# A Firn Model for CLM

William Lipscomb
August 2013

## Introduction

Glaciers and ice sheets form when snowfall survives the melt season, accumulates over many years, and eventually turns to ice under its own weight. Old snow (i.e., snow that has survived a melt season but not yet turned to ice) is called *firn*. Firn is granular and compacted as a result of metamorphic processes, including recrystallization. Compared to fresh snow, firn is relatively impermeable to air and water. The firn layer can be tens of meters thick, with typical densities of ~500 kg/m$^3$ or more and grain sizes of 0.5 to 5 mm. As the firn densifies, the grains become larger and air pockets become smaller, until the air is cut off from contact with the atmosphere. At this point the firn has turned to glacier ice with a density of ~850 to 900 kg/m$^3$. (The maximum density of pure ice is 917 kg/m$^3$.) See Cuffey and Paterson (2010) for details.

We want to turn CLM's existing snowpack model into a firn model for glacier and ice sheet simulations. Initially we would like to develop a firn model with a minimum of physics changes. That is, we would use CLM's existing mechanisms for snow densification and meltwater percolation and refreezing. Changes would be primarily numeric—for instance, allowing more and thicker snow layers.

Currently there is a maximum of 5 snow layers above the ice or soil. Snow is assumed to turn to ice when it exceeds a depth of 1 m water equivalent (w.e.). Reviewers of the recent *J. Climate* papers have pointed out that this is not realistic. In particular, the thin snowpack limits the amount of meltwater retention and refreezing. Thus the low depth threshold for snow-to-ice conversion could bias the glacier surface mass balance, especially in temperate regions where annual accumulation and ablation can exceed 1 m w.e.

The depth limit can easily be increased in the current model, but all additional snow is put in the lowest layer, which becomes very thick and has poor vertical temperature resolution. In the new model we want to add layers while choosing layer depths intelligently. The distinction between fresh snow and firn is somewhat arbitrary, but generally the upper snow layers (down to depths of ~1 m) will be fresh snow and the lower snow layers will be firn.

Some questions to answer:

- What should be the maximum number of snow layers, to ensure we have adequate vertical resolution through the firn?
- What rules do we use to add and take away layers?
- What should be the maximum snow depth (in meters w.e.)?

- Should we prescribe a maximum snow density, above which firn turns to ice?
- What CLM code changes are needed?

*Notes:*

- From CLM's point of view, firn is just snow that happens to be dense. We will use the existing snow data structure to model firn, rather than add a new data structure for firn.

- If snow compaction is treated realistically, we might not need to prescribe a maximum snow depth. Instead, we could convert firn to ice when it reaches a density threshold (e.g., 800 kg/m$^3$). But when a layer many meters thick reaches the threshold, it is not obvious how to convert it to ice; it might be numerically dangerous to convert the full layer to ice instantaneously. So it may be better to retain a maximum snow depth as the criterion for snow-ice conversion. In this case we should verify that snow densities remain physically reasonable (< 900 kg/m$^3$).

- These changes would likely be applied only in runs with glacier_mec landunits.

- We want not only to model the firn layer of glaciers and ice sheets, but also to allow vegetated landunits to develop a thick firn layer that turns to ice.

- We will probably want to change the definition of the SMB in CLM so that it includes mass changes for the entire snow-ice column, not just changes in the mass of ice.

## Current snowpack scheme in CLM

Here we briefly describe CLM's current snowpack scheme, focusing on the numerics. See the CLM 4.5 Technical Note (Oleson et al., 2013), especially Section 7.2, for details.

The snowpack can have up to 5 layers, indexed from -4 (the top layer) to 0 (the layer just above the soil or ice). Layer node depth is $z$, layer interface depth is $z_h$ (with the snow/soil interface at $z_{h0}$), and layer thickness is $\Delta z$. State variables in each layer are water mass $w_{\mathrm{liq}}$ (kg/m$^2$), ice mass $w_{\mathrm{ice}}$ (kg/m$^2$), thickness $\Delta z$, and temperature $T$. Water vapor is ignored. For snow thinner than 0.01 m, there are no explicit layers, and the state variable is the snow mass $W_{\mathrm{sno}}$ (kg/m$^2$).

The ground fraction covered by snow, $f_{\mathrm{sno}}$, is computed based on Swenson and Lawrence (2012). There are conservation equations for $w_{\mathrm{ice}}$ (incorporating phase changes and surface snowfall, deposition, and sublimation) and $w_{\mathrm{liq}}$

(incorporating phase changes, downward percolation, and surface rainfall, condensation, and evaporation). When $w_{\mathrm{liq}}$ is greater than a layer's holding capacity, excess water is added to the layer below, limited by that layer's porosity. Liquid water reaching the soil surface (for vegetated landunits) can either infiltrate or run off, whereas liquid water reaching the ice surface (for glacier and glacier_mec landunits) is assumed to run off.

Particles of black carbon, organic carbon, and mineral dust can be deposited from the atmosphere. These particles affect the snow radiative properties, including albedo. Particle mass is conserved when snow layers are divided or combined. Particles reaching the bottom of the snow column run off with the meltwater and do not enter the soil or ice.

If there are no existing snow layers, but snow depth $z_{sno} > 0.01$ m following snowfall, then layer 0 (the bottom layer) is initialized with $\Delta z_0 = z_{\mathrm{sno}}$, $w_{\mathrm{ice},0} = W_{\mathrm{sno}}$, and $w_{\mathrm{liq},0} = 0$.

Snow is compacted after the soil hydrology calculations are done. Three processes contribute to snow compaction: (1) destructive metamorphism of new snow (i.e. breakdown of crystals), (2) snow overburden pressure, and (3) melting. The layer compaction rate, $(1/\Delta z)(\partial \Delta z/\partial t)$, is given by the sum of these processes. Compaction does not occur if the layer is saturated or if $w_{\mathrm{ice}} < 0.1$.

The key compaction process for firn is snow overburden. The overburden compaction rate, based on Anderson (1976), is a linear function of the snow load pressure $P_s$ (kg/m$^2$):

$$C_R = \frac{1}{\Delta z}\frac{\partial \Delta z}{\partial t} = \frac{P_s}{\eta}, \tag{1.1}$$

where $\eta$ is a viscosity coefficient (kg s m$^{-2}$) that varies with density and temperature:

$$\eta = \eta_0 \exp\left[ c_5 (T_f - T) + c_6 \frac{w_{ice}}{f_{sno}\Delta z} \right]. \tag{1.2}$$

Here, $\eta_0 = 9 \times 10^5$ kg s m$^{-2}$, $c_5 = 0.08$ K$^{-1}$, and $c_6 = 0.023$ m kg$^{-1}$ are prescribed constants, and $T_f$ is the freezing point of water in Kelvin. The pressure is computed as a sum of the ice and water mass above the layer, plus half the mass of the layer being compacted. This equation implies that cold snow is more viscous (and thus compacts more slowly) than warm snow, and dense snow is more viscous than light snow.

Let's do a back-of-the-envelope calculation to make sure this compaction rate will turn firn to glacier ice in a reasonable time. Let $P = 5\times10^4$ kg m$^{-2}$, corresponding to a 100-m firn layer with density 500 kg m$^{-3}$. Let $T_f - T = 20$ K, and let $w_{\text{ice}}/(f_{\text{sno}}\Delta z) = 500$ kg m$^{-1}$. Then the term in brackets in (1.2) is 13.1, with the dominant contribution (11.5) from the second (density) term. The exponential is then $\sim 5\times10^5$, giving $\eta \sim 5\times10^{11}$ kg s m$^{-2}$. We obtain C$_R$ ~ $10^{-7}$ s$^{-1}$, implying a compaction time scale of $\tau = 1/C_R \sim 0.3$ yr, which is quite short.

But increasing the density term to $w_{\text{ice}}/(f_{\text{sno}}\Delta z) = 900$ kg m$^{-1}$, the exponential is ~ $5\times10^9$, giving $\tau = 3000$ yr, which is a long time. For every increase of 100 kg m$^{-1}$ in the density term, the time scale increases by a factor of ~10 (whereas the time scale decreases by only a factor of 2 for a doubling of the firn depth.) Thus the time required to turn firn into ice is sensitive to the threshold density at which the conversion occurs.

For temperate snow (e.g., in mountain glaciers) we can ignore the first term in brackets. This results in a viscosity that is ~5 times smaller for a given snow thickness and density, and thus a compaction time scale that is ~5 times shorter. But for mountain glaciers, the pressure term would typically be ~5 times smaller than assumed above (e.g., we might have a 20-m firn layer instead of a 100-m firn layer), so the compaction time would have a similar order of magnitude at given density. (We could turn this logic around to say that the reason firn layers are thinner in mountain glaciers is that the snow is warmer and less viscous, hence lower pressure is needed to compact snow into ice.)

Next we describe how layers are combined and divided in CLM, following Jordan (1991).

Table 7.2. Minimum and maximum thickness of snow layers (m)

| Layer | $\Delta z_{\min}$ | N$_l$ | N$_u$ | $(\Delta z_{\max})_l$ | $(\Delta z_{\max})_u$ |
|---|---|---|---|---|---|
| 1 (top) | 0.010 | 1 | >1 | 0.03 | 0.02 |
| 2 | 0.015 | 2 | >2 | 0.07 | 0.05 |
| 3 | 0.025 | 3 | >3 | 0.18 | 0.11 |
| 4 | 0.055 | 4 | >4 | 0.41 | 0.23 |
| 5 (bottom) | 0.115 | 5 | >5 | — | — |

Each layer has a minimum thickness as shown in Table 7.2 of the CLM 4.5 Tech Note, reproduced above. (Note that the table uses layer indices of 1 to 5 instead of -4 to 0. Also note that these are actual thicknesses, as opposed to depths in water equivalent.) The minimum thickness is smallest for the top layer and increases with depth. If a layer's thickness is smaller than the minimum, then it is combined with a neighboring layer. The combination rules are as follows:

- If the top layer is being removed, it is combined with the underlying layer.
- If the underlying layer is not snow (i.e., it is the top soil layer), the layer is combined with the overlying layer.
- If the layer is nearly completely melted, the layer is combined with the underlying layer.
- If none of the above rules apply, the layer is combined with the thinnest neighboring layer.

Layer combination conserves liquid mass, ice mass, and heat content. After a layer is removed, the number of snow layers decreases by one and the layer indices are altered accordingly. Node depths and layer interfaces are then recomputed.

Table 7.2 also shows the maximum layer thicknesses, which depend on the current number of layers. If the number of layers $N = N_l$, then the maximum thickness is given by ( $\Delta z_{\max}$ )l, but if $N > N_l$, then the maximum thickness is given by ( $\Delta z_{\max}$ )u. For example, layer 1 has a maximum thickness of 0.03 m if it is the only layer, but only 0.02 m if there is a layer beneath.

Layers are checked in order from top to bottom. If a layer thickness is greater than the maximum, it is subdivided according to the following rules:

- If there is just one layer with a thickness greater than 0.03 m, it is divided into two layers of equal thickness, liquid water and ice mass, and temperature.
- If an underlying layer exists, then the thickness, liquid water and ice mass, and temperature of the excess snow (i.e., the snow exceeding the max thickness) are added conservatively to the underlying layer.
- If there is no underlying layer after this adjustment (i.e., the excess snow is not thick enough to form a new layer), then the layer is subdivided into two layers of equal thickness, liquid water and ice mass, and temperature.

Node depths and layer interfaces are then recalculated. The CLM Tech Note describes how temperature is reconstructed to maintain the vertical temperature profile while preserving heat content.

## Modified snowpack scheme for CLM

From a numerical point of view, the key to developing a firn model is to generalize Table 7.2 to an arbitrary number of snow layers. For layers 3 and higher, the entries can be generated recursively as follows:

$$\Delta z_{\min}^{n} = \frac{\left(\Delta z_{\max}\right)_u^{n-1}}{2} \tag{1.3}$$

$$\left(\Delta z_{\max}\right)_u^n = 2\left(\Delta z_{\max}\right)_u^{n-1} + 0.01 \tag{1.4}$$

$$\left(\Delta z_{\max}\right)_l^n = \left(\Delta z_{\max}\right)_u^n + \left(\Delta z_{\max}\right)_l^{n-1} \tag{1.5}$$

Using these relationships we can extend Table 7.2, as shown below for the case of 12 layers.

Extended Table 7.2. Minimum and maximum thickness of snow layers (m)

| Layer | $\Delta z_{\min}$ | $N_l$ | $N_u$ | $(\Delta z_{\max})_l$ | $(\Delta z_{\max})_u$ |
|---|---|---|---|---|---|
| 1 (top) | 0.010 | 1 | >1 | 0.03 | 0.02 |
| 2 | 0.015 | 2 | >2 | 0.07 | 0.05 |
| 3 | 0.025 | 3 | >3 | 0.18 | 0.11 |
| 4 | 0.055 | 4 | >4 | 0.41 | 0.23 |
| 5 | 0.115 | 5 | >5 | 0.88 | 0.47 |
| 6 | 0.235 | 6 | >6 | 1.83 | 0.95 |
| 7 | 0.475 | 7 | >7 | 3.74 | 1.91 |
| 8 | 0.955 | 8 | >8 | 7.57 | 3.83 |
| 9 | 1.915 | 9 | >9 | 15.24 | 7.67 |
| 10 | 3.835 | 10 | >10 | 30.59 | 15.35 |
| 11 | 7.675 | 11 | >11 | 61.30 | 30.71 |
| 12 | 15.355 | 12 | >12 | 122.73 | 61.43 |

With *N* = 12 this scheme supports a snowpack thickness of ~100 m, which should be adequate for a firn model.

Let's return to the questions posed in the introduction, with some provisional answers:

*Q:* What should be the maximum number of snow layers, to ensure we have adequate vertical resolution through the firn?

*A:* If we extend the existing scheme using the recursion relations (1.3)–(1.5), then about 12 layers should be sufficient, assuming a maximum allowed snow depth of ~100 m (~50 m w.e. assuming a mean snow density of 500 kg m$^{-3}$). For a maximum snow depth of 20 m (~10 m w.e.), about 10 layers should be sufficient. We should run the model with some different values to test the sensitivity to the maximum number of snow layers.

But a remaining question is whether the recursion scheme, in which each layer is roughly twice as thick as the one above, gives adequate vertical resolution as the snowpack becomes thick (~10 m or more). We could compare this scheme to another scheme in which layer thicknesses asymptote to a uniform value in the lower layers. But I'm unclear on whether the existing combination and subdivision subroutines would work for layers of uniform thickness, so I'll leave that question open for now.

*Q:* What rules do we use to add and take away layers?

*A:* If we simply extend Table 7.2, then we can use the same rules as in current CLM.

*Q:* What should be the maximum snow depth (in meters w.e.)?

A: If the answer is based on observations, we might want a maximum of ~100 m w.e. for ice sheets, ~10 m w.e. for glaciers. (See Cuffey and Paterson, 2010.) But it's possible that for firn thicker than ~10 m w.e., the results will asymptote, and it will not matter if we make the layers any thicker. So it may be reasonable to start with a limit of 10 m w.e. (i.e., 10 times greater than the current limit) and then test the sensitivity to larger and smaller values.

*Q:* Should we prescribe a maximum snow density, above which firn turns to ice?

*A:* Based on the back-of-the-envelope calculation, it will take a long time (centuries to millennia) for firn to reach a density of > 900 kg $m^{-3}$. We should verify that the snow density does not exceed this value when we run the model. Prescribing a maximum snow depth of ~10 m w.e. would probably ensure that densities remain physically reasonable.

If we were to use a density threshold instead of a thickness threshold for snow-ice conversion, then we would have to work out how to smoothly turn snow to ice in thick lower layers that reach the density threshold at a given time.

*Q:* What CLM code changes are needed?

A: Hopefully not too many. Here are the ones I can think of:

- Increase `nlevsno` (the maximum number of snow layers) in *clm_varpar.F90* to the appropriate value. If it is desired to maintain nlevsno = 5 for landunits other than glacier_mec, then we could define a separate parameter (`nlevsno_glcmec`?), but this would lead to some additional logic elsewhere in the code. My preference would be to allow more snow layers for all landunits, given that in practice the number of layers with nonzero snow depth will be determined by the value of `h2osno_max`, which can remain the same for landunits other than glacier_mec.

- Increase `h2osno_max` (the maximum snow depth in w.e.) in *clm_varcon.F90* to an appropriate value for glacier_mec landunits. Maybe we should define a new parameter (`h2ofirn_max`?) that applies only to glacier_mec landunits, so that users have the option of running with one value for glacier_mec landunits and a smaller value for other landunits. But when we have dynamic landunits, we may want the same maximum snow depth for glacier_mec and vegetated landunits, so that vegetated regions will glaciate given sufficient snow accumulation.

Note that `h2osno_max` is used in two places: *mkarbinit_mod.F90* and *clm_driver_init.F90*. In *clm_driver_init.F90,* we set `do_capsnow(c) = .true.` if and only if `h2osno(c) > h2osno_max`. Here we would need some additional logic to deal with glacier_mec landunits limited by `h2ofirn_max`.

In *mkarbinit_mod.F90,* `h2osno_max` is used to initialize the snow depth for glacier landunits (`h2osno(c) = h2osno_max`) and glacier_mec landunits (`h2osno(c) = 0.5 * h2osno_max`). If glacier_mec landunits are allowed to have thick firn layers, then it might be better not to initialize these landunits with a snow depth equal to a significant fraction of `h2ofirn_max` (since `h2ofirn_max` can be large). For simplicity, maybe it's better to initialize glacier and glacier_mec landunits to the same value, possibly a new parameter called `h2osno_init` (which could be declared in *clm_varcon.F90*). Then we can test the sensitivity of the spun-up model state to `h2osno_init`, which would likely be in the range 1 to 10 meters.

- In *SnowHydrologyMod.F90*, subroutine *Combine_Snow_Layers*, we should declare `dzmin` and `dzminloc` to have dimension `nlevsno`, instead of hardwiring the dimension to 5. The data statement for `dzmin` should include all the values in the extended table above.

- In *SnowHydrologyMod.F90*, subroutine *Divide_Snow_Layers,* more work will be needed, because the existing subroutine has unfortunate hardwiring. Instead of declaring arrays `dzmax_l` and `dzmax_u`, the values of these parameters are hardwired in the code. There are several 'if' blocks corresponding to the cases of `msno` = 1, >1, >2, >3, and >4 snow layers. We could simply add more 'if' blocks, but it would be better to declare `dzmax_l` and `dzmax_u` and to consolidate the subdivide logic into a single 'do' loop from 1 to `msno`. (Life is never easy.)

- Make sure the value `nlevsno = 5` is not hardwired elsewhere in the code. I grepped on '5' and found these minor issues:
    - Subroutine *DivideSnowLayersLake* has hardwired layer numbers similar to subroutine Divide SnowLayers.

  - Subroutine *snow_depth2levLake* is hardwired with the current values in Table 7.2, assuming a maximum 5 snow layers. Since this is just an initialization routine it may not matter, but it would be good coding practice to remove the hardwiring of parameters.
  - A comment in subroutine *Biogeophysics2* states that "Soil / snow temperature is predicted from heat conduction in 10 soil layers and up to 5 snow layers." There is a similar comment in subroutine *SoilTemperature*.

- Make the interpinic changes needed to restart the model from a file with a different value of `nlevsno`.

## Snowpack schemes in regional climate models

Here I'll review some other models, mainly the detailed snowpack model used in the regional climate model MAR (Modèle Atmosphérique Régionale). The point of this review is to better understand how CLM's snowpack scheme compares to what's considered a top-of-the-line scheme in a regional climate model.

The MAR snowpack model is described here (thanks to Xavier Fettweis for these references):

- Vionnet et al. (2012): http://www.geosci-model-dev.net/5/773/2012/gmd-5-773-2012.html

- Lefebvre et al. (2003): http://www.agu.org/pubs/sample_articles/cr/2001JD001160/2001JD001160.pdf

- Xavier Fettweis's PhD thesis: http://orbi.ulg.ac.be/bitstream/2268/36720/1/These-Xavier_Fettweis-2006.pdf

- Reijmer et al. (2012): http://www.the-cryosphere.net/6/743/2012/tc-6-743-2012.pdf

**Vionnet et al. (2012)** characterize CLM's snowpack model (Oleson et al. 2010) as a model of *intermediate complexity*:

> Acknowledging the limitations of single-layer schemes, snowpack schemes of intermediate complexity were developed to account for some internal processes such as snow settling, water percolation and refreezing. These schemes generally vertically discretize the snowpack with a prescribed number of layers (from 2 to 5, generally) (Boone and Etchevers, 2001; Loth

> and Graf, 1998; Lynch-Stieglitz, 1994). In these schemes, most snowpack physical properties are parameterized as a function of snow density, which is a surrogate for taking into account snow ageing (Boone and Etchevers, 2001).

These intermediate models are distinguished from more detailed models such as Crocus:

> A few detailed snowpack models belong to the third class and account explicitly for the layering of its physical properties. They include a more or less explicit description of the time evolution of the snow microstructure. This includes the models SNTHERM (Jordan, 1991), Crocus (Brun et al., 1989, 1992) and SNOWPACK (Bartelt and Lehning, 2002).

Sections 2 and 3 of Vionnet et al. (2012) describe Crocus in detail. Briefly, Crocus is a 1D multilayer snow scheme that simulates snow evolution as a function of energy and mass exchange between the snow and atmosphere. Input variables for Crocus are (as for CLM) near-surface temperature, specific humidity, wind speed, incoming shortwave (direct and diffuse) and longwave, precipitation rate (both rain and snow), and surface air pressure.

Snow is stratified parallel to the local slope. (So there is a slope angle not included in CLM.) Primary state variables for each layer are thickness $D$, heat content $H$ (i.e., enthalpy, from which temperature and water content are diagnosed), density $\rho$, and age $A$. Other variables (dendricity $d$, sphericity $s$, grain size $g_s$, and a historical variable $h$ describing whether there was once liquid water or faceted crystals) describe snow grain evolution under metamorphism. The density of new snow is a function of wind speed and air temperature (eq. 1), with a prescribed minimum of 50 kg/m$^3$ and a maximum (in practice) of ~200 kg/m$^3$.

Section 3.2 describes the vertical discretization: "The dynamical evolution of the number and thicknesses of the numerical snow layers is a key and original feature of Crocus, which aims at simulating the vertical layering of natural snowpacks in the best possible way (Brun et al., 1992)." The number of snow layers $N$ is user-defined, with a minimum $N = 3$. There is no limit on the maximum, but typical Crocus values are $N_{\max} = 20$ or 50. The authors note that "the snowpack scheme dynamically manages a different vertical grid mesh, in terms of the number and the thickness of snow layers, for each grid point when it is run in parallel mode for a spatially distributed simulation." Indexing is from 1 at the top to $N$ at the bottom.

The rules (quoted verbatim from the paper) are as follows:

- For snowfall over a bare soil, the snowpack is built up from identical layers, in terms of thickness and state variables. Their number, $N$, depends on the amount of fresh snow, $D_{\text{new}}$, and on the maximum number of layers, $N_{\text{max}}$:

$$N = \max[N_{\text{min}}, \min(N_{\text{max}}, \ 100D_{\text{new}})]$$

where $D_{\text{new}}$ is in meters.

- For snowfall over an existing snowpack, it is first attempted to incorporate the freshly fallen snow into the existing top layer, provided its grain characteristics are similar and its thickness is smaller than a fixed limit. The similarity between two adjacent layers is determined from the value of the sum of their differences in terms of $d$, $s$ and $g_s$, each weighted with an appropriate coefficient. If the merging is not possible, a new numerical layer is added to the preexisting layers. If the number of layers then reaches its maximum, a search is carried out to identify two adjacent layers to be merged. This is done by minimizing a criterion balancing the similarity between their respective grain characteristics and their thicknesses.

- For no snowfall, a check is carried out to see whether it is convenient to merge too thin snow layers or to split those which are thick. This is achieved by comparing the present thickness profile to an idealized profile, which acts as an attractor for the vertical grid. This idealized thickness profile depends on the current snow depth and on the user-defined maximal number of layers. Figure 4 shows two examples of such an idealized profile. Merging two layers is only possible for those which are similar enough in terms of grain characteristics. Grid resizing affects only one layer per time step, with a priority given to the surface and bottom layers, in order to accurately solve the energy exchanges at the surface and at the snow/soil interface.

  *Note: Figure 4 of the paper shows that the model favors several thin layers near the surface, followed by several thicker, equally spaced layers, and finally a single thin layer at the snow-soil interface. I'm unclear on the importance of the single thin layer at the lower interface.*

- For most time steps, no grid resizing is carried out, except that the thickness of each layer decreases according to its compaction rate.

When two or more layers are combined, their average depth-weighted optical grain size is conserved, to ensure smooth evolution of the surface albedo.

The parameterizations of snow metamorphism and compaction, wind drift, albedo, surface fluxes, heat diffusion, snow melting, and percolation/refreezing are described in Sections 3.2 to 3.10. Since we're hoping not to change what CLM does already, I'll skip the descriptions.

I could not find any discussion of what happens if and when the snow density reaches a threshold for conversion to ice.

**Lefebre et al. (2003)** describe an application of Crocus to West Greenland. Some numerical parameters of interest:

- A fresh snow layer is added when there is a snowfall event of at least 2 mm w.e.
- $N_{\max} = 20$
- When an internal layer becomes thinner than 5 mm, it is combined with the layer below.

The validation experiments are short-term, and there is no mention of snow compaction to form ice.

Meltwater reaching the snow-ice interface is assumed to run off. But since it does not run off instantly, there is time for superimposed ice to form. In reality, the authors say, meltwater at the snow-ice interface percolates into the ice (which is somewhat porous) and refreezes.

**Reijmer et al. (2012)** compare parameterizations of refreezing on the Greenland ice sheet. Both RACMO2 and MAR, in which refreezing is calculated explicitly, are used as benchmarks to evaluate six parameterizations. Results from RACMO2 are similar to results from MAR. RACMO2 uses the SOMARS snowpack model (Simulation Of glacier surface Mass balance And Related Sub-surface processes, Greuell and Konzelman, 1994). MAR uses Crocus (Brun et al., 1992), as described above.

In SOMARS, layer thicknesses range from 6.5 cm near the surface to 4 m at 30 m depth. Layer state variables are temperature, density, liquid water content, depth, and thickness. Layer thicknesses can change at each timestep. Details of meltwater percolation and snow densification are given in the paper.

In Crocus, layer thicknesses range from < 1 cm near the surface to ~1 m at 10 m depth, with thicknesses changing at each timestep. Layer state variables are temperature, density, liquid water content, depth, thickness, and three snow grain parameters: dendricity, sphericity and descriptive grain size. Other physics details are given in the paper (and in the Vionnet et al. paper summarized above).

**Xavier Fettweis's thesis** (2006) gives a detailed summary of MAR, including the 1D surface model SISVAT (Soil Ice Snow Vegetation Atmosphere Transfer). The snow component of SISVAT is described as follows:

> The SISVAT snow-ice model is an one-dimensional multi-layered energy

> balance model that determines the exchanges between the sea ice, the ice sheet surface, the snow-covered tundra, and the atmosphere. It consists of a thermodynamic module, a water balance module, a turbulence module, a snow metamorphism module, a snow/ice discretization module, a blowing snow module, and an integrated surface albedo module. It is based on the CEN (Centre d'Etudes de la Neige) snow model called CROCUS (Brun et al., 1992) and its physics and validation are described in details in Gallée and Duynkerke (1997), Gallée et al. (2001), and Lefebre et al. (2003).

More quotations from Xavier's thesis:

> The snow metamorphism parametrizations are taken from the CROCUS model. The snow pack is described by its (gradient of) temperature, its liquid water content, its density, its age as well as the size and the form of the snow grains. Freshly fallen snow (called dendritic snow) is described by its dendricity and sphericity. Dendricity describes the part of the original crystal shapes which are still remaining in a snow layer and always decreases from 1 for fresh dendritic-shaped crystals to 0. Sphericity describes the ratio of rounded versus angular shapes. The dendritic snow grains evolve rapidly through disintegration and combined sublimation-deposition processes which also tend to dissipate the smaller particles in favour of bigger ones.
>
> When dendricity becomes equal to 0, the snow grains arrive at the stage of rounded (sphericity = 1) crystals, faceted (sphericity = 0) crystals or at an intermediate state, depending on the temperature gradients that were present in the snow pack. The snow grains are now called non-dendritic snow grains and are characterized by their sphericity and their descriptive grain size. Sphericity again describes the ratio of rounded versus angular shapes while the descriptive grain size indicates the average size of the snow crystals (Lefebre, 2002).
>
> The snow/ice discretization module manages the snow pack vertical discretization. The total number of snow layers may change during the simulation. The snow grid has a maximum of 20 snow layers which have a variable thickness and the splitting or aggregation of snow layers is controlled by the CROCUS snow metamorphism laws. This is done in such a way that the natural stratigraphy of the snow pack is preserved. A fresh snow layer is added to the snow pack when enough snow is available and the CROCUS parametrizations are used to determine the density, dendricity and sphericity of the fresh snow layer.