## 3. The Mesh refinement scheme

The 2-way interactive mesh refinement scheme is constructed to allow for an arbitrary number of overlapping and translating rectangular grids with an arbitrary number of refinement levels. The grids must be aligned with the model coordinates (no rotating meshes), and the mesh refinement ratio of the temporal and spatial grid increments is common for all meshes, and currently set to three. Vital parts of this refinement scheme are the interpolation routines (Smolarkiewicz and Grell, 1992), which are used upon initialization of new nests as well as for defining the boundaries of the fine meshes. If the user can supply his own analysis for the finer grids (or a finer grid), the interpolated fields can be overwritten. In the following section we describe the heart of the scheme, the monotone interpolation routines.

### 3.1 The monotone interpolation routines

The most vital element of any mesh refinement scheme is an accurate and efficient interpolation procedure. A complaint about traditional polynomial-fitting methods used for interpolating scalar fields defined on a discrete mesh is that they often lead to spurious numerical oscillations in regions of steep gradients of the interpolated variables. In order to suppress computational noise, which is characteristic of quadratic and higher-order interpolation schemes, one often implements a smoothing procedure or increased diffusion. These, however, introduce excessive numerical diffusion that smears out sharp features of interpolated fields. A more advanced approach invokes the so-called shape-preserving interpolation, which incorporates appropriate constraints on the derivative estimates used in the interpolation schemes (see Rasch and Williamson (1990) for a review). In MM5 we consider as an alternate approach a class of schemes derived from monotone advection algorithms (Smolarkiewicz and Grell, 1992). Smolarkiewicz and Grell (1992) show that the interpolation problem becomes identical to the advection problem, when the distance vector is replaced by the velocity vector (see also the end of this section). Here we will describe the implementation of the advection algorithm used in MM5. The interested reader is referred to Smolarkiewicz and Grell (1992) for a detailed derivation of the "advection-interpolation" equivalence.

Since shape preservation and monotonicity are important in the interpolation problem, we chose the Flux Corrected Transport (FCT) scheme that uses the high-order accurate constant-grid-flux dissipative algorithms developed by Tremback *et al.* (1987). We will first describe, in abbreviated form, a general FCT algorithm, as used in MM5. Given the exactness of the alternate-direction representation of the interpolation algorithm, it is sufficient to consider only one-dimensional FCT schemes. Starting with the one-dimensional advection equation in flux-form

$$\frac{\partial \phi}{\partial t} = -\frac{\partial u\phi}{\partial x}, \tag{3.1.1}$$

where $\phi$ is a scalar variable advected with a flow field $u(x,t)$, an FCT advection scheme may be compactly written as

$$\phi_i^{n+1} = \Phi_i^{n+1} - (\tilde{A}_{i+1/2} - \tilde{A}_{i-1/2}), \tag{3.1.2}$$

where $\Phi$ denotes a low-order, monotone solution to (3.1.1), and $\tilde{A}$ is the antidiffusive flux, limited such as to ensure that the solution (3.1.2) is free of local extrema absent in the low-order solution. Note that

$$\tilde{A}_{i+1/2} = min\left(1, \beta_i^{\downarrow}, \beta_{i+1}^{\uparrow}\right)\left[A_{i+1/2}\right]^{+} + min\left(1, \beta_i^{\uparrow}, \beta_{i+1}^{\downarrow}\right)\left[A_{i+1/2}\right]^{-}, \tag{3.1.3}$$

where

$$A_{i+1/2} \equiv FH_{i+1/2} - FL_{i+1/2}, \tag{3.1.4}$$

with $FH$ and $FL$ denoting fluxes from a high-order and a low-order advection scheme, respectively. $[\ ]^{+} \equiv max(0,\ \ )$ and $[\ ]^{-} \equiv min(0,\ \ )$ are the positive- and the negative-part operators, respectively, and

$$\beta_i^{\uparrow} \equiv \frac{\phi_i^{MAX} - \Phi_i^{n+1}}{A_i^{IN} + \varepsilon} \;\; ; \;\; \beta_i^{\downarrow} \equiv \frac{\Phi_i^{n+1} - \phi_i^{MIN}}{A_i^{OUT} + \varepsilon}, \tag{3.1.5a, b}$$

where $A_i^{IN}$, $A_i^{OUT}$ are the absolute values of the total incoming and outgoing $A$-fluxes, (3.1.4), from the $i$-th grid box, respectively. $\varepsilon$ is a small value, e.g. $\sim 10^{-15}$, and allows for efficient coding of $\beta$-ratios when $A_{\mathbf{i}}^{IN}$ or $A_{\mathbf{i}}^{OUT}$ vanish. The limiters $\phi_i^{MAX}$ and $\phi_i^{MIN}$ define monotonicity of the scheme (i.e., by design $\phi_i^{MIN} \leq \phi^{n+1} \leq \phi_i^{MAX}$), and they are traditionally specified (Zalesak 1979) as

$$\phi_i^{MAX} = max\left(\phi_{i-1}^n, \phi_i^n, \phi_{i+1}^n, \Phi_{i-1}^{n+1}, \Phi_i^{n+1}, \Phi_{i-1}^{n+1}\right) \tag{3.1.6a}$$

$$\phi_i^{MIN} = min\left(\phi_{i-1}^n, \phi_i^n, \phi_{i+1}^n, \Phi_{i-1}^{n+1}, \Phi_i^{n+1}, \Phi_{i-1}^{n+1}\right). \tag{3.1.6b}$$

A shape-preserving interpolation scheme requires less restrictive monotonicity constraints than a conservative advection scheme. The minima over $\beta$ ratios appearing in (3.1.3) ensure that the antidiffusive flux attributed to the $i+1/2$ position on the grid does not contribute to the generation of spurious extrema, either in gridbox $i$ or in gridbox $i+1$. However, monotonicity of the interpolation scheme only requires that $\phi_i^{n+1} = \psi(x_o)$ is free of spurious extrema. Consequently, equation (3.1.3) may be replaced by

$$\tilde{A}_{i+1/2} = min\left(1, \beta_i^{\downarrow}\right)\left[A_{i+1/2}\right]^+ + min\left(1, \beta_i^{\uparrow}\right)\left[A_{i+1/2}\right]^-. \tag{3.1.3'}$$

Furthermore, since the effective flow field is constant, and therefore incompressible, the limiters in (3.1.6) may be simplified to

$$\phi_i^{MAX} = max\left(\phi_{i-1}^n, \phi_i^n, \phi_{i+1}^n, \Phi_i^{n+1},\right) \;;\; \phi_i^{MIN} = min\left(\phi_{i-1}^n, \phi_i^n, \phi_{i+1}^n, \Phi_i^{n+1}\right), \tag{3.1.6'a,b}$$

where the redundant dependence of the limiters on $\Phi_i^{n+1}$ has been retained to ensure strictly nonnegative values of the $\beta$ ratios in (3.1.5) (cf., Section 3.1 in Smolarkiewicz and Grabowski, 1990). Since the low-order solution may always be written as an old value, minus the divergence of fluxes from the low-order scheme, the entire algorithm consisting of (3.1.2), (3.1.3′), (3.1.4), (3.1.5), and (3.1.6′) is in the form of a general advection scheme.

The advection schemes used to calculate the high- and low-order fluxes for the above equations are from Tremback *et al.* (1987). They derive as follows. Starting with the flux form of the one-dimensional advection equation (3.1.1) in finite difference form

$$\begin{aligned}\phi_i^{n+1} &= +\,\phi_i^n + \frac{\Delta t}{\Delta x}[F_{i+1/2} - F_{i-1/2}] \\ &= \phi_i^n + \frac{\Delta x}{\Delta t}\left[\sum_m b_m \phi_{i+1+m}^n - \sum_m b_m \phi_{i+m}^n\right],\end{aligned} \tag{3.1.7}$$

where

$$F_{i+1/2} = \sum_m b_m \phi_{i+1+m}^n \tag{3.1.8}$$

and

$$F_{i-1/2} = \sum_m b_m \phi_{i+m}^n \tag{3.1.9}$$

were used. Following Tremback *et al.* (1987), the solutions for the even-order schemes which are used in the mesh refinement scheme are then given by

$$\begin{aligned} F_{i+1/2}\frac{\Delta t}{\Delta x} = &+ \frac{\alpha}{2}(-\phi_i - \phi_{i+1}) + \\ &+ \frac{\alpha^2}{2}2(-\phi_i + \phi_{i+1}) \end{aligned} \tag{3.1.10}$$

for second order accuracy;

$$\begin{aligned} F_{i+1/2}\frac{\Delta t}{\Delta x} = &+ \frac{\alpha}{12}(\phi_{i-1} - 7\phi_i - 7\phi_{i+1} + \phi_{i+2}) \\ &+ \frac{\alpha^2}{24}(\phi_{i-1} - 15\phi_i + 15\phi_{i+1} - \phi_{i+2}) \\ &+ \frac{\alpha^3}{12}(-\phi_{i-1} + \phi_i + \phi_{i+1} - \phi_{i+2}) \\ &+ \frac{\alpha^4}{24}(-\phi_{i-1} + 3\phi_i - 3\phi_{i+1} + \phi_{i+2}) \end{aligned} \tag{3.1.11}$$

and for fourth order accuracy;

$$\begin{aligned} F_{i+1/2}\frac{\Delta t}{\Delta x} = &+ \frac{\alpha}{60}(-\phi_{i-2} + 8\phi_{i-1} - 37\phi_i - 37\phi_{i+1} + 8\phi_{i+2} - \phi_{i+3}) \\ &+ \frac{\alpha^2}{360}(-2\phi_{i-2} + 25\phi_{i-1} - 245\phi_i + 245\phi_{i+1} - 25\phi_{i+2} + 2\phi_{i+3}) \\ &+ \frac{\alpha^3}{48}(\phi_{i-2} - 7\phi_{i-1} + 6\phi_i + 6\phi_{i+1} - 7\phi_{i+2} + \phi_{i+3}) \\ &+ \frac{\alpha^4}{144}(\phi_{i-2} - 11\phi_{i-1} + 28\phi_i - 28\phi_{i+1} + 11\phi_{i+2} - \phi_{i+3}) \\ &+ \frac{\alpha^5}{240}(-\phi_{i-2} + 3\phi_{i-1} - 2\phi_i - 2\phi_{i+1} + 3\phi_{i+2} - \phi_{i+3}) \\ &+ \frac{\alpha^6}{720}(-\phi_{i-2} + 5\phi_{i-1} - 10\phi_i + 10\phi_{i+1} - 5\phi_{i+2} + \phi_{i+3}) \end{aligned} \quad , \tag{3.1.12}$$

for sixth order accuracy; $\alpha$ is defined as

$$\alpha = U\frac{\Delta t}{\Delta x}. \tag{3.1.13}$$

In MM5, equations (3.1.10 - 3.1.12) are used together with (3.1.1), (3.1.3$'$), (3.1.4), 3.1.5), and (3.1.6$'$) to solve the interpolation problem. Note that the velocity vector, is replaced by the distance vector, $X_d$, which, with a mesh-refinement ratio of three, simply becomes 1/3 or 2/3. For interpolating boundary conditions to the finer meshes, fourth order accuracy is used, while for new nest initialization, sixth order accuracy is used. While the new nest initialization covers the whole domain, boundary interpolation is performed for the outermost 2 rows and columns of the nest. Two rows were necessary to ensure that the same operators were applied to each nested grid-point (including fourth-order diffusion).

### 3.2 Overlapping and moving grids

The mesh-refinement scheme allows for overlapping grids on the same nest-level. To ensure numerical stability, the solution in the overlap region has to be identical. To accomplish this, after each time-step of the grids in question, the boundary conditions in the overlap regions are provided by the overlapping mesh. It is very important that this procedure be performed at every timestep.

Nests can also be moved at any time in the forecast. This can be done many times, and for any distance (integer number of grid points). The user may also move the nests automatically if he supplies an algorithm to do so. Upon a move, a new nest initialization is performed first. Then all high-resolution information from the previous location of the mesh is used to overwrite the fields of the newly initialized mesh. Therefore, to ensure best use of high resolution information, it is better to move a nest more often and for a smaller distance.

### 3.3 The feedback to the coarser grids

Since the mesh refinement ratio in MM5 is set to three, a higher resolution mesh has to be integrated three times as often as its “Mother Domain”(MD), where MD means the coarser domain from which it gets its boundary conditions. To keep the solutions in a 2-way interaction from diverging, the meteorological fields have to be fed back from the higher-resolution mesh to its MD. This is done at the end of the three time-step integration. Naturally, when this is done without smoothing or averaging, the solution on the MD will appear somewhat noisy (diluted with small-scale information). To avoid numerical instability, the following methods are supported in MM5 to remove non-resolvable noise from the MD. Note that these smoothers are only applied over an interior area that is completely determined by the higher resolution domain. It is important that input into the nest, and feedback back to the MD does not overlap. The smoother that is used by the MM5 system in various forms was defined by Shapiro (1970) as

$$\begin{aligned}\bar{\alpha}(i,j) =& \alpha(i,j) \\ &+\frac{\nu}{2}(1-\nu)(\alpha(i+1,j)+\alpha(i-1,j)+\alpha(i,j+1)+\alpha(i,j-1)+4\alpha(i,j)) \\ &+\frac{\nu^2}{4}(\alpha(i+1,j+1)+\alpha(i+1,j-1)+\alpha(i-1,j+1)+\alpha(i-1,j-1)-4\alpha(i,j))\end{aligned} \tag{3.3.1}$$

### 3.3.1 A Nine-point averager

This method was in the original MM4 nested version (Zhang *et al.* 1986). It is a feedback method that averages information for a whole MD grid box (surrounding and centering on the nested grid point). However, it does not take out all non-resolvable information on the MD. It also imposes a severe restriction on the terrain for the hydrostatic model. In case of overlapping and moving nests on several nest levels, it is very elaborate and complicated to apply. It is still an option in the model, because it may be useful for simpler applications (like one coarse and one nested domain). However, care must be taken to create a terrain data set that is consistent with this method. The operator that is applied to the nested grid-points (note that nothing is done to the MD) is defined by using $\nu = 0.5$ in (3.3.1).

### 3.3.2 A Smoother-Desmoother

The smoother-desmoother is a filter that removes $2\Delta x$ waves and damps short waves, but leaves long waves almost unaffected. It is much more selective than diffusive smoothers. It is applied to the "coarser grid" only in the area where the coarse grid values are overwritten with the nested grid values.

A single pass of the smoother-desmoother involves two steps. Equation (3.3.1) is used first to smooth the fields, then to desmooth the fields. $\nu_1 = 0.50$ is used for the smoothing coefficient, and $\nu_2 = -0.52$ for the desmoothing coefficient. The first step strongly smoothes the field, completely removing the $2\Delta x$ wave, and the second step attempts to restore the other waves to their original amplitudes. There are two passes of the smoother-desmoother applied in the model.