Numerical smog prediction II: grid refinement and its application to the Dutch smog prediction model

M. van Loon

Department of Numerical Mathematics

NM-R9523 1995

Report NM-R9523
ISSN 0169-0388

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

CWI is the National Research Institute for Mathematics and Computer Science. CWI is part of the Stichting Mathematisch Centrum (SMC), the Dutch foundation for promotion of mathematics and computer science and their applications.
SMC is sponsored by the Netherlands Organization for Scientific Research (NWO). CWI is a member of ERCIM, the European Research Consortium for Informatics and Mathematics.

Numerical Smog Prediction II: Grid Refinement and its application to the Dutch Smog Prediction Model

M. van Loon
e-mail: vanloon@cwi.nl

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

Abstract

This paper describes a finite-volume grid refinement technique and its application to the Dutch Smog Prediction Model. This technique is used to improve the numerical quality of the model calculations, especially in areas with large spatial gradients (for example in regions with strong emissions) and areas of interest (to be specified by the user). To illustrate the effect of the refinement strategy, results of model computations and comparisons with measurements are presented.

AMS Subject Classification (1991): Primary: 65M50. Secondary: 80A32
CR Subject Classification (1991): G.1.8
Keywords & Phrases: local uniform grid refinement, numerical modeling, air pollution.
Note: The research reported belongs to the project EUSMOG which is carried out in cooperation with the Air Laboratory of the RIVM - The Dutch National Institute of Public Health and Environmental Protection. The RIVM is acknowledged for financial support.

1. Introduction

In [8] we described the physical aspects of the Dutch Smog Prediction Model including the choice of the base grid. As base grid we have a uniform grid with mesh widths of $.55^o$ in both horizontal directions. This means that the physical grid distances are about 61 km in North-South direction, and vary between 56 km. and 61 km. in East-West direction. This grid is too coarse to represent local phenomena well enough. For example the enhancement in concentrations in urban areas, resulting from local emissions, will not be resolved in the present grid. In general, point source emissions will directly be smeared out over a single grid cell, introducing an unnatural amount of diffusion into the model. To represent such local phenomena, a much finer grid is necessary. However, a uniform grid with mesh widths of, say, 10 km would already require more than 30 times as much grid cells as we have presently on the base grid and the computation time would increase with approximately the same factor. For routine smog predictions, this would be too expensive in terms of computation time, as the model calculations have to be done within a few hours on a workstation. Yet we need more resolution to represent local phenomena more precisely. This leads us to the concept of local grid refinement. The basic idea behind this technique is that a higher resolution is only needed in certain areas of the model domain, for example near (point) sources and strong gradients in the concentration field. In these areas the grid is refined and a more accurate solution is obtained. In other parts of the model domain the solution on the base grid will be good enough and no refinement is necessary there. By refining the grid only in areas where it seems necessary, much less grid cells are needed to obtain a solution comparable to the solution obtained by using the fine grid on the whole model domain. This requires that the areas in the model domain where the grid will be refined can be determined in a dynamic way. In other words, it should be possible to create a new refined grid every time step again. The reason for this is that the need for more resolution is not restricted to the direct surroundings of the location of the sources. Generally speaking, refinement will be necessary in areas with large solution gradients. Large gradients are likely to occur not only in areas with strong emissions but also (under certain circumstances) in other areas downwind from sources. This will certainly be the case during smog episodes.

The aim of this paper is to describe the grid refinement technique and its application to the Dutch Smog Prediction Model. First, the choice for a specific method is described. Next, a general description of the refinement technique is given, followed by the actual application to the four layer model with 15 components described in [8]. In the final sections, numerical results and conclusions are presented.

2. Choice of method

When choosing a grid refinement method one should, of course, take into account the specific application the method will be used for. In the smog model (see [8]) processes are present that satisfy mass conservation relations, for example horizontal advection and diffusion and also emission. For these processes numerical schemes are selected that conserve mass as well. Therefore, it is natural to require that the grid refinement technique does not disturb the conservation of any conservative integration scheme, or disturbs it only to a very limited extent. A grid point approach does not satisfy this requirement, as we will show later, and therefore we choose the finite-volume approach. This choice is in accordance with the physical model (see [8]) which is explicitly in terms of vertically averaged concentrations, but also in horizontal direction due to the way emissions have to be modeled.

Because the smog model is a four layer model in which vertical processes are parametrized, grid refinement in vertical direction makes no sense. For this reason only grid refinement in horizontal direction is applied. Moreover, in each layer the same grid structure will be used, thus avoiding complicated (and therefore expensive) treatment of vertical processes due to different grid structures. In the description of the grid refinement technique it therefore suffices to consider two-dimensional problems with one component only. Extension to four layers with a number of components is straightforward and will only briefly be discussed.

As we chose the finite-volume approach, the grid refinement techniques developed by Trompert & Verwer [9, 10], Arney and Flaherty [1], Berger and Oliger [3], Gropp [5, 6, 7] in their present form are not suited for our purpose. Yet, these methods are of interest because they can easily be adapted for the finite-volume context. We will not discuss the differences between the various methods as, in our opinion, these methods are comparable and mainly differ in the way the datastructure is built up and consequently how grid structures may look like. A finite-volume grid refinement algorithm is presented by Berger and Colella [2]. The basic idea of this method is the same as of the others, only their choice of datastructure is heavily dominated by practical considerations and the actual application. As a consequence, each grid level consists of a set of rectangular subgrids, possibly overlapping. The reason to require (sub)grids to be rectangular is to avoid (user) implementation of complicated numerical algorithms for a specific grid structure and loss of computational efficiency because otherwise the algorithms have to be applied on irregular meshes. The consequence of this requirement is an increased overhead, due to the complicated way the grid structures are created and, in addition, extra computations in overlapping grid cells. We do not follow this approach for two reasons. First of all, a number of processes in the smog model [8] has no horizontal coupling and just a simple loop over all grid cells is sufficient. Secondly, even if there is horizontal coupling (e.g. in advection and diffusion) it depends on the specific datastructure and numerical scheme whether or not loss of computational efficiency will occur. In view of our experience with the method of Trompert & Verwer [9, 10], we do not, generally speaking, expect a significant loss of computational efficiency. We therefore adopted their basic ideas and used them to construct our own finite-volume grid refinement algorithm. The datastructure however has been adapted for specific use in the smog model. Also a few changes had to made because of the formulation in spherical instead of cartesian coordinates. The actual implementation of the datastructure is based on a code written by Blom, described in [4], in which the changes in the datastructure are carried through.

3. Local Uniform Grid Refinement

In this section the local uniform grid refinement technique will be described. First a global outline is given and next each step of the refinement procedure will be discussed in more detail. The datastructure will not be discussed here. The interested reader is referred to Appendix A where a short description of the datastructure is given.

3.1 Algorithmic Outline

The first step of the algorithm is integration from time level T to $T + \Delta T$ on the base grid. Next, the algorithm checks whether grid refinement is necessary. If so, a new, fine grid is created and the integration from T to $T + \Delta T$ is redone on this grid, if necessary in more than one time step. Missing initial values on the fine grid are obtained by interpolation. After integration on the fine grid, the algorithms checks whether further refinement is necessary and if so, a second level of refinement is created which is treated in the same way as the first level. This process of creating even finer grids is continued until the solution on a certain level becomes acceptable or until a prescribed maximum number of levels is reached. At this point the time integration step is finished. Now the solution on each grid level is injected into the solution on the next lower grid level. This means that the solution values in grid cells of the fine grid are used to obtain solution values in the grid cells of the underlying coarse grid. This has to be done in a top down manner, of course, because the solution at the finest grid is expected to be the most accurate solution. A schematic representation of the algorithm is given by Algorithm 3.1.

```
1.   integrate on base grid
2.   check if and where refinement is necessary
3.   if no refinement necessary goto 9
4.   create datastructure of new grid
5.   determine initial values on new grid
6.   determine boundary conditions
7.   integrate on new grid
8.   if #level < max_level goto 2
9.   inject solution
```

Algorithm 3.1: The grid refinement procedure for one time step

3.2 How to refine?

Suppose we have integrated on the base grid from T to $T + \Delta T$. Using some criterion (to be specified later) the algorithm decides in which grid cells the solution is not good enough. These cells together with their eight neighboring cells are flagged and refined by bisecting all sides, see Fig.1. Cells at and near grid boundaries may not have eight neighbors. In that case only the existing neighboring cells are refined. This procedure implies that one flagged cell already results into a 4×4 grid on the next grid level in case of a boundary cell, and into a 6×6 grid in case of an interior cell. On top of this all cells that abut a flagged cell, are also refined. In this way a safety buffer is created that will prove useful later on. Moreover, the smallest possible grid structure on the next grid level is a 6×6 grid, ensuring that numerical methods with stencils of six or less points (cells) in one direction can be applied in a consistent way. Note that this procedure allows the refined grid to consist of several disjunct subgrids, see for example Fig. 2.

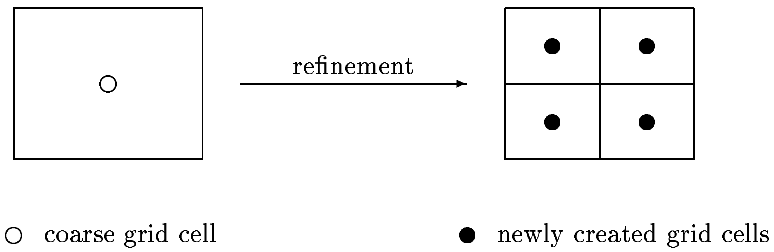


Figure 1: Refinement of a grid cell

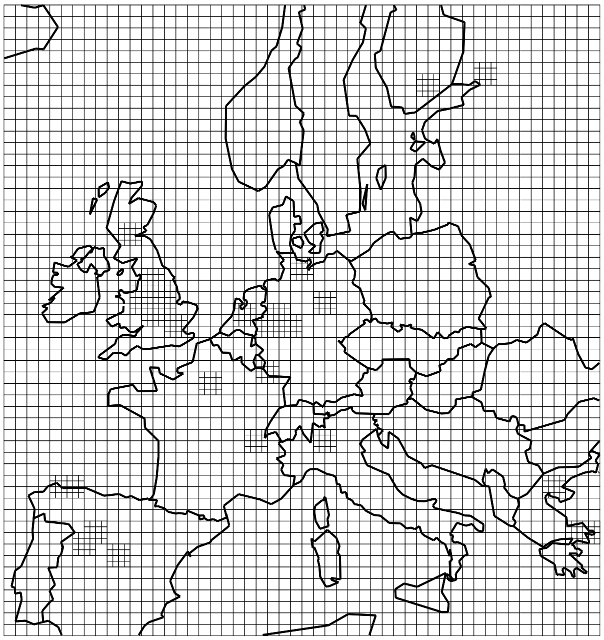


Figure 2: Example of grid refinement with two grid levels

3.3 Where to refine?
Suppose we have integrated on a certain grid level l and we have obtained a solution c^l on this level. To determine where refinement is necessary, some error estimate is needed. We use the curvature of the solution as error indicator, similar as in [9], based upon the second order derivatives of the solution. Since we only need an expression for the curvature of the solution with respect to the coordinate system, or rather the computational grid, we simply use the expression

$$(\Delta\phi)^2|c^l_{\phi\phi}| \; + \; (\Delta\theta)^2|c^l_{\theta\theta}|, \tag{3.1}$$

which is approximated by standard second order central finite differences in internal grid cells and by uncentered first order differences in boundary cells. The longitude and latitude coordinates are denoted by ϕ and θ respectively. A grid cell is flagged if in this cell

$$(\Delta\phi)^2|c^l_{\phi\phi}| \; + \; (\Delta\theta)^2|c^l_{\theta\theta}| > tol \; \min(c^{max}, c^l_{max}), \tag{3.2}$$

where *tol* is a tolerance value specified by the user, c^l_{max} is the maximum value of c^l at time $T+\Delta T$, and $c^{max} > 0$ is a user defined value. The factor $\min(c^{max}, c^l_{max})$ in the right hand side of (3.2) is added in order to make the refinement criterion independent of the scaling of the problem. The reason why the minimum of the maximum value of the solution and the parameter c^{max} is taken, is that due to emissions c^l_{max} may become so excessively high, especially when $l > 1$, that refinement outside the emission area(s) would not occur otherwise. This consideration suggests a value for c^{max} in the order of magnitude of the natural background value for species c.

In addition to the refinement criterion (3.2), the user has the possibility to enforce refinement in certain areas. To achieve this, the code not only flags cells satisfying (3.2) but also flags cells indicated by the user. These cells may correspond to certain longitude-latitude coordinates, specifying an area of special interest, for example The Netherlands, or to certain emission areas if not already flagged due to (3.2). In the same way the user might unflag cells in areas in which he does not want the code to create refined grids, albeit that this has to be done very carefully and with good knowledge of the physical model.

3.4 Interpolation of initial values

If a fine grid level l has been created, initial values are needed for the start of the integration. Three different situations are distinguished:

- If it is the initial time step, the initial values on each grid level are supposed to be specified by initial conditions. A subroutine is available that assigns initial values to grid cells.
- If level l also existed in the previous time step, values in coinciding cells are copied.
- For all other cells interpolation has to be carried out.

If interpolation has to be carried out, we require that the interpolation procedure is mass conserving, i.e. the sum of the mass in the four fine grid cells should be equal to the mass in the underlying coarse grid cell. This is evidently the case when using constant interpolation. For higher order interpolation methods, it is more complicated to achieve conservation of mass. Suppose a coarse grid cell with value c_0 is refined and the values in the fine grid cells need to be obtained from interpolation. Let us denote them by c_1, c_2, c_3, c_4. The coordinates of the cell centers are given by (ϕ_i, θ_i), with $i = 0, \ldots, 4$. In spherical coordinates, the mass in a grid cell $[\phi_i - \frac{1}{2}\Delta\phi, \phi_i + \frac{1}{2}\Delta\phi] \times [\theta_i - \frac{1}{2}\Delta\theta, \theta_i + \frac{1}{2}\Delta\theta]$ is given by (taking c_i to constant or as averaged concentration over the cell) $c_i S_i$ where S_i is the surface integral over the cell defined by

$$S_i(\Delta\phi, \Delta\theta) \;=\; \int\limits_{\phi_i-\frac{1}{2}\Delta\phi}^{\phi_i+\frac{1}{2}\Delta\phi} \int\limits_{\theta_i-\frac{1}{2}\Delta\theta}^{\theta_i+\frac{1}{2}\Delta\theta} \cos\bar{\theta} d\bar{\theta} d\bar{\phi} \;=\; \Delta\phi\Delta\theta\cos\theta_i \frac{\sin(\frac{1}{2}\Delta\theta)}{\frac{1}{2}\Delta\theta}. \tag{3.3}$$

From expression (3.3) it can be seen that the surface integral is not only proportional to $\Delta\phi\Delta\theta$, as would be expected from the corresponding expression in cartesian coordinates, but depends also on the latitude and the actual mesh size in latitude direction. Imposing a mass balance for the coarse grid cell gives the condition for mass conserving interpolation

$$\sum_{i=1}^{4} c_i S_i(\frac{1}{2}\Delta\phi, \frac{1}{2}\Delta\theta) \;=\; c_0 \sum_{i=1}^{4} S_i(\frac{1}{2}\Delta\phi, \frac{1}{2}\Delta\theta) \tag{3.4}$$

which can be simplified to

$$\sum_{i=1}^{4}(c_i - c_0)\cos\theta_i \;=\; 0. \tag{3.5}$$

The condition for mass conserving interpolation (3.5) suggests interpolation of the function $c(\theta)\cos\theta$ in order to obtain values c_i, $i = 1, \ldots, 4$. Condition (3.5) is equivalent with

$$\sum_{i=1}^{4} c_i \cos\theta_i \;=\; 4c_0 \cos\theta_0 \cos(\frac{1}{4}\Delta\theta). \tag{3.6}$$

If the function $\tilde{c}{=}c\cos\theta$ is used in the following way to obtain interpolated values c_i

$$c_i \cos\theta_i \;=\; c_0 \cos\theta_0 \;+\; \tilde{c}_\theta(\theta_i - \theta_0) \;+\; \tilde{c}_\phi(\phi_i - \phi_0) + \text{ h.o.t.} \tag{3.7}$$

with $\tilde{c}_\phi$ and $\tilde{c}_\theta$ approximations for the derivatives of the function $\tilde{c}$ in (ϕ_0, θ_0), we obtain

$$\sum_{i=1}^{4} c_i \cos\theta_i \;=\; 4c_0 \cos\theta_0. \tag{3.8}$$

Hence, the conservation condition (3.6) is slightly violated. Exact conservation can be imposed by multiplying the interpolated values c_i by a factor $\cos(\frac{1}{4}\Delta\theta)$. For the base grid, this factor is already larger than 0.99999 so that omitting this factor will hardly be felt.

In the model however, we simply use constant interpolation. In practise, interpolation of initial values will only be necessary near the boundaries of new created grid levels where the solution is to be expected relatively smooth. Recall that the refinement procedure causes a safety buffer to be created by refining cells that need not to be refined based on the error estimate only.

3.5 Injection of solution values

When the time integration step is finished, either because the algorithm decided that no further refinement is necessary, or because a prescribed maximum number of grid levels has been reached, the solution values of the fine grid cells have to be injected into the solution values of the underlying coarse grid cells. Again the requirement is that this has to be done in a mass conserving way. The obvious way to perform the injection is to sum up the mass for each set of four fine grid cells that form one coarse grid cell and assign this mass to the coarse grid cell. In terms of average concentrations over a grid cell, mass conserving injection of four fine grid values to one coarse grid value, using the notation of the previous subsection, can be written as

$$c_0 \;=\; \frac{1}{4}\sum_{i=1}^{4} \frac{\cos\theta_i}{\cos\theta_0 \cos(\frac{1}{4}\Delta\theta)} c_i \tag{3.9}$$

Again we note that omitting the factor $\cos(\frac{1}{4}\Delta\theta)$ will hardly be felt. However, this factor has to be computed only once per injection step, so it does not require much extra computation time to take this factor into account. The same holds for $\cos\theta_i$, $i = 0, \ldots, 4$, since an array containing the cosines of all cell center coordinates is stored if a new grid is created, because they are also needed in other parts of the model computation. Therefore, there is no reason to apply a different formula than (3.9.

It is the injection step in which a grid point approach fails to be conservative. In such an approach, a quarter (ignoring boundaries) of the fine grid point coincides with coarse grid points. Injection of solution values then is straightforward. Values in fine grid points coinciding with coarse grid points are copied, other values in fine grid points do not influence the solution on the coarse grid. Though formally we cannot speak about mass on a grid in the grid point approach, we define the mass on a grid with mesh widths (for simplicity in Cartesian coordinates) Δx and Δy as the sum of all solution values times $\Delta x \Delta y$, being a second order approximation of the integral of the concentration function over the domain. It is now easy to see that a different mass is injected from the fine grid than the mass present on the fine grid. The mass on the fine grid M_F is given by the sum of the concentration values times $\frac{1}{4}\Delta x \Delta y$. The injected mass M_I, however, is equal to the sum of the concentration values in grid points coinciding with coarse grid points times $\Delta x \Delta y$. In general, these two sums are not equal. A simple example clearly shows the inconsistency of the grid point approach at this point. Consider a point source on the fine grid. The only possible way to treat a point source is to assign the emitted mass M_E to the nearest grid point. If this grid point is not coinciding with a coarse grid point, the mass M_E is not injected into the coarse grid solution and the emission is lost on the coarse grid. On the other hand, if this grid point does coincide with a coarse grid point, the solution value is copied. In that case the increase of mass due to the emission of M_E is equal to $4M_E$, because the grid point on the coarse grid represents a four times larger area.

The above considerations show that a grid point approach is not suited for this application. The present finite-volume approach deals with mass conservation in a quite natural and consistent way, and in our opinion this approach is to be preferred in atmospheric models.

3.6 Boundary conditions

Assuming boundary conditions are prescribed for cells that abut the physical boundary, no problems arise when integrating on the base grid. At this grid level we only have physical boundaries. When integrating on grid level $l > 1$, not all boundary cells will abut the physical boundary (see Fig. 2). Those cells that do not abut the physical boundary are called internal boundary cells and for these cells additional boundary conditions have to be specified. Following Trompert and Verwer we prescribe, when necessary, Dirichlet boundary conditions for these cells which can be derived from the solutions at time T and time $T + \Delta T$ at the underlying coarser grid level $l - 1$ by first applying spatial interpolation followed by temporal (linear) interpolation.

In case of advection it seems natural to have flux conditions at physical inflow boundaries. However, no information is available about these fluxes. We observed, for example, that assuming inflow of "clean" air may lead to unnecessary grid refinement in case the concentrations in grid cell next to boundary cells are higher than the computed concentrations in the boundary cells. For this reason, values in cells that abut the physical boundary are computed by extrapolation in case of inflow. We use constant extrapolation to prevent new extrema in the solution due to the extrapolation.

3.7 Mass conservation

We have seen that the grid refinement technique will disturb the mass conservation to a very limited extend. In general, this happens when the mass on the grid changes during the integration step due to in- and outflow. The change in mass is equal to the integral of the fluxes over the boundary of the computational grid.

To illustrate how mass conservation may be slightly disturbed, consider an integration step on a domain Ω using a coarse grid. On the subdomain $\Omega^{'} \subset \Omega$ the coarse grid is refined. The exact change in mass ΔM on the subdomain is given by the integral of the fluxes over the boundary $\partial\Omega^{'}$. On both the coarse and the fine grid this integral is approximated numerically in different ways, so in general, the approximations of ΔM will not be the same for both grids. Since the mass of the fine grid is injected into the coarse grid solution, the mass on the coarse grid will change with $\Delta M_F - \Delta M_C$ where ΔM_F and ΔM_C denote the change in mass on $\partial\Omega^{'}$ using the fine grid and using the coarse grid, respectively.

Of course, it is possible to enforce exact mass conservation by imposing a suited flux condition at $\partial\Omega^{'}$ for the integration on the fine grid. This condition may be derived from the fluxes over $\partial\Omega^{'}$ during the coarse grid computation. However, this leads to a lot of extra overhead. In addition, negative values or underhoot may occur in advection computations. Further, the underlying coarse grid cells of the boundary cells of the fine grids have not been flagged by the space monitor, so strong gradients are not expected at fine grid boundaries. At this point the usefulness of the safety buffer, described in Section 3.2, becomes clear. Because we suppose the numerical solutions to be smooth at the fine grid boundaries no significant gain or loss of mass is expected due to the boundary treatment.

3.8 Time stepping

The model takes fixed overall time steps of half an hour (see [8]). As operator splitting is applied, each subprocess is integrated separately. For each subprocess as many time steps can be taken as necessary to integrate from time level T to $T + \Delta T$. The number of time steps can also vary per grid level, as the integration on a fine grid is independent of integration on the underlying coarse grid. For example, in case of advection, the time step on the base grid may be the overall time step, whereas on the finer grids two or more time steps are necessary in order to satisfy the Courant condition on these levels.

3.9 The datastructure

The datastructure is based upon the one described in [4] and is closely related to the one in [9]. Some modifications were necessary. The solution on each grid level (including the base grid) is stored row-wise in a one-dimensional array. Information about the structure of each grid level is also stored in a one-dimensional integer array. This array consists of several (sub)arrays that actually describe the grid

- an array containing the number of rows in the grid and pointers to the start of each row,
- an array containing for each row the row number corresponding to its θ-coordinate,
- an array specifying for each grid cell its column number in a virtual rectangle, corresponding to its ϕ-coordinate,
- an array containing the number of physical and interior boundary cells and pointers to these cells,
- an array with for each grid cell a pointer to the underlying coarse grid cell it is part of,
- an array with for each grid cell a pointer to the cell directly above it,
- an array with for each grid cell a pointer to the cell directly below it,
- an array with for each grid cell a pointer to the lower left cell of the four cells on the next finer grid that form the present coarse grid cell.

In case cells, as indicated in the three last descriptions, do not exist, the corresponding pointers are set to zero. With the aid of the arrays listed above the implementation of numerical algorithms becomes only slightly more difficult in comparison to using just a uniform grid. Creating the fine grid structure is, in our opinion, not a very complicated the task and, in our experience, only requires a few percent of the total computation time, including the construction of initial values and the injection procedure. The latter two processes can be implemented in a straightforward manner because of the pointers to underlying coarse grid cells and to finer cells on the next grid level. These pointers were not present in the datastructure of Trompert & Verwer and therefore they had to implement the interpolation of initial values and the injection procedure in a more complicated and time consuming way. For our application it is worthwhile to have these pointers, because we will have to perform injection relatively often due to the operator splitting approach.

For a more detailed description of the datastructure, see Appendix A.

4. Application to the four-layer model

In the previous section a description was given of the grid refinement algorithm for a two dimensional problem in space with only one solution component. The purpose of this section is to explain how we applied this technique within our smog prediction model with four layers in vertical direction and 15 species, described in [8]. Recall that the model is not really a 3D model. The physical description is in terms of vertically averaged concentrations and therefore it makes no sense to refine in vertical direction. This explains why we restrict ourselves to a 2D refinement technique.

4.1 The grid

Each layer is numerically represented by a two-dimensional grid. Because of the several exchange processes between the layers (vertical diffusion, deposition, fumigation), it would be very inconvenient if in different layers different grids were created. An additional interpolation procedure would be necessary to interpolate the concentration function in the layer above and the layer below. Therefore, we work with the same grids in all layers. An additional advantage is that only one grid structure has to be created for all layers and components, thus reducing overhead and saving memory space.

4.2 When to refine?

As already mentioned, the numerical integration is performed in an operator splitting setting. In the odd integration steps we perform the subprocesses in the order: advection, diffusion, emission, deposition, fumigation and chemistry. In the even integration steps we perform them in reversed order, after an update of the model parameters. All processes are integrated on all grid levels. The remaining question is: should regridding take place in all processes or just in one of them? And, if we choose the latter, in which one? First of all, it is clear that avoiding regridding in each substep saves computation time. So it is worthwhile to look at each of the processes listed above and check whether they require regridding or not. Observing that all processes except advection and emission do not introduce (new) sharp gradients or move existing sharp gradients, we conclude that only advection and emission are candidates for regridding. As we can combine both by steering the grid refinement within the advection step, this step is the obvious choice. As mentioned earlier, refinement can be imposed not only according to criterion (3.2) but also to other, problem dependent criteria. If we also refine based on emission data, introduction of sharp gradients due to emission can be anticipated within the advection step. For another reason it also natural to have the refinement taking place in the advection step. In this step peaks in the solution are moved to other locations in the model area and the only possibility to follow this with grid refinement is by letting the advection determine the refinement.

4.3 The refinement criterion

The extension of criterion (3.1) and (3.2) to a four-layer model with 15 species is straightforward. First, we calculate the space monitor $spcmon(j,l)$ for each component j and each layer l according to (3.1). Then we calculate total space monitor SPC

$$SPC \;=\; \max_{i,j} \left\{ \frac{w_{j,l} * spcmon(j,l)}{c_{j,l}^{max}} \right\}, \tag{4.10}$$

where the $w_{j,l}$ are weight factors corresponding to component j and layer l, with $0 \le w_{j,l} \le 1$; $c_{j,l}^{max}$ is the maximum value of component j in layer l on the grid level considered. Finally, the refinement criterion reads

$$SPC \; > \; tol, \tag{4.11}$$

where *tol* is a tolerance value to be defined by the user.

4.4 Computational efficiency

In case a new grid level is created in the advection step, a complete integration step including all subprocesses will be done on this new grid. In case of horizontal coupling between the grid cells the solution values in all grid cells on all grid levels are updated. In principle this is not necessary. In fact, only values in grid cells which are not further refined really need to be updated. All other values are obtained by injection from the next finer grid. However, for processes with horizontal coupling a complicated (and unphysical) procedure for the boundary conditions would be necessary since boundary conditions can no longer be derived from the just finished update on the next lower grid. Therefore, only in subprocesses with vertical coupling or no coupling at all between the grid cells, this (computational) advantage will be used. Its implementation is quite easy due to the pointer array pointing to cells on the next finer grid in case of further refinement, and to zero otherwise, see Section 3.9.

5. Experiments I: Winter smog

In this section, results of model runs using meteorological data from a winter smog episode will be presented. The selected episode is November/December 1989. For this period hourly measurements for SO_2 in a number of stations in The Netherlands are available. Also measurements in EMEP stations are available, but only on daily averaged basis. Figure 3 shows the distribution of the stations of the Dutch National Air Quality monitoring Network and the distribution of the EMEP stations over Europe. Unfortunately, for a large number of the EMEP stations no measurements are available or cannot be used because the altitude of the stations is too far above sea level. From the few remaining measurements a concentration distribution over Europe has to be derived. This will be a very crude estimation of the real distribution and comparison of the model output with measurements on European scale will therefore only be indicative. For that reason we will restrict the comparisons in this paper to comparisons with measurements of the Dutch monitoring network. Another reason to do so is that the model has been developed to give predictions for The Netherlands.

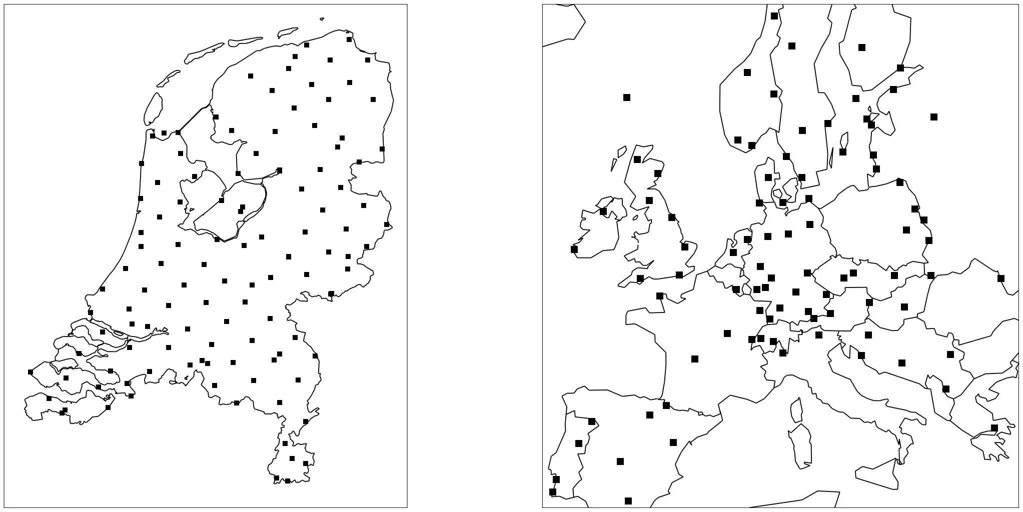


Figure 3: Stations in The Netherlands (l) and EMEP stations (r)

In this section, results of experiments will be presented. First, a comparison is made between the results of model runs with EUROS (the original model) and CWIROS, in order to check whether CWIROS produces comparable results as EUROS. Next, the virtue of grid refinement is illustrated by comparing results obtained by using a number of refined grids and results without using grid refinement. Finally a comparison will be made with measurements.

5.1 Comparison between EUROS and CWIROS

As the original model EUROS is a model for winter smog, both models are compared using data from the November/December 1989 smog episode. When running CWIROS without grid refinement, the differences between EUROS and CWIROS are:

- different advection schemes are used,
- CWIROS uses a much more complicated chemical model and consequently solves the resulting chemical kinetics problem in a different way,
- CWIROS uses other emission data than EUROS,
- CWIROS applies different deposition parameters for land and sea.

With respect to the chemical model, it should be noted that the more complicated chemical model in CWIROS reduces to a very simple model in wintry conditions when only considering SO_2 and SO_4 and involves the same reactions as the simpler chemical model in EUROS. Therefore, both chemical models are considered to be comparable when modeling SO_2 and SO_4 in winter. For this comparison however, we modeled the OH concentration in CWIROS in the same way as it is done in EUROS, where the OH concentration is prescribed as a function of time. This approach decouples SO_2 and SO_4 from the other species and thus simplifies the chemistry if only results for SO_2 and SO_4 are desired (see [8].

The operational code EUROS is run every day simulating a period of 5 days (or 120 hours). Therefore we chose a five days period from the selected episode, from 27 November 1989 12:00 GMT till 1 December 12:00 GMT (for initialization purposes, the model starts one day earlier than the specified time). In figure 4 solution plots of both model runs at 1 December 12:00 GMT are given. Apart from differences which are probably caused by the above mentioned reasons, both plots are in good agreement with each other. Both plots show an increased SO_2 level mainly in or close to the North-Eastern part of The Netherlands. This is in accordance with actual measurements, as we will see later on.

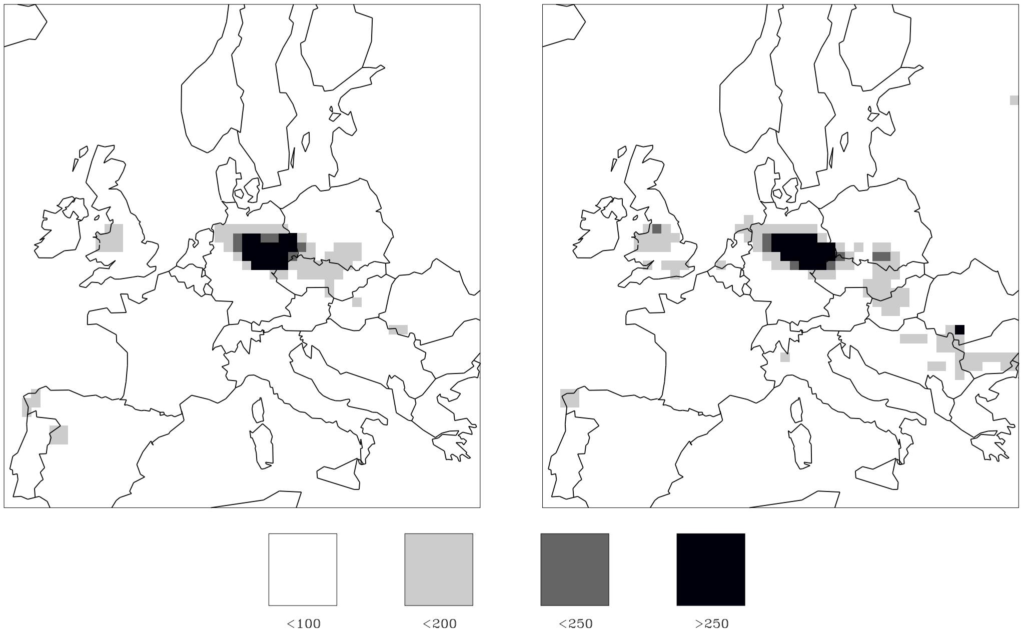


Figure 4: Solution plots for EUROS (l) and CWIROS (r) without grid refinement, SO_2 in $\mu g\, m^{-3}$ for 1 December 1989, 12:00 GMT.

5.2 Grid refinement

The model runs have also been performed using 2, 3 and 4 grid levels. On grid level 2 and 3 we always refined a rectangular area containing The Netherlands. Further, grid refinement is enforced around all sources with the restriction that grid refinement is never applied *outside* the region defined by $[0^o, 15^o] \times [-15^o, 0^o]$ to prevent unnecessary and time consuming grid refinement. In the solution plot the area in which refinement is allowed, is indicated by a rectangle. In Figure 5 the solutions of two model runs are plotted. Both solution plots are in good qualitative agreement with each other, again indicating a proper functioning of the grid refinement. The plots also show that grid refinement results into a (slightly) different solution, as was to be expected. Furthermore, as in Figure 4, the solution plots show a cloud with polluted air over central and north-west Europe that just passes through the north-eastern part of The Netherlands. The latter observation is in accordance with Dutch measurements, as we will show in Section 5.3. Table 1 gives some statistical information about the four model runs. The numbers in the last column may serve as a measure for the efficiency of the refinement procedure. The numbers specify the percentage of grid cells used in the model runs relative to the number of cell necessary in case of a uniform fine grid on the maximum grid level used. In the model runs, refining only The Netherlands results into 196 cells on level 2, 616 cells on level 3. These numbers are relatively small compared to the average number of cells used on these grid levels, according to Table 1. If The Netherlands are further refined on level 4, it would take 1824 extra cells. However, we do not refine The Netherlands on level 4 automatically. If The Netherlands need further refiment the space monitor is supposed to take care of that. This prevents unnecessary refinement in (parts of) The Netherlands, and thus saves some computation time.

		grid level						
MAXLEV	TOL	1	2	3	4	total	uniform	%
1	-	2860	-	-	-	2860	-	-
2	0.5	2860	2084	-	-	4944	11440	43.2
3	0.5	2860	2054	4189	-	9103	45760	19.9
4	0.75	2860	1909	3323	7242	15334	183040	8.4

Table 1: Average number of cells for the model runs

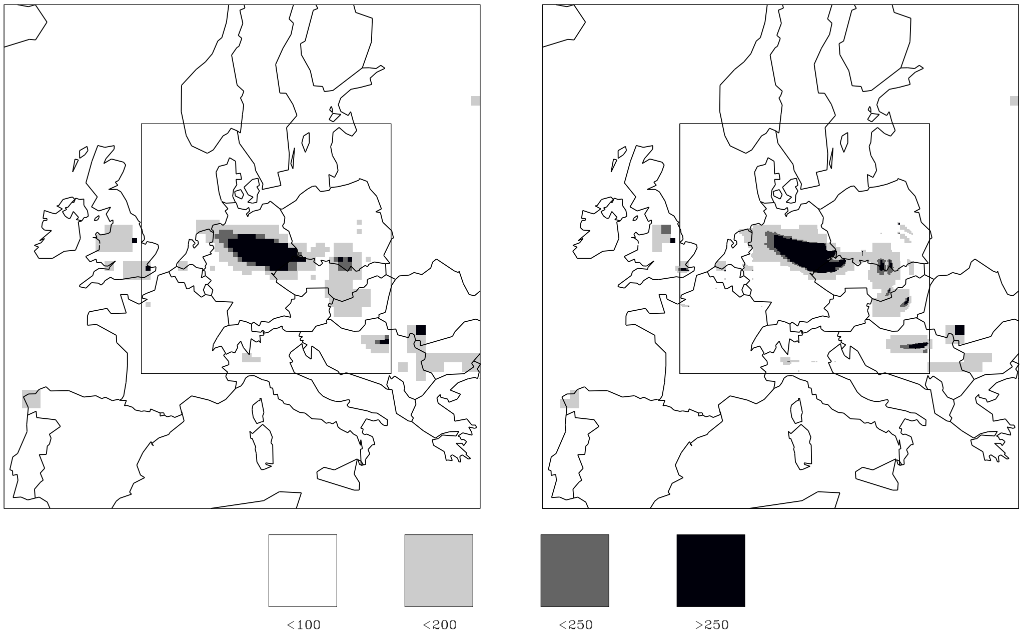
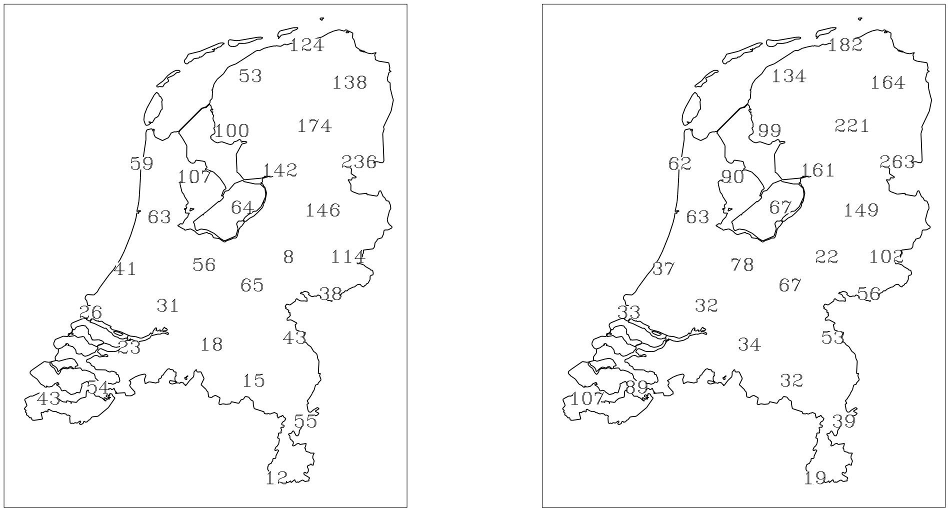


Figure 5: Solution plots for CWIROS using 2 grid levels (left) and 4 levels (right)
SO_2 in $\mu g\, m^{-3}$ at 1 December 1989, 13:00 MET

5.3 Comparison with Dutch measurements

Comparing model results with observed concentrations is difficult. Not only model errors are present, but we also have to deal with uncertainties in the input parameters. An important parameter in the present experiment is the r_c value (the surface resistance) of SO_2. This parameter influences the dry deposition: the higher the surface resistance, the lower the deposition velocity. As SO_2 is not very reactive (it is only slowly transformed into SO_4 in our chemical model), dry deposition may cause significant removal of SO_2. However, the surface resistance of SO_2 is strongly dependent of the soil condition. In case of a frozen soil and snow this value is about five times higher than the default value 100 s/m. Because we do not have detailed information about the surface conditions in the selected period, an estimation for the surface resistance for SO_2 had to be made. Fortunately, temperature fields are available. We therefore estimated the surface resistance of SO_2 based on the local temperature. If the temperature is lower than -1^oC we take $r_c = 540$, if the temperature is higher than 1^oC we take $r_c = 100$ and between -1^oC and $+1^oC$ the surface resistance varies linearly between 540 and 100.

In Figure 6 measurements for The Netherlands are plotted. As can be seen from figure 6, the highest concentrations are observed in the North eastern part of The Netherlands. From figure 6 it can also be seen that a relatively large concentration gradient is present in The Netherlands, compared to the spatial resolution on the base grid. Model calculations on the base grid only will therefore not be able to reproduce this concentration gradient. However, we may expect that using a number of fine grid level improves the results especially near the gradient. Figure 7 shows the computed distributions above The Netherlands using no grid refinement and using four grid levels. Clearly the grid refinement delivers a much more realistic solution plot. It also nicely shows local sources that are invisible on the coarse grid solution. However, comparing Figure 7 with Figure 6 reveals that the computed concentration cloud in the northern part of The Netherlands has a somewhat different position than the observed cloud. This also seems to be the case with the small cloud in the south-west. Computed and observed concentrations in a measurement station will therefore not match, just because the computed concentration cloud may miss the station whereas in reality it just passes the station. In particular, this will happen for stations at the edge of the cloud.



12:00 MET

13:00 MET

Figure 6: SO_2 measurements in $\mu g\, m^{-3}$ for 1-12-1989

The following will show that grid refinement sometimes gives less well agreement with observations. Possible explanations will be given. These explanations will reveal that in order to exploit the virtue of a higher numerical resolution, the model input and model coefficients become more critical. Our first comparison concerns three measurement stations from the Dutch *Air Quality Monitoring Network*,

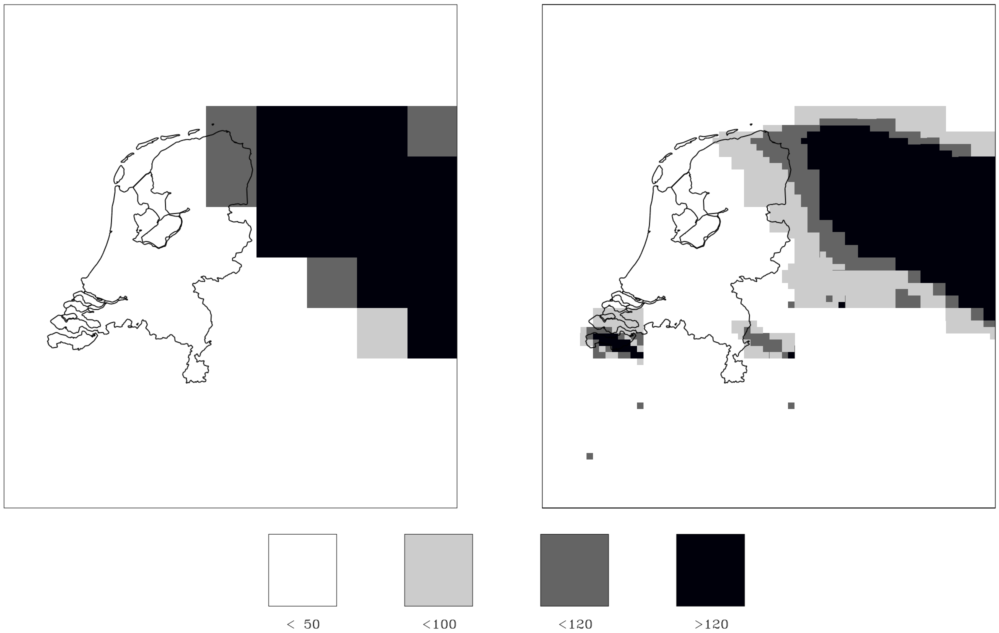


Figure 7: Solution plots for SO_2 in $\mu g\, m^{-3}$ at 1 December 1989, 13:00 MET.
left: 1 grid level; right: 4 grid levels

the stations 1-3 specified in Table 2. These three stations are located in the same grid cell on the base grid. Moreover, the three stations are very close to the observed concentration gradient. As can be seen from Table 2, station 3 is the closest one. Also in station 3 the highest SO_2 concentration is observed. Figure 8 shows the measured concentrations in the three stations. Figure 8a shows a peak in the observed concentrations on 1 December at 12:00 for station 3 and at 13:00 for the other two stations. Although the three stations are located in a small geographical area, the observed peak values are quite different. On the other hand, the time behavior for the three stations is comparable, so that one may expect a similar time behavior for the model calculations in the coarse grid cell. From figure 8b it is seen that this is indeed the case and also that the model predicts the peak at the right time, though the modeled peak value is lower than actually observed. After reaching the peak values, the observed concentrations decrease rapidly, whereas the modeled concentrations only slowly decrease. The reason for this is not clear. Wind directions may have changed very quickly. The model applies time interpolation between two 6-hour wind fields and wind variations in the model are therefore always smooth. Another explanation could be the occurrence of precipitation. According to the available precipitation fields derived from synoptic measurements, there was no rain in the area of

number	location	θ	ϕ
1	Sappemeer	53.14	6.80
2	Hoogersmilde	52.90	6.40
3	Wijerswold	52.66	6.81
4	Kloosterburen	53.40	6.41
5	Cornjum	53.24	5.61

Table 2: Some measurements stations and their coordinates in degrees

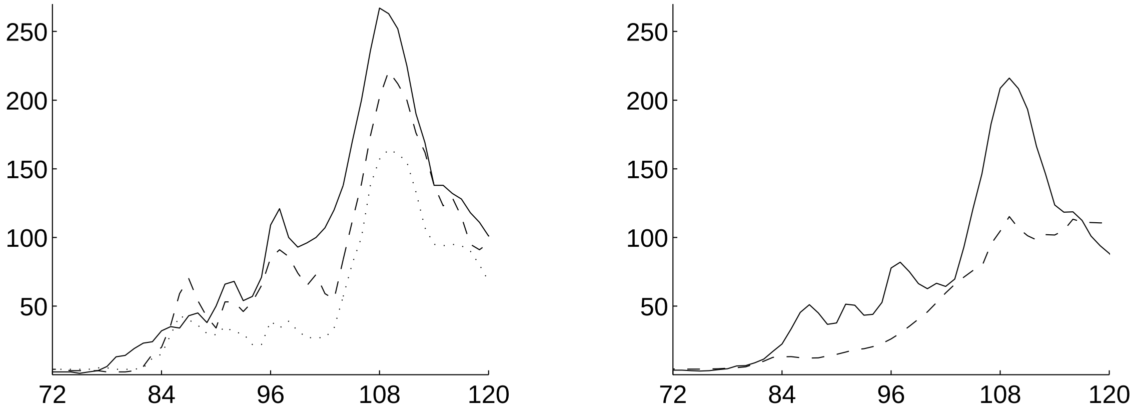


Figure 8: SO_2 measurements in $\mu g\, m^{-3}$ for 30-11 and 1-12 1989
left: station 1 (dotted), 2 (dashed) and 3 (solid); right: averaged measured concentrations over the 3 stations (solid) and modeled concentration in the coarse grid cell using 4 grid levels (dotted). The horizontal axis represents time in hours, 30-11-1989 0:00 GMT = 72.

interest in the selected period. If there has been significant local rainfall, that has not been resolved by the measurements, upwind of (or at) the stations, efficient wash-out would have taken place causing a drop in concentrations as observed, which then would have become visible in the model results, since wet deposition is included in the model. Another possible explanation is a change in mixing height. If the mixing height increases due to a change in weather conditions, the pollutants will be diluted and their concentration will decrease. The model, however, will not follow such a change in mixing height, since the mixing height has a predescribed profile and is taken constant in space.

On higher grid levels, the three stations are located in different grid cells. Figure 9 contains the modeled concentrations plots for station 1 and 3 for two different grid levels. The prediction for station 1 shows no visible improvement, whereas for station 3 shows less good agreement when using more grid levels. A similar result is obtained for station 2 (not plotted). A possible explanation is the presence of a local source upwind from station 3, that has not been resolved in the emission inventory of the model. In that case the model can be expected to predict lower values when refining the grid. Another explanation could be the fact that station 3 lies relatively close to the concentration gradient that is present above The Netherlands, as can be seen from the concentration plots in the Figures 4-7. If the wind fields used in the model are slightly more in northern direction than was actually the case, large differences like we encounter now may readily occur. Recall that the wind fields have to be obtained by time and spatial interpolation from other wind fields in a different coordinate system, and then have to be made divergence free. The latter process does not only affect the wind speeds, but also (slightly) changes the wind directions. In our experiments, we observed that the wind speeds at the cell centers (of the base grid) are corrected by about 10% on average.

Two other station were considered for comparing observations with model predictions. It concerns the stations 4 and 5, also specified in Table 2. The observed and modeled concentration profiles can be found in Figure 9. Again we see that the peak values seem to be modeled at the right time. For station 4 the grid refinement does not result into any improvement of the modeled concentration profile. For station 5 however the result of grid refinement is even more disappointing: instead of a closer resemblance with the observed concentrations, the prediction becomes worse, similar as was seen with the modeled concentrations for station 3. The same possible explanations for the differences

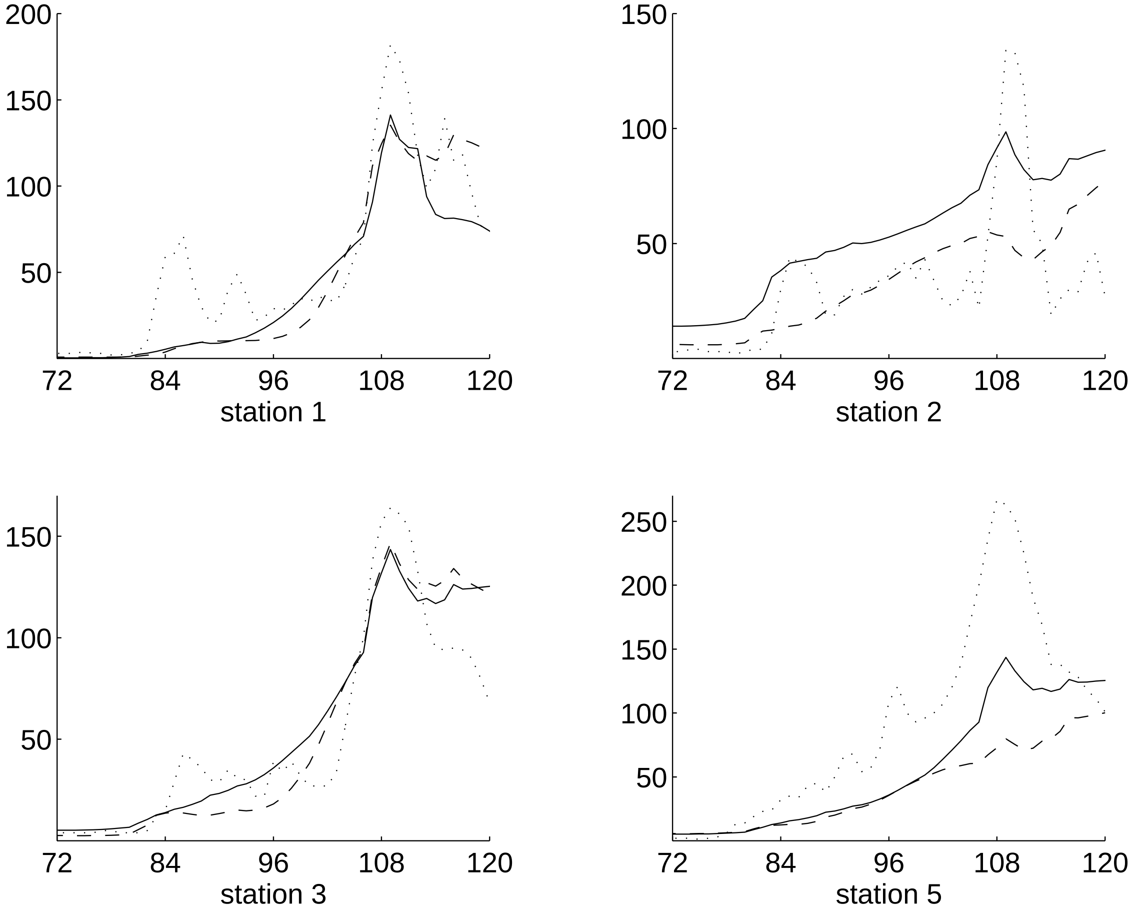


Figure 9: Observed (dotted) and modeled SO_2 concentrations in the stations 1,2,3 and 5 using 1 grid level (solid) and 4 grid levels (dashed) in $\mu g\, m^{-3}$ for 30-11 and 1-12 1989

between modeled and observed concentrations are valid as before.

6. Experiments II: summer smog

For summer smog, no comparison can be made between EUROS and CWIROS. It would make no sense to let EUROS perform ozone calculations as the modeled species in EUROS do not even include the VOCs which are essential in ozone formation. In this section, we present results of model runs for a smog episode from July 1989. The selected period is from 19 July until 24 July, a time interval of 144 hours.

An interesting numerical observation has been made during the experiments. The chemistry has to be solved such that the NO_x balance in each single cell is not disturbed (too much). If one does not take care of this, very different model results are obtained. As the nonlinear system in the chemical kinetics equations, arising from the BDF2 formula, is solved by Gauss-Seidel iteration, see [11, 12], this observation provides a possible stopping criterion for the iteration. In some cells probably the NO_x balance is not disturbed much and two iterations suffice, whereas in other cells 4 or 5 iterations are necessary. In the present experiments the number of iterations was taken equal to 5 in all grid cells. By specifying a stopping criterion, the algorithm may be made more efficient.

6.1 Grid refinement

Apart from runs on the base grid, runs with 2, 3 and 4 grid levels have been performed. Table 3 gives some information about the number of cells used by the algorithm for the model computations. A comparison with the corresponding Table 1 for the winter episode shows that in the present computations more grid cells have been used, especially on the grid levels 3 and 4. Both tables show the grid refinement becomes more efficient for MAXLEV larger than two. However, on our fastest workstation, the computations for MAXLEV>2 took much more computing time than the allowed 3 or 4 hours. This means that the restriction on the area in which the algorithm is allowed to refine the grid, is necessary and even then we have to further restrict the number of cells in order to meet the restriction on the computation time. Of course it is possible to increase the tolerance on the higher grid levels, but for the present experiments this has not been done in order to be sure of the quality of the fine grid solutions.

		grid level						
MAXLEV	TOL	1	2	3	4	total	uniform	%
1	-	2860	-	-	-	2860	-	-
2	0.5	2860	2232	-	-	5092	11440	44.7
3	0.5	2860	2291	6597	-	11747	45760	25.7
4	0.75	2860	2276	6557	12207	23900	183040	13.1

Table 3: Average number of cells for the model runs

Figure 10 shows the computed concentration distributions over Europe according to a coarse and fine grid computation. Figure 10 clearly shows that grid refinement results into higher ozone concentrations in Europe, especially in The Netherlands and in Great Britain. From the comparisons in Section 6.2 we will see that this means a better agreement with observed concentrations.

6.2 Comparison with Dutch measurements

While the winter smog episode was mainly restricted to the North-Eastern part of The Netherlands, in the present summer episode increased ozone concentrations are observed in the whole country. The highest concentrations occur in the South. Therefore we took measurements from three stations in the South together with one station in the North for comparisons with model calculations. The stations are listed in Table 4. Figure 11 shows the concentrations in the stations 6-9. Note that the measurements series for station 6 is incomplete, unfortunately. Figure 11 clearly shows that grid refinement improves the model predictions. Especially the peak values on the last day of the simulations are much better represented in the 4-level computation than in the coarse grid computation. We also see that the

number	location	θ	ϕ
6	Braakman	51.30	3.75
7	Wijnandsrade	50.90	5.88
8	Houtakker	51.52	5.15
9	Hellendoorn	52.39	6.40

Table 4: Some measurements stations and their coordinates in degrees

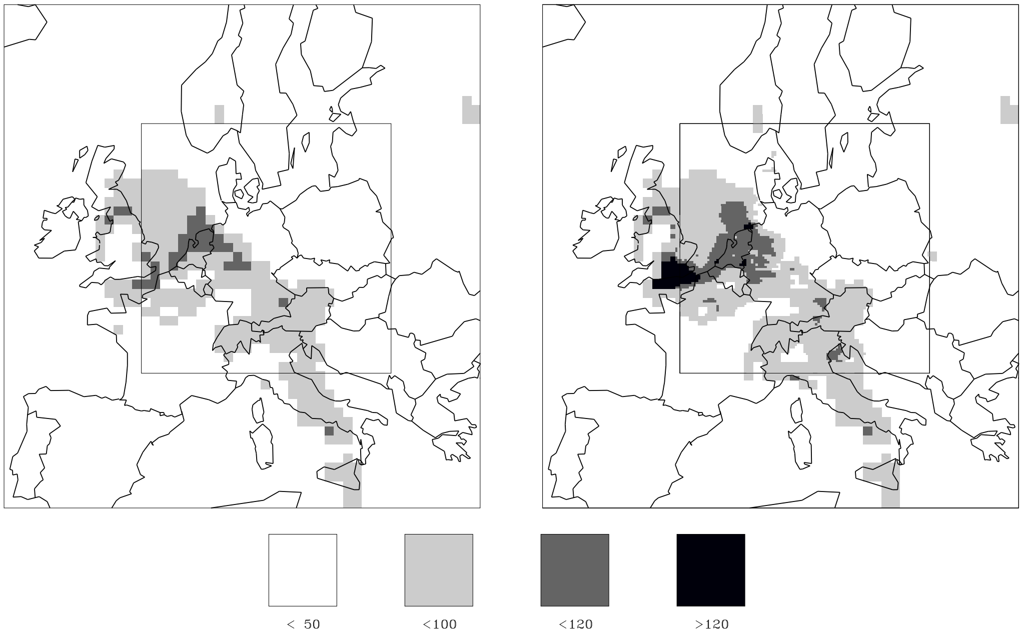


Figure 10: Computed O_3 distribution in $\mu g\ m^{-3}$ over Europe for 24-7-1989, 14:00 MET. For the coarse grid (l) and for 4 levels (r)

observed high values on the third day of the simulation are not predicted by the model and that grid refinement does not show improvement for that particular day. The nightly minimum values for station 9 (the Northern station) seem to be systematically too high. This gives rise to some questions concerning some modeling aspects and not concerning the grid refinement. In general we consider the model results in this comparison to be quite good.

7. Summary and conclusions

In the present paper a finite-volume, local uniform grid refinement technique has been described and applied to the Dutch Smog Prediction Model using data from a winter and a summer smog episode. From these experiments the following conclusions are drawn:

- CWIROS is in accordance with EUROS in a qualitative way. Differences are observed but seem to be caused by different (meteorological) input and not due to (errors in) the computational process.
- The time behavior of the model seems in order. The experiments in Section 5 show that the model is able to predict concentration peaks at the right time. This indicates that the emission inventory as well as the advection and emission/deposition routines in the model are correct.
- The grid refinement technique works properly, but a better agreement with observed concentration has not been obtained by applying this technique in the winter smog episode from November/December 1989. In the July 1989 summer smog episode, the refinement does produce concentration profiles that are in significantly better agreement with measurements.
- Restriction on the number of grid cells of the fine grids may be necessary for summer smog computations due to operational constraints. The present approach of restricting the area in which the algorithm may refine the grid is probably not sufficient.

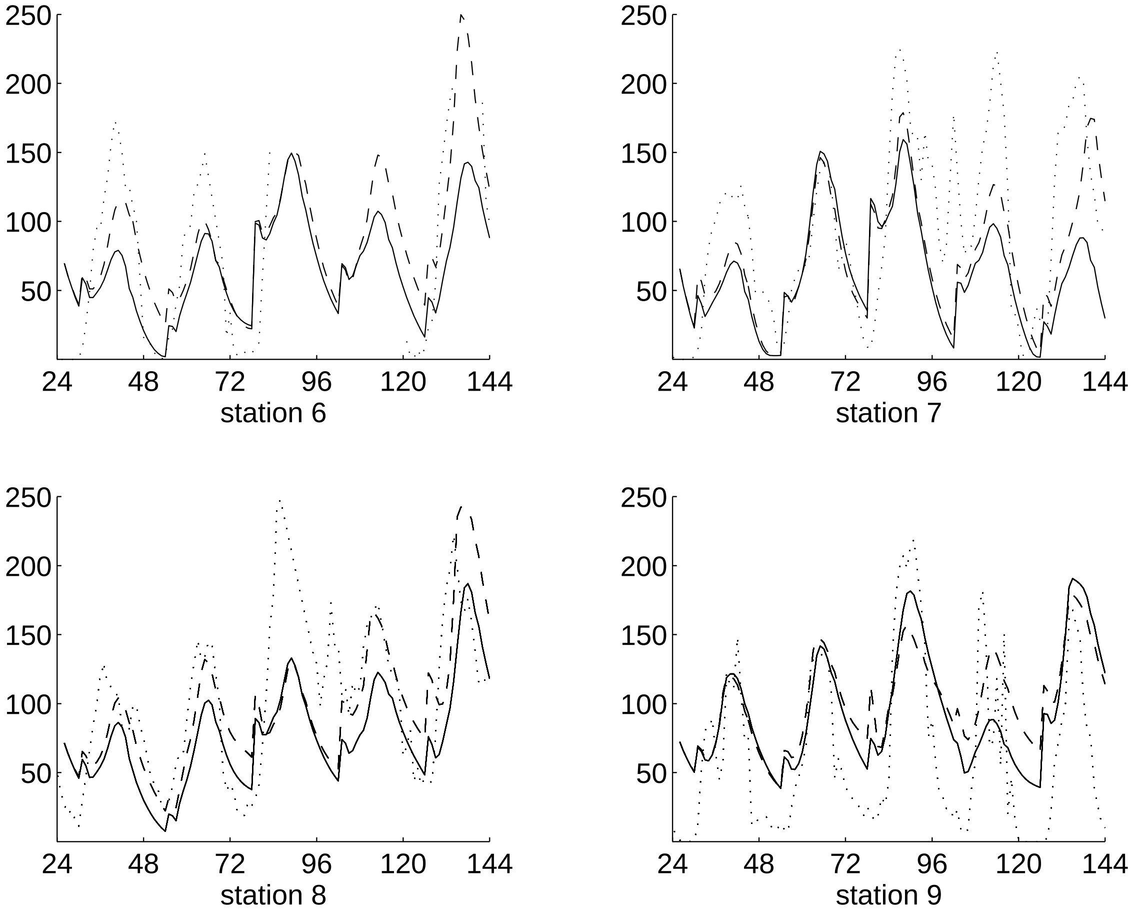


Figure 11: Observed (dotted) and modeled concentrations using 1 grid level (solid) and 4 grid levels (dashed) $\mu g\, m^{-3}$

- The model seems to be sensitive for meteorological input. Because of uncertainty about the wind fields (and how they are used in the model) combined with uncertainty in other meteorological parameters, the comparisons of modeled concentrations with observed concentrations in Section 5 are only indicative. The same uncertainty holds for the summer smog episode, see Section 6, even though in that case the model computations are in good agreement with the measurements.

- It has been shown that grid refinement did not result in a (significant) better agreement between observed and modeled concentrations in all cases. A reason for this might be that the improved spatial resolution has not been used in an optimal way. More attention should be paid to bringing the spatial resolution in the description of other (atmospheric) processes in line with the spatial resolution of the fine grids, for example, by using emission inventories with the same resolution as the adaptive grid.

ACKNOWLEDGEMENT
I would like to thank Christoph Kessler for taking the time to discuss some results of model runs and for doing some useful suggestions.

1. D.C. Arney and J.E. Flaherty. An adaptive local mesh refinement method for time-dependent partial differential equations. *Appl. Numer. Math.*, 5: 257 – 274, 1989.
2. M.J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82: 64 – 84, 1989.
3. M.J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53: 484 – 512, 1984.
4. J.G. Blom, R.A. Trompert, and J.G. Verwer. VLUGR2: A vectorizable adaptive grid solver for PDEs in 2D. Report NM–R9306, CWI, Amsterdam, 1993.
5. W.D. Gropp. A test of moving mesh refinement for 2D-scalar hyperbolic problems. *SIAM J. Sci. Statist. Comput.*, 1: 191 – 197, 1980.
6. W.D. Gropp. Local uniform mesh refinement on vector and parallel processors. In P. Deuflhard and B. Engquist, editors, *Large–Scale Scientific Computing*, Birkhäuser Series Progress in Scientific Computing 7, pages 349 – 367. Birkhäuser, Basel, 1987.
7. W.D. Gropp. Local uniform mesh refinement with moving grids. *SIAM J. Sci. Statist. Comput.*, 8: 292 – 304, 1987.
8. M. van Loon. Numerical smog prediction I: The physical and chemical model. Report NM–R9411, CWI, Amsterdam, 1994.
9. R.A. Trompert and J.G. Verwer. A static–regridding method for two–dimensional parabolic partial differential equations. *Appl. Numer. Math.*, 8: 65 – 90, 1991.
10. R.A. Trompert and J.G. Verwer. Analysis of the implicit Euler local uniform grid refinement method. *SIAM J. Sci. Comput.*, 14: 259 – 278, 1993.
11. J.G. Verwer. Gauss-Seidel iteration for stiff ODEs from chemical kinetics. *SIAM J. Sci. Comput.*, 15: 1243 – 1250, 1994.
12. J.G. Verwer and D. Simpson. Explicit methods for stiff ODEs from atmospheric chemistry. Report NM–R9409, CWI, Amsterdam, 1994. to appear in Appl. Numer. Math.

Appendix A. Datastructure for grid refinement

A.1 approach

The way we look at a grid at a certain level is in fact not different from the way we treat a standard rectangular uniform $M \times N$ grid where each node is identified by its row- and column number. Although the grid cells are numbered from 1 to $M \times N$ instead of using a double index, information about the row- and column number of each cell is available. The row- and column number of a certain fine grid cell are equal to the row- and column number the same cell would have if it was part of a uniform fine $M_{fine} \times N_{fine}$ grid over the whole model domain, where rows and columns are numbered from 0 to $M_{fine} - 1$ and $N_{fine} - 1$, respectively. In Figure 12 an example is given of a rectangular 6 × 6 base grid (dashed lines) with one level of refinement, consisting of 40 cells. The bold numbers indicate the cell numbers in the fine grid, the other numbers are coarse grid cell numbers. The idea is to store the solution values row-wise. For each grid level the number of really existing rows is specified (6 for the fine grid in Figure 12) and indicate for each row where it starts. In addition, for each cell the indices of the cells directly above and below it are specified. Also for each grid cell its column number has to be specified. With all this information, discretizations of differential equations on the fine grid can be implemented in quite a simple way. Below, a specification of all (integer) arrays used to describe the grid structure, is given.

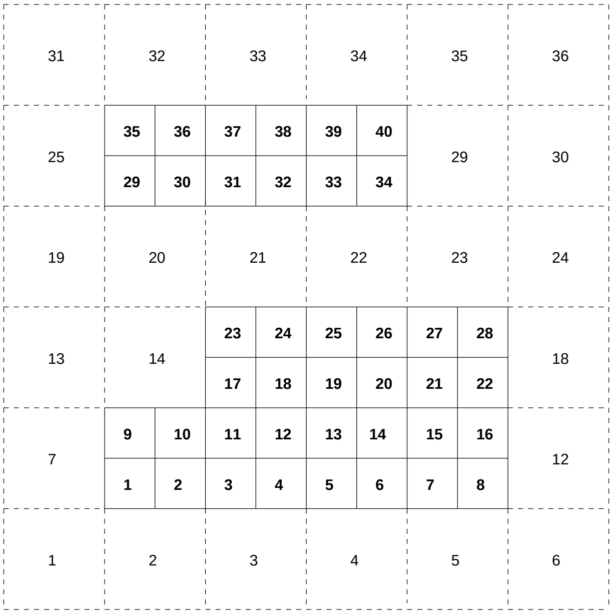


Figure 12: Example of a two level grid structure

A.2 the integer array(s)

The grid structure for each grid level is contained in one large integer array ISTRUC consisting of the following arrays:

- LROW(0:LROW(0)+1)
  LROW(0): the number of rows in the grid
  LROW(1:NROWS): pointers to the start of each row in the grid
  LROW(NROWS+1): NPTS + 1, the total number of grid cells + 1

- IROW(NROWS): the row number of each row in a virtual rectangle, corresponding to its θ-coordinate,
- ICOL(NPTS): the column number of a grid cell in the virtual rectangle, corresponding to its ϕ-coordinate
- IPREV(NPTS): pointers to the underlying coarse grid cells; filled with zeros for the base grid,
- LBND(0:LBND(0)+LBND(1)+1)
  LBND(0): NFBPTS, the total number of physical boundary cells in the actual grid level
  LBND(1): NIBPTS, the total number of internal boundary cells in the actual grid level
  LBND(2:NFBPTS+1): pointers to the physical boundary cells in the grid
  LBND(NFBPTS+2:NFBPTS+NIBPTS): pointers to the internal boundary cells in the grid,
- LABOVE(0:NPTS): pointers to the node directly above each grid cell, zero otherwise.
  LABOVE(0) is zero, which makes recursive use of LABOVE possible,
- LBELOW(0:NPTS): identical to LABOVE but pointing to the cell directly below,
- INEXT(NPTS): if a cell is refined the corresponding entry of INEXT points to the lower left of the four created fine grid cell on the next level. Otherwise the pointer is set to zero. The pointers are only set when actually creating a next finer grid level, so they are unknown when integrating on the present grid level.

For each of the arrays listed above, we will give (some) values for the fine grid in Figure 12

- LROW[0] = 6; LROW[1..6] = [1,9,17,23,29,35,41]
- IROW[1..6] = [2,3,4,5,8,9]
- ICOL[1..8] = [2,3,4,5,6,7,8,9]
- IPREV[1] = 8; IPREV[39]=28
- LBND[0,1] = [0,30]; LBND[i+1] = i, $i = 1 \ldots 10$
- LABOVE[0] = 0; LABOVE[i] = $i + 8$, $i = 1 \ldots 8$; LABOVE[9]=0
- LBELOW[0..8] = 0; LBELOW[i] = $i - 8$, $i = 9 \ldots 16$
- INEXT[i] = 0, $i = 1 \ldots 40$
  N.B. for the base grid we have INEXT[7..12] = [0,1,3,4,7,0]