap SMB between the land (CLM) and glc (CISM) grids such that

1. The remapping is smooth and continuous on the CISM grid, without obvious imprinting of the CLM grid.
2. The remapping is exactly conservative. By exact conservation, we mean that the sum over CLM grid cells of the SMB sent to the ice sheet model (and therefore removed from the water budget of the land model) is equal (within machine roundoff) to the sum over CISM grid cells of the SMB received from the land model.
3. To the extent possible, the SMB values applied in CISM at a given location are close to the SMB values computed at that location for that elevation class in CLM. Further, SMB is remapped without sign changes; any positive (negative) SMB in CLM maps to a positive (negative) SMB in CISM.

To preserve the sign of SMB as described in (3), we divide CLM domain into an accumulation region and an ablation region. Both regions are restricted to SMB-eligible grid cells: i.e., grid cells with min(lfrac, Sg_icemask_l) > 0. The accumulation region consists of all elevation classes in all SMB-eligible grid cells with SMB > 0, and the ablation regions consists of all elevation classes in all SMB-eligible grid cells with SMB < 0. It does not matter where we assign elevation classes with SMB = 0.

Let a_i = min(lfrac, Sg_icemask_l) and let A_i = the area of grid cell i. Let q_{ik} = the SMB in elevation class k of cell i, and let f_{ik} = the fractional area occupied that elevation class. The total SMB over the CLM accumulation region is

$$Q_{acc} = \sum_i a_i A_i \sum_{k \in q > 0} f_{ik}\, q_{ik}.$$

The notation beneath the second summation sign can be read as "sum over all elevation classes k with q > 0 (i.e., positive SMB). The expression for the total SMB over the ablation region is identical, except for the sign of q in the second sum:

$$Q_{abl} = \sum_i a_i A_i \sum_{k \in q < 0} f_{ik}\, q_{ik}.$$

On the CISM grid, there are no elevation classes; we simply sum the SMB over grid cells. The total SMB over the accumulation region is

$$Q^*_{acc} = \sum_{j \in q > 0} A_j\, q_j,$$

where A_j is the area of grid cell j, and the notation beneath the summation sign can be read as "sum over all cells j with q > 0." Similarly, the total SMB over the ablation region is

$$Q^*_{abl} = \sum_{j \in q < 0} A_j\, q_j.$$

The SMB remapping is done in coupler module prep_glc_mod.F90, in a new subroutine called prep_glc_map_qice_conservative_lnd2glc. For the SMB, this subroutine is called from prep_glc_calc_l2x_gx in lieu of the simpler subroutine prep_glc_map_one_field_lnd2glc. The basic procedure is as follows:

1. Compute the integrals Q_{acc} and Q_{abl} on the CLM grid.
2. Use the smooth, non-conservative bilinear remapper to map q_{ik} from the CLM grid to the CISM grid. Do this for each elevation class k.
3. For each CISM grid cell j, use linear interpolation in the vertical to compute the SMB at the cell elevation h_j.
4. Compute the integrals Q^*_{acc} and Q^*_{abl} for the remapped SMB on the CISM grid.
5. For each cell j in the accumulation region, multiply q_j by $Q_{acc} / Q^*_{acc.}$ For each cell j in the ablation regions, multiply q_j by Q_{abl} / Q^*_{abl}.

This procedure ensures that the total SMB in the CISM accumulation region is equal to the total SMB in the CLM accumulation region, and similarly for the ablation regions. Below, I describe each step in detail.

The integrals in (1) are straightforward to compute given all the required terms on the land grid, but it is non-trivial to ensure that all terms are present in all CESM configurations. Apart from the SMB, Flgl_qice, the required terms are as follows:

- a_i = min(lfrac, Sg_icemask_l) for each land cell i.
    - The land fraction lfrac is exported from attribute vector fractions_lx, which is passed in through the argument list.
    - Sg_icemask_l is obtained by a conservative (flux-type) horizontal mapping of Sg_icemask_g, which is native to the glc grid. (In principle, Sg_icemask_g can evolve in time as ice shelves advance and retreat, so it cannot be computed once and for all at initialization.) In some configurations, Sg_icemask_l may already be present and up to date in the coupler. But in other configurations (e.g., TG), Sg_icemask might not be mapped from the glc grid to the land grid. To handle all cases with an active ice sheet (including TG), I wrote some code to compute Sg_icemask_l by (1) exporting Sg_icemask_g from attribute vector g2x_gx, (2) importing Sg_icemask_g into a temporary attribute vector, Sg_icemask_g_av, (3) doing a flux-type mapping of Sg_icemask_g to temporary attribute vector Sg_icemask_l_av, and (4) exporting Sg_icemask_l to a local field. We should probably revisit this procedure in the future. It might be more straightforward to force a mapping of Sg_icemask_g from the glc to the land grid for all configurations with an active ice sheet.

- A_i = grid cell area for each land cell i. In the code, this field is called area_l, and it is obtained from dom_l, the derived-type mct_ggrid object associated with the land grid.
- f_{ik} = fractional area in each elevation class on the land grid. This is contained in fields Sg_ice_covered00, Sg_ice_covered01, …, Sg_ice_covered10 (i.e., 11 fields for the case of 10 elevation classes plus a bare-land class). In some configurations, these fields are obtained by mapping Sg_ice_covered (which has a value of 0 or 1 for each glc grid cell) from the glc grid to the land grid. But since this is not always the case (e.g., for TG configurations), I wrote some code to map Sg_ice_covered from the glc grid to the land grid. This is done by (1) creating a temporary attribute vector, g2x_lx, to hold the remapped fields Sg_ice_covered** and Sg_topo**, (2) calling the existing subroutine map_glc2lnd_ec to map from attribute vector g2x_gx to g2x_lx, and (3) exporting Sg_ice_covered** to a local array called frac_l. We should revisit this procedure too, since it may be more straightforward to force a mapping of Sg_ice_covered from the glc to the land grid for all configurations with an active ice sheet.
- q_{ik} = SMB in each elevation class on the land grid. This field is exported from the attribute vector l2gacc_lx, which contains the glc-specific fields accumulated and averaged by the land model.

The sums Q_{acc} and Q_{abl} are done on the local processor, followed by a call to subroutine shr_mpi_sum to obtain the global values.

Operations (2) and (3) are carried out in module map_lnd2glc.F90. The main subroutine, map_lnd2glc, is given the attribute vector l2x_l as an input argument. This attribute vector contains the SMB fields (Flgl_qice00, Flgl_qice01, etc.), surface temperature fields (Sl_tsrf00, Sl_tsrf01, etc.) and topography fields (Sl_topo00, Sl_topo01, etc.) for each elevation class on the land grid. Here we consider the remapping of the SMB. Surface temperature is treated similarly, but since it is a state rather than a flux, it does not require a post-hoc correction for conservation.

Subroutine map_lnd2glc first calls subroutine map_bare_land, which remaps the bare-land SMB from the land grid to the glc grid. CISM cells are initially given bare-land values, but these values are subsequently overwritten in ice-covered cells.

Next, subroutine map_lnd2glc calls a new subroutine, map_ice_covered, which supersedes subroutine map_one_elevation_class. (The older subroutine does conservative but non-smooth vertical interpolation.) The new subroutine creates a temporary attribute vector, l2x_g_temp, which contains the SMB and topo fields for each elevation class (though not bare land) on the glc grid. It uses the existing subroutine seq_map_map to do a billinear (smooth but not conservative) mapping of the SMB and topo fields from the land grid to the glc grid.

Once we have l2x_g_temp, it is straightforward to do the vertical interpolation that formerly was done by Glint. The fields in l2x_g_temp are exported into local arrays. For each CISM grid cell, the code determines the two adjacent elevation classes (one above and the other below the CISM elevation), and linearly interpolates the SMB between the values at adjacent elevations. If a CISM cell's elevation lies below the lowest EC in CLM, it is given the SMB value from the lowest EC, and similarly for CISM cells lying above the highest EC.

After looping through all the CISM cells, the resulting SMB is imported into attribute vector l2x_g, which is returned to subroutine prep_glc_map_qice_conservative_lnd2glc.

Subroutine prep_glc_map_qice_conservative_lnd2glc then does step (4), computing the integrals Q^*_{acc} and Q^*_{abl} on the CISM grid. In addition to the remapped SMB, these integrals require CISM grid cell areas. These areas, called area_g, are contained in dom_g, the derived-type mct_ggrid object associated with the glc grid. After the sums are computed on the local processor, they are converted to global values with a call to shr_mpi_sum.

Step (5) is straightforward, consisting of a multiplicative correction for each cell on the CISM grid.