**Recap**

Giving this another try. Starting with the simpler constraint given in the paper,

(1)

The idea is to begin with the requirement that

(2)

where depends on the usual diffusion and source terms, and the right-hand-side is the function specified in Eq. 1 but with replacing . The value of is arbitrary, since the intention is that it will go away by the time we’re done. If we solve the above equation for and take a Taylor expansion of Eq. 1 about , we find that really does cancel out, giving

(3)

where is meant to remind us that this is the result of a Taylor expansion. By contrast, the original algorithm (in the paper) is very similar to this,

(4)

The “source” term in Eq. 3 is identical to the one in Eq. 4 (omitting a factor of that I think shouldn’t be there -- in any case, in our simulations, so in practice it doesn’t make a difference). The big difference between Eqs. 3 and 4 is .

|  |  |
| --- | --- |
|  |  |
| **Figure 1**. Steady state using the original formulation (i.e., Eq. 4), with . | |

In numerical simulations, I’ve found that controls whether the system achieves steady-state: the smaller , the more steps in the steady state. Fig. 1 shows the results when , which is 1/1000 the value used in the paper’s “reference state.” This run exhibited a diffusive slowdown of about 2%.

Figure 2 shows results with the *biggest* value of I can specify and still get converged results without interrupting to update (again assuming other parameters assume their “reference” values), , which is 1/100 the value used in the paper’s “reference state.” Here, the simulation runs fast enough that it’s easy to go well past the first point of “steady state” (here, at ) shown. What is revealed here, is that the steady state is not actually steady: the profile exhibits an inversion. This run also exhibited a diffusive slowdown of about 2%.

|  |
| --- |
|  |
| **Figure 2**. Layer difference (left), and profiles of states close to (middle) and the ending state (right), using the original formulation for (i.e., Eq. 4), and as shown. |

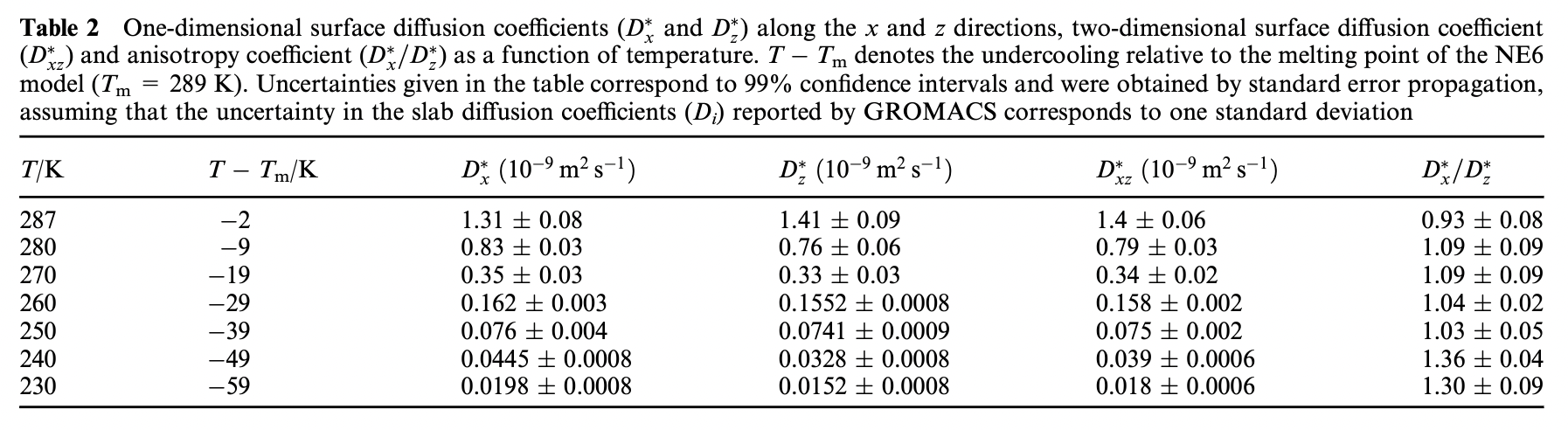
If we make the size of the crystal bigger, we can get to smaller diffusion coefficients. Figure 3 shows a run having a crystal twice as long as the previous, and , which is 1/20 the value used in the paper’s “reference state.”

|  |
| --- |
|  |
| **Figure 3**. Layer difference (left), and profiles of states close to (right), using the original formulation for (i.e., Eq. 4), as shown, but a bigger crystal. |

**Focusing on 260 K**

Now we’re going to see if we can be more specific about the temperature, say . According to Gladich et al’s, Table 2 (reproduced below), at that temperature we should have .

*From Gladich et al paper:*



We also need the kinetic deposition rate. The formula is given by Neshyba et al’s Eq. 2, which will be our Eq. 5 here:

(5)

where is the density of ice, is the density of water vapor in equilibrium with the solid, and is the mass of a water molecule. Using this, we get . That’s a lot bigger than what is used in the paper, , but that value would apply to a lower temperature, probably .

Bearing in mind that computational stability requires that source term in Eq. 4 needs to be big compared to the diffusion term, (variable *Doverdeltax2* in the code), we can note the following:

1. The source term, (which remember we want to be big), increases with bigger , so it’s good that we’re up to now. The supersaturation, designated in the paper as (see Eq. 13 of the paper, and appearing in the old code as variable *sigmastepmax, revised to sigmaIcorner*), had a value of for the reference system in the paper. If we bump this up to , it buys us some maneuvering room. As long as , we’ll get growth.
2. In the diffusion term, the key quantity is (which remember we want to be small). We’re stuck with if we want to adhere to what Gladich et say for in their Table 2. We do have some control over , however: it’s determined by how we discretize. If we choose a bigger crystal with the same number of points, is bigger and therefore is smaller, so that’s good. There’s a limit to this, however, because if gets too small, we lose the ability to resolve the steps. It seems like needs to no greater than to achieve this.

Figure 4 shows a run having a crystal (so ), with parameters corresponding to a temperature of . It achieves what we’ve been calling a steady state, although now we recognize that it inverts.

|  |
| --- |
|  |
| **Figure 4**. Layer difference (left), and profile close to (right), with , , and as shown. For this discretization, . |

*My lazy way of doing Clausius-Clapeyron*

*(https://byjus.com/clausius-clapeyron-equation-calculator/)*

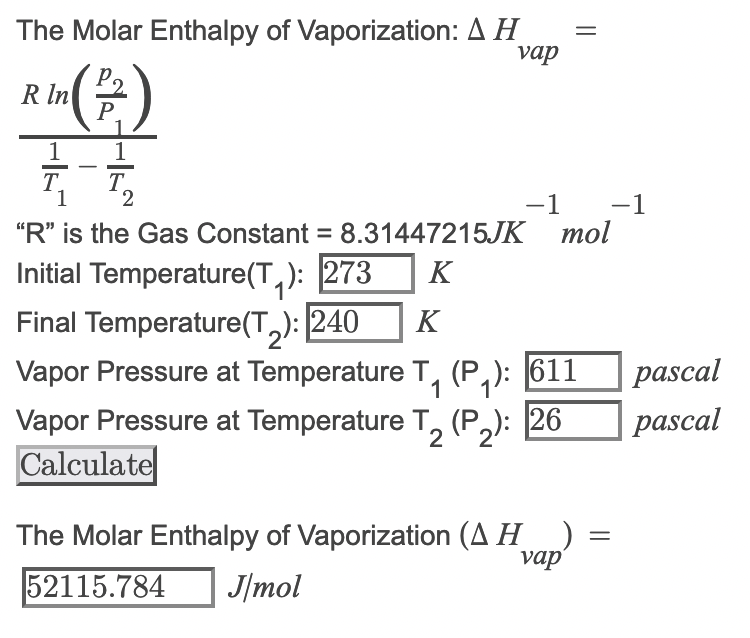
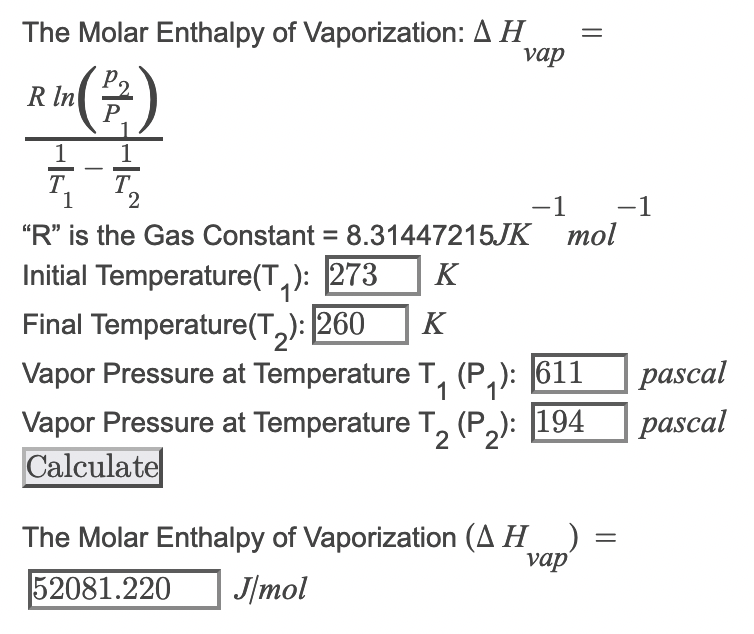
 

Figure 5 shows results of another run with smaller , just to make sure that the inversion also happens with strong V-shaped profiles (it does). What’s also striking to me is the way has a different behavior around the times that the surface is flat (layer difference close to zero).

|  |
| --- |
|  |
| **Figure 5**. Layer difference (left), profile close to 10 (center), and , with , , and as shown. |