The vertical diffusion module

May 21, 2001

1 Introduction

The vertical diffusion module is used in Atmospheric GCMs to add the tendency due to vertical diffusion to the existing tendencies in the equations for zonal and meridional momenta, temperature, water vapor mixing ratio, and other tracers. The diffusivities are assumed known.

We first describe the algorithm 1) for a tracer with an explicitly specified surface flux, then 2) for a tracer in which the dependence of the flux on the lowest level atmospheric data is implicit, and finally 3) for tracers in which the dependence of the surface flux on both atmosperic values and surface properties are treated implicitly. We then describe the some specific considerations for diffusion of heat, moisture, and momentum.

Our starting point is the advection-diffusion equation in a compressible fluid for the concentration of a tracer ξ with source per unit mass S. Only vertical diffusion is considered here. We can write this equation in flux form

$$\frac{\partial \rho\xi}{\partial t} = -\nabla \cdot (\rho v \xi) - \frac{\partial F}{\partial z} + \rho S$$

$$F \equiv -\mathcal{D}(z)\rho\frac{\partial \xi}{\partial z}$$

or advective form

$$\frac{\partial \xi}{\partial t} = -v \cdot \nabla \xi - \frac{1}{\rho}\frac{\partial F}{\partial z} + S \tag{1}$$

Here $\mathcal{D}$ is the kinematic diffusivity and ρ is the density of the air. ξ is a mixing ratio: the amount of a substance per unit mass of air. F is the upward diffusive flux of tracer, with value F_s at the surface. In some cases, i.e. temperature, there can also be a distinction between the quantity whose gradient determines the diffusive flux and the quantity itself, a point that we return to below.

The diffusion is computed in advective form. This need present no difficulty with conservation of tracer as long as one is careful that the advection is treated consistently with the treatment of conservation of mass. Note, however, that if one sets $v = 0$ in the advective form of the tracer equation, while still allowing the flow to change the mass distribution, ρ, then total tracer substance will no longer be conserved, whereas conservation would still hold if one set $v = 0$ in the flux form of the equation. So if one wants to diffuse a tracer but not advect it, this must be done with care.

2 Diffusion of heat

The thermodynamic equation of the model is assumed to be in the standard form obtained by considering the rate of change of enthalpy per unit mass $h = c_p T$,

$$c_p \frac{DT}{Dt} = \frac{1}{\rho} \frac{Dp}{Dt} + Q \tag{2}$$

where D/Dt is the material derivative, c_p the heat capacity at constant pressure per unit mass, and Q is the heating rate per unit mass. Equivalently,

$$\frac{DT}{Dt} = \frac{\kappa T \omega}{p} + Q/c_p \tag{3}$$

where $\kappa = R/c_p$, $\omega = Dp/Dt$, and R is the gas constant.

We add to the RHS of (26) the effect of mixing the the "dry static energy temperature" σ

$$\sigma \equiv T + \frac{gz}{c_p} \tag{4}$$

where g/c_p is the dry-adiabatic lapse rate. Thus, the temperature equation is modified to read

$$\frac{DT}{Dt} = \frac{\kappa T \omega}{p} + \frac{Q}{c_p} + \frac{1}{\rho} \frac{\partial}{\partial z} \mathcal{D}(z) \rho \frac{\partial \sigma}{\partial z} \tag{5}$$

Potential temperature is conserved if the flow is adiabatic, and one expects the fast turbulence being modeled here to mix this conserved quantity. However, one would also like to conserve energy. For a hydrostatic ideal gas, the internal energy $c_v T$ plus the potential energy gz, when integrated over the entire atmospheric column, equals $c_p T$ integrated over the column. So conservation of internal plus potential energy is guaranteed if one conserves the mean of $\int \rho T dz$

If one simply diffuses potential temperature, Θ,

$$\frac{\partial \Theta}{\partial t} = \frac{1}{\rho} \frac{\partial}{\partial z} \rho \mathcal{D} \frac{\partial \Theta}{\partial z} \tag{6}$$

one will not conserve energy. One alternative is to write

$$\frac{\partial T}{\partial t} = \frac{1}{\rho} \frac{\partial}{\partial z} \rho \mathcal{D} \frac{\partial \Theta}{\partial z} \tag{7}$$

but then the implied dependence on the reference temperature used in the definition of potential temperature seems strange. A nice alternative is to utilize the fact that, in a hydrostatic ideal gas

$$\frac{\partial \sigma}{\partial z} = \frac{T}{\Theta}\frac{\partial \Theta}{\partial z} \tag{8}$$

so that σ is well-mixed when Θ is well-mixed.

In a future release, this scheme will be modified so that the moist static energy and and total (vapor plus liquid) water content are homogenized by diffusion, rather than dry static energy and specifi humidity.

3 Time differencing – two-time level models

A fully backwards time-step for diffusion is the standard choice in atmospheric GCMs as the diffusivities are invariably large enough that explicit time steps are too restrictive. Centered implicit steps, while more accurate, can produce damped oscillations when the correct solution is monotonically damped. Fully backwards steps are the safest.

Consider for a moment the generic equation, with backwards differencing,

$$\frac{\xi^{i+1} - \xi^{i}}{\Delta t} = L(\xi^{i+1}) + S$$

The superscript refers to the time step, and L is a linear operator (the diffusion operator in our case). S is the rest of the RHS of the equation, assumed to be known at time step i. Equivalently, we can write

$$\xi^{i+1} - \xi^{i} = \Delta t L(\xi^{i+1} - \xi^{i}) + E$$

where E includes the tendency due to L as computed explicitly.

$$E = \Delta t(S + L(\xi^{i}))$$

To simplify notation, we set $\Delta\xi$ equal to the increment in tracer mixing ratio over one time step.

$$\Delta\xi \equiv \xi^{i+1} - \xi^{i}$$

Then the equation to be solved is

$$\Delta\xi = \Delta t L(\Delta\xi) + E$$

This equation is solved by inverting the operator $(1 - L\Delta t)$:

$$\Delta\xi = (1 - \Delta t L)^{-1} E$$

It may at times be useful to split the source term S into a part that is conveniently computed before the diffusion is performed, and a part that is computed after the diffusion $S = S_{before} + S_{after}$, resulting in

$$\frac{\Delta\xi}{\Delta t} = S_{after} + (1 - \Delta t L)^{-1}(S_{before} + L(\xi^i)) \tag{9}$$

which differs from the result that would be obtained if one computed all of the explicit tendencies before the implicit diffusion step

$$\frac{\Delta\xi}{\Delta t} = (1 - \Delta t L)^{-1}(S_{before} + L(\xi^i) + S_{after})$$

Splitting may be unavoidable at times, but it is potentially problematic. The difficulty arises when there is close compensation between S_{before} and S_{after}, and the implicit treatment of the equation then creates tendencies that are too large.

The diffusion module updates the tendency by replacing the old tendency S_{before} with $(1 - \Delta t L)^{-1}(S_{before} + L(\xi^i))$. It does not know anything about S_{after}

4 Leapfrog

Using backwards diffusion in the context of the leapfrog step used in many GCMs, we have

$$\frac{\xi^{i+1} - \xi^{i-1}}{2\Delta t} = L(\xi^{i+1}) + S$$

With the same manipulation as before, we obtain the same equation

$$\Delta\xi = (1 - 2\Delta t L)^{-1} E$$

where now

$$\begin{aligned} E &= (2\Delta t) S + L(\xi^{i-1})) \\ \Delta\xi &\equiv \xi^{i+1} - \xi^{i-1}. \end{aligned}$$

The diffusion module does not need to know whether or not a leapfrog step is being used. In the leapfrog case, the field ξ^{i-1} is input to the module rather than ξ^i, and the time step passed as input is $2\Delta t$ rather than Δt. We use the notation δt to stand for the time step Δt in the two time level scheme and for $2\Delta t$ in the leapfrog scheme. Also, the symbol $\delta\xi$ refers to $\xi^{i+1} - \xi^{i-1}$ in the leapfrog case and $\xi^{i+1} - \xi^i$ for the two-time level version.

5 Vertical differencing

Using the hydrostatic equation, we can write the diffusion as

$$g\frac{\partial F}{\partial p}$$

Dividing the atmosphere into N layers, with $k = 1$ the top layer and $k = N$ the bottom layer, the diffusive flux is defined at the interfaces, with $F_{k+1/2}$ between the k and $k+1$ layers, so that $F_{1/2} = 0$ and $F_{N+1/2} = F_s$, the surface flux. We also require the pressure defined at the interface $p_{k+1/2}$. The simplest differencing then yields

$$\mu_k(F_{k+1/2} - F_{k-1/2})$$

where

$$\mu_k \equiv \frac{g}{p_{k+1/2} - p_{k-1/2}}, \quad k = 1, N \tag{10}$$

Note that the fluxes are considered positive when directed upwards, even though the indexing starts from the top (to be consistent with standard practice in GCM codes).

The fluxes, in turn, are given in terms of the diffusivity $\mathcal{D}_{k+1/2}$ and density $\rho_{k+1/2}$ defined at the interfaces and the heights z_k of the full levels,

$$\begin{aligned} F_{k+1/2} &= -\mathcal{D}_{k+1/2}\rho_{k+1/2}\frac{\xi_{k+1} - \xi_k}{z_{k+1} - z_k} \\ &\equiv \nu_{k+1/2}(\xi_{k+1} - \xi_k) \end{aligned} \tag{11}$$

Note that ν is positive (as is μ) since k increases downwards.

6 Tridiagonal solver with explicit surface flux

We define E to be the increment in ξ due to all terms treated explicitly plus the explicit contribution from diffusion in the interior. We do not include the effect of the surface flux F_s itself on the increment in the lowest model layer, even when explicit, in the expression for E.

With

$$\mu' \equiv (\delta t)\mu$$

we have

$$\begin{aligned}
\delta\xi_k &= E_k + \mu'_k(\nu_{k+1/2}(\delta\xi_{k+1} - \delta\xi_k) - \nu_{k-1/2}(\delta\xi_k - \delta\xi_{k-1})) \quad k = 2, N-1 \\
\delta\xi_1 &= E_1 + \mu'_1\nu_{3/2}(\delta\xi_2 - \delta\xi_1) \\
\delta\xi_N &= E_N + \mu'_N F_s - \mu'_N\nu_{N-1/2}(\delta\xi_N - \delta\xi_{N-1})
\end{aligned} \tag{12}$$

So then

$$A_k\delta\xi_{k+1} + B_k\delta\xi_k + C_k\delta\xi_{k-1} = E_k, \quad k = 1, N$$

where

$$\begin{aligned}
A_k &= -\mu'_k\nu_{k+1/2}, \quad k = 1, N-1 \\
A_N &= 0 \\
C_k &= -\mu'_k\nu_{k-1/2}, \quad k = 2, N \\
C_1 &= 0 \\
B_k &= 1 + \mu'_k(\nu_{k+1/1} + \nu_{k-1/2}), \quad k = 2, N-1 \\
B_1 &= 1 + \mu'_1\nu_{3/2} \\
B_N &= 1 + \mu'_N\nu_{N-1/2}
\end{aligned}$$

To solve these equations with the standard tridiagnonal substitution, we set

$$\delta\xi_k = e_k\delta\xi_{k+1} + f_k, \quad k = 1, N \tag{13}$$

Substituting, we have

$$A_k\frac{\delta\xi_k - f_k}{e_k} + B_k\delta\xi_k + C_k(e_{k-1}\delta\xi_k + f_{k-1}) = E_k$$

so that

$$e_k = \frac{-A_k}{B_k + C_k e_{k-1}} \quad f_k = \frac{E_k - C_k f_{k-1}}{B_k + C_k e_{k-1}}$$

Starting at the top of the atmosphere, we have

$$e_1 = -\frac{A_1}{B_1}, \quad f_1 = \frac{E_1}{B_1}$$

One can then progress downwards, computing e's and f's up to e_N and f_N. The last step in this sweep is

$$\begin{aligned} e_N &= 0 \\ f_N &= \delta\xi_N = -\frac{E_N + \mu'_N F_s - C_N f_{N-1}}{B_N + C_N e_{N-1}} \end{aligned} \tag{14}$$

The upward part of the tridiagonal reduction proceeds by returning to (6) and sweeping recursively up the atmosphere to $k = 1$.

Substituting the values of A_N and B_N into (7)

$$\delta\xi_N = \Gamma(E_N^* + \mu'_N F_s) \tag{15}$$

where

$$E_N^* \equiv E_N + \mu'_N \nu_{N-1/2} f_{N-1} \tag{16}$$

$$\Gamma \equiv \frac{1}{1 - \mu'_N D^T} \tag{17}$$

$$D^T \equiv -\frac{\partial F_{N-1}}{\partial \xi_N} \equiv -\nu_{N-1/2}(1 - e_{N-1}) \tag{18}$$

$$\tag{19}$$

D^T is the sensitivity of the *downward* flux through the top of the lowest model layer to the value of ξ in the lowest model layer, taking into account the implicit corrections to all of the other layers.

7 Implicit dependence of surface flux on lowest atmospheric layer

Using a linear Taylor's expansion we assume that the surface flux can be written in the form

$$F_s = F_s^{ex} + D^B \delta\xi_N \tag{20}$$

where F_s^{ex} and D^B are known explicitly. Following through the derivation in the preceeding section, the only change in the resulting algorithm is that in the final step in the downward sweep we need to replace F_s by F_s^{ex} in (8) and we need to change D_T to $D_T + D_B$ in the definition of Γ.

$$\begin{aligned} \delta\xi_N &= \Gamma(E_N^* + \mu'_N F_s^{ex}) && (21) \\ \Gamma &\equiv \frac{1}{1 - \mu'_N(D^T + D^B)} && (22) \end{aligned}$$

8 Implicit dependence of surface flux on surface parameters

Now suppose that the surface flux also depends on some surface parameter ζ. (This could also be a set of parameters, but it is easier to think about the case in which there is only 1). A linear Taylor expansion now reads

$$F_s = F_s^{ex} + D^B \delta\xi_N + D^S \delta\zeta \tag{23}$$

Revising (14), we have

$$\delta\xi_N = \Gamma(E_N^* + \mu'_N(F_s^{ex} + D^S \delta\zeta)) \tag{24}$$

or

$$\delta\xi_N = e_N \delta\zeta + f_N \tag{25}$$

where

$$\begin{aligned} e_N &= \Gamma \mu'_N D^s && (26) \\ f_N &= \Gamma E_N^* + \mu'_N F_s^{ex} && (27) \end{aligned}$$

Substituting back into the expression for the flux,

$$F_s = \alpha + \beta\delta\zeta \tag{28}$$

where

$$\begin{aligned} \alpha &\equiv F_s^{ex} + D^B f_N && (29) \\ \beta &\equiv D^S + D^B e_N && (30) \end{aligned}$$

So we can ask the surface modules to compute $\delta\zeta$ given a boundary comdition of the form (21). One can then compute $\delta\xi_N$ from (18) and proceed with the upward sweep.

9 Implementation

If there is no implicit dependence of surface fluxes on surface parameters, the vertical diffusion can update the model tendencies to take into account the vertical diffusion by itself. This is done with a single call to the subroutine $vert\_diff$ for a single field, or gcm_vert_diff which handles u, v, T, q, and tracers simultaneously.

If there is implicit dependence of surface fluxes on surface parameters, then the job is split between the vertical diffusion module, the flux exchange module, and the surface module(s)

The "GCM" calls assume the specific humidity and heat are diffused with the same diffusivity, but momentum can potentially have a different diffusivity. Other tracers share the diffusivity for heat and moisture.

The GCM calls also assume that the surface fluxes of momentem have no implicit dependence on surface parameters, but that the surface heat and moisture fluxes potentially do. (All other tracers are like momentum in this regard.) The module is not aware of the details of this dependency.

One first calls $gcm\_vert\_diff\_down$ to do the downward sweep. The values of e_k and f_k for $k = 1, N-1$ are saved as module variables for later use by the upward sweep. The variable e_k is the same for moisture and temperature, but there is are separate f_k fields for moisture and for temperature. (If temperature and moisture are given different diffusivities, then the code will have to be modified to save different e_k fields for the different variables.) Additionally, the values of the following fields are output

$$\begin{aligned} \mu'_N &\equiv \delta t \mu_N \\ E^*_N &\equiv E_N + \mu_N \nu_{N-1/2} f_{N-1} \\ D^T &\equiv -\nu_{N-1/2}(1 - e_{N-1}) \end{aligned} \tag{31}$$

The latter two are computed separately for temperature and for moisture, resulting in five fields that are placed in a $type(surf\_diff\_type)$.

μ'_N is the atmospheric time step ($2\ \Delta t$ for leapfrog) divided by the mass of the lowest atmospheric layer. E^*_N is the increment (of temperature or specific humidity) in the lowest atmospheric layer, due to all terms computed explicitly, including the diffusive flux at the top of this layer (but not including the surface flux), corrected for the implicit treatment of the diffusive fluxes throughout the atmosphere (by adding the term $\mu_N \nu_{N-1/2} f_{N-1}$). D^T is the sensitivity of the *downward* flux through the top of the lowest model layer to the value of tempeature or specific humidity in the lowest model layer, taking into account the implicit corrections to all of the other layers.

The calling program requests that a variable of $type(surf\_diff\_type)$ be allocated when initializing $vert\_diff\_mod$. Let's call this variable $Surf$. Then the five fields output from $gcm\_vert\_diff\_down$ are

$$\begin{aligned}
Surf\%dtmass &= \mu_N \\
Surf\%dlfux_t &= D^T|_{temperature} \\
Surf\%dlfux_q &= D^T|_{specific_humidity} \\
Surf\%delta_t &= E_N^*|_{temperature} \\
Surf\%delta_q &= E_N^*|_{specific_humidity}
\end{aligned}$$

All surface fluxes and derivatives are computed in the flux exchange module. It computes e_N and f_N and then α and β. The surface modules then do their job, after which the flux exchange module computes the final fluxes and the final increments in temperature and moisture for the lowest atmospheric layer. These increments are passed back to the vertical diffusion module in $Surf\%dflux\_t$ and $Surf\%dflux\_q$. The subroutine $gcm\_vert\_diff\_up$ then computes the increments for all of the other atmospheric layers.