A Zooming Technique for Wind Transport of Air Pollution[*]

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

CWI, P.O. Box 94079
1090 GB Amsterdam, The Netherlands
http://www.cwi.nl/
e-mail: [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.*

KEY WORDS: *finite volumes, advection schemes, local refinement, air pollution, high performance computations*

1 Introduction

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

$$\begin{aligned} &\frac{\partial c_s}{\partial t} + \nabla \cdot (\underline{u} c_s) = V(\underline{u}, c_s) + R_s(c), \\ &c = \{c_s(\underline{x}, t)\} \in \mathbb{R}^m, \quad \underline{x} \in \Omega \subset \mathbb{R}^3, \end{aligned} \tag{1}$$

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) 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

[*] 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.

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) is discretized in space with a finite-volume technique. For typical grid resolutions and numbers of chemical species m, (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 [4, 11] and the survey [7] giving an overview of numerical techniques used in the field.

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 [5], several modern advection schemes are used. The two basic schemes are the Slopes scheme [9] and the Split scheme, a third order flux-limited upwind scheme [6, 8]. 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, 3]). 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

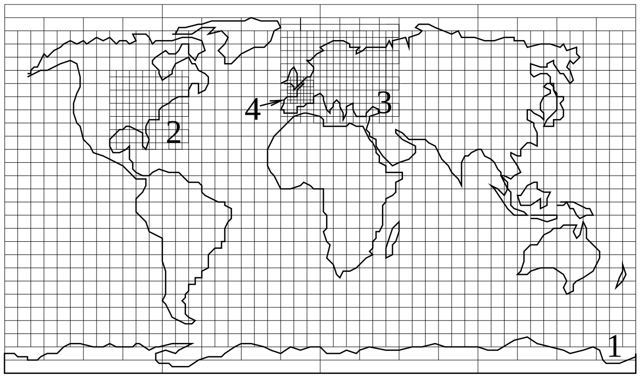

Figure 1: An example of zooming. Zoom regions are numbered.

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.

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 $F_{i\pm 1/2}^n$ 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 $\texttt{ref} = 3$ (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

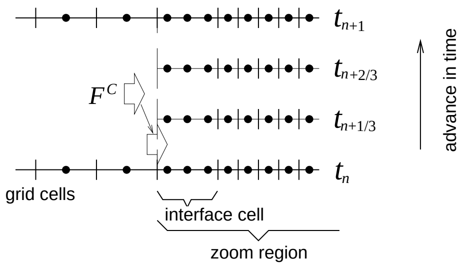

Figure 2: 1D zooming: one time step $t_n \rightarrow t_{n+1}$ corresponds to three time substeps in the zoom region. The "coarse" flux F^C is applied on the first substep.

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 [2]). 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.

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

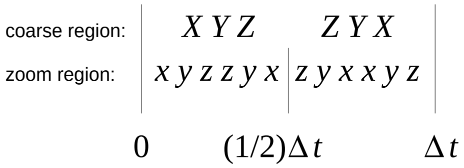

Figure 3: The sequence of advection steps in a coarse and a fine region with refinement factor 2.

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 [2].

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 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 [10]. In this test the advection equation on the (unit) sphere

$$c_t + \cos^{-1} y \, ((uc)_x + (v \cos y \, c)_y) = 0$$

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,

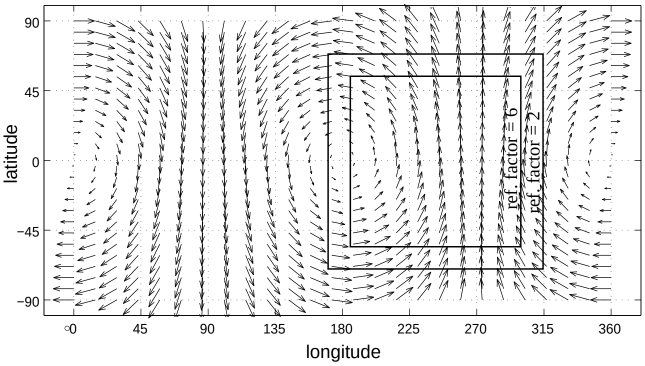

Figure 4: The velocity field and the position of the zoom regions.

with resolution $\Delta x = \Delta y = 4.5^{\mathrm{o}}$ and two zoom regions refined with factors 2 and 6 (resolutions $\Delta x = \Delta y = 2.25^{\mathrm{o}}$ and $\Delta x = \Delta y = 0.75^{\mathrm{o}}$ 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 ($\Delta x = \Delta y = 4.5^{\mathrm{o}}$), the zoom, and the overall fine ($\Delta x = \Delta y = 0.75^{\mathrm{o}}$) grids. In each case, we have measured accuracy by comparing the solution c^n 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

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

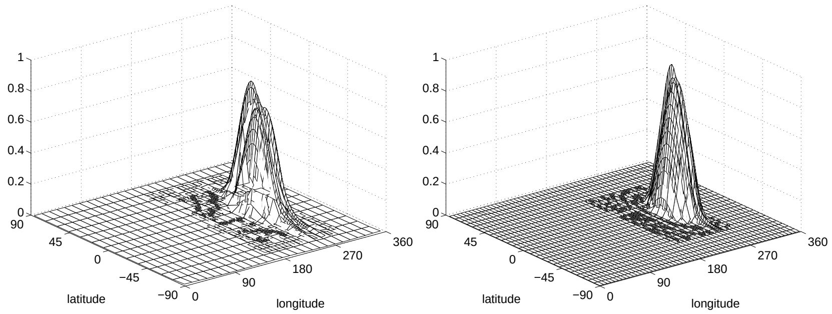

Figure 5: The cone after the full rotation

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^{o} 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, cone is to travel through the grid that is only factor 3 coarser, with basic resolution $\Delta x = \Delta y = 2.25^{\text{o}}$). As expected, 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 as 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 again refer to [2].

Our zoom code will be incorporated in the air pollution model TM3, a recent successor of [5]. 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).

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] P. J. F. Berkvens and M. A. Botchev, W. M. Lioen and J. G. Verwer. A zooming technique for wind transport of air pollution. Report MAS-R99xx, CWI, 1999.

[3] J. H. Flahery, P. J. Paslow, M. S. Shephard, and J. D. Vasilakis, editors. *Adaptive methods for Partial Differential Equations*. SIAM, Philadelphia, PA, 1989.

[4] T. E. Graedel and P. J. Crutzen. *Atmosphere, Climate and Change*. Scientific American Library. Freeman and Company, New York, 1995.

[5] M. Heimann. The global atmospheric tracer model TM2. Technical Report 10, Deutches Klimarechenzentrum (DKRZ), Hamburg, 1995.

[6] W. Hundsdorfer and B. Koren, M. van Loon and J. G. Verwer. A positive finite-difference advection scheme. *Journal of Computational Physics*, 117:35–46, 1995.

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

[8] 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.

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

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

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