I spent some time trying to characterize the consequences of constraining the (called Fliq in the code) outside the integrator (odeint, from scipy.integrate). The constraint is

(1)

|  |  |
| --- | --- |
|  |  |
| **Figure 1**. with regular updates (left) and without updating (right). | |

As the comparison in Fig. 1 shows, it’s clear that by , there’s a lot of drift. Therefore, some kind of constraint is essential. But how to apply it in a way that we don’t require regular updates to outside of the integrator?

One approach, carried out *inside* the first derivative functions (f0d and f1d, located in diffusionstuff6\_old.py), begins 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. So, if we solve the above equation for and take a Taylor expansion of Eq. 1 about , we can get to

(3)

So how does this work? Using the above equation (specified as variable dFliq\_dt in diffusionstuff6\_old.py), we get the results shown in Fig. 2.

|  |  |
| --- | --- |
|  |  |
|  |  |
| **Figure 2**. and computed with regular applications of Eq. 1 (left), and without updating (right), both employing Eq. 3 in the integrator. | |

Inspecting the top row of Fig. 2, it’s evident that updates are not needed at all for the first (30 layers). However, it’s clear from the bottom row of Fig. 2, at (50 layers), that significant differences between the updated vs non-updated results now exist.

|  |  |
| --- | --- |
|  |  |
| **Figure 3**. when employing Eq. 3 in the integrator, with and without regular application of a constraint outside the integrator. | |

Inspection of Fig. 3 shows that is well-behaved all the way up to (50 layers), but after that time, the integration becomes unstable.