Centrum voor Wiskunde en Informatica
REPORT*RAPPORT*

A Zooming Technique for Wind Transport of Air Pollution

P.J.F. Berkvens, M.A. Botchev, W.M. Lioen, J.G. Verwer

Modelling, Analysis and Simulation (MAS)

MAS-R9921 August 31, 1999

Report MAS-R9921
ISSN 1386-3703

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

A Zooming Technique for Wind Transport of Air Pollution*

P.J.F. Berkvens, M.A. Botchev, W.M. Lioen and J.G. Verwer
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
www.cwi.nl
[*berkvens, botchev, walter, janv*]*@ cwi.nl*

ABSTRACT

In air pollution dispersion models, typically systems of millions of equations that describe wind transport, chemistry and vertical mixing have to be integrated in time. To have more accurate results over specific fixed areas of interest—usually highly polluted areas with intensive emissions—a local grid refinement or zoom is often required. For the wind transport part of the models, i.e. for finite volume discretizations of the transport equation, we propose a zoom technique that is positive, mass-conservative and allows to use smaller time steps as enforced by the CFL restriction in the zoom regions only.

1991 Mathematics Subject Classification: 65M06 65M50, 65C20
1991 ACM Computing Classification System: G.1.8, J.2
Keywords and Phrases: finite volumes, advection schemes, local refinement, air pollution, high performance computations
Note: Part of this report has been published as [3].

1. Introduction

Mathematical problems often encountered in air pollution modelling are transport-reaction problems of the form

$$\begin{aligned} &\frac{\partial c_s}{\partial t} + \nabla \cdot (\underline{u} c_s) = V(\underline{u}, c_s) + R_s(c), \\ &c = \{c_s(\underline{x}, t)\} \in \mathbb{R}^m, \quad \underline{x} \in \Omega \subset \mathbb{R}^3, \end{aligned} \tag{1.1}$$

where c_s are the concentrations of m chemical species in the atmosphere. The species under consideration are not only pollutants, but all the main chemical substances present in the atmosphere. In real applications m lies between 25 and 100. The first term in (1.1) denotes the time rate of change of c_s, the term $\nabla \cdot (\underline{u} c_s)$ describes the transport of the species by a given wind field $\underline{u}$. The term $V(\underline{u}, c_s)$ on the right-hand side appears as a result of parameterization of transport processes not resolved on the grid. It will be referred to as vertical mixing. Vertical mixing is often modelled by means of turbulent diffusion parameterization. The stiff reaction term R describes the chemical reactions among the species c_s. Usually, (1.1) is discretized in space with a finite-volume technique. For typical grid resolutions and numbers of chemical species m, (1.1) yields a system of millions of equations that has to be integrated in time. More background information on air pollution modelling can be found in the recent books [5, 15] and the survey [13] giving an overview of numerical techniques used in the field.

*This work has been done within the program "Wiskunde Toegepast" ("Mathematics Applied") of NWO, the Netherlands Organization for Scientific Research, project no. 613-302-040. This work was supported by NCF, the National Computing Facilities Foundation, under Grant NRG 98.02.

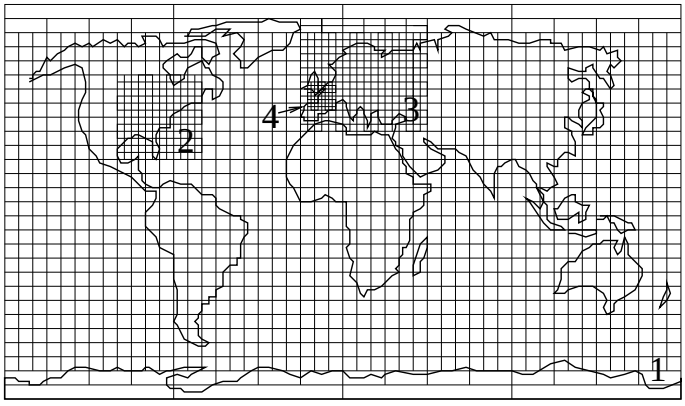

Figure 1: An example of zooming. Zoom regions are numbered.

1.1 Model

In air pollution models problem (1) is typically discretized in space with a finite-volume technique and then integrated in time with operator splitting. This means that the whole advance in time consists of separate advection, vertical mixing and chemistry advances. Chemistry and vertical mixing are integrated in time implicitly, to avoid a severe restriction on the time step Δt imposed by the chemistry-mixing stiffness.

Advection is usually integrated in time explicitly. In a particular atmospheric dispersion model we are working with, a successor of [6], several modern advection schemes are used. The two basic schemes are the Slopes scheme [12] and the Split scheme, a third order flux-limited upwind scheme [7, 8, 9]. Both schemes are one-dimensional and applied with directional splitting.

For the atmospheric application it is important that mass is conserved and concentrations remain positive ($c_s(t) \geqslant 0$, $t \geqslant 0$). Both advection schemes mentioned have these properties on uniform grids, even for divergent flows, which occur in the atmosphere. Positivity of species concentrations is guaranteed for time-step size restrictions such that no negative air masses can occur during any one of the split advection steps, which is a natural constraint.

1.2 Why zooming?

To capture local phenomena—as occurring over highly polluted areas—without increasing the cost too much, it is often desirable to have a nonuniform grid with local refinement (zooming) over the areas of interest.

The local grid refinement or zoom methods have been subject of active research (see e.g. relatively early works [1, 4]). Most of the effort has been to develop powerful *adaptive* grid refinement. In our case, however, adaptivity is not needed and not even welcome. First of all, areas of interest are known in advance (that can be Europe, for example). Second, meteorological data as e.g. wind fields are often not available on a fine grid in the whole domain (see Figure 1). Third, the position of the zoom regions is often determined by the chemistry part of the model: for example, one has to refine in the regions with high emission activity. Thus, our goal is an efficient robust zoom algorithm for *fixed* zoom regions and we propose a simple strategy for this.

2. Zooming technique

2.1 Requirements for zooming

Since mass conservation and positivity are important for the atmospheric application, we wish our zooming technique to preserve *both* these properties (something we have not managed to find in the literature). Due to the smaller size of the finite volumes, the CFL restriction is more severe in zoom

regions and a smaller time step Δt needs to be taken. Positivity and mass conservation are more difficult to preserve if we want to have smaller step sizes in the zoom regions only, so that e.g. in a zoom region with refinement factor two, two time steps are to be done within each global advance in time. Our zooming technique gives a simple and reliable way to have smaller time-step sizes in the zoom regions *only* while preserving positivity and mass conservation across zoom interfaces.

In the literature, the question of preserving positivity and mass conservation in grid refinement techniques seems to be left without much attention. The classical adaptive mesh refinement (AMR) algorithm of M. Berger [1, 2] employs a nested hierarchy of directional substeps similar to the one used in our algorithm (see Section 2.3). However, with the coarse-to-fine grid treatment in the AMR method (namely, sum of fine grid fluxes is taken as a flux on the border a zoom region), preservation of positivity appears problematic. In the AMR algorithm described in [10, 11], on the coarse-to-fine grid interface so-called dummy cells are used, which are similar to our interface cells (see Section 2.2). In [10] one particular second order advection scheme is shown to be monotonicity preserving via the coarse-to-fine grid interface provided that the CFL number is restricted appropriately. In contrast, with our algorithm no further CFL restrictions are necessary. Furthermore, the question of preserving positivity in the framework of directional splitting on the coarse-to-fine grid interface is not discussed in [10].

2.2 How to zoom in 1D

We explain first how to perform zooming for the one-dimensional (1D) transport equation $c_t+(uc)_x = 0$ and then how to extend this approach to the three-dimensional (3D) case. We use advection schemes in the mass conservative flux formulation. Each cell i contains a mass μ_i being the integral of c over the ith cell volume. In 1D fluxes $F_{i+1/2}$ are calculated giving the amount of chemical species transported per time step between cells i and $i+1$. They depend on the given air-mass fluxes determined from the velocity field and on the masses μ_i. The time advance has the form

$$\mu_i^{n+1} = \mu_i^n + F_{i-1/2}^n - F_{i+1/2}^n,$$

where $F_{i\pm1/2}^n$ are the fluxes at time level n.

On the boundaries of the zoom region its advection scheme is adjusted to the scheme on the coarse level. As an example, consider the situation with refinement factor `ref` $= 3$ (see Figure 2). This adjustment consists of two elements. First, the three outermost cells of the zoom region are lumped together and considered to be one single cell (the interface cell). The tracer mass in the interface cell is assumed to be distributed uniformly (or in accordance with its slope for the slope scheme) over its fine grid subcells, and the fluxes between these subcells are never computed. Second, on the boundary of the zoom region we calculate the flux F^C that is coarse in both space and time. For each coarse time step, there are three corresponding time substeps that have to be done in the zoom region. The coarse boundary flux F^C is applied at once at the first fine time substep, for the second and for the third substep zero flux is taken at the boundary of the zoom region.

It can be shown that this simple strategy guarantees mass conservation and positivity of the chemical species for many modern advection schemes including the Slopes and Split schemes (for more details see Appendix A). The time-step restriction for positivity is basically the same as in the uniform case, namely that the coarse air mass flux on the one wall plus the three fine air mass fluxes on the other wall do not take out more mass than there was in the cell. The strategy is also easily generalized to the multidimensional case, with a similar time-step restriction per direction.

Another choice would be to equidivide F^C in three parts and apply them consecutively at the same times as the fine fluxes on the other side of the interface cell. In 1D as well as in multidimensional cases this gives no longer a positive scheme, unless a more severe time-step restriction is enforced. The problem is that parts of the coarse flux, which was calculated with the *old* concentration field, are applied to a *new* concentration field. The latter may have changed in such a way that positivity is lost.

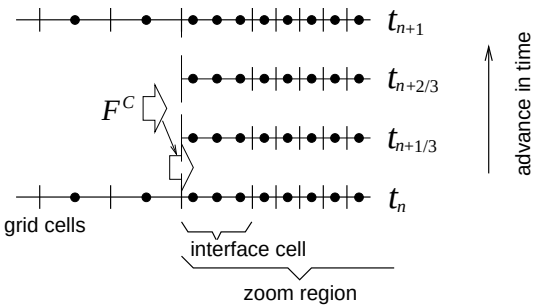

Figure 2: 1D zooming: one time step $t_n \rightarrow t_{n+1}$ corresponds to three time substeps in the zoom region. The "coarse" flux F^C is applied on the first substep.

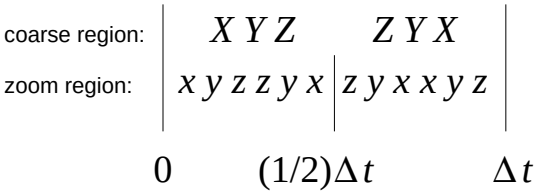

Figure 3: The sequence of advection steps in a coarse and a fine region with refinement factor 2.

2.3 How to zoom in 3D

We now give more details about our zooming algorithm in 3D. For simplicity reasons we restrict ourselves to block-shaped zoom regions in 3D. For our applications this is sufficiently general. A zoom region has interface cells all along its edges forming the boundaries with its parent region. The 1D algorithm described above can then be applied in a split manner. The only complication as compared with the 1D case is caused by the fact that the advection scheme is directionally split. We use a Strang (symmetric) splitting scheme consisting of six split steps as follows: $\hat{X}$-$\hat{Y}$-$\hat{Z}$-$\hat{Z}$-$\hat{Y}$-$\hat{X}$, where $\hat{X}$ denotes the application of the advection operator in the x-direction during half a time step, $\Delta t/2$, etc. For one triplet of advection steps—one in each direction—on the coarse level, e.g. $\hat{X}$-$\hat{Y}$-$\hat{Z}$, in the zoom region we perform `ref` triplets of advection steps in the same and the opposite order, thus $\hat{x}$-$\hat{y}$-$\hat{z}$-$\hat{z}$-$\hat{y}$-$\hat{x}$ in case `ref`$= 2$. Here $\hat{x}$ denotes the application of the advection operator in the x-direction during a quarter time step, $\Delta t/4$, etc. For example, in Figure 3 we show the sequence of the directional substeps in a zoom region with refinement factor 2. Clearly, Strang splitting order is preserved in the zoom region, which is important for accuracy reasons.

The coarse fluxes computed by the parent at the boundary with the child region are used as boundary conditions for the child region and are applied at once at the first time substep per direction. After the zoom or child region has carried out its `ref` triplets, the concentration field in the child region is copied back to its parent after suitable coarsening. This then allows the parent region to advance another three directional substeps and to provide new boundary conditions for the child region.

This 3D zooming algorithm can also be shown to be mass-conservative and positive for the advection schemes we are interested in. The complete description of our algorithm can be found in the appendices.

3. Experiments

With our tests we want to demonstrate that (i) our zooming technique gives an accuracy comparable with the accuracy one would get on the overall fine grid provided that important phenomena remain

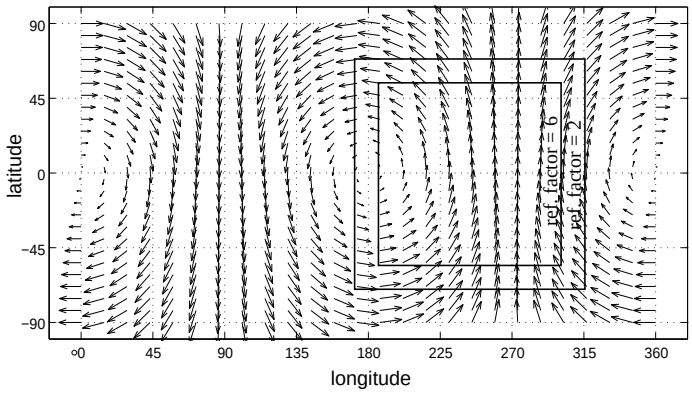

Figure 4: The velocity field and the position of the zoom regions.

in the zoom regions; (ii) this accuracy is achieved with significantly less computational effort; (iii) even when important phenomena occur outside the zoom regions, our algorithm still gives a solution which is at least as accurate as one would get on the overall coarse grid. The situation (i) is quite common for atmospheric modelling since one of the typical needs for the local refinement is concentration of the emission sources in highly polluted areas. We note in passing that zooming also leads to a more economical memory use, compared to the overall fine grid.

We present 2D numerical results obtained with our zoom implementation for the Slopes advection scheme. As a model problem for our experiments we take the solid-body rotation test commonly known as the Molenkamp-Crowley test [14]. In this test the advection equation on the (unit) sphere

$$c_t + \frac{1}{\cos y}\left((uc)_x + (v\cos y\, c)_y\right) = 0$$

is solved for the velocity field $u = 2\pi\cos x \sin y$, $v = -2\pi \sin x$. Here $x \in [0; 2\pi]$ and $y \in (-\pi/2; \pi/2)$ are the longitude and the latitude coordinates respectively. The chosen velocity field provides a solid-body nature of the air rotation over the sphere, so that after one full rotation (at time $t = 1$) the solution will coincide with the initial distribution $c(x, y, 0)$. For the initial distribution in our tests we take a cone of height 1 centered at the point $(x, y) = (3\pi/2, 0)$. We stress that the cone provides a severe test for any advection scheme. The radius of the cone base is taken to be 21 grid cells with respect to the finest zoom region used.

In Figure 4, we show the velocity field and the position of the zoom regions used in the tests. Overall, we have a nonuniform grid consisting of three regions: the coarsest, with resolution $\Delta x = \Delta y = 4.5°$ and two zoom regions refined with factors 2 and 6 (resolutions $\Delta x = \Delta y = 2.25°$ and $\Delta x = \Delta y = 0.75°$ respectively).

First of all we have compared the zoom grid solution with the solution obtained on the overall fine grid (with the same resolution as in the finest zoom region). As desired, it turned out that the difference between them is negligible as long as the moving cone remains in the zoom region. The difference in the solutions is explained by the fact that the order of the directional substeps in the zoom algorithm alternates (as in Figure 3) whereas it remains the same for the uniform grid algorithm.

Performing one full rotation ($0 \leqslant t \leqslant 1$) on the nonuniform zoom grid means that the cone has to travel over the whole sphere, leaving and then entering again the zoom regions. Of course, with the *local* zoom regions fixed in space, i.e. not moving with the cone, *globally* one can not expect much better accuracy than on the overall coarse grid. This is confirmed in our tests. We have performed one full rotation on the overall coarse ($\Delta x = \Delta y = 4.5°$), the zoom, and the overall fine ($\Delta x = \Delta y = 0.75°$) grids. In each case, we have measured accuracy by comparing the solution c^n

Table 1: Errors. "Local error" means with the cone cone still being within the finest zoom region, "global error" means after one full rotation.

Errors	local:			global:		
		emin	emax	err0	err1	err2
fine	$\leqslant$global	−5.9e−3	−3.1e−2	2.5e−3	1.4e−3	−3.5e−5
zoom	same as fine	−1.8e−2	−0.12	1.7e−2	1.8e−3	−0.12
coarse		−1.9e−2	−0.21	3.3e−2	2.6e−3	−0.15

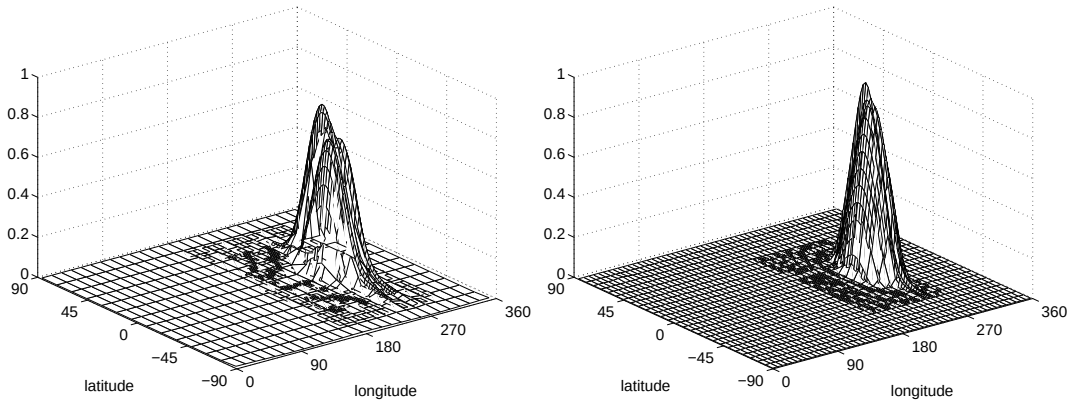

Figure 5: The cone after the full rotation.

after one full rotation with the initial distribution c^0 (the exact solution would give no difference with c^0). We have computed the following errors (representing minimum, maximum, scaled l_2, mean and variance errors respectively):

$$\text{emin} = \frac{\min c^n_{i,j} - \min c^0_{i,j}}{\max c^0_{i,j}}, \quad \text{emax} = \frac{\max c^n_{i,j} - \max c^0_{i,j}}{\max c^0_{i,j}},$$

$$\text{err0} = \frac{\sqrt{\sum \gamma_j (c^n_{i,j} - c^0_{i,j})^2}}{\max c^0_{i,j}}, \quad \text{err1} = \frac{\sum \gamma_j c^n_{i,j}}{\sum \gamma_j c^0_{i,j}} - 1, \quad \text{err2} = \frac{\sum \gamma_j (c^n_{i,j})^2}{\sum \gamma_j (c^0_{i,j})^2} - 1,$$

where γ_j is the (i,j)-cell air mass divided by the total mass of the air (cells near the poles have less mass), and the sums and the minima/maxima are taken for all the grid cells within the finest zoom region. To be able to compare errors of different space resolutions, on the overall fine and on the zoom grids we computed the errors on the data coarsened up to the coarse grid resolution. We have summarized our accuracy observations in Table 1.

In Figure 5 (left picture), we plot the zoom solution after the full rotation. As we see, the cone shape is strongly deformed. This is not only because it has travelled through the grid that is 6 times coarser (we should emphasize that the resolution 4.5° is quite coarse for this test) but also because of the accuracy losses near the poles. For comparison reasons, at the right plot we present the solution for the same test but with half the grid size in the coarse region (in other words, the cone is to travel through the grid that is only factor 3 coarser, with basic resolution $\Delta x = \Delta y = 2.25°$). As expected, the deformation is significantly decreased.

Finally, we comment briefly on the computational expenses. Rough estimates taking into account total number of grid cells and the CFL restriction on the time step size show that the computations

on the overall fine grid would be 5.2 times more expensive than on the zoom grid. This is however a too optimistic speed-up estimate which does not take into account communication overhead for the zoom algorithm. With the current implementation (SGI workstation) we observed that our zoom algorithm is approximately 2.6 times faster than the uniform algorithm on the fine grid. However, code optimization has to be performed yet.

4. Conclusions

We have presented a positive and mass-conservative local grid refinement (zoom) algorithm for advective transport. The algorithm can be applied to many modern advection schemes with directional splitting in space and explicit advance in time. With our approach, a smaller time step (due to the stricter CFL condition) is taken within the zoom regions only.

The described zoom algorithm has been implemented in a code that allows to use an arbitrary number of zoom regions that can lie inside each other in an arbitrary way, provided they are of block shape and strictly embedded. To have high performance of the uniform grid code preserved as much as possible, the work of the zoom code is organized in a uniform region-wise manner, i.e. the zoom grid is split into a cascade of uniform grid regions. For more details on the implementation we refer to Appendix B.

Our zoom code will be incorporated in the air pollution model TM3, a recent successor of [6]. The TM3 model is operational at the Institute of Marine and Atmospheric Research (IMAU, Utrecht University), the Dutch Royal Meteorological Center (KNMI) and the Dutch National Institute of Public Health and the Environment (RIVM)[1].

References

1. M. J. Berger. *Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations.* PhD thesis, Department of Computer Science, Stanford University, Stanford, CA 94305, Aug. 1982.
2. M. J. Berger and P. Collela. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82(1):64–84, 1989.
3. P. J. F. Berkvens, M. A. Botchev, W. M. Lioen and J. G. Verwer. A zooming technique for wind transport of air pollution. In R. Vilsmeier, D. Hänel, and F. Benkhaldoun, editors, *Finite Volumes for Complex Applications*, pages 499–506. Hermes Science Publications, 1999.
4. J. H. Flahery, P. J. Paslow, M. S. Shephard, and J. D. Vasilakis, editors. *Adaptive methods for Partial Differential Equations.* SIAM, Philadelphia, PA, 1989.
5. T. E. Graedel and P. J. Crutzen. *Atmosphere, Climate and Change.* Scientific American Library. Freeman and Company, New York, 1995.
6. M. Heimann. The global atmospheric tracer model TM2. Technical Report 10, Deutches Klimarechenzentrum (DKRZ), Hamburg, 1995.
7. W. Hundsdorfer, B. Koren, M. van Loon and J. G. Verwer. A positive finite-difference advection scheme. *Journal of Computational Physics*, 117:35–46, 1995.
8. B. Koren. A robust upwind discretization method for advection, diffusion and source terms. In C. B. Vreugdenhill and B. Koren, editors, *Numerical methods for advection-diffusion problems*, volume 45 of *Notes on Fluid Mechanics*, pages 117–138. Braunschweig, 1993.
9. A. C. Petersen, E. J. Spee, H. van Dop, and W. Hundsdorfer. An evaluation and intercomparison of four new advection schemes for use in global chemistry models. *Journal of Geophysical Research*, 103(D15):19,253–19,269, Aug. 1998.
10. J. J. Quirk. *An adaptive grid algorithm for computational shock hydrodynamics.* PhD thesis, Cranfield institute of technology, College of Aeronautics, 1991.

[1] For more information, see URLs `http://www.cwi.nl/cwi/projects/TM3/` and `http://www.phys.uu.nl/~peters/TM3/TM3S.html`

11. J. J. Quirk. A parallel adaptive grid algorithm for computational shock hydrodynamics. *Appl. Numer. Math.*, 20(4):427–453, 1996.

12. G. L. Russell and J. A. Lerner. A new finite-differencing scheme for the tracer transport equation. *J. Appl. Meteor.*, 20:1483–1498, 1981.

13. J. G. Verwer, W. Hundsdorfer and J. G. Blom. Numerical time integration for air pollution models. Report MAS-R9825, CWI, 1998.

14. D. L. Williamson and P. J. Rasch. Two-dimensional semi-Lagrangian transport with shape-preserving interpolation. *Monthly Weather Review*, 117:102–129, 1989.

15. Z. Zlatev. *Computer treatment of large air pollution models*. Kluwer Academic Publishers, 1995.

A. Positivity and mass conservation of the zoom scheme

Here we show that, if the original advection scheme on the uniform grid is positive and mass conservative, our zoom approach leads to an advection zoom scheme that is positive and mass conservative too. The advection schemes used in the zoom code (the slopes and the split schemes) are described in Appendix C.

A.1 Positivity on a uniform grid

Consider the 1D transport equation $c_t + (u(x,t)c)_x = 0$. In finite volume discretizations we work with the tracer mass μ_i^n which is the (approximate) integral of $c(x,t_n)$ over the grid cell i. We call an advection scheme

$$\mu_i^{n+1} = \mu_i^n + F_{i-1/2}^n - F_{i+1/2}^n \tag{A.1}$$

positive if for all grid cells i $\mu_i^n \geqslant 0 \Rightarrow \mu_i^{n+1} \geqslant 0$ at any time step n. For many advection schemes, including the schemes we use (see Appendix C),

$$\begin{aligned} F_{i-1/2} < 0 \quad &\Rightarrow \quad \mu_i + F_{i-1/2} \geqslant 0, \\ F_{i+1/2} > 0 \quad &\Rightarrow \quad \mu_i - F_{i+1/2} \geqslant 0, \end{aligned} \tag{A.2}$$

which means that tracer mass leaving the grid cell through each of the grid cell walls can not exceed tracer mass of the grid cell. Therefore, aiming at positivity, one often is content with the condition

$$\mu_i^n + F_{i-1/2}^n \geqslant 0 \quad \text{and} \quad \mu_i^n - F_{i+1/2}^n \geqslant 0. \tag{A.3}$$

This condition is not sufficient for positivity. (In view of (A.2), it is, obviously, a necessary condition for positivity.) In practice, however, condition (A.3) leads to a scheme which is very likely to be positive. Indeed, if (A.3) is true then positivity in (A.1) may only be lost at a certain grid cell i if

$$F_{i-1/2}^n < 0 \quad \text{and} \quad F_{i+1/2}^n > 0. \tag{A.4}$$

To see this, assume that positivity is lost, i.e.

$$(\mu_i^{n+1} =) \quad \mu_i^n + F_{i-1/2}^n - F_{i+1/2}^n < 0, \tag{A.5}$$

whereas (A.2) is still true. Rewriting (A.5) as $F_{i-1/2}^n < -(\mu_i^n - F_{i+1/2}^n)$ we notice that the right-hand side in this inequality is nonpositive because of (A.2). This means $F_{i-1/2}^n < 0$. The inequality $F_{i+1/2}^n > 0$ can be proven in the same way. Relations in (A.4) mean that the velocity changes its sign inside the grid cell i and hence, provided that the grid is fine enough to resolve this velocity sign change smoothly, $|u_{i\pm1/2}|$ is small as compared with $\max_i |u_{i+1/2}|$. The same, of course, is true for the local CFL numbers $\mathcal{C}_{i\pm1/2} = |u_{i\pm1/2}|\Delta t/\Delta x$: they are small with respect to the global CFL number $\mathcal{C} = \max_i \mathcal{C}_{i+1/2}$.

This local smallness of the CFL number is already sufficient for many advection schemes to remain positive. For example, if for the split scheme condition (A.4) is true at some grid cell i, the scheme will remain positive up to $\mathcal{C}_{i\pm1/2} \leqslant \frac{1}{2}\mathcal{C}$. The same is true for the slopes scheme and for the simple upwind donor-cell scheme.

A.2 Positivity and mass conservation in 1D

To make clear that both positivity and mass conservation are preserved with our zoom approach, we now add some detail to our explanation of Section 2.2. Figure 6 represents the way the fluxes are computed and applied at the interface of the zoom region. This is depicted for a particular value of the refinement factor `ref` $= 3$.

The key idea is to have a so-called interface cell at the zoom region boundary. This is the cell which is just one single cell with respect to the coarse grid and, coupled together, `ref` grid cells with respect

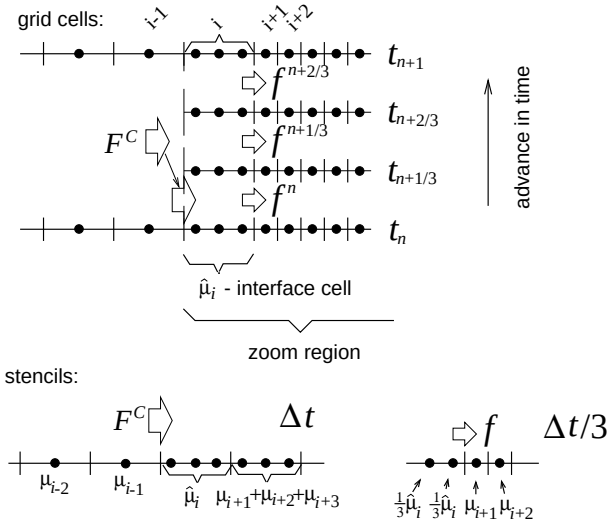

Figure 6: Time step advance $t_n \to t_{n+1}$ and fluxes of the interface cell.

to the zoom region. The tracer mass in the interface cell is assumed to be distributed uniformly (or in accordance with its slope for the slope scheme) over its fine grid subcells, and the fluxes between these subcells are never computed. Since both to the left and to the right of the interface cell we have uniform grid advection schemes, positivity and mass conservation may be corrupted only by the zooming at the interface cell.

Since we do not want to have smaller time steps imposed by the CFL restriction in the zoom region everywhere, for each "coarse" time step `ref` substeps will be done in the zoom region. As shown in Figure 6, at the wall next to the coarse region the coarse flux F^C is applied on the first time substep. This flux is coarse in both time and space, i.e. it is computed with respect to the coarse grid time step Δt and on the coarse grid stencil as shown in Figure 6. In all the remaining substeps zero fluxes are applied at the next-to-coarse wall of the interface cell. On the opposite wall of the interface cell, fine grid fluxes f are applied. These are fine both in time and space (see Figure 6).

We denote the tracer mass of the interface cell by $\hat{\mu}_i$, with the hat ˆ to emphasize that it is actually a couple of fine grid cells. To summarize, we advance $\hat{\mu}_i$ in time in the following way:

$$\begin{aligned} \hat{\mu}_i^{n+\frac{1}{\mathtt{ref}}} &= \hat{\mu}_i^n + F^C_{i-1/2} - f^n_{i+1/2}, \\ \hat{\mu}_i^{n+\frac{k}{\mathtt{ref}}} &= \hat{\mu}_i^{n+\frac{k-1}{\mathtt{ref}}} + 0 - f^{n+\frac{k-1}{\mathtt{ref}}}_{i+1/2}, \quad k = 2, \ldots, \mathtt{ref}, \end{aligned} \tag{A.6}$$

where `ref` is the refinement factor and the fluxes $F^C_{i-1/2}$ and $f_{i+1/2}$ are computed as shown in Figure 6. Figure 6 and update (A.6) describe the update on the left edge of the zoom region. The work on the right edge is done in a similar way, with obvious symmetric adjustments.

The update (A.6) is mass-conservative because of its flux formulation (i.e. the same fluxes are applied to the interface cell as well as to its neighbouring cells). Positivity immediately follows from the fact that the fluxes in (A.6) are computed naturally, i.e. in a way they would be computed on a uniform grid[2]. Since the advection scheme used for zoom is assumed to be positive on a uniform (on

[2]Of course, for preserving positivity it is also important that the fluxes are applied to the cells before they have been changed by other processes.

the fine as well as on the coarse) grid, we have

$$\hat{\mu}_i^n + F_{i-1/2}^C \geqslant 0, \qquad \frac{1}{\texttt{ref}}\hat{\mu}_i^n - f_{i+1/2}^n \geqslant 0, \tag{A.7}$$

$$\frac{1}{\texttt{ref}}\hat{\mu}_i^{n+\frac{k-1}{\texttt{ref}}} - f_{i+1/2}^{n+\frac{k-1}{\texttt{ref}}} \geqslant 0, \quad k = 2, \ldots, \texttt{ref}. \tag{A.7$'$}$$

Relations (A.7) are in fact the desired positivity condition (A.3) for the first substep in time, and relations (A.7′) give (A.3) for the other substeps. Therefore, we conclude the 1D zoom advection scheme is positive as long as the original advection scheme on the uniform grid is positive.

A.3 Positivity and mass conservation in 3D

The zoom region is now a box which has interface cells along each of its edges. The interface cells form walls around the zoom region and we will call these interface cell walls x-, y-, and z-walls.

To demonstrate positivity and mass conservation for the 3D case, we give a full description of our algorithm. The algorithm is based on one simple principle: to be able to use our 1D zoom algorithm in the directional splitting, at each directional substep we have to apply the outer coarse fluxes to the interface cells not only of this direction but also of the directions that are still to be advanced. For example, for the sequence $\hat{x}$-$\hat{y}$-$\hat{z}$- … of the substeps in a zoom region that corresponds to the coarse grid advance $\hat{X}$-$\hat{Y}$-$\hat{Z}$, coarse x-fluxes will be applied to x-, y- and z-walls, coarse y-fluxes will be applied to y- and z-walls, and coarse z-fluxes to z-walls only. Thus, at each directional substep interface cells of this direction have been already updated in the directions of the previous substeps (if there have been any directional substeps before). This guarantees positivity without loosing mass conservation because the interface cells are handled in a way they would be handled with directional splitting on the uniform coarse grid.

Let us assume that the full advection step consists of the Strang sequence $\hat{X}$-$\hat{Y}$-$\hat{Z}$-$\hat{Z}$-$\hat{Y}$-$\hat{X}$, so that the time step size for all the directional substeps is $\Delta t/2$.

As explained in Section 2.3, for each triplet of the directional substeps (e.g. $\hat{X}$-$\hat{Y}$-$\hat{Z}$) in the coarse region, `ref` triplets will be done in the zoom region. In fact, it suffices to consider only one such triplet advance including all the substeps of the zoom region. For the coarse grid sequence $\hat{X}$-$\hat{Y}$-$\hat{Z}$, the first triplet in the zoom region will have the same sequence: $\hat{x}$-$\hat{y}$-$\hat{z}$. At these first three steps the entire coarse outer fluxes will be applied, just as in the 1D case. The sequence of the rest of zoom substeps is important neither for positivity nor for mass conservation (to have the Strang second order splitting, we determine this sequence from symmetry considerations).

The coarse fluxes from the first substep $\hat{X}$ will be applied to x-, y- and z-walls (as shown in Figure 7). This means that the first zoom substep $\hat{x}$ will be done for the interior of the zoom region exluding its y- and z-walls. The interface cells of these walls will be updated with the coarse x-fluxes and therefore for this update considered to be coarse in all directions. Evidently, positivity is lost nowhere since the y- and z-walls are updated by the uniform grid advection scheme, whereas the rest of the cells are updated by the positive 1D zoom algorithm. All the remaining $\texttt{ref} - 1$ substeps $\hat{x}$ for the zoom region will be done with zero outer fluxes, exactly as in the 1D case. These substeps will be carried out for the interior cells excluding the y- and z- walls (which have been already advanced in x-direction for $\Delta t/2$).

Next, at the substep $\hat{y}$, the coarse fluxes from the coarse substep $\hat{Y}$ are applied to y- and z-walls (see Figure 8). The cells of the x-walls are now updated on the fine scale (in the same way as all the internal interface grid cells). Furthermore, as can be seen in Figure 8, coarse y-fluxes are applied on the cells of the walls that have been updated on both sides by the coarse x-fluxes. This means that positivity can not be lost in those cells. Positivity is also preserved in the internal and x-wall cells—here the positive 1D zoom algorithm is employed. At this moment, the z-walls have already been advanced in time for $\Delta t/2$ in both x- and y-directions. As prescribed by our 1D zoom scheme, the remaining $\texttt{ref} - 1$ substeps $\hat{y}$ will be performed with zero outer y-fluxes.

Finally, the $\hat{z}$ update is performed on all the internal cells and both x- and y-walls (their grid cells

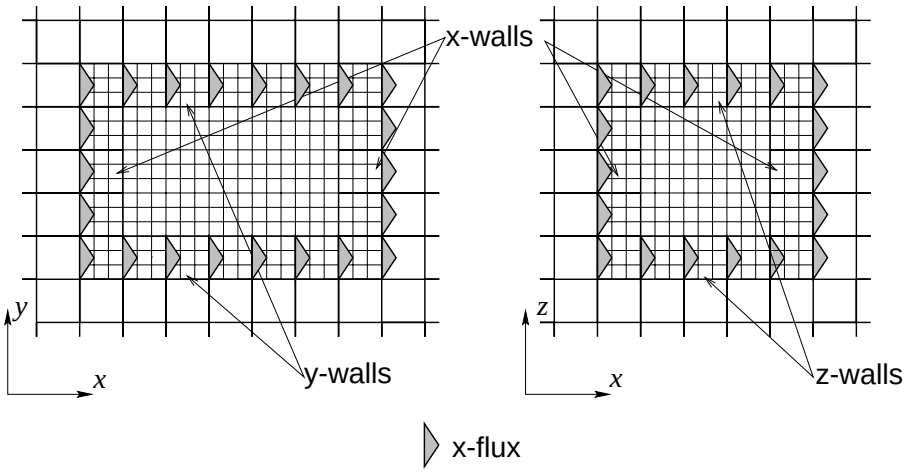

Figure 7: First substep $\hat{x}$: the coarse x-fluxes are applied to x-, y-, and z-walls.

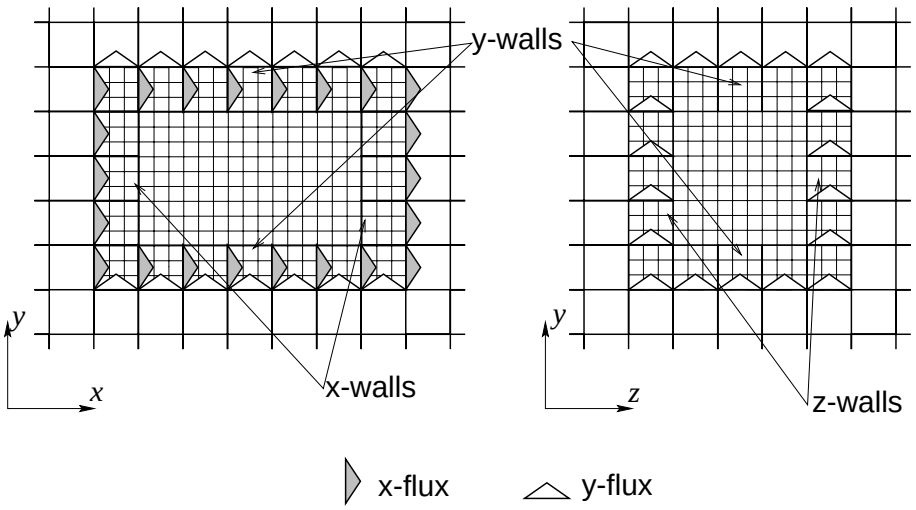

Figure 8: Second substep $\hat{y}$: the coarse y-fluxes are applied to y- and z-walls.

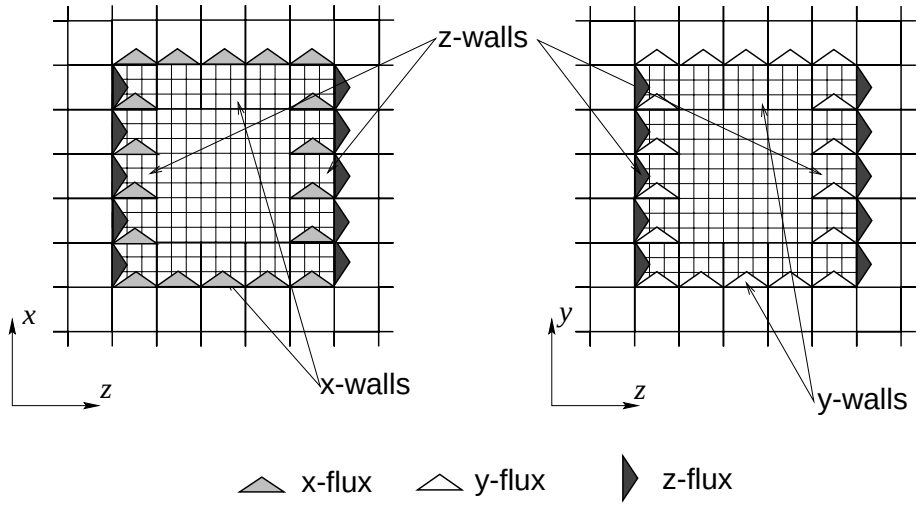

Figure 9: Second substep $\hat{z}$: the coarse z-fluxes are applied to z-walls only.

are updated on the fine scale). In fact, all the cells are treated now according to the 1D zoom scheme (see Figure 9). The coarse z-fluxes are applied to z-walls only.

Summarizing, each of the rest of the substeps $\hat{x}$, $\hat{y}$, $\hat{z}$ in the zoom region is carried out on the same cells as at the first substep for this direction. The only difference is that zero outer flux is now taken.

B. Zoom code

Our zoom algorithm can be implemented with relatively low programming efforts. Here we explain how to do this by giving some details on zoom implementation for the TM3 global atmospheric model, a recent successor of [6].

B.1 Data structure of the zoom code

The main variable in the TM3 model is the tracer mass per grid cell μ which is simply tracer concentration c integrated over the grid cell. In the code μ is stored in the array `rm`. Declared as a one-dimensional array, `rm` contains tracer mass of all regions (i.e., of the coarsest grid region and all the zoom regions) one by one. For example, to perform an advection substep in the direction x for a certain region, a subroutine `advectxzoom` is called with the `rm` section of the region as a parameter. Inside `advectxzoom`, a formal parameter `rm` is declared as

```
rm(-1:im(reg)+2,-1:jm(reg)+2,-1:lm(reg)+2,ntrace),        (B.1)
```

where `im`, `jm`, and `lm` are the numbers of grid cells in region `reg` for directions x, y, and z respectively and `ntrace` is the number of tracers.

If a zoom region `reg` has in turn more zoom regions inside (we will call them children), this does not affect the data structure (B.1). In other words, for each of the subdomains of the children zoom regions two grids are stored: the coarser grid of the parent region `reg` and the child region grid (see Figure 10). This allows to work with uniform data structures, thus avoiding indirect addressing.

The same data structure is used for the other grid data in the TM3 as e.g. air mass m (array `m`), air mass fluxes $A_{i\pm1/2,j,l}$, $B_{i,j\pm1/2,l}$, $C_{i,j,l\pm1/2}$ (arrays `am`, `bm`, `cm`), etc.

A section of `rm` containing data of a particular region `reg` is accessed as `rm(addr_rm(reg))` where `addr_rm` is a precomputed address array.

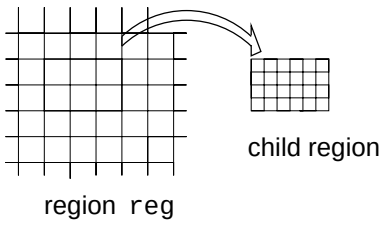

Figure 10: There are two grids for zoom subdomains: a coarse and a fine.

B.2 Algorithmic structure of the code
The constants that control the work of the zoom code are:

`integer nregions` Number of zoom regions including the coarsest grid.

`real,parameter,dimension(nregions):: xbeg,xend,ybeg,yend,zbeg,zend` Physical coordinates of the zoom regions.

`integer,parameter,dimension(0:nregions):: xref,yref,zref,tref` Refinement factors in each of the space directions and time specified for each region. To comply with the CFL condition, `tref` should be chosen such that `tref` $\geqslant \max\{$`xref`, `yref`, `zref`$\}$. Note that region `0` does not exist, we introduced zero entries in the refinement factors for coding convenience.

`integer,parameter,dimension(nregions):: parent` Parent regions specified for each of the zoom region. By definition, `parent(1)=0`.

These are the constants that are set by the user of the zoom code. Other important zoom variables are:

`integer,dimension(nregions,0:nregions):: children` For region `reg` `children(reg,0)` is the total number of children in the region, `children(reg,1)` is number of the first child of region `reg`, `children(reg,2)` is number of second child, etc.

`nsplitsteps` Number of split steps in the full advection step. We use splitting x-y-z-z-y-x and thus `nsplitsteps` $= 6$.

`character splitorder(nsplitsteps)` order of substeps in the directional splitting of the advection advance, usually `['x', 'y', 'z', 'z', 'y', 'x']`.

`character splitorderzoom(nregions,nsplitsteps*maxref)` Here `maxref` is maximal possible refinement factor (can be any number, in principle). `splitorderzoom` is the `splitorder` but specified per region according to its refinement factor `tref`. `splitorderzoom` is computed by the code automatically. For the situation as shown in Figure 3 we would have

```
splitorderzoom(1,:)=(/'x','y','z','z','y','x',' ',.../),
splitorderzoom(2,:)=(/'x','y','z','z','y','x',
                      'z','y','x','x','y','z',' ',.../).
```

The rest of the array is filled with `' '`.

To give an impression of how the code works, in Figure 11 we give a fragment of the code containing a subroutine `advect_region`. This is the key recursive subroutine of the zoom code that performs the advection advance for the next three substeps x, y, and z in the order prescribed by

```
recursive subroutine advect_region(region,rm,rxm,rym,rzm,am,bm,cm,m)
! for a given region, performs tref(region)/tref(parent(region))
! advection steps for each direction x,y,z
! including all region's children, granchildren, etc.
use prepareZoom
implicit none
integer region
real,dimension(*):: rm,rxm,rym,rzm,am,bm,cm,m
! local:
integer child,i,ichild,tref_

! determine refinement factor with respect to the parent
tref_ = tref(region)/tref(parent(region))

do i=1,tref_
   ! do next three substeps x, y, z...
   call do_next3(region,rm,rxm,rym,rzm,am,bm,cm,m)

   ! call advect_region for all the children (if there are any)
   ichild = 0
   do while(ichild<children(region,0))
      ichild = ichild + 1
      child = children(region,ichild)

      call advect_region(child,rm,rxm,rym,rzm,am,bm,cm,m)
   enddo
enddo

call update_parent(region,m(addr_m(region)),rm(addr_rm(region)),      &
                   rxm(addr_rxm(region)),rym(addr_rxm(region)),       &
                   rzm(addr_rxm(region)),m,rm,rxm,rym,rzm             )

end subroutine advect_region
```

Figure 11: Subroutine `advect_region`.

the `splitorderzoom` array. The advance is done for the region including all its zoom subregions (children). First, in the current region, the next three directional substeps are carried out in the order prescribed by the `splitorderzoom`. This is done by the subroutine `do_next3`. Next, `advect_region` is called recursively for all the children regions. The whole procedure is repeated `tref_` times,

$$\texttt{tref\_} = \texttt{tref(region)/tref(parent(region))},$$

so that the region reaches the same level in time as its parent. The subroutine is completed with the call to `update_parent` which puts the zoom region data into the coarser cells of the region's parent.

We now briefly comment on the way the updates of the interface cell walls are implemented (cf. Section A.3). Assume, for example, that the sequence of substeps in a zoom region is $\hat{x}$-$\hat{y}$-$\hat{z}$- ... and the first substep $\hat{x}$ is to be done. According to our zoom algorithm, at the first directional substep the coarse x-fluxes are to be applied at the x-, y-, and z-walls (as shown in Figure 7), and the interior zoom region cells are updated with regular uniform grid scheme. In the code, this work is split into two parts. First, subroutine `advectxzoom` handles all the zoom region cells except y- and z-walls. Next, the y- and z-walls have to be updated with the coarse x-fluxes. The actual update is not necessary to be done since it has already taken place in the parent region. Therefore, we simply have the parent region fill the y- and z- walls of the region with the updated values.

We note also that, in fact, `advectxzoom` applies the 1D zoom algorithm in direction x with possibly restricted (to leave out y- and z-walls) y- and z-scopes. Whether the y- and/or z- scope are/is to be restricted is determined by the number of the current substep in the directional splitting.

C. Zoom advection schemes in the TM3 model

Several modern advection schemes are available in the TM3 model. Instructive comparisons of them can be found in [9]. Here we give a description of two basic TM3 advection schemes available in the zoom code.

For simplicity, we will write down the schemes for the 1D case. This is also natural because both schemes are one-dimensional and applied directionally split.

Advection schemes in the TM3 model are used in the air mass flux formulation, i.e. the air transport process is prescribed as

$$m_i^{n+1} = m_i^n + A_{i-1/2}^n - A_{i+1/2}^n, \tag{C.1}$$

where $A_{i\pm1/2}^n$ are the air mass fluxes. Note that (C.1) is a discrete version of the continuity equation

$$\rho_t + (u\rho)_x = 0, \qquad \rho = \rho(x,t), \quad u = u(x,t), \tag{C.2}$$

where ρ is the air density and u is the velocity. Evidently, (C.2) would be recovered from the 1D transport equation

$$\begin{aligned} &c_t + (uc)_x = 0, \quad \text{or} \\ &(\rho\chi)_t + (u\rho\chi)_x = 0 \quad (c = \rho\chi) \end{aligned} \tag{C.3}$$

in case the mixing ratio $\chi \equiv 1$. Both the advection schemes are discrete formulations of (C.3), they describe the explicit time stepping relation

$$\mu_i^{n+1} \quad \Leftarrow \quad \mu_i^n,$$

where μ is tracer mass in the grid box i: $\mu_i = \chi_i m_i$.

C.1 Slopes scheme

In the slopes scheme [12], mixing ratio χ is considered to be linearly distributed within each grid cell. This means that for the grid cell i, we have two parameters describing the mixing ratio: the average

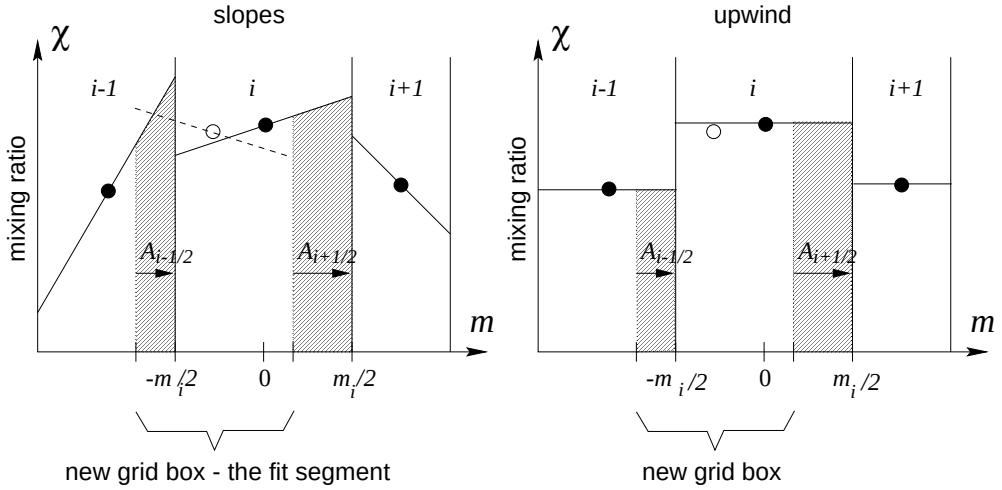

Figure 12: The slopes scheme vs. the upwind donor cell scheme. The abscissa is the air mass. The hatched areas are the tracer fluxes $F^n_{i\pm1/2}$, i.e. tracer masses moved into and out of the grid box i. The open circle is the tracer mixing ratio at the end of the time step χ_i^{n+1}. The dashed line is a result of the least squares fitting.

mixing ratio χ_i and the slope s_i. A natural way to appreciate the idea behind the slopes scheme is to consider a new frame with a mass coordinate m having zero at the center of the grid box i. Using the continuity equation (C.2), we introduce the m coordinate as a Lagrangian coordinate:

$$\frac{\partial m}{\partial x} = \rho, \qquad \frac{\partial m}{\partial t} = -\rho u.$$

In Figure 12 we have depicted the slopes scheme update as compared to the upwind donor-cell scheme update. As we see, the only difference is that the slope s_i takes part in the computations of fluxes $F^n_{i\pm1/2}$, for the zero slope the slopes scheme would degenerate into the donor-cell scheme. We determine the new slope value s_i^{n+1} by finding a linear function $\chi_i^{n+1} + s_i^{n+1} m$ that would fit in the least squares sence to the piecewise linear distribution of the mixing ratio on the current time step n. The fitting is made within the grid box i of the time step $n+1$, i.e. for the segment $[-\frac{1}{2}m_i - A_{i-1/2}; \frac{1}{2}m_i - A_{i+1/2}]$ (see Figure 12).

In overall, denoting $\sigma_i = \frac{1}{2} s_i m_i$ (in the TM3 code, σ_i is stored in the array `rxm`), we obtain the

following time-step update for the slopes scheme:

$$
\begin{aligned}
m_i^{n+1} &= m_i^n + A_{i-1/2}^n - A_{i+1/2}^n,\\
\mu_i^{n+1} &= \mu_i^n + F_{i-1/2}^n - F_{i+1/2}^n,\\
F_{i+1/2}^n &= \begin{cases} \dfrac{A_{i+1/2}^n}{m_i^n}\left(\mu_i^n + \left(1 - \dfrac{A_{i+1/2}^n}{m_i^n}\right)\sigma_i^n\right) & \text{if } A_{i+1/2}^n \geqslant 0,\\ \dfrac{A_{i+1/2}^n}{m_{i+1}^n}\left(\mu_{i+1}^n - \left(1 + \dfrac{A_{i+1/2}^n}{m_{i+1}^n}\right)\sigma_{i+1}^n\right) & \text{if } A_{i+1/2}^n < 0,\end{cases}\\
\sigma_i^{n+1} &= \sigma_i^n + \Big\{P_{i-1/2}^n - P_{i+1/2}^n - (A_{i-1/2}^n - A_{i+1/2}^n)\sigma_i^n +\\
&\qquad 3\big[(A_{i-1/2}^n + A_{i+1/2}^n)\mu_i^{n+1} -\\
&\qquad (F_{i-1/2}^n + F_{i+1/2}^n)m_i^n\big]\Big\}\Big/ m_i^{n+1}.\\
P_{i+1/2}^n &= \begin{cases} A_{i+1/2}^n\left(\left(\dfrac{A_{i+1/2}^n}{m_i^n}\right)^2\sigma_i^n - 3F_{i+1/2}^n\right) & \text{if } A_{i+1/2}^n \geqslant 0,\\ A_{i+1/2}^n\left(\left(\dfrac{A_{i+1/2}^n}{m_{i+1}^n}\right)^2\sigma_{i+1}^n - 3F_{i+1/2^n}\right) & \text{if } A_{i+1/2}^n < 0.\end{cases}
\end{aligned}
$$

Since the slopes scheme can be readily used on a nonuniform grid[3] the zoom algorithm can be slightly simplified for this scheme. At the first substep, the entire coarse flux F^C is applied to the interface cell (cf. Figure 6), but the fine flux f used at tne opposite wall of the interface cell is computed in a different way. It is computed fine in time (i.e. with respect to $\Delta t/\texttt{ref}$ rather than with respect to Δt) and half-coarse in space (i.e. the grid cell on one side is coarse, this is the interface cell, and on another is fine). The remaining $\texttt{ref} - 1$ time substeps in the zoom region are carried out with zero outer fluxes instead of F^C (we set mass outer fluxes to zero: $A_{i-1/2}^{n+\frac{k-1}{\texttt{ref}}} = 0$, $k = 2, \ldots, \texttt{ref}$) while f fluxes are computed in the same way.

C.2 Split scheme

The split scheme is a flux-limited third order upwind scheme introduced in [7]. We use the split scheme in the following formulation:

$$
\begin{aligned}
m_i^{n+1} &= m_i^n + A_{i-1/2}^n - A_{i+1/2}^n,\\
\mu_i^{n+1} &= \mu_i^n + F_{i-1/2}^n - F_{i+1/2}^n,\\
F_{i+1/2}^n &= \begin{cases} \dfrac{A_{i+1/2}^n}{m_i^n}\left(\mu_i^n + \psi\left(\dfrac{A_{i+1/2}^n}{m_i^n}, \theta_i^n\right)\left(\mu_{i+1}^n\dfrac{m_i^n}{m_{i+1}^n} - \mu_i^n\right)\right) & \text{if } A_{i+1/2}^n \geqslant 0,\\ \dfrac{A_{i+1/2}^n}{m_{i+1}^n}\left(\mu_{i+1}^n + \psi\left(\dfrac{|A_{i+1/2}^n|}{m_{i+1}^n}, \dfrac{1}{\theta_{i+1}^n}\right)\left(\mu_i^n\dfrac{m_{i+1}^n}{m_i^n} - \mu_{i+1}^n\right)\right) & \text{if } A_{i+1/2} < 0,\end{cases}\\
\theta_i^n &= \frac{\chi_{i-1}^n - \chi_i^n}{\chi_i^n - \chi_{i+1}^n}, \qquad \chi_i^n = \mu_i^n/m_i^n.
\end{aligned}
\tag{C.4}
$$

Here $\psi(\nu, \theta)$ is the Koren-Sweby limiter [8]:

$$
\begin{aligned}
\psi(\nu,\theta) &= \max\left\{0, \min\left\{1, d_0(\nu) + d_1(\nu)\theta, \frac{1-\nu}{\nu}\theta\right\}\right\},\\
d_0(\nu) &= \frac{1}{6}(2-\nu)(1-\nu), \qquad d_1(\nu) = \frac{1}{6}(1-\nu^2).
\end{aligned}
$$

[3] We emphasize, however, that preservation of positivity and mass conservation is still not straightforward for the slopes scheme when smaller time-step sizes are used in the zoom regions and directional splitting is employed.

The flux limiting in (C.4) guarantees that if the mixing ratio is constant through out the grid, it will be preserved constant:

$$\mu_i^n/m_i^n = \text{const}(i) \quad \Rightarrow \quad \mu_i^{n+1}/m_i^{n+1} = \text{const}(i).$$

D. Conservation of Three Physical Quantities by the Zoom Scheme

The aim of this appendix is twofold. The first is to prove the conservation of three relevant physical quantities, namely mass, positivity, and constant mixing ratio, by the present zoom scheme, which provides nested Strang. The second is to enhance understanding of what causes the conservation of these quantities: what are the necessary constraints and what are the remaining freedoms in the design of schemes which conserve these quantities. In Subsection D.1 we describe the zoom scheme. In Subsection D.2 we prove conservation of mass. In Subsection D.3 we prove and study conservation of positivity. In Subsection D.4 we prove conservation of constant mixing ratio.

D.1 Description of the Zoom Scheme

We give an alternative description of the zoom algorithm developed for combining of directionally split advection schemes with nested Strang splitting in zoom regions. The description in this subsection is general in the sense that it can be applied to any advection scheme in flux formulation which is used in a directionally split manner.

D.1.1 General Lay-Out of Grids Th lay-out consists of a main grid with one or more recursively embedded subgrids. All grids are block-shaped. Mathematically speaking the grid is nonuniform both in space and in time. Algorithmically speaking the grid consists of several uniform grids, but with space and time resolutions differing per grid. The refinement factors of a certain (fine) grid F with respect to the next coarser grid C are r_x, r_y, and r_z, respectively, for the mesh sizes in the directions x, y, z, and r_t for the time-step size. All refinement factors are integers greater than or equal to one. The numbers of cells per direction on a grid are indicated by n_x, n_y, and n_z, with $n_d \geq 3r_d$, $d = x, y, z$.

D.1.2 Interface Edges Each grid F, except the coarsest main grid, has interface cells along its edges corresponding to cells on the next coarser grid C. (F is nested in C.) These cells are treated in a special way during the advection process and they serve as communication interfaces between F and C. F is divided into four regions in two different ways. Both have the same interior region I:

$$I = \{(i,j,k)|(i,j,k) \in [r_x+1, n_x-r_x] \times [r_y+1, n_y-r_y] \times [r_z+1, n_z-r_z]\} \ , \tag{D.1}$$

where (i,j,k) are cell numbers or coordinates. In the first division the edges are divided as follows:

$$E_x^{(1)} = \{(i,j,k)|(i,j,k) \in ([1,r_x] \cup [n_x-r_x+1, n_x]) \times [r_y+1, n_y-r_y] \times [r_z+1, n_z-r_z]\} \tag{D.2}$$

$$E_y^{(1)} = \{(i,j,k)|(i,j,k) \in [1,n_x] \times ([1,r_y] \cup [n_y-r_y+1, n_y]) \times [r_z+1, n_z-r_z]\} \tag{D.3}$$

$$E_z^{(1)} = \{(i,j,k)|(i,j,k) \in [1,n_x] \times [1,n_y] \times ([1,r_z] \cup [n_z-r_z+1, n_z])\} \tag{D.4}$$

Thus $I \cup E_x^{(1)} \cup E_y^{(1)} \cup E_z^{(1)}$ is a representation of the entire grid F. In the second division the edges are divided as follows:

$$E_x^{(2)} = \{(i,j,k) | (i,j,k) \in ([1,r_x] \cup [n_x - r_x + 1, n_x]) \times [1,n_y] \times [1,n_z]\} \quad \text{(D.5)}$$

$$E_y^{(2)} = \{(i,j,k) | (i,j,k) \in [r_x + 1, n_x - r_x] \times ([1,r_y] \cup [n_y - r_y + 1, n_y]) \times [1,n_z]\} \quad \text{(D.6)}$$

$$E_z^{(2)} = \{(i,j,k) | (i,j,k) \in [r_x + 1, n_x - r_x] \times [r_y + 1, n_y - r_y] \times ([1,r_z] \cup [n_z - r_z + 1, n_z])\} \quad \text{(D.7)}$$

Also $I \cup E_x^{(2)} \cup E_y^{(2)} \cup E_z^{(2)}$ is a representation of the entire grid F.

D.1.3 Initialization and Restriction The initial air and tracer masses (and slopes) are defined in all cells of each grid. Next, starting from the finest grid, the data are recursively restricted from a finer grid F to the corresponding part of the next coarser grid C. This is done by adding air and tracer masses in a set of $r_x r_y r_z$ fine cells and attributing these values to the corresponding cell on C. We call this restriction. (For the Slopes scheme new slopes are also restricted, but the details are not relevant here.)

Air-mass fluxes are defined on the walls between the cells on each grid. They are prescribed in a way such that the net flow of air mass through a set of e.g. $r_y \times r_z$ x-walls on a finer grid F during r_t time steps is exactly equal to the flow of air through the corresponding x-wall on a coarser grid C during one time step. This ensures that the air-mass fluxes on F add up to exactly the same value as in the corresponding part of C.

D.1.4 Order of Advection Steps We now describe a full advection step of size Δt on a (coarse) grid C in combination with the corresponding time steps on an embedded (fine) grid F. The splitting scheme on C exists of two triplets $T^{(m)}$ of one-dimensional advection steps: $\hat{X}^{(1)}$-$\hat{Y}^{(1)}$-$\hat{Z}^{(1)}$-$\hat{Z}^{(2)}$-$\hat{Y}^{(2)}$-$\hat{X}^{(2)}$, where $\hat{X}^{(m)}$, $m \in N$ represents an advection step in the x-direction during half a (coarse) time step $\Delta t/2$, etc., and m indicates the triplet number. In this way, after $T^{(1)}$, C is advanced to $t^{n+1/2} = t^n + \Delta t/2$ and after $T^{(2)}$ to $t^{n+1} = t^n + \Delta t$. Correspondingly, r_t time steps of size $\Delta t/r_t$ are performed on F. There are two constraints on the order of the advection steps on F. From the initial time t^n onwards the order of advection steps in the first triplet on F is the same as the order in the first triplet on C with appropriate time refinement, namely $\hat{x}^{(1)}$-$\hat{y}^{(1)}$-$\hat{z}^{(1)}$. Here $\hat{x}^{(m)}$ now denotes an advection step in the x-direction during half a (fine) time step $\Delta t/(2r_t)$, etc. Similarly, the first triplet after $t^{n+1/2}$ is the same as the second triplet on C, thus $\hat{z}^{(r_t+1)}$-$\hat{y}^{(r_t+1)}$-$\hat{x}^{(r_t+1)}$. For the rest the order of advection steps on F is free, provided that the same time advance Δt is made in all three directions. In particular, we have the freedom to choose a Strang or symmetric order of advection steps for the r_t time steps on F. If r_t is odd, exactly the same order as on C may be chosen for each small time step. If r_t is even, the order of directional steps may be alternated from time step to time step.

D.1.5 Zoom Algorithm The zooming process consists of communication between C and F in the following way. We start in C where $\hat{X}^{(1)}$ is applied to $C \cup E_y^{(1)} \cup E_z^{(1)}$ where coarse cells in $E_y^{(1)}$ and $E_z^{(1)}$ are used. The coarse fluxes that are calculated on the boundaries between C and $E_x^{(1)}$ are stored. Next $\hat{Y}^{(1)}$ is applied to $C \cup E_z^{(1)}$ where coarse cells in $E_z^{(1)}$ are used. The coarse fluxes that are calculated on the boundaries between C and $E_y^{(1)}$ are stored. Finally $\hat{Z}^{(1)}$ is applied to C and the coarse fluxes that are calculated on the boundaries between C and $E_z^{(1)}$ are stored. Thus C has been advanced to time level $t^{n+1/2}$.

Now we proceed with F, where $\hat{x}^{(1)}$ is applied to $I \cup E_x^{(1)}$, where again coarse cells are used in $E_x^{(1)}$. At the same time, the stored coarse $\hat{X}^{(1)}$-fluxes are applied at the outer boundaries of $E_x^{(1)}$. Next $\hat{y}^{(1)}$ is applied to $I \cup E_x^{(1)} \cup E_y^{(1)}$ where coarse cells are used in $E_y^{(1)}$ and fine cells in $E_x^{(1)}$, after prolongation of the data from the coarse cells in $E_x^{(1)}$ to the fine cells by distributing air and tracer

mass (and slopes). Finally $\hat{z}^{(1)}$ is applied to $F = I \cup E_x^{(1)} \cup E_y^{(1)} \cup E_z^{(1)}$ where coarse cells are used only in $E_z^{(1)}$ and fine cells in $E_x^{(1)}$ and $E_y^{(1)}$ after prolongation.

After this first fine triplet, the remaining $r_t - 1$ fine triplets until $t^{n+1/2}$ are applied to F with zero fluxes on its outer boundaries. Then the data on F are restricted to the corresponding part of C. (The data on I restricted to C will only be used to provide data in stencils for the calculation of the coarse fluxes on the outer boundary of F and possibly a little further away.)

For the advance from $t^{n+1/2}$ to t^{n+1} the process is very similar. The only difference is the reverse in the order of the advection steps and the use of edge regions $E_d^{(2)}$ rather than $E_d^{(1)}$.

D.2 Conservation of Mass in the Zoom Scheme

We only have to consider mass conservation in the region $C \cup E_x^{(m)} \cup E_y^{(m)} \cup E_z^{(m)} \cup I$ for $m = 1, 2$, because this is the nonuniform mathematical grid. First of all we note that the only exchange of air and tracer mass between cells takes place by fluxes from one cell to another. Because the fluxes merely move mass from one cell to another, mass is conserved in this process.

The only other aspect we need to consider is the restriction and prolongation between data in coarse and fine representations of the edge regions. Since for a coarse cell e with fine subcells $e; I, J, K$ in an edge region we always guarantee:

$$\sum_{(I,J,K)=(1,1,1)}^{(r_x,r_y,r_z)} \mu_{e;I,J,K} = \mu_e \quad , \tag{D.8}$$

no mass is created or destroyed in this process and mass is conserved.

D.3 Conservation of Positivity in the Zoom Scheme

A particular aim in designing the zoom scheme was conservation of positivity for the tracer mass in all cells, which is a desirable property in air-pollution modelling. This is not straightforward. For example, a simple way for applying a (coarse) flux F_c in direction d at the outer edges of a fine region F would be to apply F_c/r_t every time a fine advection step in this direction is made in F. With this approach, however, positivity is no longer guaranteed.

Positivity is preserved for the restriction and prolongation in the zooming process by taking $\mu_{e;I,J,K} \geq 0$ for $(I, J, K) \in [1, r_x] \times [1, r_y] \times [1, r_z]$ for a coarse cell e in the edge regions.

To prove positivity for the advection process we will prove that each of the advection steps is always positive (results in a positive state $q+1$ of tracer mass values when the previous state q contains only positive values) for a certain cell i. It then follows that any sequence of one-dimensional advections steps gives a positive result.

First we define an advection step in Subsubsection D.3.1, which consists of the calculation of fluxes (described in Subsubsection D.3.2) and an update of the (tracer) mass (described in Subsubsection D.3.3). Next we introduce two flux schemes, namely Slopes and Split, in Subsubsection D.3.4. Then we discuss the handling of resolution changes in Subsubsection D.3.5. In Subsubsection D.3.6 we study positivity from the expression for the updated mass. The main conclusions are summarized in Subsubsection D.3.7.

D.3.1 Definition of an Advection Step An advection step has the following properties:

1. An advection step handles one direction at a time;
2. A flux through a wall is calculated from the air-mass flux through it and the values of air and tracer masses in a one-dimensional stencil of cells in its neighbourhood;
3. The resolution of the stencil for the calculation of the flux through a wall is equal to the resolution of the smallest of the two cells on either side of the wall;

4. A flux through a wall is always applied to the masses in the two neighbouring cells at the same time;

For each cell the advection process consists of a series of one-dimensional advection steps in arbitrary order.

We will study positivity for a one-dimensional advection step for a cell. Consider a cell i with walls $i-1/2$ and $i+1/2$ and (immediate) neighbours $i-1$ and $i+1$. An advection step consists of calculating fluxes through the two (opposite) cell walls $i-1/2$ and $i+1/2$ from neighbouring data when these data are in a state p and updating the mass in the enclosed cell i from a state $q \geq p$ with this flux to a state $q+1$. Although the data in other cells may have changed between states p and q, the data in the cells i and its two neighbours may not have changed between the calculation of the fluxes and their application. Only if the air-mass flux to one side of the cell equals zero, the data in the neighbouring cell to that side may have changed between states p and q. These conditions on the states of the data will be shown to be important for positivity.

D.3.2 Flux Calculation The (time-integrated) flux F_-^q through a wall $i-1/2$ between levels q and $q+1$ is calculated when the tracer masses on the (composite) grid are in a certain state $p \leq q$. It is applied to state q. In the case of a Direct Space Time (DST) scheme it can be expressed as follows:

$$F_-^q = F(A_-^q, \boldsymbol{D}_{i-1}^p, \boldsymbol{D}_i^p, \boldsymbol{E}_-^p) \ , \tag{D.9}$$

Here the integrated air-mass flux $A_- = \int_{t_-^q}^{t_-^{q+1}} a_- dt$, where a_- is the air-mass flow rate through wall $i-1/2$, and t_-^q and t_-^{q+1} are times at this wall corresponding to states q and $q+1$ respectively. They may be different for different walls. $\boldsymbol{D}_{i-1}$ and $\boldsymbol{D}_i$ represent the air mass, tracer mass, and slope (only in the slopes scheme) in the two neighbouring cells. In general $\boldsymbol{D} = (m, \mu(, \sigma))^T$. $\boldsymbol{E}_-^p$ represents the data in the remaining cells in the stencil. The lengths of these vectors depend on the advection scheme.

D.3.3 Mass Update In the case of a Direct Space Time (DST) scheme the updates of the air mass m_i and the tracer mass μ_i in cell i from state q to state $q+1$ read:

$$m_i^{q+1} = m_i^q + A_-^q - A_+^q \ , \tag{D.10}$$

$$\mu_i^{q+1} = \mu_i^q + F_-^q - F_+^q \ . \tag{D.11}$$

Insight may be gained by considering that air-mass fluxes A_w^q through a wall w may be zero because $t_w^{q+1} = t_w^q$. In the zoom algorithm we take the freedom to put e.g. $t_-^{q+1} > t_-^q$ on the left wall and $t_+^{q+1} = t_+^q$ on the opposite wall for certain cells. We call this lateral splitting. This can be considered as a further generalization of the concept of directional splitting, where $t_-^q = t_+^q$ and $t_\pm^{q+1} = t_\pm^q + \tau$, $\tau > 0$ is taken on both walls in one direction and $t_w^{q+1} = t_w^q$ on any remaining walls w.

D.3.4 Two Particular Flux Schemes: Slopes and Split We limit our positivity analysis to flux functions for the Slopes and the Split scheme.

Slopes Scheme For Slopes we have:

$$F(A_-, \boldsymbol{D}_{i-1}, \boldsymbol{D}_i) = m_{i-\beta}\nu_-[s_-\chi_{i-\beta} + (1-\nu_-)\phi(\chi_{i-\beta}, \sigma_{i-\beta})] \ , \tag{D.12}$$

see[12], where $s_- \equiv \mathrm{sgn}(A_-)$, $\beta \equiv (1+s_-)/2$, $\nu_- \equiv |A_-|/m_{i-\beta}$, and the mixing ratio $\chi_i \equiv \mu_{i-\beta}/m_{i-\beta}$. The limiter function $\phi(\chi, \sigma)$ reads:

$$\phi(\chi, \sigma) = \max(\min(\chi, \sigma), -\chi) \ . \tag{D.13}$$

This limiter ensures a natural condition on the flux, namely that F has the same sign as A_- for arbitrary values of ν, χ, and σ. Because of this limiter, the actual value of σ_i (as well as of the slopes in the other two directions) turns out to be irrelevant for the positivity study. Therefore we do not have to consider their updates.

Split Scheme For Split we have:

$$F(A_-, \boldsymbol{D}_{i-2}, \boldsymbol{D}_{i-1}, \boldsymbol{D}_i, \boldsymbol{D}_{i+1}) = m_{i-\beta}\nu_-[s_- X_{i-\beta} + \psi(\nu_-, \Theta_{i-\beta}^{s_-})(X_i - X_{i-1})] \ , \tag{D.14}$$

see [7], where $X_{i-\beta} \equiv \mu_{i-\beta}/m_{i-\beta}$ are the tracer mixing ratios and $\Theta_{i-\beta} \equiv (X_{i-\beta} - X_{i-\beta-1})/(X_{i-\beta+1} - X_{i-\beta})$ is a measure of the trend change. (We will use upper-case symbols for mixing ratios and trend changes pertaining to data in the stencil S_- for the left flux, and lower-case symbols for those in the stencil S_+ for the right flux.)

The limiter function $\psi(\nu, \theta)$, see [8], with value between 0 and 1, ensures (among other things) that F and A_- carry the same sign for arbitrary values of ν, X_{i-2}, X_{i-1}, X_i, X_{i+1}, and X_{i+2}. It reads:

$$\psi(\nu, \theta) = \max\{0, \min[(1-\nu)\theta/\nu, (1/6)(1-\nu)((2-\nu) + (1+\nu)\theta), 1]\} \in [0, 1] \ . \tag{D.15}$$

In the positivity study below we want to have positivity for a given arbitrary value of ν and for all possible values for data in the stencil. Three trivial results which will prove useful are:

$$\forall \nu \in [0,1] \quad \exists \theta \quad | \ \psi(\nu, \theta) \leq \max\{0, (1-\nu)\theta/\nu\} \ , \tag{D.16}$$

$$\forall \nu \in [0,1] \quad \exists \theta \quad | \ \psi(\nu, \theta) \geq \max\{0, (1-\nu)\theta/\nu\} \ , \tag{D.17}$$

$$\nu\psi(\nu, \theta)\delta = \varsigma \max\{0, \min[\varsigma(1-\nu)\theta\delta, (1/6)\varsigma\nu(1-\nu)((2-\nu)\delta + (1+\nu)\theta\delta), \varsigma\nu\delta]\} \ , \tag{D.18}$$

where $\varsigma = \mathrm{sgn}(\delta)$ and $\delta \in \mathbb{R}$.

D.3.5 Resolution Changes We study resolution changes within the stencils for the fluxes through both walls of a cell in a particular direction. The calculation of a flux through a wall is always based on a stencil with cells with the size of the smallest neighbouring cell(s), or with the smallest transverse size for Slopes. For the positivity analysis for a cell i it will turn out that we only have to consider resolution changes within the triplet of consecutive cells $i-1$, i, and $i+1$. We limit ourselves to resolution changes where refinement in one or more directions is not accompanied by simultaneous coarsening in another direction. Thus in case of refinement in at least one direction we have $r_d \geq 1$ for all d. Furthermore we assume that, in all directions, at least two consecutive cells with the same size (resolution) occur before the resolution changes again. We distinguish two cases which represent all allowed cases. In the first case the cell under consideration, cell i, is fine and has a coarse neighbour $i-1$ to its left and a fine neighbour $i+1$ of its own size to its right. In the second case cell i is coarse and has a coarse neighbour $i-1$ of its own size to its right and a block $(i+1)$ of $r_x \times r_y \times r_z$ fine cells numbered as $i+1; I, J, K$ to its right, where $(I, J, K) \in \{1, \ldots, r_x\} \times \{1, \ldots, r_y\} \times \{1, \ldots, r_z\}$. The fine cells are refined with factors r_x, r_y, and r_z with respect to the coarse cell(s) in the respective directions. The case of no refinement results as a special case, namely when $r_x = r_y = r_z = 1$.

Fine cell i. For positivity in cell i it does not matter how the left air-mass flux A_- and data in pseudo cells to its left — which are needed for S_- and possibly for S_+ too — are chosen, as long as they satisfy general conditions which should hold for all cells. The first is that $m \geq 0$ and $\mu \geq 0$ in all cells. The second is that $|A_-|$ is not larger than the air mass in cell i or that in the pseudo cell to its left. The situation is then equivalent to that of a uniform-grid cell, which will be studied in Subsubsection D.3.6.

Coarse cell i. In this case we divide wall $i+1/2$ into $r_y \times r_z$ subwalls with subfluxes $F_{+;J,K}$, $(J,K) \in \{1,\ldots,r_y\} \times \{1,\ldots,r_z\}$. For positivity of the tracer mass in cell i we need:

$$\mu_i^{q+1} = \mu_i^q + F_-^q - \sum_{(J,K)=(1,1)}^{(r_y,r_z)} F_{+;J,K} \geq 0 \ , \tag{D.19}$$

which should hold for all possible values of the air and tracer masses and fluxes (and of the slopes). For the study of positivity in cell i we consider it to be subdivided transversely into $r_y \times r_z$ subcells and we will study positivity for each of these subcells. The cells to its left, as far as necessary to provide data in stencils S_- and S_+ for the subfluxes $F_{-;J,K}$ and $F_{+;J,K}$, are similarly subdivided. If the tracer mass $\mu_{i;J,K}$ remain positive, then positivity is certainly guaranteed for cell i if we take $\mu_i \equiv \sum_{(I,J)=(1,1)}^{(1,1)} \mu_{i;J,K}$. We choose to equidistribute air mass and air-mass fluxes, thus $m_{\ell;J,K} = m_\ell/(r_y r_z)$ for all necessary ℓ and $A_{\pm;J,K} = A_\pm/(r_y r_z)$. This means that $s_{\pm;J,K} = s_\pm$ and $\nu_{\pm;J,K} = \nu_\pm$. From here we treat Slopes and Split separately.

Slopes For Slopes the subdivision of cells $i-1$ and i should be such that $\sum_{(I,J)=(1,1)}^{(r_y,r_z)} F_{-;J,K} = F_-$. Using equation (D.12) and equidistribution of m_ℓ and $A_\pm$ we rewrite this requirement as:

$$\begin{aligned}
&\sum_{(I,J)=(1,1)}^{(r_y,r_z)} \frac{m_{i-\beta}}{r_y r_z} \nu_- [s_- \chi_{i-\beta;J,K} + (1-\nu_-)\sigma_{i-\beta;J,K}] = \\
&\qquad m_{i-\beta}\nu_- [s_- \chi_{i-\beta} + (1-\nu_-)\sigma_{i-\beta}] \Leftrightarrow \\
&\nu_- \left[s_- \sum_{(I,J)=(1,1)}^{(r_y,r_z)} \chi_{i-\beta;J,K} + (1-\nu_-) \sum_{(I,J)=(1,1)}^{(r_y,r_z)} \sigma_{i-\beta;J,K} \right] = \\
&\quad r_y r_z \nu_- [s_- \chi_{i-\beta} + (1-\nu_-)\sigma_{i-\beta}] \ .
\end{aligned} \tag{D.20}$$

Here we have assumed that $\sigma_{i-\beta}$ and $\sigma_{i-\beta;J,K}$ are the limited values of the slopes. This condition is satisfied for all values of s_- and ν_- if and only if for $\ell = i-1, i$ the following two conditions are satisfied:

$$\sum_{(I,J)=(1,1)}^{(r_y,r_z)} \chi_{\ell;J,K} = r_y r_z \chi_\ell \ , \tag{D.21}$$

$$\sum_{(I,J)=(1,1)}^{(r_y,r_z)} \sigma_{\ell;J,K} = r_y r_z \sigma_\ell \ , \tag{D.22}$$

Such a distribution can always be found. One choice which satisfies both conditions is $\chi_{\ell;J,K} = \chi_\ell$ and $\sigma_{\ell;J,K} = \sigma_\ell$.

Split For Split we need data in supercell $i+1; J, K$ to calculate the coarse flux $F_{-;J,K}$. We leave this data unspecified for now, but we will find conditions for this data in the positivity analysis below. The subdivision of cells $i-2$, $i-1$ and i should be such that $\sum_{(I,J)=(1,1)}^{(r_y,r_z)} F_{-;J,K} = F_-$. Using

equidistribution of m_ℓ and $A_\pm$ we rewrite this as:

$$\sum_{(I,J)=(1,1)}^{(r_y,r_z)} \frac{m_{i-\beta}}{r_y r_z}\nu_-[s_- X_{i-\beta;J,K} + \psi(\nu_-,\Theta^{s_-}_{i-\beta;J,K})(X_{i;J,K} - X_{i-1;J,K})] =$$
$$m_{i-\beta}\nu_-[s_- X_{i-\beta} + \psi(\nu_-,\Theta^{s_-}_{i-\beta})(X_i - X_{i-1})] \Leftrightarrow$$
$$\nu_-\left[s_- \sum_{(I,J)=(1,1)}^{(r_y,r_z)} X_{i-\beta} + \sum_{(I,J)=(1,1)}^{(r_y,r_z)} \psi(\nu_-,\Theta^{s_-}_{i-\beta;J,K})(X_{i;J,K} - X_{i-1;J,K})\right] =$$
$$r_y r_z[s_- X_{i-\beta} + \psi(\nu_-,\Theta^{s_-}_{i-\beta})(X_i - X_{i-1})] \ , \tag{D.23}$$

where we have again used the fact that the air-mass fluxes are equidistributed over the subwalls. This condition is satisfied for all values of s_- and ν_- if and only if for $\ell = i-1, i$ the following two conditions are satisfied:

$$\sum_{(I,J)=(1,1)}^{(r_y,r_z)} X_{\ell;J,K} = r_y r_z X_\ell \ , \tag{D.24}$$

$$\sum_{(I,J)=(1,1)}^{(r_y,r_z)} \psi(\nu_-,\Theta^{s_-}_{i-\beta;J,K})(X_{i;J,K} - X_{i-1;J,K}) = r_y r_z \psi(\nu_-,\Theta^{s_-}_{i-\beta})(X_i - X_{i-1}) \ , \tag{D.25}$$

where we have again used equidistribution of air mass over the subcells. Such a distribution can always be found. A choice which satisfies both conditions is $X_{\ell;J,K} = X_\ell$, $\ell = i-2, i-1, i, i+1$. Note that this constrains the values of $X_{i+1;J,K}$ to be all equal. (Remember that $\Theta_{i-\beta}$ may depend on each of these X_ℓ.)

From the above we conclude that it is possible to find values for the data in pseudocells (for stencil values) such that the sum of subfluxes through the left wall is equal to the coarse flux which is actually used in the algorithm. From here the positivity analysis for cell i can be replaced by one for each of its subcells: if positivity holds for each of them, then it certainly holds for cell i, since $\mu_i = \sum_{(I,J)=(1,1)}^{(r_y,r_z)} \mu_{i;J,K}$.

We are left with the task of studying the case of collinear refinement to the right of cell i with a factor $r_x \geq 1$. This is done below simultaneously with the case where cells $i-1$, i, and $i+1$ have the same (collinear) resolution.

D.3.6 Positivity Analysis We wish to know the constraints on the (time integrated) air-mass fluxes $A^q_\pm$ (or time-step sizes $t^{q+1}_\pm - t^q_\pm$) and stencil values in $S_\pm$ such that positivity of μ^{q+1}_i in equation (D.11) is always guaranteed for arbitrary tracer-mass distributions. We do not only consider non-divergent flows, but also divergent ones. They are relevant in a physical sense in the application of air-pollution modelling, as well as in a numerical sense because the directional splitting and even lateral splitting occur in the present approach. This causes changes in the air mass in a cell within an advection step.

For an arbitrary (sub)cell i with at least one neighbour of its own size in direction d we will study the positivity of μ^{q+1}_i as the result of an advection step in direction d applied to μ^q_i.

First we define some quantities which we will use in the analysis. We define $N_\pm \equiv |A^q_\pm|/m_i \geq 0$ with respect to the mass in the concerning cell i, not as usually with respect to the upwind cell. $N_\pm$ are equal to the conventional CFL-number ν if cell i is the upwind cell. We assume $N_- \leq 1$ if $A^q_- < 0$, and $N_+ \leq 1$ if $A^q_+ \geq 0$, and $N_- + N_+ \leq 1$ if $A^q_- < 0$ and $A^q_+ \geq 0$ at the same time (implying that no negative air mass should result in going from state q to state $q+1$) for reasons of stability. Furthermore, $s_\pm = \mathrm{sgn}(A_\pm)$. In the equations below we will omit the superscript q whenever possible without causing confusion.

Slopes We begin by noting that the Slopes scheme is not based on cell sizes, but on mass coordinates, see [12]. This means that we need not distinguish between the cases with and without refinement to the right of cell i. The following analysis therefore applies to all cases $r_x \geq 1$. We distinguish four cases for the signs (s_-, s_+).

The case $s_- = 1$ and $s_+ = -1$ Since the sign of a tracer-mass flux is always equal to that of the corresponding air-mass flux, the case $s_- = 1$ and $s_+ = -1$ always results in a positive value of μ_i^{q+1} if $\mu_i^q \geq 0$.

The case $s_- = -1 = s_+$. The worst case is where $A_+ = 0$. We prove positivity for this case. After some elementary algebra the new tracer mass is expressed as:

$$\begin{aligned} \mu_i^{q+1} &= m_i[(1-N_-)\chi_i + N_-(1-N_-)\phi(\chi_i,\sigma_i)] \\ &= m_i(1-N_-)(\chi_i + N_-\phi(\chi_i,\sigma_i)) \ . \end{aligned} \tag{D.26}$$

This is always positive since each of the terms on the right side is positive. Positivity is then certainly guaranteed if $A_+ < 0$.

The case $s_- = 1 = s_+$. Handling this case in a similar way, we find that μ_i^{q+1} always positive.

The case $s_- = -1$ and $s_+ = 1$. This is the most severe case. We have for the new tracer mass:

$$\begin{aligned} \mu_i^{q+1} &= m_i[(1-N_--N_+)\chi_i + N_-(1-N_-)\phi(\chi_i,\sigma_i) - N_+(1-N_+)\phi(\chi,\sigma_i)] \\ &= m_i(1-N_--N_+)[\chi_i + (N_--N_+)\phi(\chi,\sigma_i)] \ , \end{aligned} \tag{D.27}$$

which is certainly positive if:

$$(N_- - N_+)\phi(\chi_i,\sigma_i) \geq -\chi_i \ . \tag{D.28}$$

But this is true since $\phi(\chi,\sigma_i) \geq -\chi_i$ and since $-1 \leq N_- - N_+ \leq 1$.
In conclusion, under the assumed conditions, the Slopes scheme is positive for all possible values of m_ℓ, μ_ℓ, and σ_ℓ, $\ell = i-1, i, i+1$.

Split Instead of a priori relating the data in the coarse left stencil S_- and those in the fine right stencil S_+ with each other and with the data on the grid, we will keep most of them formally unrelated for the time being. The positivity analysis will result in conditions which take the form of such relations. The conditions will show the constraints as well as the freedoms one has to choose the stencil data. The only a priori relations are that we set X_i in cell i in S_- and χ_j in cell j (see below) equal to the actual value of the mixing ratio in cell i which we will henceforth indicate with X_i. Furthermore we require that $m_j \leq m_i$. The analysis uses several properties the Split scheme explicitly, so the results hold only for schemes which share these properties.

For the left stencil we take values in stencil cells $i-2$ and $i-1$ left from wall $i-1/2$ and stencil cells i and $i+1$ right from that wall. The left flux reads:

$$F_-^q = F(A_-, \boldsymbol{D}_{i-2}, \boldsymbol{D}_{i-1}, \boldsymbol{D}_i, \boldsymbol{D}_{i+1}) = m_{i-\beta}\nu_-[s_-X_{i-\beta} + \psi(\nu_-, \Theta_{i-\beta}^{s_-})(X_i - X_{i-1})] \ . \tag{D.29}$$

For the right stencil we take values in stencil cells $j-1$ and j left from wall $i+1/2$ and stencil cells $j+1$ and $j+2$ right from that wall. The right flux reads:

$$F_+^q = F(A_+, \boldsymbol{d}_{j-1}, \boldsymbol{d}_j, \boldsymbol{d}_{j+1}, \boldsymbol{d}_{j+2}) = m_{j+\gamma}\nu_+[s_+\chi_{j+\gamma} + \psi(\nu_+, \theta_{j+\gamma}^{s_+})(\chi_{j+1} - \chi_j)] \ . \tag{D.30}$$

Here $\boldsymbol{d}_\ell$ represents data in cell ℓ, $\gamma \equiv (1-s_+)/2$, $\nu_+ \equiv |A_+|/m_{j+\gamma}$, and the tracer mixing ratios $\chi_{j+\gamma} \equiv \mu_{j+\gamma}/m_{j+\gamma}$.

To facilitate the analysis, we introduce several new quantities:

$$\begin{aligned} U &\equiv \mathrm{sgn}(X_i - X_{i-1}) \ , && \text{(D.31)}\\ G &\equiv |X_i - X_{i-1}| = U(X_i - X_{i-1}) \ , && \text{(D.32)}\\ V &\equiv \mathrm{sgn}(X_{i+1} - X_i) \ , && \text{(D.33)}\\ H &\equiv |X_{i+1} - X_i| = V(X_{i+1} - X_i) \ , && \text{(D.34)}\\ u &\equiv \mathrm{sgn}(\chi_{j+1} - \chi_j) \ , && \text{(D.35)}\\ g &\equiv |\chi_{j+1} - \chi_j| = u(\chi_{j+1} - \chi_j) \ , && \text{(D.36)}\\ v &\equiv \mathrm{sgn}(\chi_j - \chi_{j-1}) \ , && \text{(D.37)}\\ h &\equiv |\chi_j - \chi_{j-1}| = v(\chi_j - \chi_{j-1}) \ , && \text{(D.38)} \end{aligned}$$

where upper-case symbols refer to quantities in S_- and lower-case symbols to quantities S_+. We wish to point out that G, H, g, and h are independent of each other if $r_x > 1$.

A major simplification results from using statements (D.16) and (D.17) and considering that positivity is required for all allowed values of N_- and N_+ and for all (positive) values of the mixing ratios in the stencils S_- and S_+, and that with arbitrary values for the mixing ratios in the stencil all values in $\mathbb{R}$ can be obtained for $\Theta_{i-\beta}$ and for $\theta_{j+\gamma}$. This means that we only have to consider the first argument of the min-function in the definition of the limiter function ψ, and we define a modified limiter and a modified updated or new tracer-mass:

$$\begin{aligned} \tilde{\psi}(\nu,\theta) &\equiv \max\{0,(1-\nu)\theta/\nu\} > 0 \ , && \text{(D.39)}\\ \tilde{\mu}_i^{q+1} &\equiv m_i[(1-N_-)X_i - N_+\chi_j + UN_-\tilde{\psi}(\nu_-,\Theta_i^{-1})G - uN_+\tilde{\psi}(\nu_+,\theta_i)g] \ . && \text{(D.40)} \end{aligned}$$

If we can find conditions for $\tilde{\mu}_i^{q+1}$ to be positive, then certainly $\mu_i^{q+1} \geq 0$ under these conditions. We distinguish for cases for the signs (s_-, s_+).

The case $s_- = 1$ and $s_+ = -1$. Since $\mathrm{sgn}(F_\pm) = s_\pm$, this case always results in $\mu_i^{q+1} \geq 0$ if $\mu_i^q \geq 0$.

The case $s_- = -1 = s_+$. The worst case is where $F_+ = A_+ = 0$. We prove positivity for this case. After some elementary algebra the modified new tracer mass is expressed as:

$$\tilde{\mu}_i^{q+1} = m_i[(1-N_-)X_i + UN_-\tilde{\psi}(N_-,\Theta_i^{-1})G] \ . \qquad \text{(D.41)}$$

We distinguish the following cases.

- If $U = 1$, this is certainly positive.
- If $U = -1$ we have:

$$\tilde{\mu}_i^{q+1} = m_i[(1-N_-)X_i - \max\{0,(1-N_-)(-V)H\}] \ . \qquad \text{(D.42)}$$

  - If $V = 1$ then this expression is positive.
  - If $V = -1$ we find after a little algebra:

$$\tilde{\mu}_i^{q+1} \geq m_i(1-N_-)X_{i+1} \ , \qquad \text{(D.43)}$$

    which is positive.

Since we have studied all possible cases for U and V, this completes the positivity proof for $s_- = -1 = s_+$.

The case $s_- = 1 = s_+$. We consider only the worst case $F_- = A_- = 0$. The analysis is very similar to the one above except that for the first argument of the limiter we need to use the value of the conventional CFL-number $\nu_+ = \lambda_j^{-1} N_+$ instead of N_+. Using $\chi_j = X_i$ we find for the modified new tracer mass:

$$\tilde{\mu}_i^{q+1} = m_i[(1 - N_+)X_i - uN_+\tilde{\psi}(\lambda_j^{-1}N_+, \theta_j)g] \ . \tag{D.44}$$

We distinguish the following cases.

- If $u = -1$ this is certainly positive.
- If $u = 1$ we have:

$$\tilde{\mu}_i^{q+1} \geq m_i[(1 - N_+)X_i - \max\{0, (\lambda_j - N_+)vh\}] \ . \tag{D.45}$$

  - If $v = -1$ and $N_+ \leq \lambda_j$ (which is a natural condition which we will assume henceforth) this is positive.
  - If $v = 1$ we have:

$$\tilde{\mu}_i^{q+1} \geq m_i[(1 - \lambda_j)X_i + (\lambda_j - N_+)\chi_{j-1}] \ , \tag{D.46}$$

    which is positive.

Since we have studied all possible cases for u and v, this completes the positivity proof for $s_- = 1 = s_+$.

The case $s_- = -1$ and $s_+ = 1$. This is the most severe case. We find for the modified new tracer mass:

$$\tilde{\mu}_i^{q+1} = m_i[(1 - N_-)X_i - N_+\chi_j + UN_-\tilde{\psi}(N_-, \Theta_i^{-1})G - uN_+\tilde{\psi}(\lambda_j^{-1}N_+, \theta_i)g] \ . \tag{D.47}$$

We distinguish the following cases.

- If $U = 1$ and $u = -1$ or if $UV = -1 = uv$, then using $\chi_j = X_i$ we find:

$$\tilde{\mu}_i^{q+1} \geq m_i(1 - N_- - N_+)X_i \ , \tag{D.48}$$

  which is positive.

- If $U = -1 = u$ we have:

$$\tilde{\mu}_i^{q+1} \geq m_i[(1 - N_- - N_+)X_i - \max\{0, (1 - N_-)(-V)H\}] \ . \tag{D.49}$$

  - If $V = 1$ this is positive.
  - If $V = -1$ then we have:

$$\tilde{\mu}_i^{q+1} \geq m_i[(1 - N_-)X_{i+1} - N_+X_i] \ . \tag{D.50}$$

    This is positive if:

$$X_{i+1} \geq \frac{N_+}{1 - N_-}X_i \ . \tag{D.51}$$

    Since the expression on the right is larger than or equal to X_i and can indeed reach this value, this implies $X_{i+1} \leq X_i$ if this inequality is to hold for all values of $N_\pm$. But since we have assumed $V = -1$ we need:

$$X_{i+1} = X_i \ , \tag{D.52}$$

    for positivity for arbitrary values of $N_\pm$.

- If $U = 1 = u$ we have:

$$\tilde{\mu}_i^{q+1} \geq m_i[(1 - N_- - N_+)X_i - \max\{0, (\lambda_j - N_+)vh\}] \ . \tag{D.53}$$

  – If $v = -1$ this is positive.

  – If $v = 1$ we have:

$$\tilde{\mu}_i^{q+1} \geq m_i[(1 - N_- - \lambda_j)X_i + (\lambda_j - N_+)\chi_{j-1}] \ , \tag{D.54}$$

  which is positive.

- If $U = -1$ and $u = 1$ we have:

$$\tilde{\mu}_i^{q+1} \geq m_i[(1 - N_- - N_+)X_i - \max\{0, (1 - N_-)(-V)H\} - \max\{0, (\lambda_j - N_+)vh\}] \ . \tag{D.55}$$

  – If $V = 1$ and $v = -1$ we find one special case of the cases satisfying with $UV = -1 = uv$ under equation (D.47) and positivity was shown to be guaranteed.

  – If $V = -1 = v$ we have:

$$\tilde{\mu}_i^{q+1} \geq m_i[(1 - N_-)X_{i+1} - N_+X_i] \ . \tag{D.56}$$

  This was studied in equation (D.50) and we have positivity if $X_{i+1} = X_i$.

  – If $V = 1 = v$ we find:

$$\tilde{\mu}_i^{q+1} \geq m_i[(1 - N_-\lambda_j)X_i + (\lambda_j - N_+)\chi_{j-1}] \ , \tag{D.57}$$

  which is positive.

  – If $V = -1$ and $v = 1$ we have:

$$\tilde{\mu}_i^{q+1} \geq m_i[(1 - N_- - N_+)X_i - (\lambda_j - N_+)(X_i - \chi_{j-1})] \ , \tag{D.58}$$

  where we have used $X_{i+1} = X_i$. This is positive if:

$$X_i - \chi_{j-1} \leq \frac{1 - N_- - N_+}{\lambda_j - N_+} X_i \ . \tag{D.59}$$

  Since the expression on the right is less than or equal to 0 and can indeed attain this value, this implies $X_i - \chi_{j-1} \leq 0$ if this inequality is to hold for all values of $N_\pm$. But since we have assumed $v = 1$ we need:

$$\chi_{j-1} = X_i \ , \tag{D.60}$$

  for positivity.

In conclusion, under the assumed conditions, the Split scheme is positive for all possible values of m_ℓ, μ_ℓ, and σ_ℓ, $\ell = i-2, i-1, i, i+1, j-1, j, j+1, j+2$ if $N_+ \leq \lambda_j$ (if $A_+^q > 0$), and $X_{i+1} = X_i = \chi_j = \chi_{j-1}$ is taken. The latter extra conditions come down to using the first-order upwind scheme on either side of cell i.

However, in the uniform-grid case $r_x = 1$ we take $\lambda_j = 1$, $\chi_{j-1} = X_{i-1}$, $\chi_j = X_i$, and $\chi_{j+1} = X_{i+1}$, so we have $v = U$ and $u = V$. Then the estimates in equations (D.50) and (D.57) can be relaxed, so that $X_{i+1} = X_i$ and $\chi_{j-1} = \chi_j$ are no longer required. We can use the Split scheme for cell i without loosing positivity.

Under the above conditions, $\mu_i^{q+1} = 0$ may occur for certain allowed values of air and tracer masses and air-mass fluxes. This means that for positivity in cell i the tracer mass μ_i^q should not be less than m_iX_i, which is used in the calculation of F_-^q and F_+^q. But since the fluxes $F_\pm^q$ — which are calculated from X_{i-1} and χ_{j+1} among other values — are also applied to cells $i-1$ and $i+1$, we should have $\mu_{i-1}^q \geq m_{i-1}X_{i-1}$ and $\mu_{i+1}^q \geq m_{j+1}\chi_{j+1}$. Otherwise, if $F_-^q = m_{i-1}X_{i-1}$ or $F_+^q = -m_{j+1}\chi_{j+1}$ (which may happen), the tracer masses in these cells would become negative.

D.3.7 Conclusion For accuracy we identify the air and tracer masses in the two coarse cells left from wall $i-1/2$ in S_- with the actual air and tracer masses m^p_{i-2}, m^q_{i-1} and μ^p_{i-2}, μ^q_{i-1} in the cells $i-2$ and $i-1$. Similarly if $r_x > 1$ we identify the air and tracer mass values in the two fine cells right from wall $i+1/2$ in S_+ with the actual air and tracer masses m^q_{j+1}, m^p_{j+2} and μ^q_{j+1}, μ^p_{j+2} in the cells $i+1;1$ and $i+1;2$. Furthermore we take $m_{i+1} = m_i$, $\mu_{i+1} = \mu_i$ and $m_{j-1} = m_j = m_i/r_x$ and $\mu_{j-1} = \mu_j = \mu_i/r_x$. If $r_x = 1$ we identify the air and tracer mass values in the two fine cells right from wall $i+1/2$ in S_+ with the actual values m^q_{i+1}, m^p_{i+2} and μ^q_{i+1}, μ^p_{i+2} in cells $i+1$ and $i+2$. Also we take then $m_{j-1} = m_{i-1} = m_j = m_i$ and $\mu_{j-1} = \mu_{i-1} = \mu_j = \mu_i$.

Note that no conditions on the values of X_{i-2} and χ_{j+2} are necessary for positivity, so they may have changed between state p used for the calculation and state $q \geq p$ where the fluxes $F^q_\pm$ are applied.

We repeat that the only further conditions we have to impose for positivity are $N_- \leq 1$ if $A^q_- < 0$, and $N_+ \leq 1$ if $A^q_+ \geq 0$, and $N_- + N_+ \leq 1$ if $A^q_- < 0$ and $A^q_+ \geq 0$ at the same time, and $N_+ \leq \lambda_j = 1/r_x$ if $A^q_+ > 0$. These are no extra conditions since they are already necessary for stability. In words: if no unilateral air-mass flux nor the sum of two bilateral fluxes takes more air mass out of a cell during an advection step, positivity is guaranteed for the tracer mass under advection according to the Slopes and the Split schemes.

D.4 Conservation of Constant Mixing Ratio in the Zoom Scheme

A physical property of advection is that if initially the mixing ratio is constant everywhere in a gas, then advection alone cannot lead to spatial differences in the mixing ratio, nor can it lead to changes with time. The reason is that mixing N arbitrary amounts of air m_i (with arbitrary densities) with the same tracer mixing ratio χ yields a new amount of air with the same mixing ratio as in each the original amounts. If μ_i are the tracer masses in each amount of air, we have for the new mixing ratio X:

$$X \equiv \frac{\sum_{i=1}^N \mu_i}{\sum_{i=1}^N m_i} = \frac{\sum_{i=1}^N \chi m_i}{\sum_{i=1}^N m_i} = \chi \ , \tag{D.61}$$

which proves our statement. The same reasoning holds for subtraction of amounts of air from another larger amount. In the remainder of this subsection we will assume that the mixing ratio is constant, i.e. the same in each cell.

As we have seen in Subsection D.2, the zoom scheme merely restricts and prolongates data in the edge cells and reorders advection steps for these cells. Restriction does not change mixing ratios if the mixing ratios in the $r_x \times r_y \times r_z$ fine cells that make up a coarse cell all have the same mixing ratio. Prolongation does not change mixing ratios if each fine cell gets the same portion of the air mass as of the tracer mass in the coarse cell.

We only have to prove now that the advection schemes do not change a constant mixing-ratio field. Since an advection step consists of nothing more than subtraction and addition of amounts of air with certain mixing ratios, we have to prove that each flux through a wall moves an amount of air with the same mixing ratio as in the two neighbouring cells.

D.4.1 The Slopes Scheme From the Slopes expression (D.12) for the flux through a wall $i-1/2$ we see that, if $(1-\nu_-)\phi(\chi_{i-\beta}, \sigma_{i-\beta}) = 0$, the mixing ratio of the amount of air moved by the air-mass flux is equal to that of the upwind cell. In that case the mixing ratio is not changed in either of the cells neighbouring the wall. The condition is satisfied by $\sigma_{i-\beta} = 0$ which is natural in a constant mixing-ratio field.

D.4.2 The Split Scheme From the Split expression (D.14) for the flux through a wall $i-1/2$ we see that, if $X_i - X_{i-1} = 0$ (which is assumed), the mixing ratio of the amount of air moved by the air-mass flux is equal to that of the upwind cell. In that case the mixing ratio is not changed in either of the cells neighbouring the wall.

We conclude that the application of arbitrary numbers of restriction, prolongation, and Slopes and Split advection steps according to the schemes in mixing-ratio formulation, as given in equations (D.12) and (D.14), conserves a constant mixing-ratio field.