Review of A. Beljaars et al 'On the numerical stability of surface-atmosphere coupling in weather and climate models'

Alex West, Met Office Hadley Centre

This concise study addresses the problem of instability due to explicit coupling across a surface boundary, and proposes a method by which this instability can be greatly reduced or eliminated. In this method, the coupling is made effectively semi-implicit by allowing the surface / boundary layer solver to make an educated guess of the future bottom boundary condition.

I found this study reasonably well argued and set out, with clear additional evidence provided from one-dimensional simulations, and am persuaded that the method described provides a good approximation to a semi-implicit scheme, with significant improvements over a traditional explicit scheme. The study makes a useful addition to the literature. My only comments relate to ways in which the method might perhaps be explained more clearly and consistently. In particular, I am not sure that the algorithm can be described as acting like a 'fully' implicit coupling scheme (see point 2).

1. **Page 3, line 7: 'The matrix equations can be solved by successive elimination of the C-coefficients from the bottom upward'.**

   This is a crucial step as it provides the initial linear relation between surface flux and top layer temperature; however, I had to work through it quite carefully to understand how this produced equation (8). It is also a little confusingly written as strictly speaking it is the variables that are eliminated, not the coefficients.

   I wonder if it would be worth expanding this line to demonstrate the elimination of bottom layer temperature from the lowest pair of equations, and its result:

   $$A_{NL}T_{NL-1}^{n+1} + B_{NL}T_{NL}^{n+1} = D_{NL} \qquad (1)$$

   $$A_{NL-1}T_{NL-2}^{n+1} + B_{NL-1}T_{NL-1}^{n+1} + C_{NL-1}T_{NL}^{n+1} = D_{NL-1} \qquad (2)$$

   $$\Rightarrow \quad \frac{A_{NL}}{B_{NL}}T_{NL-1}^{n+1} - \frac{A_{NL-1}}{C_{NL-1}}T_{NL-2}^{n+1} - \frac{B_{NL-1}}{C_{NL-1}}T_{NL-1}^{n+1} = \frac{D_{NL}}{B_{NL}} - \frac{D_{NL-1}}{C_{NL-1}} \qquad (3)$$

   Which is an equation of the form

   $$T_{NL-2}^{n+1} = \lambda T_{NL-1}^{n+1} + \mu \qquad (4)$$

   It is then easy to see that by repeating upwards, the linear relation $T_1^{n+1} = \alpha G_0 + \beta$ can be obtained, as G_0 effectively takes the part of the upper boundary condition (i.e. instead of 'T_0^{n+1}') in the topmost equation.

2. **Page 4, line 18: 'Together with Eq. (13), T_1^{n+1} and G_0 can be computed:' (and following equations, (14), (15).**

   It is probably a trivial point, but the solution of equations (14) and (15) actually depends on the future air temperature, T_a^{n+1}, already being known. I would guess that it is assumed in the method that the atmospheric simulation is solved first, using bottom boundary conditions from the previous timestep, hence T_a^{n+1} is known when the time comes to calculate the surface flux.

   With this understanding, the algorithm works as follows:

   i) Atmospheric simulation is advanced one timestep, using boundary conditions from the previous timestep;
   ii) Implicit approximation to future surface flux is calculated, using new atmospheric temperature and coupling fields received from the snow below according to parameterisation described in Section 5;
   iii) 'Future' surface flux is passed to the snow model below;
   iv) Snow model is advanced one timestep, using the future surface flux as its upper boundary condition.

   With this algorithm, while the snow model is advanced in time using forcing valid at the **end** of the timestep, the atmosphere model is still advanced in time using forcing valid at the **beginning** of the timestep. It may just be semantics, but I would argue that this method should be described as semi-implicit, rather than fully-implicit.

   The implications of this point for stability of the simulation are probably rather limited. Because of the different timescales under which the atmosphere and snow simulations work, the forcing of the snow on the atmosphere (top layer temperature) would probably be liable to change less rapidly than the forcing of the atmosphere on the snow (surface flux). An implicitly calculated bottom boundary condition for the atmosphere is probably somewhat less important for stability than the top boundary condition for the snow.

   This objection therefore does not represent a fundamental flaw in the method, more really an issue with how it is described. It would probably be good if the authors could state explicitly somewhere that T_a^{n+1} is solved as part of the atmosphere simulation based on previous values of the top snow layer temperature.

3. **Page 6, line 30 – page 7, line 8 (discussion of scaling behaviour of surface temperature evolation).**

   This section (Section 5) contains the central idea of the study, which is impressive in its simplicity. The idea is based on the fact that dimensional constraints on solutions to the diffusion equation mean that in the case of an initially isothermal profile, the evolution of the surface temperature in response to any surface flux G_0 must take a very specific form, described in equation (18) of Beljaars et al.

My issue is that having derived equation (18), the authors appear to digress into a discussion of the surface temperature simulation before arriving at equation (22), which describes the form the coefficient α must take. I think that this is unnecessary (see below). It also appears to conflate two different concepts: the surface skin temperature (SSKT) and top layer temperature (TLT). It is the TLT that the authors are trying to approximate using the scaling arguments, but in equations (19) and (20) they effectively derive the form α would take if they were instead trying to approximate the SSKT. The authors then state that this form must also take account of Δz for an approximation of TLT, due to the system being discretised. But I think that the form must take account of Δz for the (simpler) reason that it is TLT they are trying to approximate, not SSKT. It is possible that the authors view TLT as being the 'discrete' approximation to SSKT, but I think that it is much more accurate to view it as the discrete approximation to temperature at a depth of $\frac{\Delta z}{2}$.

The same conclusions can be arrived at from equation (18) more smoothly, as follows:

Rearranging equation (18) of Beljaars et al, and rewriting it in the notation of section 2, leads to the form

$$T_j^{n+1} = \frac{\delta G_0}{K} h\left(\frac{z_j}{\delta}\right) + T_j^n \qquad (5)$$

Here δ is the scaling depth $\left(\frac{K\Delta t}{\rho C}\right)^{1/2}$, z_j the midpoint of vertical layer *j*, and h() some universal function. It should be noted that (5) still describes the evolution of the temperature according to the continuous diffusion equation, rather than being a description of the discretised solution (even though it is written in this notation).

In particular, setting *j*=1, to estimate the TLT:

$$T_1^{n+1} = \frac{\delta G_0}{K} h\left(\frac{\Delta z}{2\delta}\right) + T_1^n \qquad (\text{using } z_1 = \Delta z/2) \qquad (6)$$

Equation (6) shows that in estimating the discretised solution the quantity $\frac{\delta}{K} h\left(\frac{z_j}{\delta}\right)$ naturally takes the place of the coefficient α (the '2' in the denominator of the function argument can be ignored, by scaling *h* and relabeling). As *h* is dimensionless, this implies that α scales like $\left(\frac{\Delta t}{K\rho C}\right)^{1/2}$ (the same conclusion as the authors reach in their equation (20)). This shown, the function can be 'measured'.