WORKING PAPER

Numerical Approximation of the Transport Equation: Comparison of Five Positive Definite Algorithms

Jerzy Bartnicki
Krzysztof Olendrzynski
Krzysztof Abert
Peter Seibert
Bogdan Morariu

February 1990
WP-90-10

IIASA
International Institute
for Applied Systems Analysis

Numerical Approximation of the Transport Equation: Comparison of Five Positive Definite Algorithms

Jerzy Bartnicki
Krzysztof Olendrzynski
Krzysztof Abert
Peter Seibert
Bogdan Morariu

February 1990
WP-90-10

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

Authors

Drs. K. Olendrzynski and K. Abert are from the Institute for Meteorology and Water Management, Warsaw, Poland.

Drs. P. Seibert and B. Morariu are from the Central Institute for Meteorology and Geodynamics, Vienna, Austria.

Dr. J. Bartnicki is from the International Institute for Applied Systems Analysis, Laxenburg, Austria.

Foreword

The RAINS model will be used to develop and assess international control strategies to reduce emissions of acidifying pollutants. These strategies will involve the expenditure of large sums of money; it is important, therefore, to assess the effect of uncertainties in the model on its results. An important component of the RAINS model is its atmospheric transport component; this paper reports the results of examining several algorithms for solution of the atmospheric transport equation. It also represents a joint effort between IIASA scientists and those in the Institute of Meteorology and Water Management in Warsaw and Central Institute for Meteorology and Geodynamics in Vienna.

Roderick W. Shaw
Leader
Transboundary Air Pollution Project

Bo R. Döös
Leader
Environment Program

Acknowledgements

The authors would like to thank Joseph Alcamo and Roderick Shaw from IIASA for their helpful discussions, comments and review of this manuscript. We are also indebted to Maximilian Posch, who developed 3-D graphical software used in this paper. Finally, the first three authors wish to acknowledge Stefan Reichhart of the Institute of Meteorology and Water Management in Warsaw who supported a part of the study in Poland.

Table of Contents

Numerical Approximation of the Transport Equation: Comparison of Five Positive Definite Algorithms

Jerzy Bartnicki, Krzysztof Olendrzynski, Krzysztof Abert, Peter Seibert and Bogdan Morariu

1. INTRODUCTION

The RAINS (Regional Acidification INformation and Simulation) model developed at IIASA (Alcamo *et. al.*, 1988) is an important policy-oriented tool for analysis of acidification in Europe. According to different sulfur or nitrogen emission scenarios, the RAINS model can compute sulfur and nitrogen depositions in Europe and its impact on soil, forest and lakes. There are three basic parts of RAINS: (1) emission submodels describing sulfur and nitrogen input to atmosphere, (2) atmospheric transport modules relating emissions to concentrations and depositions and (3) impact submodels. An important additional part, is the optimization algorithm which allows the user to find an optimal emission reduction strategy for assumed distribution of sulfur in Europe (Batterman et al. 1988; Batterman, 1988).

The atmospheric part of the RAINS is represented by so-called source-receptor matrices (SR) computed by the long-range transport air pollutant models (LRTAP). The SR matrices used in RAINS were computed by the EMEP MSC-W (Meteorological Synthesising Centre - West) model (Eliassen and Saltbones, 1983; Eliassen *et. al.*, 1988) for a four year period starting from 1978. The nitrogen SR matrices were computed for the 9 year period: 1978 to 1986 by the climatological model developed at IIASA (Bartnicki and Alcamo, 1989). Recently, a nitrogen SR matrix s been included, based on computed by the new version of the EMEP MSC-W model (Eliassen *et. al.*, 1988), but only for one meteorological year - 1985.

When the RAINS model is used by policymakers, the possible financial consequences of suggested emission strategies can be very large. Therefore it is quite natural to ask about the uncertainty of the RAINS computations. The answer to such question is not easy and depends on the uncertainty of the three RAINS components: emissions, atmospheric transport and impact. Most of the work, concerning uncertainty, has been done at IIASA on the atmospheric part of the RAINS. The general framework for the uncertainty analysis of the LRTAP model was developed by Alcamo and Bartnicki (1985,1987). In this framework, all sources of uncertainty were divided into five categories: model structure, model parameters, model forcing functions, initial state and model operation. Among many different sources of uncertainty which belong to the last group - model operation, the possible errors caused by a numerical algorithm applied to the transport equation may be quite significant. Unfortunately, the general analytical solution for the transport equation does not exist. In this case, reduction of uncertainty due to the approximate numerical solution can be achieved by the use of more accurate algorithms.

There is a number of existing numerical methods for the solution of the transport equation; however not all of them can be applied to certain classes of the LRTAP models. For example, when modeling photo-oxidants or mercury transport in the atmosphere, negative values created by some of the numerical schemes are not permitted. It means that only algorithms which do not produce false negative concentrations - so-called positive definite algorithms - can be used. Having in mind the active role of the IIASA

TAP project in modeling both photo-oxidants and heavy metals, the problem of the accuracy of such positive definite algorithms can be an important issue for the future.

Until now we have discussed the applications of positive definite algorithms to LRTAP models. These schemes are also very important for numerical solutions of the transport equations in the mesoscale. For example, some of the algorithms presented in this paper were used in an air pollution model for the Linz Region in Austria (Pechinger *et al.*, 1987).

Each scheme or method for the numerical solution of the transport equation may belong to one of the three categories: (1) Lagrangian, (2) Semi-Lagrangian and (3) Eulerian. In the Lagrangian category, two approaches are especially interesting: trajectory (Eliassen and Saltbones, 1983) and Monte Carlo (Zannetti, 1981). Trajectory models are relatively efficient for application in the LRTAP modeling, but only for single layer models, and therefore we eliminated the trajectory method from the comparison. A serious drawback of the Monte Carlo Method is the problem of incorporating chemical reactions into a huge number of small single particles. Any solution requires a long computer time, and in addition, large computer memory to run the programs. Also none of the authors have enough experience with the method, so we finally decided not to take it into account in our comparison. Thus, in this study, we have concentrated on five schemes from the second and third category. The question why we have chosen these particular schemes will be answered in Section 3 of the paper together with detailed description of each algorithm.

In the next Section we will briefly review the transport equation which is used both in long range (1000-10000 km) and mesoscale (< 500 km) models.

2. ADVECTIVE PART OF THE TRANSPORT EQUATION

A general equation which describes transport of pollutants in the atmosphere has to take into account several important physical processes such as: natural and anthropogenic emissions, horizontal and vertical motion of the air, turbulent diffusion, chemical reactions and both wet and dry deposition of pollutants. For most of the transport models, such mathematical formulation has the form of a three-dimensional advection-diffusion equation

$$\frac{\partial c}{\partial t} + \sum_{i=1}^{3} u_i \frac{\partial c}{\partial x_i} - \sum_{i=1}^{3}\sum_{j=1}^{3} \frac{\partial}{\partial x_i} K_{i,j} \frac{\partial c}{\partial x_j} = R(c) + Q \qquad (1)$$

where $c = c(x,y,z,t)$ is the concentration of pollutant, $u_i = (u,v,w)$ is the velocity field, $K_{i,j}$ is the diffusivity tensor, R is an operator describing chemical reactions and $Q = Q(x,y,z,t)$ is responsible for pollutant sources and sinks. It should be stressed here that there is one basic assumption which permits us to use equation (1). Namely, it is assumed that all pollutants involved in this equation are passive in the sense that they cannot change meteorological fields, like for example, precipitation or transport wind. This assumption is, more or less, reasonable in the case of long-range transport models but may be incorrect for subregional transport where concentrations of pollutants may be high enough to influence precipitation or the radiation balance.

The right side of equation (1) depends on the chemical scheme used in particular transport model and on the parameterization of pollutant sources and sinks. Generally, it can be a nonlinear function of the concentration c. However, compared to the terms on the left side of this equation, numerical problems associated with the right side of this equation are relatively easy to solve. Therefore, for the remainder of this paper we will neglect the right side of equation (1).

If in addition, diffusivity tensor $K_{i,j}$ is isotropic and uniform in the horizontal direction, which is the case for almost all transport models, equation (1) can be rewritten as

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} + v\frac{\partial c}{\partial y} + w\frac{\partial c}{\partial z} - (\frac{\partial}{\partial x}K_h\frac{\partial c}{\partial x} + \frac{\partial}{\partial y}K_h\frac{\partial c}{\partial y} + \frac{\partial}{\partial z}K_v\frac{\partial c}{\partial z}) = 0 \qquad (2)$$

where K_h and K_v are horizontal and vertical components of the diffusivity tensor, respectively.

There are three parts on the left side of equation (2): the rate of change of the concentration represented by term $\frac{\partial c}{\partial t}$, the advective transport represented by terms with the velocity components, and diffusion described by terms which include K_h and K_v.

From the numerical point of view, diffusion is much easier to simulate than advection (Zlatev, 1988). In addition, scale analysis (Bartnicki, 1983) and numerical experiments (Christensen and Prahm, 1976) indicate that horizontal diffusion in LRTAP models has much less influence on the results than advection. In some LRTAP models (e.g. Eliassen et al., 1988), the diffusion part is even neglected. From the physical point of view, both horizontal and vertical diffusion are quite important for regional transport models. However, also in this case, crucial numerical difficulties arise in the advective part of the transport equation. This is the main reason why, in the following chapters of the paper, we will concentrate on the advective part of the transport equation; The three-dimensional form of this part of the transport equation is

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} + v\frac{\partial c}{\partial y} + w\frac{\partial c}{\partial z} = 0 \qquad (3)$$

In the next Section, we describe the five algorithms for solving equation (3) which have been chosen for comparison.

3. NUMERICAL ALGORITHMS COMPARED IN THE STUDY

A critical review of the numerical methods for the solution of the advection-diffusion equation is outside the scope of this paper. The reader can find such a review Chock and Dunker (1983) and Chock (1985), for example. In this study we concentrate on following five Eulerian methods for the numerical solution of the transport equation:

(1) Positive Definite Pseudospectral Method (PDPS).

(2) Positive Definite Spline Method (PDS).

(3) Positive Definite Galerkin Method (PDG).

(4) Antidiffusion Correction Method (AC).

(5) Flux Correction Method (FC).

There were several reasons for selecting these algorithms:

- They do not produce negative concentrations. This is particularly important in transport models with complicated nonlinear chemistry or re-emission of pollutants.
- They are relatively simple and comprehensive, both in mathematical formulation and computer applications. All necessary programs can be run on IBM compatible PC microcomputers.
- These algorithms give relatively accurate numerical solutions, and can be applied to a wide class of problems, not only to air pollution transport.

As mentioned above, all the considered schemes eliminate negative values from the concentration field. In order to protect the solution from negative concentrations, two different techniques are used: filtering and flux correction. Another technique - antidiffusive correction was applied to the "upwind" finite difference scheme which does not produce negative values but does create a substantial amount of numerical diffusion. All of these techniques will be described in this Section together with particular schemes to which they were applied.

3.1. Positive Definite Pseudospectral Method (PDPS)

A PDPS method developed by Bartnicki (1986; 1989) involves two basic steps at each time interval when solving the advective part of equation (3):

(1) The pseudospectral method is applied to the advection equation at time t, and a solution is achieved. This solution may contain negative concentration values.

(2) The filtering procedure removes all negative values of the concentration and computes a new solution at time $t+\Delta t$.

The pseudospectral method can be represented by operator $\hat{P}$ which, when applied to the discrete concentration field c^m at time $m\Delta t$, produces the concentration $\tilde{c}^{m+1}$ at time $(m+1)\Delta t$:

$$\tilde{c}^{m+1} = \hat{P}(c^m) \tag{4}$$

The concentration $\tilde{c}^{m+1}$ may still include negative values. The filtering procedure can be represented by operator $\hat{F}$ which transforms $\tilde{c}^{m+1}$ to c^{m+1} containing only non-negative values :

$$c^{m+1} = \hat{F}(\tilde{c}^{m+1}) \tag{5}$$

Thus, the positive definite pseudospectral method can be defined as:

$$c^{m+1} = \hat{F}.\hat{P}(c^m) \tag{6}$$

A multidimensional form of the advection equation will be used to describe the PDPS algorithm:

$$\frac{\partial c}{\partial t} + \sum_{j=1}^{N} u_j \frac{\partial c}{\partial x_j} = 0 \tag{7}$$

where $c = c(\boldsymbol{x},t)$ is the concentration (which can be arbitrarily scalar) assumed to be non-negative, $u_j = u_j(\boldsymbol{x},t)$ is the j-th velocity component, $(\boldsymbol{x},t) = (x_1,...,x_N,t)$ is the space and time coordinates.

In order to find the numerical solution of the advection equation (7) it is necessary, at first, to introduce a discrete grid system. Let $c^m = c(\boldsymbol{x},m\Delta t)$ be the concentration field with periodic boundary conditions at time $m\Delta t$. We are looking for the concentration $c^{m+1} = c(\boldsymbol{x},(m+1)\Delta t)$ at time $(m+1)\Delta t$ in the uniform mesh of size $L_1 \times L_2 \times ... \times L_N$ where the location of the mesh points is given by:

$$x_j = l_j \Delta x_j; \quad l_j = 0,1,2,\ldots, L_j - 1 \tag{8}$$

where

$$\Delta x_j = \frac{2\pi}{L_j} \tag{9}$$

for any $j = 1,2,\ldots,N$.

The grid system defined above will be used to explain the PDPS method. We start this description with the pseudospectral operator $\hat{P}$.

3.1.1. Pseudospectral Solution

The pseudospectral approach developed by Gazdag (1973) has been chosen as the operator $\hat{P}$, because of its high accuracy. The principle of Gazdag's method is to approximate the time derivatives by a truncated Taylor series, and then replace the time derivatives by the space derivative terms which are computed using the spectral method. Mathematically, the method can be described as follows: Assuming that we know the concentration c^m at time $m\Delta t$, the concentration c^{m+1} at the next time step $(m+1)\Delta t$ can be approximated by the truncated Taylor series

$$c^{m+1} = c^m + \frac{\partial c^m}{\partial t}\Delta t + \frac{\partial^2 c^m}{\partial t^2}\frac{\Delta t^2}{2!} + \frac{\partial^3 c^m}{\partial t^3}\frac{\Delta t^3}{3!} + \cdots + \frac{\partial^p c^m}{\partial t^p}\frac{\Delta t^p}{p!} \tag{10}$$

Following Gazdag (1973), the time derivatives of c can be expressed in terms of the space derivatives of c and u_j by making use of equation (7):

$$\frac{\partial c}{\partial t} = -\sum_{j=1}^{N} u_j \frac{\partial c}{\partial x_j} \tag{11}$$

$$\frac{\partial^2 c}{\partial t^2} = -\sum_{j=1}^{N}\left\{\frac{\partial u_j}{\partial t}\frac{\partial c}{\partial x_j} + u_j \frac{\partial}{\partial x_j}\frac{\partial c}{\partial t}\right\} \tag{12}$$

$$\vdots \qquad \vdots$$

$$\frac{\partial^{l+1} c}{\partial t^{l+1}} = -\sum_{j=1}^{N}\sum_{r=0}^{l}\binom{l}{r}\frac{\partial^r u_j}{\partial t^r}\frac{\partial}{\partial x_j}\left[\frac{\partial^{l-r} c}{\partial t^{l-r}}\right] \tag{13}$$

The superscript m has been omitted in the above equations for convenience. Equations (11-13) show how to compute any order time derivative of c from the lower order time derivatives of u_j and c. The first order time derivative of c can be computed directly from the basic advection equation. It remains only to compute space derivatives of c. This is done with the spectral method. Denoting the set of all grid points (equations 8-9) by R, the finite Fourier transform C of c can be written as

$$C(\boldsymbol{k},t) = \frac{1}{L_1 L_2 \cdots L_N}\sum_{\boldsymbol{x}\in R} c(\boldsymbol{x},t)\, e^{-i\boldsymbol{k}\cdot\boldsymbol{x}} \tag{14}$$

where $i = \sqrt{-1}$ and $\boldsymbol{k}$ is the wave vector

$$\boldsymbol{k} = (k_1,...,k_j,...,k_N) \tag{15}$$

whose components assume integer values within the limits.

$$-K_j < k_j \leq K_j; \quad K_j = \frac{L_j}{2} \tag{16}$$

From $C(\boldsymbol{k},t)$ the partial derivatives of $c(\boldsymbol{x},t)$ can be computed as

$$\frac{\partial c(\boldsymbol{x},t)}{\partial x_j} = \sum_{|k_j|<K_j} ik_j C(\boldsymbol{k},t)\, e^{i\boldsymbol{k}\cdot\boldsymbol{x}} \tag{17}$$

The numerical computation of the space derivatives described by equations (14-17) can be carried out sufficiently fast by the use of the numerical Fast Fourier Transform (FFT, Cooley and Tukey, 1965). According to Gazdag (1973) such computation gives very accurate results which prompted him to call it the Accurate Space Derivative (ASD) method. The stability condition of the ASD method depends on the order of expansion p in equation (10) and has the following form:

$$\left| \sum_{l=0}^{p} \frac{(i\varphi)^l}{l!} \right| \leq 1 \tag{18}$$

where

$$\varphi = -\boldsymbol{k}\cdot\boldsymbol{v}\Delta t \tag{19}$$

For example, stability condition (18) is satisfied by ASD schemes of orders 3,4,7 and 8. In the case of third order expansion, the Courant number associated with the pseudospectral method is 0.5. It should be mentioned here that performance of the pseudospectral method does not depend too much on the Courant number. Some experiments (Gazdag, 1973; Bartnicki 1983) indicate that differences in analytical and numerical solutions, in rotational test described in Section 4, are relatively small for the wide range of the Courant numbers: 0.1 - 2.0. Thus, the method gives accurate results also for the cases when the time step exceeds the stability condition.

In this paragraph we described the first operator in the PDPS method. In the next, we will describe the second operator - $\hat{\boldsymbol{F}}$ which represents the filtering procedure.

3.1.2. Filtering procedure

The concept of the filtering procedure (Bartnicki, 1989), represented by the operator $\hat{\boldsymbol{F}}$ can be explained as follows: Let c_j be the concentration in the j-th point in the grid system of arbitrary dimension consisting of N points ($j = 1,...,N$). If all c_j values are non-negative the filter does not change them. Now, let us assume that the concentration field has N_1 positive values ($c_j > 0$), N_2 zero values ($c_j = 0$) and N_3 negative values ($c_j < 0$). Obviously

$$N_1 + N_2 + N_3 = N \tag{20}$$

We assume that

$$M_1 > M_3 \tag{21}$$

where

$$M_1 = \sum_{\substack{j=1 \\ c_j>0}}^{N} c_j \tag{22a}$$

is the "positive" mass and

$$M_3 = \left| \sum_{\substack{j=1 \\ c_j<0}}^{N} c_j \right| \tag{22b}$$

is the "negative" mass. With the above assumptions the filtering procedure is defined by the following algorithm:

1. Compute the negative mass M_3 and check if it is greater than zero. If not, stop.
2. Compute the number of positive concentrations N_1.
3. Check the sign of the concentration c_j for $j = 1,...,N$
   (a) If $c_j > 0$, subtract the negative mass divided by the number of positive concentrations: $c_j := c_j - \frac{M_3}{N_1}$
   (b) If $c_j = 0$, do nothing.
   (c) If $c_j < 0$, set it to zero: $c_j := 0$.
4. Go to 1.

The algorithm presented above is convergent (Bartnicki, 1989) and typically only two iterations are necessary to remove all negative values.

A different filtering procedure will be implemented in the Positive Definite Spline method, described below.

3.2. Positive Definite Spline Method (PDS)

Cubic splines have often been used in the numerical solutions of the partial differential equations (e.g. Price and MacPherson, 1973; Purnell, 1976; Pepper *et. al.*, 1979). Solutions obtained by the spline method contain nonphysical negative values due to the oscillations of the spline function. In contrast to the above mentioned algorithms, the method presented here completely eliminates unwanted negative concentrations in the second step. In the first step, cubic splines are used to obtain a preliminary solution, and then negative values in the preliminary solution are eliminated by a local filter.

To describe the method we will use N-dimensional advection equation (7):

$$\frac{\partial c}{\partial t} + \sum_{j=1}^{N} u_j \frac{\partial c}{\partial x_j} = 0$$

Solution of this equation is represented by an operator $\hat{U}$ relating concentrations at the time $m\,\Delta t$ and $(m+1)\,\Delta t$ (similar to operator $\tilde{P}$ in equation (4)):

$$c^{m+1} = \hat{U}\, c^m \tag{23}$$

where $c^m = c(m\Delta t,\boldsymbol{x})$ and $c^{m+1} = c((m+1)\Delta t,\boldsymbol{x})$. If there is more than one dimension ($N>1$) in equation (7) we use the time splitting method to compute operator $\hat{U}$. Let us assume that the solutions of the following equations

$$\frac{\partial c}{\partial t} + u_j \frac{\partial c}{\partial x_j} = 0; \quad j = 1,\cdots,N \tag{24}$$

are described in $j-th$ dimension by the associated operators $\hat{U}_j$

$$\tilde{c}_j^{m+1} = \hat{U}_j\, \tilde{c}_j^m \tag{25}$$

Operators $\hat{U}_j$ will be called partial solutions* of equation (7) which approximate operator $\hat{U}$:

$$\hat{U} = \hat{U}_N \cdot \hat{U}_{N-1} \cdots \cdots \hat{U}_1 \tag{26}$$

According to an exponential formulation of the time splitting method, equation (23) is equivalent to the following sequence of equations:

$$\begin{aligned}
&\tilde{c}_1^m = c^m \\
&\tilde{c}_1^{m+1} = \hat{U}\tilde{c}_1^m \\
&\tilde{c}_2^m = \tilde{c}_1^{m+1} \\
&\quad\cdot \qquad\cdot \\
&\quad\cdot \qquad\cdot \\
&\quad\cdot \qquad\cdot \\
&\tilde{c}_N^{m+1} = \hat{U}_N\, \tilde{c}_N^m \\
&c^{m+1} = \tilde{c}_N^{m+1}
\end{aligned} \tag{27}$$

Computation of the partial numerical solution in $j-th$ dimension defined by equation (25):

$$\tilde{c}_j^{m+1} = \hat{U}_j\, \tilde{c}_j^m$$

involves two steps:

(1) Upstream spline method represented by operator $\hat{S}$ is applied to obtain a preliminary solution:

* Generally, the velocity fields $(0,\cdots,u_j,\cdots,0)$ are divergent for each direction j, therefore no physical significance should be given to any partial solution of this equation. Also, mass conservation does not hold for the partial solutions.

$$\tilde{\tilde{c}}_j^{m+1} = \hat{S}_j \, \tilde{c}_j^m \tag{28}$$

(2) Negative values in the preliminary solution are eliminated by a local filter $\hat{F}$:

$$\tilde{c}_j^{m+1} = \hat{F} \, \tilde{\tilde{c}}_j^{m+1} \tag{29}$$

Using equations (28) and (29) operator $\hat{U}$ may be rewritten in the following form:

$$\hat{U} = \prod_{j=1}^{N} \hat{U}_j = \prod_{j=1}^{N} (\hat{F}_j \cdot \hat{S}_j) \tag{30}$$

In the next paragraph we describe the first operator in equation (31) - $\hat{S}$.

3.2.1. Upstream Spline method

The upstream spline interpolation scheme is used in the first step of the method to solve the advection part of the transport equation. We consider an air parcel moving along the wind whose trajectory $t \rightarrow \vec{r}(t)$ is described by the following equation:

$$\frac{d\vec{r}}{dt} = \vec{v} \tag{31}$$

where $\vec{v} = (u_1, u_2, \cdots, u_N)$ is the wind field and $\vec{r} = (x_1(t), x_2(t), \cdots, x_N(t))$ is the position of the air parcel. Let us consider equation (7) again:

$$\frac{\partial c}{\partial t} + \sum_{j=1}^{N} u_j \frac{\partial c}{\partial x_j} = 0$$

For solutions of the above equation, the total time derivative:

$$\frac{d}{dt}(c(\vec{r}(t))) = \frac{\partial c}{\partial t} + \sum_{j=1}^{N} v_j \frac{\partial c}{\partial x_j} \frac{\partial x_j}{\partial t} = \frac{\partial c}{\partial t} + \sum_{j=1}^{N} u_j \frac{\partial c}{\partial x_j} \tag{32}$$

and also its time integral are equal to zero:

$$\int_t^{t+\Delta t} \frac{d}{dt}(c(\vec{r}(t)))\,dt = c(\vec{r}(t+\Delta t), t+\Delta t) - c(\vec{r}(t), t) = 0 \tag{33}$$

The last identity can be rewritten as:

$$c(\vec{r}(t+\Delta t), t+\Delta t) = c(\vec{r}(t), t) \tag{34}$$

which is the basic relation for the upstream spline interpolation technique. This relation is true both for divergent and non-divergent velocity fields.

In order to use equation (34) in a discrete grid system, it is necessary to know the values of concentration between grid points. These values can be computed by the spline interpolation technique in the following way. Let us consider the grid system $\{x_i^{l_i}:(x_i^{l_i}, l_i = 1,\cdots,L_i), i = 1,\cdots,N)\}$. From now, we only take into account one component of this system, say the $j-th$, because computations for the remaining components are identical. In this case, calculation of the upstream concentrations needs $L_1 \times L_2 \times \cdots \times L_{j-1} \times L_{j+1} \times \cdots \times L_N$ splines in the grid $\{x_j^{l_j}, l_j = j,\cdots,L_j\}$. We describe only one example of such spline. For simplicity, we drop the subscript "j" and consider only a one-dimensional concentration field $c(x^i)$ in the one-dimensional grid system $\{x^i; i = j,\cdots,L\}$.

The spline function S is defined by the following conditions (Pielke, 1984):

(1) On each interval (x^{i-1}, x^i) S is a cubic polynomial:

$$S(x) = ax^3 + bx^2 + cx + d \tag{35}$$

where a,b,c and d are the constants to be determined.

(2) The values of the function S at the grid points are equal to the concentrations at these points:

$$S(x^i) = c(x^i); \quad i=1,...,N \tag{36}$$

(3) The function S and its first and second order derivatives S' and S'' are continuous.

These three conditions plus the boundary conditions (linear relations between the derivatives of the spline on a pair of points at the boundary) completely define the spline. This formulation leads to a tridiagonal linear system of equations which can be solved using the sweep method (Roache, 1976). The solution of the system, in the case of N points, requires 2N operations.

Boundary Conditions . The boundary conditions that we use in the above algorithm are of the following type:

$$\begin{cases} S'(x^1) = C^1 \\ S'(x^N) = C^N \end{cases} \tag{37}$$

where the constants C^1 and C^N are determined by linear interpolation using the concentrations at two boundary points. We distinguish two cases:

(1) *Outflow from the model area.* To formulate boundary conditions in this case we use two points within the domain; for example, the derivative at point N is:

$$S'(x^N) = \frac{c(x^N) - c(x^{N-1})}{x^N - x^{N-1}} \tag{38a}$$

(2) *Inflow to the model area.* In this case, a supplementary point is assumed outside computational domain with zero concentration (clear air inflow) which yields

$$S'(x^N) = \frac{-c(x^N)}{x^N - x^{N-1}} \tag{38b}$$

In order to find the location x from which the concentration is advected to the grid point x^i after time step Δt, we have to integrate equation (31). If we assume a constant velocity, this yields

$$x = x^i - u^i \Delta t, \tag{39}$$

or if we assume a linear behavior for the velocity field between grid points we get instead

$$x = x^i - \frac{u^i}{\frac{\partial u^i}{\partial x}}(1 - e^{\frac{\partial u^i}{\partial x}\Delta t}) \tag{40}$$

The term $\frac{\partial u^i}{\partial x}$ in equation (40) is the upstream derivative. It should be mentioned that for both algorithms (39) and (40), the total mass is not fully conserved during the integration. The main factors responsible for this situation are:

1. The total mass calculated by the spline interpolation is not equal to the mass computed by the linear interpolation,
2. Unwanted fluxes through the boundaries can appear due to the spline oscillations,
3. The time splitting is an approximate method,
4. The coordinates of the upstream point are computed with the limited accuracy.

Equations (39) and (40) have been recently tested at the Institute of Meteorology in Vienna for application to the wind field with strongly variable components. The second algorithm represented by equation (40) showed a significant reduction of the total mass deficit - only 0.1%, in comparison to the first method represented by expression (39) with a total mass deficit of 40%. Further improvements may be achieved by a better treatment of the boundary conditions (factors 1 and 2).

Both algorithms are equivalent in the case of the wind field used for advection tests described in this paper (Section 4), because the velocity components are constant along the lines parallel to x- and y-axis. Since, the first method (equation 39) is 1.3 times faster, we have used it in the advection test.

Concentrations computed by means of equations (39) or (40) are not free of negative values. In the next step, these negative values are eliminated by a local filtering procedure.

3.2.2. Local Filtering Procedure

A common property is overshooting, i.e. creating unrealistic negative values associated with oscillations of the length $2\Delta x$. Ludes and Ulrich (1988) (see also Ludes, 1987) developed a method that removes not only the negative, but also the positive part of the wiggles. We found the method to be unstable with the cone-shaped initial condition, causing the concentration field to break into a chaotic distribution after a few rotations. This is caused by adding positive residual mass resulting from the removal of a pair of spurious positive and negative values at a neighboring grid point. Therefore, the original scheme was modified by adding up all positive residuals and distributing them globally in the end. According to Seibert and Morariu (1990), the modified Ludes and Ulrich filter can

then be described as follows (V denotes the volume of a grid cell, N the number of grid points in the x-direction):

[1] Set i=1, p=1, Δm^+; go to [2].

[2] If $c_i < 0$, set $c_{i+p} = c_{i+p} + c_i \frac{V_i}{V_{i+p}}$; go to [3].

[3] Set i=i+p; go to [4].

[4] If c_i<0, go to [5]; else, go to [9].

[5] If $c_{i+p} < c_{i-p}$, go to [9]; else go to [6].

[6] Compute $\Delta m = c_{i-p} V_{i-p} + c_i V_i$; go to [7].

[7] Set $c_{i-p} = 0$ and $c_i = 0$; go to [8].

[8] If $\Delta m < 0$, set $c_{i+p} = c_{i+p} + \frac{\Delta m}{V_{i+p}}$; else, add Δm to "stock": $\Delta m^+ = \Delta m^+ + \Delta m$; go to [9].

[9] If $p = 1$ and $i < N-1$, or if $p = -1$ and $i > 2$, go to [3]; else, if $p = 1$ and $i = N-1$, set $p = -1$ and go to [3]; else, go to [10].

[10] If $c_1 < 0$, set $c_1 = 0$ and $\Delta m^+ = \Delta m^+ + c_1 V_1$.

After completion of the process, advection is computed for the next grid line and concentrations are filtered, starting at [1]. When the whole grid (2- or 3-dimensional) is treated, the "stock" Δm^+ is distributed over the whole domain:

$$c_{ijk} = c_{ijk} + \frac{(c_{ijk})^r V_{ijk}}{\sum_{ijk}(c_{ijk})^r V_{ijk}} \frac{\Delta m^+}{V_{ijk}}$$

We use r=1 for the redistribution of the positive residual mass which imparts to the filter a certain antidiffusive effect. This may be desired for a cone-shaped or delta function initial condition in order to enhance the maximum value; otherwise, if this effect is to be reduced or avoided, a value $0 \leq r < 1$ may be used.

3.3. Positive Definite Galerkin Method (PDG)

A PDG method presented in this paper was implemented by Abert (1989). As in the case of a PDPS method, first Galerkin solution is obtained at each time step, and then filtering procedure described in the first paragraph is applied. Since, the method of Fractional Steps (Yanenko, 1971) will be used to compute the solutions in N-dimensional space, for simplicity we will only use a one-dimensional version of the advection equation in the description of the PDG method,

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} = 0 \tag{41}$$

In order to solve the advection equation (41), we expand concentration c as a linear combination of the basis functions ψ_i:

$$c(x,t) = \sum_{i=1}^{N} a_i(t)\,\psi_i \tag{42}$$

where $a_i(t)$ are the time dependent coefficients of expansion and N is the number of points in the numerical grid. The Galerkin method requires that the error of the solution is orthogonal to each base function

$$\int_D \psi_i R(x,t) = 0 \tag{43}$$

where D is the area of integration and $R(x,t)$ is the error of a numerical approximation of equation (41) by expansion (42). Equation (43) leads to a set of ordinary differential equations involving coefficients $a_i(t)$. So-called Chapeau functions were chosen for expansion (42). With the above assumptions, the resulting set of ordinary differential equations has a tridiagonal form:

$$\begin{aligned} &\frac{da_{i-1}}{dt}\Delta x_- + 2\frac{da_i}{dt}(\Delta x_- + \Delta x_+) + \frac{da_{i+1}}{dt}\Delta x_+ + \\ &+ a_{i+1}(2u_i + u_{i-1}) - a_i(u_{i+1} - u_{i-1}) - a_{i-1}(2u_i + u_{i+1}) = 0 \\ &\text{for } i = 2,3,\cdots,N-1 \end{aligned} \tag{44}$$

where $\Delta x_+ = x_{i+1} - x_i$, $\Delta x_- = x_i - x_{i-1}$ and x_i is the $i-th$ point in the grid system. Open boundary conditions were used in the method which lead to the following equations for the boundary points i=1 and i=N

$$\Delta x_+ \left(\frac{da_2}{dt} + 2\frac{da_1}{dt}\right) + (2u_1 + u_2)(a_2 - a_1) = 0 \tag{45a}$$

$$\Delta x_- \left(\frac{da_{N-1}}{dt} + 2\frac{da_N}{dt}\right) + (2u_N + u_{N-1})(a_N - a_{N-1}) = 0 \tag{45b}$$

The time derivatives in the above equations were approximated by the Crank-Nicholson method (Crank and Nicholson, 1947; Richtmyer and Morton, 1967), and a method of Fractional Steps (Yanenko, 1971) was used to expand the algorithm to two dimensions. The negative values in the solution were eliminated by the global filter (Bartnicki, 1989), described in Section 3.1.2.

The strong advantage of the PDG method is its unconditional stability which means that for any arbitrary Courant number this method is stable. However, for the large Courant numbers the phase error significantly limits the accuracy of the solution. An additional advantage of the method is the possibility of applying it to an irregular grid system, which may be important for some models.

3.4. Antidiffusion Correction Method (AC)

Smolarkiewicz (1983) and Smolarkiewicz and Clark (1986) on the basis of the "upstream" scheme designed and developed a new scheme which is positive definite, efficient and does not contain strong numerical diffusion. For the simplicity we limit our considerations to the one-dimensional case, but the method, in a natural way, may be easily expanded to multidimensional problems. The basic idea of the algorithm is as follows: Let us apply the classical upstream scheme to the one-dimensional advection equation in the flux form:

$$c_i^{n+1} = c_i^n - [F(c_i^n, c_{i+1}^n, u_{i+1/2}^n) - F(c_{i-1}^n, c_i^n, u_{i-1/2}^n)] \tag{46}$$

where $c_i^n = c(n\Delta t, i\Delta x)$ and

$$F(c_i, c_{i+1}, u) = [(u+|u|)c_i + (u-|u|)c_{i+1}] \frac{\Delta t}{2\Delta x} \tag{47}$$

Expanding c_i^{n+1}, c_{i+1}^n, c_{i-1}^n in a second-order Taylor series about the point (t^n, x_i), scheme (46) can be written as:

$$\frac{\partial c}{\partial t}|_i^n = -\frac{\partial}{\partial x}(uc)|_i^n + \frac{\partial}{\partial x}[0.5(|u|\Delta x - \Delta t u^2)\frac{\partial c}{\partial x}]|_i^n \tag{48}$$

The above equation shows that scheme (46) approximates with second-order accuracy the following equation:

$$\frac{\partial c}{\partial t} + \frac{\partial}{\partial x}(uc) = \frac{\partial}{\partial x}(K_d \frac{\partial c}{\partial x}) \tag{49}$$

where: $K_d = 0.5(|u|\Delta x - \Delta t u^2)$. So in the real computational process equation (46) approximates the one-dimensional advection equation with an additional diffusive term. The idea of the scheme is to perform the advective step using scheme (46) and then, in the corrective step, reverse the effect of the diffusion equation:

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x}(K_d \frac{\partial c}{\partial x}) \tag{50}$$

By introducing artificial "diffusive" velocity u_d, equation (50) may be written as follows:

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x}(u_d c) \tag{51}$$

where:

$$u_d = \begin{cases} -\dfrac{K_d}{c}\dfrac{\partial c}{\partial x} & \text{for } c > 0 \\ 0 & \text{for } c = 0 \end{cases} \tag{52}$$

Now defining anti-diffusive velocity u:

$$\tilde{u} = -u_d \tag{53}$$

the reversal in time of the diffusion equation (50) can be carried out by the advection equation (51) with the anti-diffusive velocity $\tilde{u}$ instead of u. Thus, the corrective step has a following form:

$$c_i^{n+1} = c_i^* - [F(c_i^*, c_{i+1}^*, \tilde{u}_{i+1/2}) - F(c_{i-1}^*, c_i^*, \tilde{u}_{i-1/2})] \tag{54}$$

where: c_i^* is obtained from equation (46) and

$$\tilde{u}_{i+1/2} = \frac{(|u_{i+1/2}|\Delta x - \Delta t u_{i+1/2}^2)(c_{i+1}^* - c_i^*)}{(c_i^* + c_{i+1}^* + \epsilon)\Delta x} \tag{55}$$

and ϵ is a small value, e.g. 10^{-15} to ensure $\tilde{u} = 0$ when $c_{i+1}^* = c_i^* = 0$. The value of ϵ may, in general, depend on the precision of calculations. In the corrective step we again use an "upstream" scheme only exchanging $\tilde{u}$ for u. So the corrective step introduces numerical diffusion that can again be corrected by the next corrective step using $\tilde{\tilde{u}} = \tilde{u}(\tilde{u}, c^{**})$ where c^{**} is calculated from equation (53). Obviously there may be many corrective steps, each improving the accuracy of the solution. The same procedure can be applied to multidimensional cases. Then the relationship defining anti-diffusion velocities includes cross-derivative terms. The method was further developed by Smolarkiewicz and Clark (1986) for the cases of space and time dependent flow fields. The extension is very simple and results in additional terms in the equation defining the diffusion velocity (see equation 46). Also, more accurate methods can be obtained by expanding equation (46) in a Taylor sum using high-order terms. A comparison of several versions of the described technique carried out by Smolarkiewicz (1983) suggests that his algorithm is competitive with algorithms using the technique of flux-correction. Smolarkiewicz, by introducing anti-diffusive velocities, showed how to express the diffusion term of the transport equation as an advective term (see equation 51), making his method suitable for advection-diffusion equations as well.

Finally, it should be mentioned that the same approach may be used for schemes other than the "upstream"schemes.

3.5. Flux Correction Method (FC)

The flux-correction method for the numerical solution of the advection equation was developed by Boris and Book (1973; 1976), Book *et al.* (1975), Shir and Shieh (1974), Shieh and Shir (1976) and generalized by Zalesak (1979). This technique has been used for dispersion studies in the urban areas of New York (Bornstein *et al.*, 1987) and Linz, Austria (Pechinger *et al.*, 1987). The flux-correction algorithm is a centered-in-space, forward-in-time numerical scheme of first order accuracy with respect to time and second order accuracy with respect to space. Basically, it is formulated for the one-dimensional equation. Generalization to three-dimensional space is achieved by the Fractional Steps method (Yanenko, 1971), i.e. the advection terms for the different directions are calculated consecutively, using the results of the preceding step as input for the next. The order in which the directions are computed (x, y, z in the presented version), is kept constant. The flux-correction method is formulated in a staggered grid (Figure 1) and reads as follows:

$$c_{i,j,k}^{n'} = c_{i,j,k}^n + A_x(c_{i,j,k}^n) \tag{56a}$$

$$c_{i,j,k}^{n''} = c_{i,j,k}^{n'} + A_y(c_{i,j,k}^{n'}) \tag{56b}$$

$$c_{i,j,k}^{n+1} = c_{i,j,k}^{n''} + A_z(c_{i,j,k}^{n''}) \tag{56c}$$

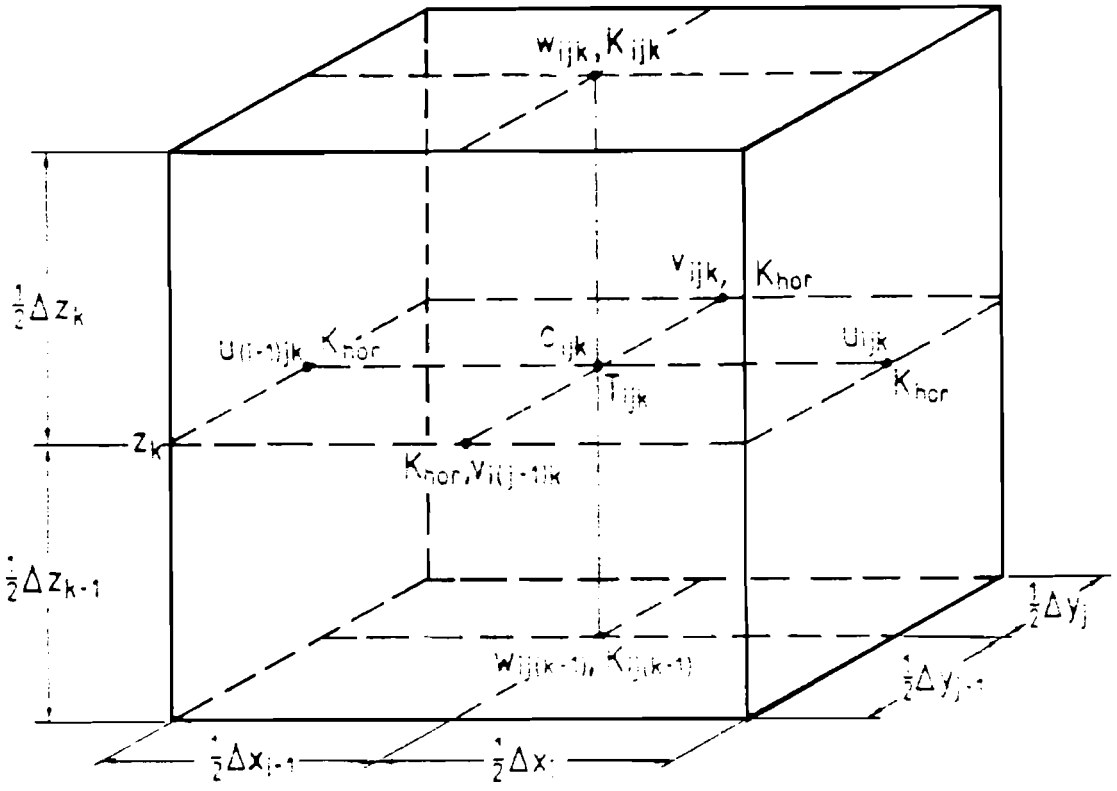

Figure 1. A staggered grid system used by the flux-correction method.

where $c^n_{i,j,k} = c(i\Delta x, j\Delta y, k\Delta z, n\Delta t)$ is the concentration and A_x, A_y, A_z are the advection operators acting in the x-, y- and z-direction, respectively. Because these operators have identical structure, we describe only one of them - A_x:

$$A_x[c_{i,j,k}] = F_x'[c_{i,j,k}] - F_x''[c_{i,j,k}] \tag{57}$$

where F_x' and F_x'' are the mass fluxes between grid (i,j,k) and grids: $(i-1,j,k)$ and $(i+1,j,k)$, respectively. They are computed in the following way:

$$F_x' = \begin{cases} \min(c_{i-1,j,k}, \tilde{F}_x') & \alpha'_i > 0 \\ \min(c_{i,j,k}, \tilde{F}_x') & \alpha'_i < 0 \end{cases} \tag{58a}$$

$$F_x'' = \begin{cases} \min(c_{i,j,k}, \tilde{F}_x'') & \alpha''_i > 0 \\ \min(c_{i+1,j,k}, \tilde{F}_x'') & \alpha''_i < 0 \end{cases} \tag{58b}$$

with supportive fluxes $\tilde{F}_x'$ and $\tilde{F}_x''$ defined as

$$\tilde{F}_x'[c_{i,j,k}] = \frac{\alpha'_i}{(1+\tau'_i)}\left[(1+\alpha'_i)c_{i-1,j,k} + (1-\alpha'_i)c_{i,j,k}\right] \tag{59a}$$

$$\tilde{F}_x''[c_{i,j,k}] = \frac{\alpha''_i}{(1+\tau''_i)}\left[(1+\alpha''_i)c_{i,j,k} + (1-\alpha'_i)c_{i+1,j,k}\right] \tag{59b}$$

and parameters τ'_i, τ''_i, α_i' and α_i'' are given by the following equations:

$$\tau'_i = \frac{\Delta x_i}{\Delta x_{i-1}} \tag{60a}$$

$$\tau''_i = \frac{\Delta x_{i+1}}{\Delta x_i} \tag{60b}$$

$$\alpha'_i = \frac{u_{i-\frac{1}{2},j,k}\,\Delta t}{\Delta x_{i-1}} \tag{60c}$$

$$\alpha''_i = \frac{u_{i+\frac{1}{2},j,k}\,\Delta t}{\Delta x_i} \tag{60d}$$

The flux correction method assumes that no more mass can be withdrawn from a grid cell than it contains; this assures the positive definiteness of the scheme.

Concerning side boundary conditions, a zero background is assumed for the upstream boundary while the downstream boundary is treated with a simple first order upstream formulation, thus avoiding any additional assumptions.

The discretization error of this scheme (Shieh and Shir, 1976) is:

$$\epsilon = \frac{(1-\alpha^2)\Delta x^2}{6}\left[u\Delta t\frac{\partial^3 c}{\partial x^3} - 3\frac{\partial N}{\partial x}\right],\quad N = \min\left[0, \frac{2c}{(1+\alpha)\Delta x} - \frac{\partial c}{\partial x}\right],\quad \alpha = \frac{u\Delta t}{\Delta x}. \tag{61}$$

The second part of that error $(-3\frac{\partial N}{\partial x})$ is due to the flux correction. If the correction is not needed (N=0), the error reads:

$$\epsilon = \frac{1-\alpha^2}{6}\,\Delta x^2\, u\frac{\partial^3 c}{\partial x^3}\,\Delta t \tag{62}$$

Expression (62) may be compared with the discretization error of the simple first order upstream scheme

$$\epsilon = \frac{\partial}{\partial x}\left[\frac{1-\alpha}{2}\,|u|\,\Delta x\frac{\partial c}{\partial x}\right]\Delta t \tag{63}$$

Computations show that the flux correction method is more accurate than the simple upstream scheme.

Tests made with a homogeneous two-dimensional flow field and a grid distance of 1 km for a one-grid block of concentration yielded a peak value reduction equivalent to a diffusion with a K ranging from 20 to 160 m^2/s (Courant numbers between .0025 and .9) (Pechinger *et al.*, 1987). This means that the numerical diffusion of that scheme is of the same order of magnitude as the natural turbulent horizontal diffusion, thus allowing its application to regional scale dispersion modeling. The phase error of the concentration peak remained less than 1–2% for transport distance of 100 km.

The scheme is numerically stable if the Courant number is less than or equal to 1.

4. ANALYTICAL TEST FOR THE NUMERICAL ALGORITHMS

As mentioned before, it is possible to solve analytically the advection equation for some special cases. A general analytical solution for the multidimensional advection equation (7) is not known. However, there are some particular cases when the analytical approach may be successful. These cases are very interesting for testing numerical algorithms. One example of such analytical solution will be used as a tool for comparison of the methods described in the previous Section.

4.1. Analytical Solution of the Advection Equation

The analytical solution that we have in mind, describes the rotation of a "frozen" initial condition around the axis of symmetry. It has been frequently used for testing numerical methods (Orszag, 1971; Gazdag, 1973; Long and Pepper, 1981; Christensen and Prahm, 1976). In two dimensions, the rotation of a frozen initial condition is described by the following equation:

$$\frac{\partial c}{\partial t} - \omega y\frac{\partial c}{\partial x} + \omega x\frac{\partial c}{\partial y} = 0 \tag{64}$$

where ω is angular velocity ($\omega = \frac{2\pi}{T}$), and T is the period of rotation.

Equation (64) was also solved numerically, using all algorithms described in this paper, and the results compared to the analytical solution. A grid consisting of 32 points, in both the x- and y-direction, was used during the experiment. The time step, for all methods was the same - $\frac{T}{400}$ which meant that one full revolution required 400 time steps. The analytical and numerical solutions were compared after each of the 10 rotations. In addition, two parameters were computed during each run. Namely:

(1) Mass conservation (in %) - M

$$M = \frac{\sum_{i=1}^{32}\sum_{j=1}^{32} c(i,j)}{\sum_{i=1}^{32}\sum_{j=1}^{32} c_o(i,j)} \times 100 \tag{65}$$

where $c_0(i,j)$ is the initial concentration, and

(2) Minimum of $c(i,j)$ - *MIN*

We compared analytical and numerical solutions for three types of the initial condition with different shapes, but all tested initial conditions had the same maximum - 100.

Before we start to describe the results of this comparison, it is necessary to make several remarks:

1. The period of integration, 10 full rotations, makes the test rather severe for all numerical methods. In reality, air pollution transport in the mesoscale does not require such long integration time. However, for the long range transport, this factor may be important because some pollutants have a very long residence time in the atmosphere. By this long integration time, even small differences between the methods are significantly amplified.
2. For the rotational velocity field used in the test the Courant number is variable in space. It is equal to zero in the center of a grid system and reaches maxima in the corners.
3. The wind field used in the test is stationary. Therefore, all conclusions from the comparison apply only to the stationary cases, and we can not generalize them to the cases with the time dependent velocity fields.

Having these remarks in mind, we begin the comparison with a cone initial condition.

4.2. Cone Shape Initial Condition

The advection equation (64) with the "cone" shape initial condition (Figure 2a) is a standard analytical test and has been applied to many numerical methods. In the grid system, the "cone" shape is defined as:

$$c_0(i,j) = \begin{cases} 100 \cdot (1 - r/4) & \text{if } r \le 4 \\ 0 & \text{if } r > 4 \end{cases} \qquad (66)$$

$$r = \sqrt{(i-8)^2 + (j-16)^2}$$

$$i, j = 1,\ldots,32$$

The numerical solutions after ten rotations are shown in Figures (2b-f). As can be seen from Figure 2a, a PDPS solution is quite close to the analytical one. Also, the shapes of the PDG and PDS solutions are not far from the analytical solution - Figures 2c and 2e, but are not as close as the pseudospectral solution. The PDG algorithm creates two small waves around the cone base which do not exist in the analytical field. A significant influence of the artificial numerical diffusion on the solution can be seen in cases of the AC and FC algorithms - Figures 2d and 2f.

Concerning mass conservation (Figure 3), both the PDPS method and the PDS method conserve 100% of the mass during the entire run. The PDG method conserves slightly worse mass - 97.4% after 10 rotations. The FC and AC schemes are worse in this respect; 89.9% and 75.5% of the total mass is conserved after 10 rotations. Saying that one of the numerical methods does not conserve the mass during the integration may be misleading. Namely, it does not mean that the particular method is inconsistent in its theoretical formulation. It means only that the mass is exported outside the model area due to the numerical diffusion and open boundary conditions.

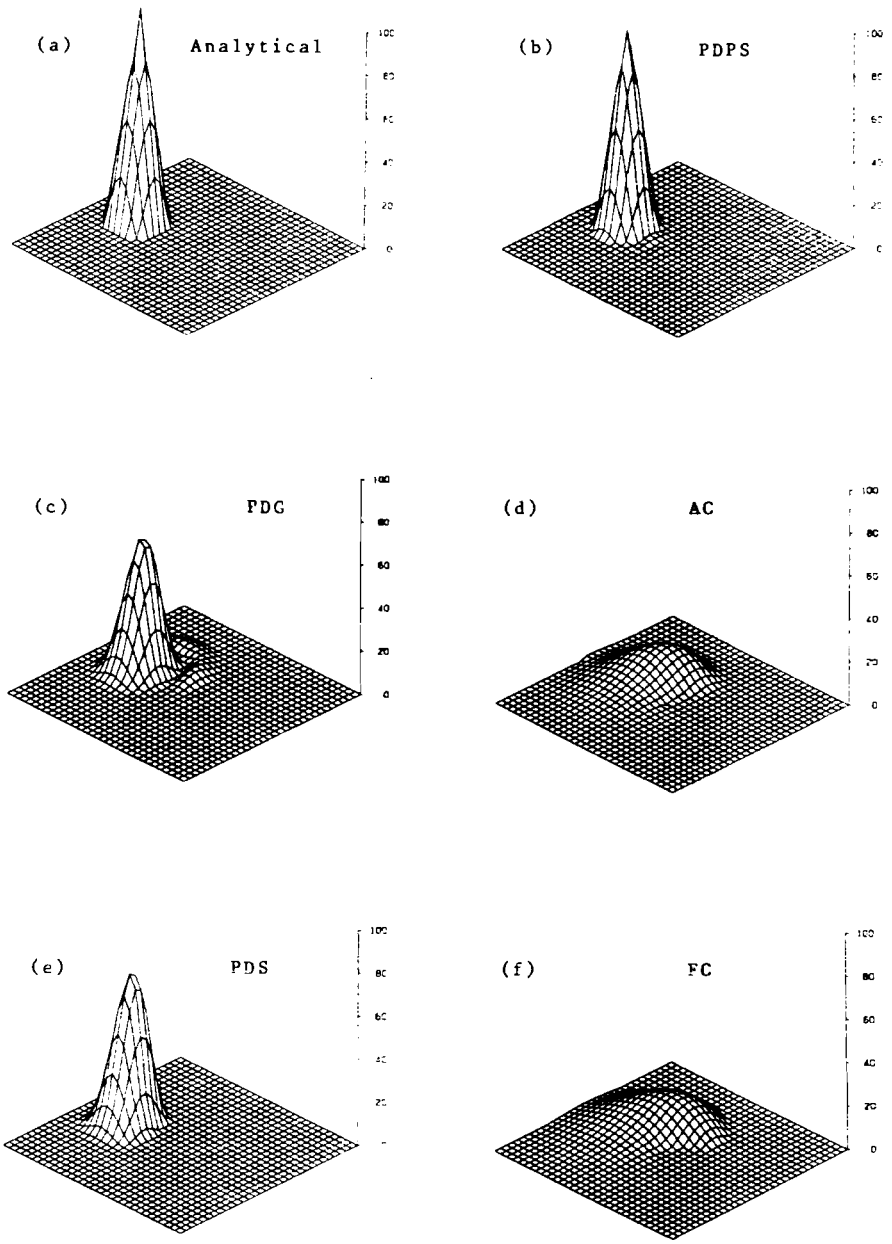

Figure 2. Analytical vs numerical solutions after 10 rotations. A cone shape initial condition. (a) Analytical solution. (b) Numerical solution computed by the positive definite pseudospectral method - PDPS. (c) Numerical solution computed by the positive definite Galerkin method - PDG. (d) Numerical solution computed by the antidiffusion correction algorithm - AC. (e) Numerical solution computed by the positive definite spline method - PDS. (f) Numerical solution computed by the flux correction algorithm - FC.

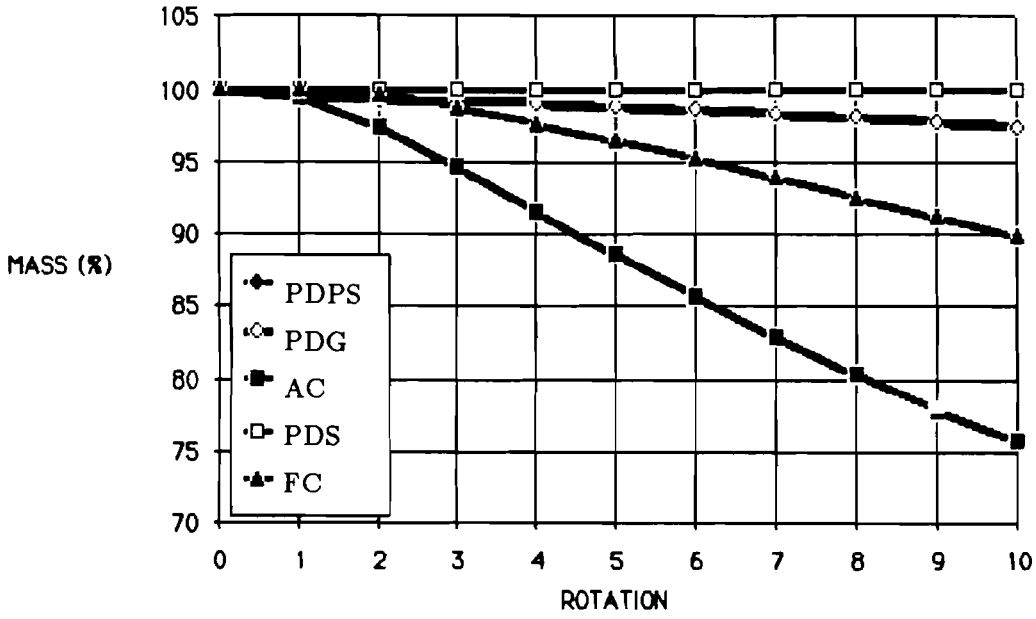

Figure 3. A cone shape initial condition: mass conservation.

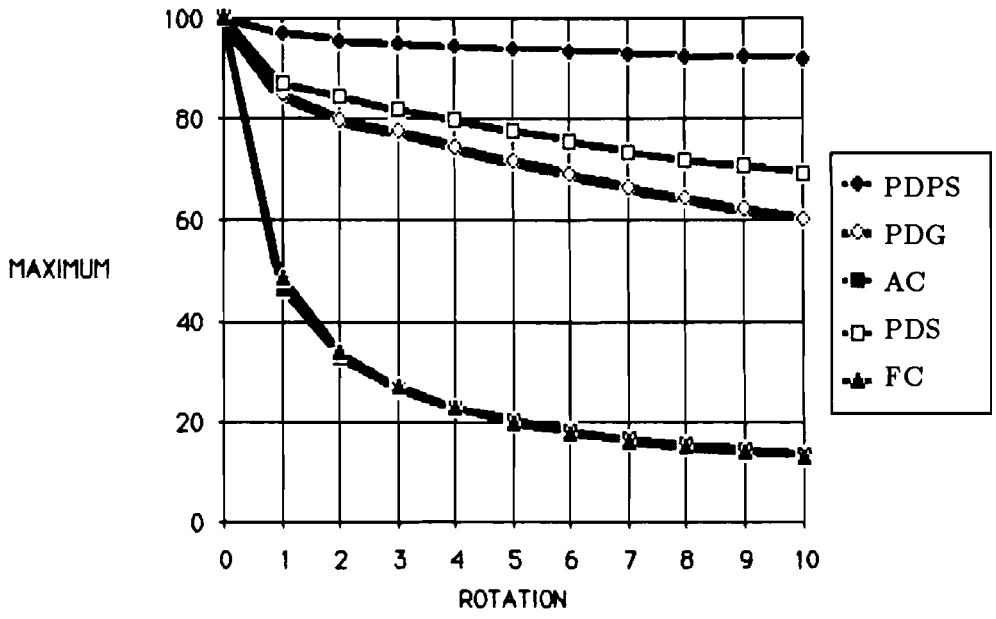

Figure 4. A cone shape initial condition: conservation of maximum.

The maximum of the cone (Figure 4) is conserved quite well by the PDPS method - 91.4% after 10 rotations and not so well by the PDS and the PDG algorithms - 68.9% and 59.7%, respectively. The results of FC and AC methods are very similar for all 10 rotations. The influence of artificial numerical diffusion is visible already after the first rotation, reducing the maximum of the cone to 48.5% and 46.2%, respectively.

4.3. Rectangular Block Initial Condition

The "Rectangular Block" initial condition is shown in Figure 5a. It is defined in the numerical grid as:

$$c_0(i,j) = \begin{cases} 100 & \text{if } 5 \le i \le 11 \text{ and } 13 \le j \le 19 \\ 0 & \text{otherwise} \end{cases} \qquad (67)$$

$$i,j = 1,...,32$$

The "Rectangular Block" initial condition is an important and difficult test because it includes relatively many points with a steep gradient. In the case of LRTAP modeling such a gradient is often visible in the emission fields (e.g. Dovland and Saltbones, 1979).

The results of analytical versus numerical computations after ten rotations are shown in Figure 5. Again, the PDPS solution (Figure 5b) is quite close to the analytical one. Though, there are some small wiggles around the rectangular block. The shapes of the PDG and PDS solutions are not so close to the initial condition, mainly due to the strong numerical diffusion which, to a large extent, is smoothing the sharp gradients of the rectangular block. In the PDG solution (Figure 5c) some artificial irregular waves are visible in the picture. The PDS solution (Figure 5c) seems to be less influenced by the numerical diffusion and does not contain any waves and wiggles. However, for this method, numerical maximum significantly exceeds analytical. Finally, the antidiffusion correction and flux correction methods (Figures 5d and 5f) give a similar picture, but in both cases, numerical diffusion is very strong.

Regarding mass conservation (Figure 6), for the PDPS method, the total mass after 10 rotations is equal to the analytical value - 100%. It is necessary to mention here that such good conservation of the total mass by the PDPS method, for all tested initial conditions, is mainly due to the periodic conditions applied at the boundaries. However, analogous tests with open boundary conditions do not show significant differences. The flux correction method and the PDG method have a similar performance - 88.7% after 10 rotations. For the antidiffusion correction method, a strong mass deficit may be observed - 74.5% after 10 rotations. There is a certain problem with the PDS algorithm. Although, the total mass is very close to 100% for each of the 10 rotations, it is at the end slightly higher (103.4% - after 10 rotations) then the analytical value. This implies that the filtering algorithm applied to the splines is probably not quite stable and should be modified according to the suggestions in Section 3.3.2.

The conservation of the maximum is shown in Figure 7. Numerical maxima computed by the PDPS and PDG methods are relatively close to the analytical maximum: 101.0 - PDPS and 96.2 - PDG, after 10 rotations. Both flux correction method and antidiffusion correction algorithm behave in a similar way increasing the maximum during the first rotation and then losing it to values: 36.7 - FC and 37.9 - AC after 10 rotations. In the case of the PDS method, after a positive jump in the first rotation, the computed maximum shows a steady increase, reaching the value 134.4 - after 10 rotations.

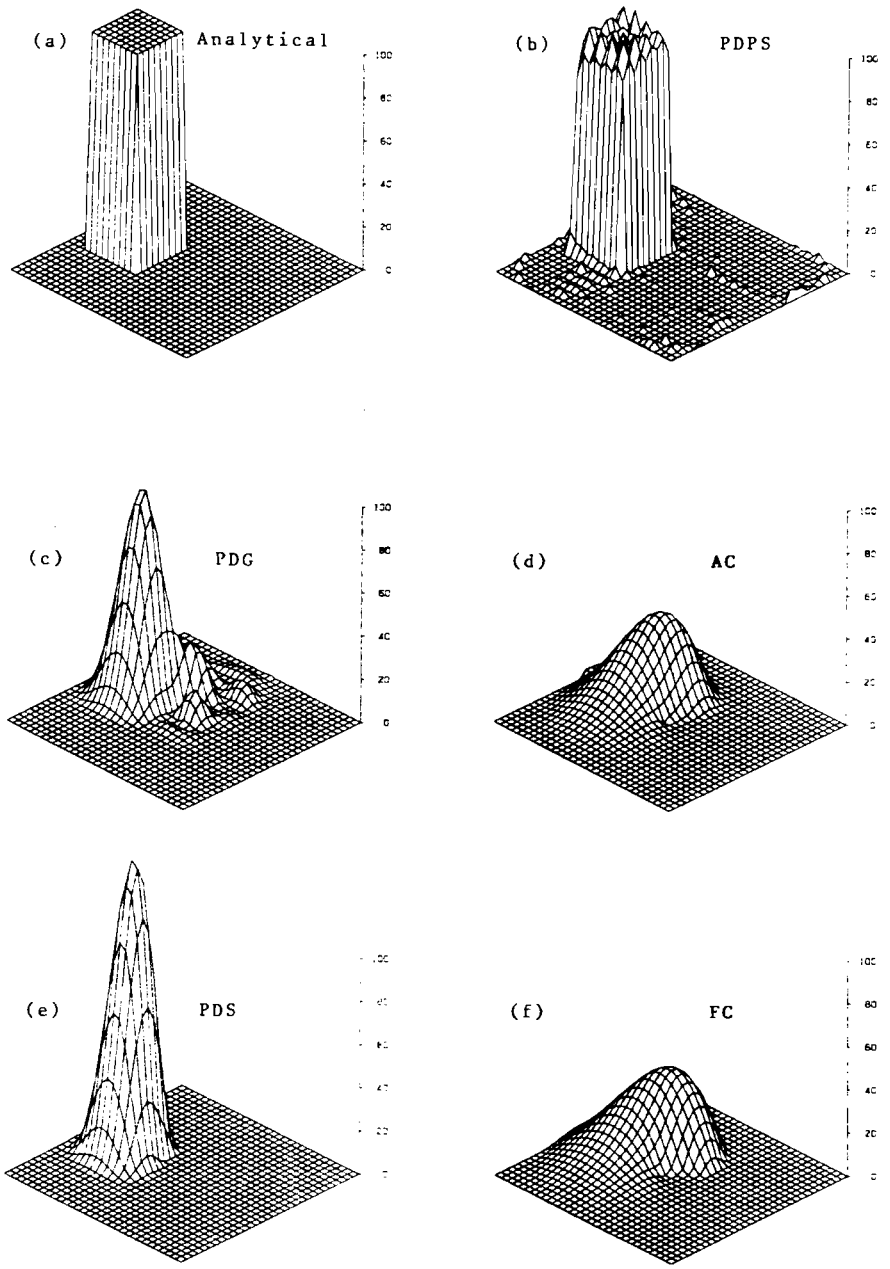

Figure 5. Analytical vs numerical solutions after 10 rotations for the rectangular block initial conditions. (a) Analytical solution. (b) Numerical solution computed by the positive definite pseudospectral method - PDPS. (c) Numerical solution computed by the positive definite Galerkin method - PDG. (d) Numerical solution computed by the antidiffusion correction algorithm - AC. (e) Numerical solution computed by the positive definite spline method - PDS. (f) Numerical solution computed by the flux correction algorithm - FC.

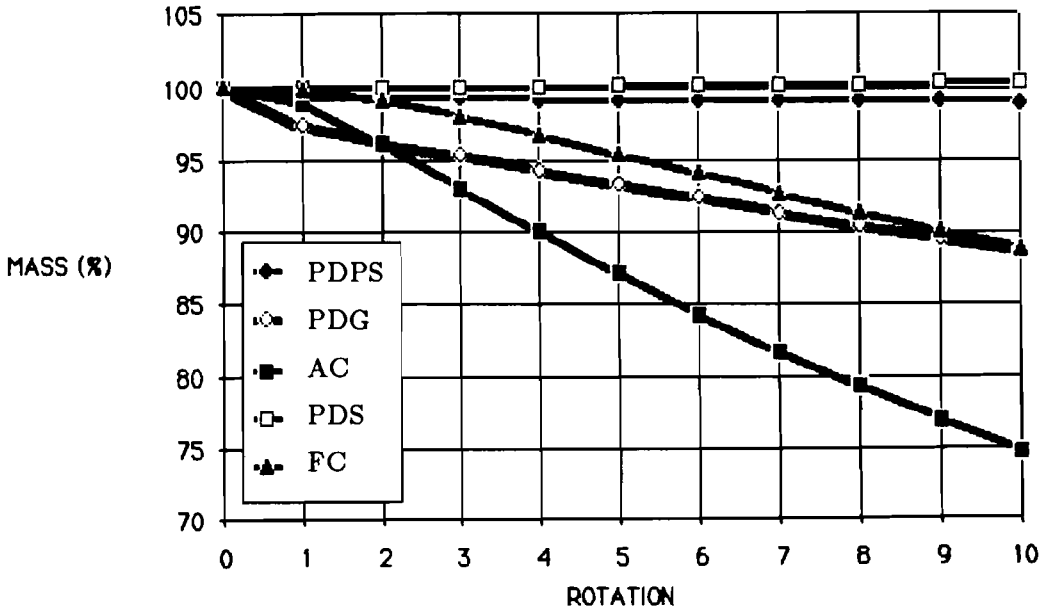

Figure 6. Rectangular block initial condition: mass conservation.

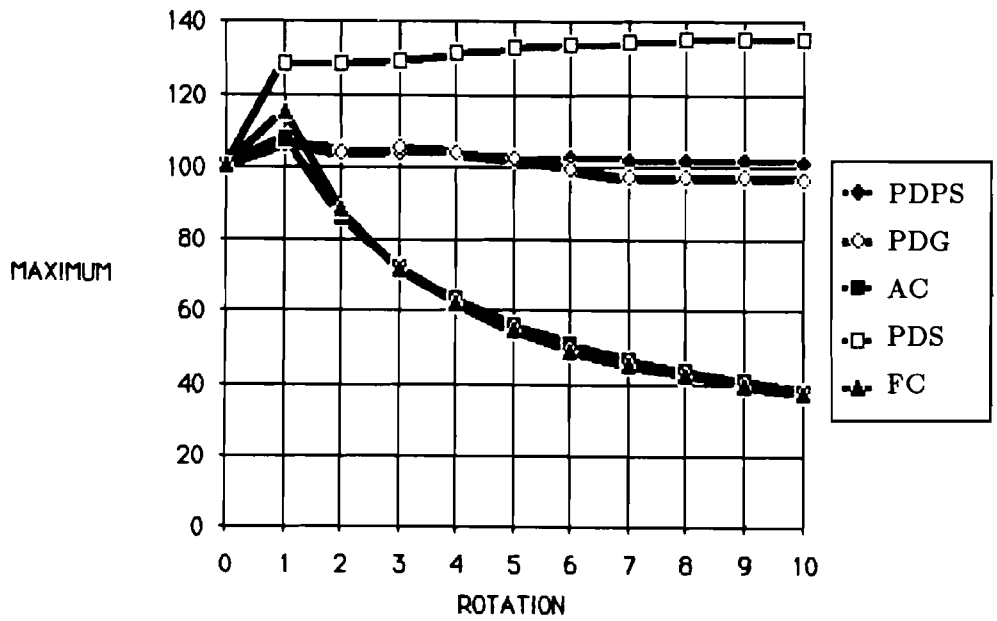

Figure 7. Rectangular block initial condition: conservation of maximum.

4.4. Delta Initial Condition

The "delta" function (Figure 7a) is probably the most difficult test case for the numerical methods and most likely this is the reason why it is hardly used and published in the open literature. Nevertheless, the "delta" type of the boundary conditions still has a practical meaning and therefore we decided to include it in our comparison. For example, in LRTAP models, "delta" can represent emissions from one small country like Luxembourg in the EMEP grid system (Dovland and Saltbones, 1979). It can also represent single emission sources in other smaller-scale models.

In the numerical grid system "delta" function is defined as follows:

$$c_0(i,j) = \begin{cases} 100 & x = 8 \text{ and } y = 16 \\ 0 & \textit{all other } i \text{ and } j \end{cases} \qquad (68)$$

$$i,j = 1,...,32$$

The numerical solutions after ten rotations are shown in Figure 8. The general impression gained from this figure is rather bad. For two methods, antidiffusion correction and flux correction, the delta initial condition was totally diffused and can not be recognized in the picture (Figures 8d and 8f). The situation is not much better for two other methods: PDS and PDG (Figures 8d and 8e) where only a trace of the initial delta is visible. In the case of the PDPS method (Figure 8b), the delta initial condition can be recognized in the solution but the shape of the function is strongly diffused to the neighboring grids.

The total mass conservation is presented in Figure 9. For the reason mentioned in the previous paragraph (periodic boundary conditions), the PDPS method conserves the total mass in 100%, also for this type of initial condition. For the PDS method, the total mass is very close to the analytical value but it is slightly higher again - 100.01%. A similar situation occurs for the PDG method, with the total mass equal: 100.06%, 100.08%, 100.02% for the first three rotations, respectively, and then is reduced to 73.9% after 10 rotations. The reason for the mass generation in this case is not quite clear, but most likely, it is related to the influence of the boundary conditions. For the remaining two methods, flux correction and antidiffusion correction, a decline of the total mass after first rotation may be observed, to 89.2% - FC and 73.9% - AC, after 10 rotations.

In general, the maximum conservation (Figure 10), for all tested methods, is very poor. It drops rapidly below 20% after the first rotation, and then slowly declines to: 16.2, 5.07, 3.85, 0.79 and 0.77 after ten rotations, for the PDPS, PDS, PDG, antidiffusion correction and flux correction methods, respectively.

4.5. Computational Time

In our numerical experiments, we have also compared the computational time for each method. Generally, it depends on the type of computer which is used to run the programs, programming language, type of compiler, etc. In our case, we used a VAX780 as a reference mainframe. However, all programs were also run on IBM compatible microcomputer without any difficulties. All schemes were coded in standard FORTRAN-77 and compiled by a standard UNIX compiler - VAX780, or PROFORT 2.1 compiler - IBM compatible microcomputer. The execution time of one full step is given, for each method, in Table 1.

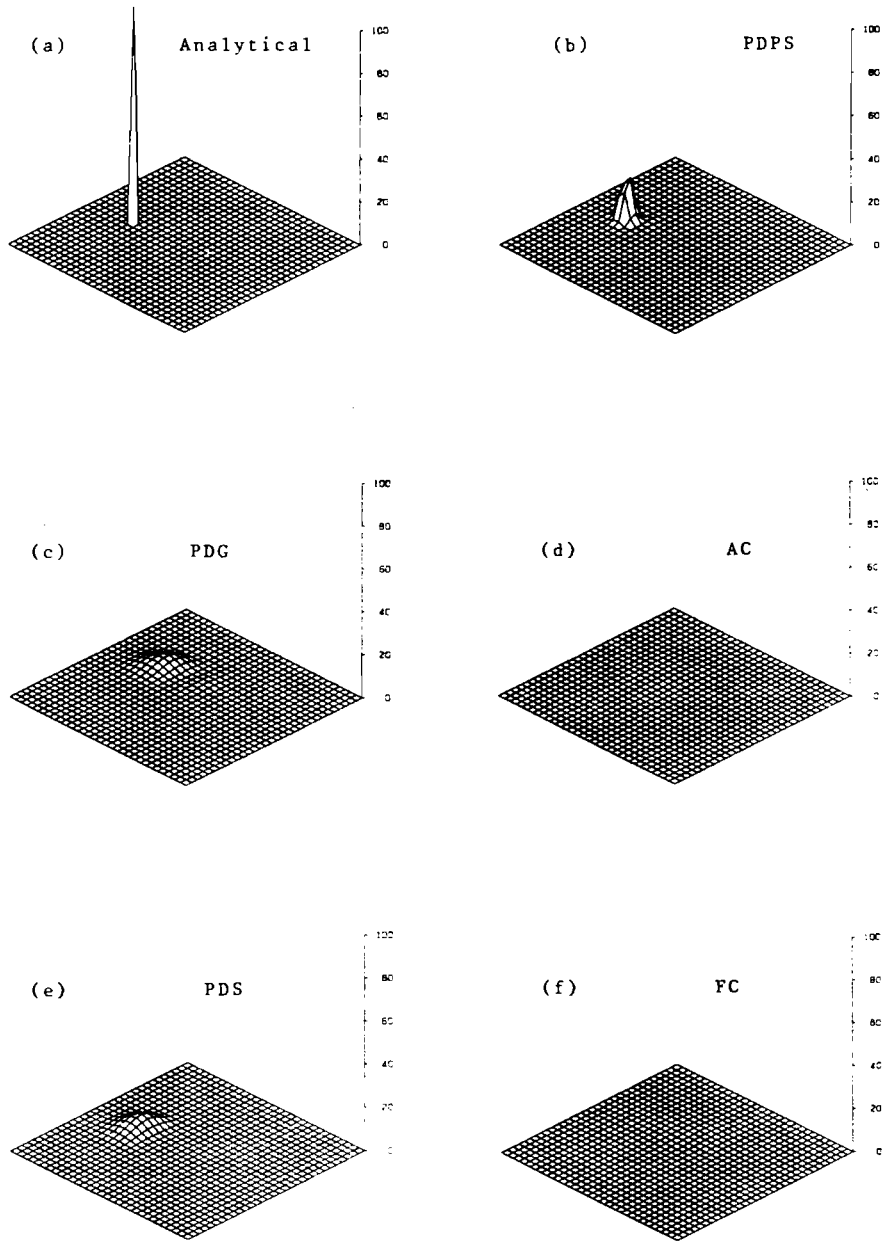

Figure 8. Analytical vs numerical solutions after 10 rotations for the delta initial conditions. (a) Analytical solution. (b) Numerical solution computed by the positive definite pseudospectral method - PDPS. (c) Numerical solution computed by the positive definite Galerkin method - PDG. (d) Numerical solution computed by the antidiffusion correction algorithm - AC. (e) Numerical solution computed by the positive definite spline method - PDS. (f) Numerical solution computed by the flux correction algorithm - FC.

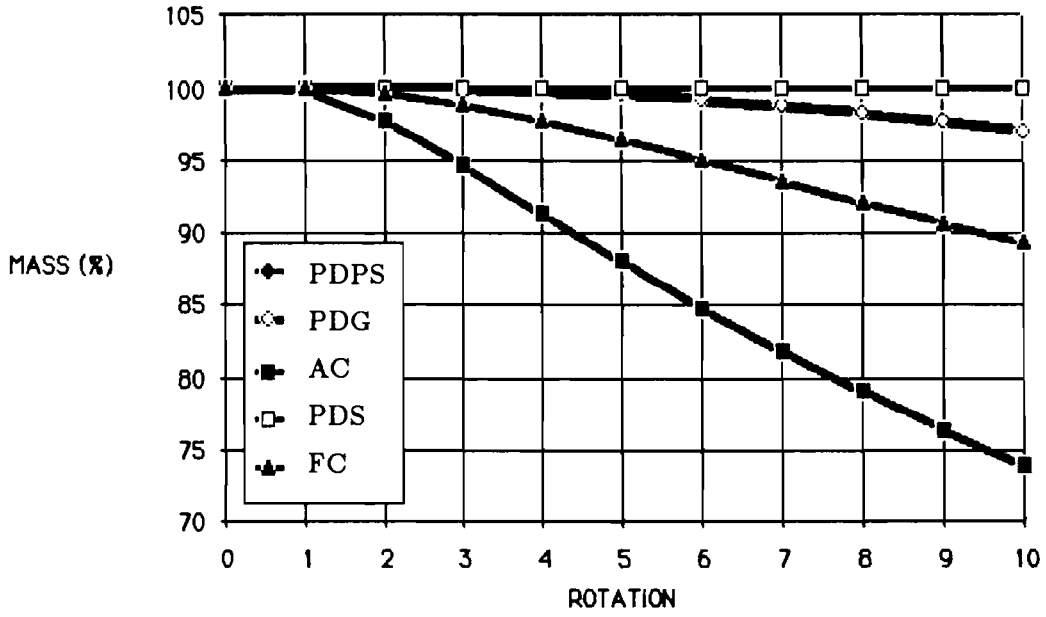

Figure 9. Delta initial condition: mass conservation.

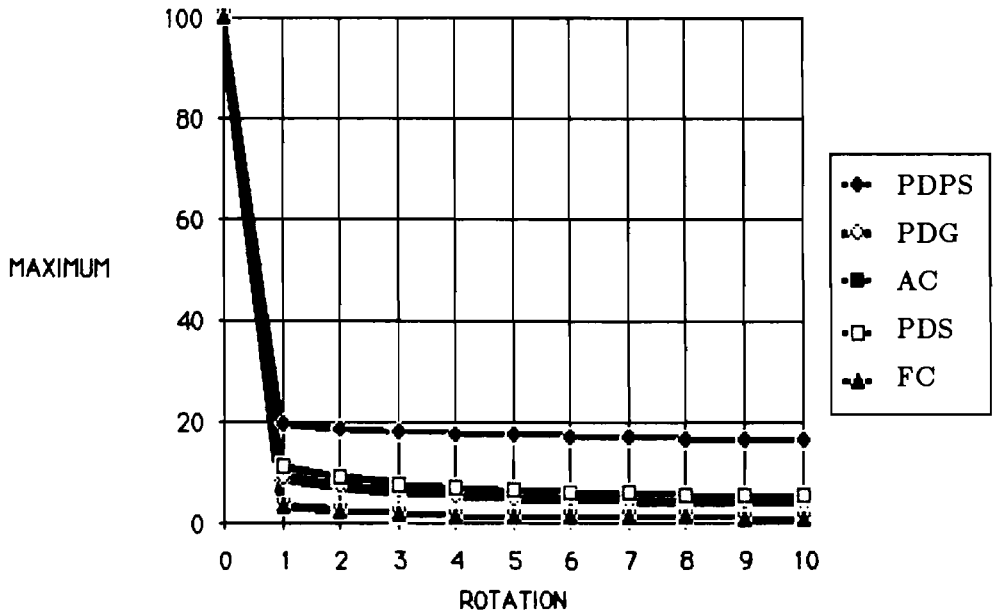

Figure 10.Delta initial condition: conservation of maximum.

Table 1. Execution of one time step for each algorithm compared in the study.

Method		Execution time
Positive Definite Pseudospectral Method (PDPS)	-	6.78 sec
Positive Definite Galerkin Method (PDG)	-	1.12 sec
Positive Definite Spline Method (PDS)	-	2.77 sec
Antidiffusion Correction Method (AC)	-	2.43 sec
Flux Correction Method (FC)	-	0.97 sec

From the methods tested, the PDPS scheme is definitely the slowest and FC is the fastest. However, the PDG method is almost as fast as FC. There are two remarks concerning computation time. The first is related to possible code optimization. Namely, all algorithms used in the experiment were coded in a relatively simple way and by no means optimally. Therefore, improvements of the code can make all compared algorithms faster. The second remark concerns a possibility of using a parallel processor in the computations. With such processor, the computer time can be much shorter because the computations can be performed simultaneously and independently for 32 rows or columns in the x- and y-directions. It means that the execution time for these three methods can be shortened by a factor of 30. It should also be mentioned that the PDPS algorithm uses complex mathematics which allows us to compute two concentration fields at the same time. It means that the effective execution time for this method should be halved. But, on the other hand, limit of the Courant number for this method is 0.5, whereas for all other algorithms it is 1.0. So, despite possible improvements, the ratios in Table 1 between execution times for different schemes should stay similar.

5. SUMMARY AND CONCLUSIONS

In this study we described five positive definite algorithms for the numerical solution of the transport equation. In order to compare these algorithms, we have applied them to a two-dimensional advection equation with a known analytical solution. Three different initial conditions were used during the experiment: cone shape, rectangular block and delta. The analytical solution was compared to the numerical solutions. When evaluating each method, we were especially interested in the differences between the shapes of the analytical and numerical solutions which were compared based on three-dimensional plots presented in the paper. We also examined the total mass and the conservation of the maximum for each numerical method. In addition, we compared the efficiency of each algorithm by calculating the execution time necessary for 10 rotations.

Based on the results of the numerical experiments we can draw the following conclusions:

1. It is necessary to use different initial conditions when testing numerical methods with the advection equation presented in this paper. As has been shown in the experiment, a popular cone shape initial condition is not the most difficult case for the comparison. We suggest two additional shapes: rectangular block and delta, for examining the simulation of sharp gradients and point sources, respectively.
2. The number of rotations in the experiments should be rather high. After the first rotation, the results looked promising for all methods, but after 10 rotations, some of them showed a significant influence of the artificial numerical diffusion in the solution. The high number of rotations is also important for the applications of the tested algorithms to air pollution transport models which include the pollutants with long residence times which is especially important for the long range air pollutant transport models. For the meso-scale models, an analytical test with ten rotations

may be too severe.

3. All methods tested in the numerical experiment do not work well in the case of the delta initial condition which corresponds to a point source in the LRTAP models.
4. For other than delta initial conditions, three methods: PDPS, PDG and PDS passed the numerical test satisfactorily. Two others: AC and FC had some difficulties with the mass conservation in the model area. Since, all tested methods were conservative in mathematical formulation, the only reason for the mass deficit could be a mass transfer through the boundaries which was the effect of significant numerical diffusion.
5. In the numerical test we did not test the performance of the boundary conditions applied to different methods. We also did not check how the results depend on the Courant number. To improve the comparison, one has to take these problems into account.
6. From the five algorithms tested, the most accurate was a PDPS scheme. The differences between analytical and numerical solutions were smallest for this algorithm and the maximum was conserved well during the runs. However, computer time for this method was 2–3 times longer than for the remaining methods.
7. Computer programs for all tested schemes are relatively simple. The FORTRAN-77 codes do not require much computer memory and allow the user to run all the algorithms on a regular PC microcomputer.

Based on the results presented in this study, besides the above specific conclusions, we can also formulate a general conclusion. Namely, when designing new air pollution transport models, it is very important to test numerical methods used for the solution of the transport equations. Otherwise, the models may compute results which have nothing to do with reality, because of numerical errors.

6. REFERENCES

Abert, K. (1989). Positive definite Galerkin method in application to the advection equation. *Acta Geophysica Polonica* (in press).

Alcamo, J., and J. Bartnicki (1985). An approach to uncertainty of a long range air pollutant transport model. IIASA Working Paper WP-85-88. International Institute for Applied Systems Analysis, A-2361 Laxenburg, Austria.

Alcamo, J., and J. Bartnicki (1987). A framework for error analysis of a long range transport model with emphasis on parameter uncertainty. *Atmos. Environ.*, **21**(10): 2121–2131.

Alcamo, J. M. Amann, J.-P. Hettelingh, M. Holmberg, L. Hordijk, J. Kämäri, L. Kauppi, P. Kauppi, G. Kornai and A. Mäkelä (1988). Acidification in Europe: a simulation model for evaluating control strategies. *Ambio*, **16**(5): 232–245.

Bartnicki, J. (1983). Operational spectral long range air pollutant transport model for Poland. Ph. D. Thesis (in Polish). Institute of Meteorology and Water Management. Warsaw, Poland.

Bartnicki, J. (1986). An efficient positive definite method for the numerical solution of the advection equation. IIASA Working Paper WP-86-35. International Institute for Applied Systems Analysis, Laxenburg, Austria.

Bartnicki, J. (1989). A simple filtering procedure for removing negative values from numerical solutions of the advection equation. *Environmental Software*, **4**: 187–207.

Bartnicki, J. and J. Alcamo (1989). Calculating nitrogen deposition in Europe. *Journal of Water, Air and Soil Pollution*, **47:101123.**

Batterman, S.A. (1988). Influential receptors in targetted emission control strategies. *Systems Analysis Modelling Simulation.* Journal of Mathematical Modelling and Simulation in Systems Analysis, 5 (6):519–532.

Batterman, S.A., M. Amann, J.-P. Hettelingh, L. Hordijk and G. Kornai (1988). Optimal SO_2 abatement policies in Europe: some examples. *Systems Analysis Modelling Simulation.* Journal of Mathematical Modelling and Simulation in Systems Analysis, 5 (6):533–559.

Bornstein, R.D., U. Pechinger, R. Salvador, L.J. Shieh, and F. Ludwig (1987). *Modeling the Polluted Coastal Urban Environment. Vol. 2: The Dispersion Model.* EPRI Research Report EA-5091.

Boris, J.P. and D.L. Book (1973). Flux-corrected transport. I. SHASTA, a transport algorithm that works. *J.Comp.Phys.*, **11**: 38–69.

Book, D.L., J.P. Boris and K. Hain (1975). Flux-corrected transport. II. Generalizations of the method. *J.Comp.Phys.*, **18**: 248–283.

Boris, J.P. and D.L. Book (1976). Flux-corrected transport. III. Minimal-error FCT algorithms. *J.Comp.Phys.*, **20**: 397–431.

Chock, D.P. and A.M. Dunker (1983). A comparison of numerical methods for solving the advection equation. *Atmos. Environ.*, **17**: 11–24.

Chock, D.P. (1985). A comparison of numerical methods for solving the advection equation - II. *Atmos. Environ.*, **19**: 571–586.

Christensen, O. and L.P. Prahm (1976). A pseudospectral model for dispersion of atmospheric pollutants. *J. Appl. Meteor.*, **15**: 1284–1294.

Cooley, J.W. and J.W. Tukey (1965). An algorithm for the machine calculation of complex Fourier series. *Math. Comp.*, **19**: 297–301.

Crank, J. and P. Nicholson (1947). A practical method for numerical integration of solutions of partial differential equations of heat-conductive type. *Proc. Camb. Philos. Soc.*, **43**: 50–73.

Dovland, H. and J. Saltbones (1979). Emissions of sulfur dioxide in Europe in 1978. Norwegian Institute for Air Research Report EMEP/CCC 2/79.

Eliassen, A. and J. Saltbones (1983). Modeling of long range transport of sulfur over Europe: a two-year model run and some model experiments. *Atmos. Environ.*, **17**: 1457–1473.

Eliassen, A. O. Hov, J. Ivensen, J. Saltbones and D. Simpson (1988). Estimates of transboundary transport of sulphur and nitrogen over Europe. *EMEP/MSC-W Report 1/88.*

Gazdag, J. (1973). Numerical convective schemes based on accurate computation of space derivatives. *J. Comput. Phys.*, **13**: 100–113.

Long, P.E. and D.W. Pepper (1981). An examination of simple numerical schemes for calculating scalar advection. *J. Appl. Meteor.*, **20**: 146–156.

Ludes, G. (1987). *Modellierung von Ausbreitungsprozessen in topographisch gegliedertem Gelände.* Diploma thesis, University of Munich.

Ludes, G. and W. Ulrich (1988). Numerical simulation of plume advection over a hill with coupled models. Part I: Numerical solution of the transport equation. *Meteor. Atmos. Phys.*.**39**: 97–107.

Orszag, S.A. (1971). Numerical simulation of incompressible flows within simple boundaries. I. Galerkin (Spectral) Representations. *Study in Appl. Math.*, **50**: 293–327.

Pechinger, U., P. Seibert, R.D. Bornstein, and M. Bogner (1987). Erstellung von numerischen Simulationsmodellen zur Berechnung der Schadstoffbelastung in Ballungsgebieten. (To be published)

Pielke, R.A. (1984). *Mesoscale Meteorological Modeling.* Academic Press.

Pepper D.W. C.D. Kern and P.E. Long (1979). Modeling the dispersion of atmospheric pollution using cubic splines and Chapeau functions. *Atmos. Environ.*, **13**: 223–237.

Price, G.W. and A.K. MacPherson (1973). A numerical weather forecasting method using cubic splines on a variable mesh. *J. appl. Met.*,**12** : 1102–1113.

Purnell, D.K. (1976). Solution of the advection equation by upstream interpolation with a cubic spline. *Mon. Weath. Rev.*, **104**: 42–48.

Richtmyer, R.D. and K.W. Morton (1967). *Difference Methods for Initial Value Problems.* Interscience, New York/London.

Roache, P.J. (1976). *Computational Fluid Dynamics.* Hermosa Publishers, Albuquerque.

Seibert, P. and B. Morariu (1990). Improvements of the upstream-spline numerical advection scheme. Submitted to *J. Appl. Meteor.*

Shaw, R.W. (1986). Optimization of emission reduction to reduce atmospheric sulphur deposition in Europe: an examination of a methodology. *Water, Air and Soil Pollution,*, **30**: 879–884.

Shieh, L.J. and C.C. Shir (1976). *Development of an Urban Air Quality Simulation Model with Compatible RAPS Data.* IBM Research Report RJ-1701.

Shir, C.C. and L.J. Shieh (1974). A generalized urban air pollution model and its application to the study of SO_2 distributions in the St. Louis metropolitan area. *J. Appl. Meteor.* **13**: 185–204.

Smolarkiewicz, P.K. (1983). A simple positive definite advection scheme with small implicit diffusion. *Mon. Wea. Rev.*, **111**: 479–486.

Smolarkiewicz, P.K. and T.L. Clark (1986). The multidimensional positive definite transport algorithm : further development and applications. *J. Comp. Phys.*, **67**: 396–438.

Yanenko, N.A. (1971). *The Method of Fractional Steps.* Springer, Heidelberg, Germany.

Zalesak, S.T. (1979). Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comp. Phys.*, **31**: 335–362.

Zannetti, P. (1981). A new Monte Carlo scheme for simulating Lagrangian particle diffusion with wind shear effects. *Applied Mathematical Modelling.*, **8**: 188–192.

Zlatev, Z. (1988). Treatment of some mathematical models describing long-range transport of air pollutants on vector processors. *Parallel Computing.*, **6**: 87–98.