Send Orders for Reprints on reprints@benthamscience.net
166 *Advanced Numerical Methods for Complex Environmental Models: Needs and Availability,* **2013, 166-202**

CHAPTER 3

Parallel Computations in a Large-Scale Air Pollution Model

Zahari Zlatev[1], Krassimir Georgiev[2,*] and Ivan Dimov[2]

[1]Department of Environmental Science, Aarhus University, Frederiksborgvej 399, P.O. 358, 4000 Roskilde, Denmark and [2]Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Acad. G. Bonchev str., bl. 25A, 1113 Sofia, Bulgaria

Abstract: Large-scale air pollution models, which are normally described mathematically as systems of partial differential equations, must very often be run **efficiently** on high-speed computer architectures. The requirement for efficiency is especially important when some fine discretization of the spatial domain is to be applied. In practice, this means that an efficient implementation of such a model on fast modern computers must nearly always be achieved, because as a rule fine grids are needed in the efforts to avoid the appearance of numerical errors that are comparable with or even larger than the errors which are caused by other reasons (uncertainties of the meteorological data, of the emission data, of the rates of the involved chemical reactions, *etc*.). The organization of the parallel computations will be discussed in this chapter of the eBook. The major principles, on which the parallelization is based, are rather general and, therefore, some of the discussed techniques can also be applied in connection with some large-scale models arising in other areas of science and engineering.

Keywords: Balkan Peninsula, Bulgaria, Denmark, England, Hungary, boundary condition, initial condition, Semi-discretization, sub-models, advection, semi-Lagrangian discretization, pseudo-spectral discretization, trigonometric polynomials, truncated Fourier series, Taylor series, QSSA (Quasi-Steady-State-Approximation), Trapezoidal Rule, Runge-Kutta method, A-stable, Strongly A-stable, L-stable method, BLAS (Basic Linear Algebra Subroutines), LAPACK, quasi-Newton iterative method, Cache memory, MPI, OpenMP.

INTRODUCTION

Five major physical and chemical processes have to be taken into account and to be described by mathematical terms when a large-scale air pollution model is to be developed. This will normally lead to a **system of partial differential equations** (PDEs). The number of equations in this system is equal to the number **q** of chemical species that are studied by the model.

***Address correspondence to Krassimir Georgiev:** Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Acad. G. Bonchev str., bl. 25A, 1113 Sofia, Bulgaria; Tel: +359 2 979 6612; Fax: +359 2 870 7273; E-mail: krtgeorgiev@gmail.com

The Unified Danish Eulerian Model (UNI-DEM), which is discussed in [1, 2, 3], will be used in this chapter. This model is described mathematically by the following system of PDEs:

$$\frac{\partial c_s}{\partial t} = -u\frac{\partial c_s}{\partial x} - v\frac{\partial c_s}{\partial y} \qquad \textbf{(1)} \text{ horizontal transport (advection)}$$

$$+\frac{\partial}{\partial x}\left(K_x\frac{\partial c_s}{\partial x}\right)+\frac{\partial}{\partial y}\left(K_y\frac{\partial c_s}{\partial y}\right) \qquad \text{horizontal diffusion}$$

$$+Q_s\left(t,x,y,z,c_1,c_2,\ldots,c_q\right)+E_s\left(t,x,y,z\right) \qquad \text{chemical reactions + emissions}$$

$$+\left(k_{1s}+k_{2s}\right)c_i \qquad \text{dry and wet depositions}$$

$$-w\frac{\partial c_s}{\partial z}+\frac{\partial}{\partial z}\left(K_z\frac{\partial c_s}{\partial z}\right), \qquad \text{vertical transport}$$

$$s=1,2,\ldots,q \qquad \text{q is number of equations (equal to the number of chemical species)}$$

The different quantities involved in (1) are briefly described below:

- $\mathbf{c_s = c_s(t,x,y,z)}$ is the concentration of the chemical species $\mathbf{s}$ at point $\mathbf{(x,y,z)}$ of the space domain and at time $\mathbf{t}$ of the time-interval,
- $\mathbf{u = u(t,x,y,z)}$, $\mathbf{v = v(t,x,y,z)}$ and $\mathbf{w = w(t,x,y,z)}$ are wind velocities (along the $\mathbf{Ox}$, $\mathbf{Oy}$ and $\mathbf{Oz}$ directions respectively) at the spatial point $\mathbf{(x,y,z)}$ and time $\mathbf{t}$,
- $\mathbf{K_x = K_x(t,x,y,z)}$, $\mathbf{K_y = K_y(t,x,y,z)}$ and $\mathbf{K_z = K_z(t,x,y,z)}$ are diffusivity coefficients at the spatial point $\mathbf{(x,y,z)}$ and time $\mathbf{t}$ (it is often assumed that $\mathbf{K_x}$ and $\mathbf{K_y}$ are non-negative constants, while the calculation of $\mathbf{K_z}$ is normally rather complicated),
- $\mathbf{k_{1s} = k_{1s}(t,x,y,z)}$ and $\mathbf{k_{2s} = k_{2s}(t,x,y,z)}$ are deposition coefficients (dry and wet deposition respectively) of the chemical species $\mathbf{s}$ at the spatial point $\mathbf{(x,y,z)}$ and time $\mathbf{t}$ of the time-interval. It should be mentioned here that for some of the species these

coefficients are non-negative constants. The wet deposition coefficients k_{2s} are equal to zero when it is not raining.

- $E_s(t,x,y,z)$ is emission source for the chemical species s at the spatial point (x,y,z) and time t of the time-interval.

- $Q_s(t,x,y,z,c_1,c_2,\ldots,c_q)$ is a non-linear term describing the chemical reactions in which the chemical species s is involved and which take place at the spatial point (x,y,z) and time t of the time-interval.

Much more details about UNI-DEM are given in the monographs [2, 3].

UNI-DEM was primarily developed for studying air pollution levels in the whole of Europe. Different features of this model are fully described in [1-16]. UNI-DEM was extensively used for performing different investigations related to the air pollution levels in

- The Balkan Peninsula ([15, 17]),
- Bulgaria ([18, 19]),
- Denmark ([2, 3, 20, 21]),
- England ([22]),
- Europe ([2, 3, 9, 20, 23, 24, 25]),
- Hungary together with its surroundings ([16, 26, 27]) and
- The North Sea ([28]).

The sensitivity of the model results to variations of some key parameters (emissions, chemical rates, *etc*.) was investigated in [2, 3, 29].

A previous version of UNI-DEM has also been used in some inter-comparisons of European large-scale air pollution models ([30, 31]).

The influence of climate changes on the pollution levels was studied in [10, 15, 16, 20, 21, 30, 32].

A lot of input data, both meteorological data and emission data, are used in UNI-DEM. These data are obtained mainly from EMEP; see [33, 34]. Some improved emission inventories from [35] were also used. Biogenic emissions were prepared following the ideas described in [25, 36-38].

In this chapter we shall start with a brief discussion the splitting procedure used in UNI-DEM (it is fully described in one of the previous chapters of this eBook) and the selection of numerical methods. Then we shall proceed with a detailed description of the organization of parallel computations. Finally, results related to the applications of UNI-DEM in connection with long-range transport and some climate change studies will be presented.

SPLITTING PROCEDURE

The following three sub-models can in an obvious way be obtained from (1):

$$\frac{\partial c_s^{(1)}}{\partial t} = -w\frac{\partial c_s^{(1)}}{\partial z} + \frac{\partial}{\partial z}\left(K_z \frac{\partial c_s^{(1)}}{\partial z}\right), \tag{2}$$

$$\frac{\partial c_s^{(2)}}{\partial t} = -u\frac{\partial c_s^{(2)}}{\partial x} - v\frac{\partial c_s^{(2)}}{\partial y} + \frac{\partial}{\partial x}\left(K_x \frac{\partial c_s^{(2)}}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial c_s^{(2)}}{\partial y}\right), \tag{3}$$

$$\frac{\partial c_s^{(3)}}{\partial t} = Q_s\left(t, x, y, z, c_1^{(3)}, c_2^{(3)}, \ldots, c_q^{(3)}\right) + E_s\left(t, x, y, z\right) + \left(k_{1s} + k_{2s}\right) c_s^{(3)} \ . \tag{4}$$

The first of these three sub-models describes the vertical exchange. The second sub-model describes the combination of the horizontal transport (the advection) and the horizontal diffusion. The last sub-model describes the chemical reactions together with the emission sources and the deposition terms.

The splitting of the original model (1) into the sub-models (2), (3) and (4) has at least three major advantages:

(a) no extra boundary conditions are needed,

(b) for each of the sub-models an optimal numerical methods can be selected and

(c) the organization of the parallel computations is considerably simpler than the attempt to organize directly parallel computations in (1), *i.e.* in the original system of PDEs.

In the following part of this chapter we shall mainly discuss the third advantage. The other two advantages are discussed in one of the other chapters of the eBook.

PARALLEL TASKS WHEN SHARED MEMORY MACHINES ARE USED

First the case where the available computer has many processors and each processor has access to common memory discs will be considered. The standard tool, which is used in this situation, is **OpenMP** (more details about this tool can be found in the web-site [39]). The programming is very straight-forward when OpenMP is used and we shall not discussed it here. It is much more important to emphasize the fact that the splitting of the original system of PDEs (1) into the three sub-systems (2)-(4) leads in a natural way to many parallel tasks.

Consider the sub-model (2). It is easy to see that this system of PDEs is in fact formed by many **independent and small** systems of PDEs. The number of these small sub-systems of PDEs is equal to the product of the number of the chemical species and number of grid-points in any of the horizontal grid-planes.

The second sub-model (3) is also formed by many independent systems of PDEs. The number of these smaller sub-systems of PDEs is equal to the product of the number of the chemical species and number of the horizontal grid-planes in the space domain.

It is nearly obvious that the third sub-model (4) can immediately be represented by a system of ordinary differential equations (ODEs), because it does not contain spatial derivatives. Furthermore, this system of ODEs contains a large number of independent and small sub-systems of ODEs. The number of these sub-systems is equal to the number of all grid-points in the space domain.

It is clear that in all three cases **each sub-system can be considered as a parallel task**. This means that many parallel tasks do appear in a quite natural way after the splitting. The number of these tasks does not depend on the particular choice

of numerical methods for the different sub-models. However, it is highly desirable to select fast and sufficiently accurate numerical methods. The choice of numerical methods is discussed in [2, 3].

The number of parallel tasks depends on four parameters: (a) on the numbers of grid-points $\mathbf{N_x}$, $\mathbf{N_y}$ and $\mathbf{N_z}$ along the coordinate axes and (b) on the number $\mathbf{q = N_q}$ of the chemical species studied by the model. In UNI-DEM the allowed values for these parameters are:

$$N_x = 96\ ,\ 288\ \textit{or}\ 480\ ,\quad N_y = N_x\ ,\quad N_z = 1\ \textit{or}\ 10\ ,\quad N_q = 35\ ,\ 56\ \textit{or}\ 168\ . \quad \textbf{(5)}$$

It should be mentioned here that the restriction of the values of the four parameters to those that are listed in (5) is determined by

(a) the distribution of the meteorological and the emission data for the first three parameters and

(b) the available chemical schemes for the fourth one.

It should also be mentioned that for $\mathbf{N_z = 1}$ the model becomes in fact two-dimensional. There is no vertical transport in this case and it is only necessary to handle the sub-models (3) and (4).

If the sub-model (2), the vertical transport, is considered and if $\mathbf{N_z = 10}$ then the number of parallel tasks is equal to the product $\mathbf{N_x \times N_y \times N_q}$ and a system of $\mathbf{N_z = 10}$ equations has to be solved in each of them.

If the sub-model (3), the horizontal transport, is considered, then the number of parallel tasks is equal to the product $\mathbf{N_z \times N_q}$ and a system of $\mathbf{N_x \times N_y}$ equations has to be solved in each of them.

If the sub-model (4), the chemical reactions (combined with emissions and the deposition), is considered, then the number of parallel tasks is equal to the product $\mathbf{N_x \times N_y \times N_z}$ and a system of $\mathbf{N_q}$ equations has to be solved in each of them.

The particular numbers of the parallel tasks for the allowed values of the four parameters are given in Table **1**.

Table 1: The numbers of parallel tasks in the different sub-models for the allowed values of the four key parameters, see (5), are given in the last six columns of this table.

N_q	$N_x = N_y$	Vertical Transport		Horizontal Transport		Chemical Reactions	
		$N_z = 1$	$N_z = 10$	$N_z = 1$	$N_z = 10$	$N_z = 1$	$N_z = 10$
35	**96**	-	322560	35	350	9216	92160
	288	-	2903040	35	350	82944	829440
	480	-	8064000	35	350	220400	2204000
56	**96**	-	516096	56	560	9216	92160
	288	-	4644864	56	560	82944	829440
	480	-	12902400	56	560	220400	2204000
168	**96**	-	1548288	168	1680	9216	92160
	288	-	13934592	168	1680	82944	829440
	480	-	38707200	168	1680	220400	2204000

The numbers of parallel tasks that are given in Table **1** show that the parallel tasks in the first and the third sub-models (the sub-models related to the vertical transport and the chemical reactions) are many (up to several millions when the fine resolution discretization is chosen). However, these tasks are very small. This fact causes a problem because some time is needed in order to start and to finish a parallel task while the time needed to perform the computations in each task is negligible. This means that the computer will use most of the time to start and to finish parallel tasks, which is obviously very inefficient. It is easy to deal with this problem: several small tasks are to be combined in a bigger task. The number of small tasks which have to be united in a bigger task depends on the available computer. It is not very difficult to find some suitable number of small tasks, so that the data (treated when the bigger task, which is obtained by combining certain number of small ones, is handled) stay in cache (it is very important for achieving efficiency of the computational process).

It is seen (from the results in the fifth and sixth columns in Table **1**) that the number of parallel tasks in the second sub-model (the horizontal transport) is not very big. Much more unfortunate is the fact that these tasks are very big when fine discretization is applied and the data, which is needed for the computations during a bigger parallel task, cannot stay in cache. Therefore, the speed of computations

is normally slow. It is necessary to split these tasks into smaller tasks in order to avoid this problem. However, it is not very easy to perform this operation. Some rules, which are similar to the rules used in the next section, can be used in connection with the second sub-model.

The short discussion given above and the results presented in Table **1** show clearly that the application of the OpenMP tool has two major advantages:

(a) it is easy to program it and

(b) many parallel tasks appear in a quite natural way when the splitting procedure from the previous section is used.

On the negative side is the problem with the big tasks in the second sub-model.

ORGANIZING THE COMPUTATIONS ON MESSAGE PASSING MACHINES

The program implementing the parallel computations on a shared memory machine consists of six major modules:

(a) Initialization: in this part the programs reads information related to discretization, first and foremost the values of the four parameters $\mathbf{N_x}$, $\mathbf{N_y}$, $\mathbf{N_z}$ and $\mathbf{N_q}$ as well as information about the location of the files from which input data are to be read and where output data are to be written.

(b) Input Operations: in this part the program reads (as a rule, not at every time-step, but only when it becomes necessary) meteorological and emission data from input files.

(c) Vertical Transport: the program performs (when $\mathbf{N_z = 10}$ and at every time-step) the calculations related to the first sub-model (2).

(d) Horizontal Transport: the program performs (at every time-step) the calculations related to the second sub-model (3).

(e) Chemistry: the program performs (at every time-step) the calculations related to the third sub-model (4).

(f) Output Operations: in this part the program writes (in general not at every time-step, but only when it becomes necessary) concentrations and some other quantities related to the concentrations on output files.

Two additional modules have to be prepared and used if the runs must be carried out of a message passing computer:

(g) Pre-Processing Module: this module is called by the code after the initialization part and before the beginning of the actual computations. The data is divided into portions and sent to the processors that have to be used. After that each processor is performing computations with the data assigned to it and preparing locally output data.

(h) Post-Processing Module: this module is called at the end of the computations. It combines the local output data files prepared from each processor during the run and stores them in common large files, which can after that be used to visualize and/or animate the results from the run.

During the computations each processor works with its own data, but the processors must exchange data from time to time. In UNI-DEM this has to be done at the end of every time-step. The standard tool, which is used to perform exchange of data among the processors is called MPI (Message Passing Interface); see [40].

OPTIMIZING THE COMPUTATIONAL PERFORMANCE

Achieving parallel computations is **necessary but not sufficient**. The computations, which are carried out at each processor, must also be optimized. The optimization of the computations will be discussed in this section. We shall use a very simple example in order to facilitate the understanding of the actions that are taken, but the ideas can easily be generalized.

Assume that

(a) the number of the selected processors is four,

(b) the space domain is discretized by using $\mathbf{N_x = N_y = 96,\ N_z = 1}$ and

(c) the chemical scheme with $\mathbf{N_q = 35}$ is selected.

Then the array, in which the concentrations are kept, is normally declared as $\mathbf{C(96, 96, 35)}$ and can be divided into four sub-arrays: $\mathbf{C_1(1:25, 96, 35)}$, $\mathbf{C_2(24:49, 96, 35)}$, $\mathbf{C_3(48:73, 96, 35)}$ and $\mathbf{C_4(72:96, 96, 35)}$ so that the first and the fourth sub-arrays contain $\mathbf{25}$ rows, while the second and the third sub-arrays contain $\mathbf{26}$ rows. Many of the other arrays used in the computational process have to be divided into four sub-arrays in a similar manner.

At each time-step the contents of rows $\mathbf{1-24}$ of array $\mathbf{C_1}$, rows $\mathbf{25-48}$ of array $\mathbf{C_2}$, rows $\mathbf{49-72}$ of array $\mathbf{C_3}$ and rows $\mathbf{73-96}$ of array $\mathbf{C_4}$ are updated (each of these sub-arrays is updated by its own processor).

The following communication tasks have to be performed at the end of the time-step:

(a) the first processor must send row $\mathbf{24}$ to the second processor, while the second processor must send row $\mathbf{25}$ to the first processor,

(b) the second processor must send row $\mathbf{48}$ to the third processor, while the third processor must send row $\mathbf{49}$ to the second processor and

(c) the third processor must send row $\mathbf{72}$ to the fourth processor, while the fourth processor must send row $\mathbf{73}$ to the second processor.

In this way row $\mathbf{25}$ is only used as an inner boundary condition on the first processor, rows $\mathbf{24}$ and $\mathbf{49}$ are used as inner boundary conditions on the second processor, rows $\mathbf{48}$ and $\mathbf{73}$ are used as inner boundary conditions on the third processor and row $\mathbf{72}$ is used as an inner boundary condition on the fourth processor. It must be mentioned here that boundary conditions are only needed in the treatment of the horizontal transport (no boundary condition is needed in the

chemical part). It should also be mentioned that the treatment of the boundary conditions is easy both because the horizontal sub-model is linear and because explicit numerical methods are normally used to handle it (see more details in [2, 3]).

MPI routines are called to perform the communications listed above. Note too that the amount of data, which have to be communicated, is not big (only one row for the first and the last processors and two rows for the remaining processors).

In order to explain better the difficulties, which appear during the computational process during the treatment of the chemical reactions, it is much more convenient to rewrite the arrays mentioned above as two-dimensional arrays $\mathbf{C(M,N)}$, $\mathbf{C_1(M_1,N)}$, $\mathbf{C_2(M_2,N)}$, $\mathbf{C_3(M_3,N)}$ and $\mathbf{C_4(M_4,N)}$ where $\mathbf{M = 9216,\ N = 35}$ and the values of $\mathbf{M_i,\ i = 1, 2, 3, 4,}$ can easily be obtained by using the above representation of the arrays $\mathbf{C_i,\ i = 1, 2, 3, 4,}$ as three-dimensional arrays. It is clear that some other arrays that are needed in the computational process should be represented in the same manner. In order to unify the considerations (in order to cover both the case where no parallel computations are to be applied and the case where computations at any of the selected processors are to be performed) we shall assume that arrays of the type $\mathbf{\overline{C}(\overline{M},N)}$ are involved in the computations. In the latter representation $\mathbf{\overline{M}}$ is the number of the involved grid-points (all grid-points when only one processor is used), while $\mathbf{N}$ is the number of the studied by the model chemical species.

Consider now the performance of the horizontal transport and assume that explicit numerical methods are used. Then the major part of the computations consists of basic linear algebra operations (mainly matrix-vector multiplications). The important issue is that the consequent components of the columns of array $\mathbf{\overline{C}(\overline{M},N)}$ and of the columns of all corresponding arrays are successively used in the computations. This is a very efficient operation when the FORTRAN language is applied because the data needed in the computations stay during a long time in cache. Thus, the calculations during the horizontal transport are not causing computational difficulties.

The treatment of the chemical sub-model is not so easy. In this part the elements of array $\mathbf{\overline{C}(\overline{M},N)}$ and all corresponding arrays are to be handled by rows and this

is very inefficient when the first dimension of arrays is very large, because the computer has frequently to change the contents of its cache memory. Therefore, some additional efforts are needed in the efforts to achieve efficiency in the treatment of the chemical sub-model (4).

Assume that a module, which will be called the **box-model**, for performing all chemical reactions at a given value $\mathbf{I}$, $\mathbf{I = 1, 2, \ldots, \overline{M}}$, has been prepared. Then the standard way for performing the chemical reactions at the current time-step can be written as shown in Fig. **1**.

```
C
C        Performance of the chemical reactions at a given time-step
C
         DO  I = 1, M
             Call the box-model
         END  DO
C
```

Figure 1: Straightforward but inefficient performance of the chemical reactions at a given time-step.

Realizing that in the box-model it is necessary to perform successfully all chemical reactions, one can try to rewrite the piece of code given in Fig. **1** as a double loop in the form shown in Fig. **2**. This will not be very helpful because the stride in the inner loop is not one but $\overline{\mathbf{M}}$ and this fact will normally cause many changes of the contents of the cache memory.

```
C
C        Performance of the chemical reactions at a given time-step
C
         DO  I = 1, M
             DO  J = 1, N
                 Perform the chemical reactions involving the J'th chemical species
             END  DO
         END  DO
C
```

Figure 2: Performing the chemical reactions in an inner loop of length **N**.

One can, of course, reverse the loops as shown in Fig. **3**. The performance will still be very slow, because while the chemical reaction number $\mathbf{J}$ is updated, the code will need the values of the concentrations of some of other chemical species, some of the species with numbers $\mathbf{1, 2, \ldots, J-1, J+1, \ldots, N}$ in row $\mathbf{I}$. This will also cause frequent changes of the contents of the cache memory.

```
C
C       Performance of the chemical reactions at a given time-step
C
        DO  J = 1, N
            DO  I = 1, M
                Perform the chemical reactions involving the J'th chemical species
            END  DO
        END  DO
C
```

Figure 3: Reversing the order of the loops in Fig. **2**.

The problem with the code shown in Fig. **3** is that

(a) the length of the inner loop is normally very large and

(b) during the performance of the computations for all values of index $\mathbf{I}$ some concentrations with indices $\mathbf{1, 2, \ldots, J-1, J+1, \ldots, N}$ in row $\mathbf{I}$ are, as was mentioned above, needed.

This means that additional improvements are needed. More precisely, it is necessary to reduce the length of this loop. In other words, instead of working with array $\mathbf{C(M, N)}$ we should try to carry out computations with several smaller arrays. This can be achieved in the following way. Let $\mathbf{NSIZE = M / NCHUNKS}$ (it is convenient here to assume that $\mathbf{NSIZE}$ is an integer, *i.e.* that $\mathbf{NCHUNKS}$ is a devisor of $\mathbf{M}$). Then we can think that the large array $\mathbf{C(M, N)}$ and all the corresponding arrays with the same dimensions (the number of the arrays in UNI-DEM the dimensions of which are $\mathbf{M}$ and $\mathbf{N}$ is eight) is divided into a series of smaller arrays which in this chapter will be called chunks. The number of chunks is $\mathbf{NCHUNKS}$ and the leading dimension of each of them is $\mathbf{NSIZE}$ while $\mathbf{N}$ is

the second dimension. It is clear now that if **NSIZE** is chosen so that the chunks can stay in the cache memory of the available computer then the device shown in Fig. **4** will be efficient.

```
C
C       Performance of the chemical reactions at a given time-step
C
        DO  ICHUNK = 1, NCHNKS
              Copy chunk  ICHUNK  from some of the eight large arrays with dimensions
              M̄  and  N  in the corresponding small arrays with leading dimension
              NSIZE  and second dimension  N
C
C       Carry out the computations in the double loop
C
        DO  J = 1, N
              DO  I = 1, NSIZE
                    Perform the chemical reactions involving the J'th chemical species
              END  DO
        END  DO
C
C       Finish the computations in the chemical sub-model
C
              Copy the information which will be needed in the further computations
              back in the right locations of the corresponding large arrays
        END  DO
C
```

Figure 4: An efficient code for the calculations in the chemical part, which is based on the use of appropriately chosen small arrays (chunks), is given in this figure.

The computational work which is carried out before the double loop in Fig. **4** and after it is extra (in comparison with the computational work used in the pieces of codes presented in Figs. **1** – **3**). Some data must be copied from the large arrays to the small arrays (the chunks) before the double loop and from the small arrays (the chunks) to the large arrays after it. However, the additional time needed to perform these copies is fully compensated by the fact that the data stays in the cache memory during the performance of the double loop and that is why the computations are performed in a more efficient way. This will be demonstrated by numerical examples in the next section.

As mentioned above the number of large arrays in UNI-DEM that are needed during the chemical part is eight. However, three of these arrays are helping

arrays and they are used only during the computations in the chemical part. Therefore, there is no need to declare these arrays as large arrays. Only three chunks are to be declared and used in the double loop in Fig. **4**. This fact has double effect:

(a) the number of copies which have to be made before the double loop is reduced from eight to five and

(b) the global storage needed in UNI-DEM is reduced because the use of three very large arrays is replaced by application of three rather small array.

Also the amount of computational work after the double loop can be reduced. The number of copies from the small arrays (the chunks) back to the large arrays that has to be carried out after the double loop can be reduced from five to three, because some information (for example, the temperatures) is needed during the treatment of the chemical reactions, but it is not modified in the double loop of Fig. **4** and, thus, there is no need to copy this information back to the large arrays.

It is quite obvious how the ideas discussed in this section for the special example can be extended for the cases where

(a) the number of processor is not four but for example $\mathbf{p}$ where $\mathbf{p}$ is greater than four and

(b) the number of grid points is not $\mathbf{N_x = N_y = 96}$ but some of the other two values which can be used in UNI-DEM.

The extension to the three-dimensional case where $\mathbf{N_z = 10}$ is slightly more complicated, but also here the approach is very straightforward. It would be convenient to declare the large arrays for example as $\hat{\mathbf{C}}(\mathbf{N_x, N_y, N_q, N_z})$ and to work in the same way as above during the treatment of the horizontal transport and the chemical reactions. It is necessary to perform some copies when the vertical transport is handled.

It is necessary to find some optimal (or at least good) value of parameter **NSIZE** or, in other words, to determine how large the small arrays (the chunks) should be.

The choice depends of the size of the cache discs on the available computer. Some experiments will be needed in order to determine a good value of this parameter.

It must be emphasized that the ideas used in the preparation of the improved devise, which was presented in Fig. **4**, are very general. All these ideas do not depend on the particular numerical methods used in the chemical sub-model. Moreover, they can be used both on shared memory parallel computers (under OpenMP) or on distributed parallel architectures (under MPI). Even if the computer is not parallel, *i.e.* if it is a sequential computer, the ideas from this section are still applicable and will lead to some savings both of the computing time and the storage requirements, because nearly all modern computers have cache memories. This is also true even for PCs.

NUMERICAL EXPERIMENTS

In this section we shall try to answer the following three important questions:

(a) Which of the two choices, OpenMP or MPI, is better when UNI-DEM is used?

(b) How to determine an optimal or at least a good value of parameter **NSIZE**?

(c) How to run the code on some computers with more advanced architectures?

The conclusions, which will be drawn in this section, are based on a long series of runs performed by using many parallel computers.

Runs Based on the Use of Open MP and MPI

It is clear that on classical distributed parallel computers, where each processor has its own memory, it is necessary to apply MPI. However, on many modern computers with shared memory, one can also simulate message passing communications. Therefore, it is worthwhile to investigate what is the better way of using UNI-DEM on such computers.

Results obtained on **8** processors of a SUN computer in the Danish Centre for Scientific Computing at the Technical University of Denmark (see [41]) are given in Table **2**. It is immediately seen that the MPI option performs much better than the OpenMP option.

The same run was performed by using **32** processors instead of **8** processors. One should expect the computing times to be reduced by a factor of **4** when the number of processors is four times larger (it is often said that the code is scalable when such a requirement is satisfied). This is not true in the case where the OpenMP option is used (the results shown in Table **3** indicate that the reductions of the computing times are by factor between two and three). For the MPI options the reduction factors are very close to four (in the chemical reactions part the factor is even slightly greater than four). This means that not only is the MPI option faster than the OpenMP option, but it also scales better when the number of the processors is increased.

Table 2: Computing times that are obtained when UNI-DEM is run with $\mathbf{N_x = N_y = 480}$ and $\mathbf{N_z = 1}$ on **8** processors are shown in this table. The computer used in this run is a SUN located in the Danish Computing Centre at the Technical University of Denmark. The number of time-steps performed in this run is **1000** and the computing times are measured in seconds.

Physical Processes (and Total Computing time)	**Running the OpenMP Option**	**Running the MPI Option**
Horizontal Transport	2618	457
Chemical Reactions	1339	688
Total Computing Time	4011	1281

Table 3: The same problem as in Table **2**, but now it is run on **32** processors. The computing times are again given in seconds. Under the columns "Speed-up" the factors by which the computing times are reduced when **32** processors are used instead of **8** processors are shown.

Physical Processes (and Total Computing Time)	**OpenMP Option**		**MPI option**	
	Time	**Speed-Up**	**Time**	**Speed-up**
Horizontal Transport	1001	2.61	120	3.81
Chemical Reactions	607	2.21	171	4.02
Total Computing Time	1771	2.26	348	3.68

It is not very clear why the MPI version of UNI-DEM performs better than its OpenMP version. The division of large array $\overline{\mathbf{C}}(\overline{\mathbf{M}},\mathbf{N})$, which contains the concentrations, and the division of all corresponding arrays with the same dimensions into much smaller arrays $\overline{\mathbf{C}}_{\mathbf{i}}(\overline{\mathbf{M}}_{\mathbf{i}},\mathbf{N})$ where $\mathbf{i}=\mathbf{1},\mathbf{2},\ \ldots,\mathbf{p}$ ($\mathbf{p}$ being the number of the processors needed) is probably the reason for this behaviour. The data is more regularly distributed when this action is carried out. The other possibility is that the MPI option is implemented in a much better way than the OpenMP option.

The MPI version of UNI-DEM is mainly used in the air pollution studies, because it performs much better than the OpenMP option. This is illustrated in this section by the results shown in Tables **2** and **3**. However, it is necessary to point out that many other numerical experiments have been carried out and the results were essentially the same.

Funding Appropriate Values for the Size of the Chunks

It is important to obtain good values of parameter **NSIZE** in the efforts to obtain efficient performance. Results obtained by running the code on several computers are given in this sub-section.

The first example is obtained by running UNI-DEM on the SUN computers at the Danish Centre for Scientific Computing ([41]). Eight processors were used. The discretization was performed by specifying $\mathbf{N_x}=\mathbf{N_y}=\mathbf{480},\ \mathbf{N_z}=\mathbf{10}$ and the chemical scheme with $\mathbf{N_q}=\mathbf{35}$ was used. A short time-interval of **42** hours was used and the results, shown in Table **4**, are given in seconds.

As should be expected reductions are obtained only in the chemical part. On this computer, the best results were achieved for **NSIZE= 24** and the computing time in this part of the code was reduced by more than **30%** (see the fifth column in Table **4**).

The case **NSIZE= 1** corresponds to the use of the smallest chunks, while **NSIZE= 28800** corresponds to the selection of the biggest chunks. The results in Table **4** indicate that the chunks should not be very small or very large. These conclusions were confirmed by many other runs.

Table 4: Numerical results obtained by using three different values of **NSIZE** on a SUN computer. "HOR", "CHEM", "VERT", "COMM" and "Total" refer respectively to horizontal transport, chemical reactions, vertical transport, communications and total computing time. Eight processors were used and the computing times are given in seconds. The relative parts of the computing times are given under "Part". The factors by which the computing times are reduced when eight processors are used instead of one processor are given under "Speed-up". Note that the speed-ups are greater than the expected values (the acceleration of the speed of computations is super-linear).

Process	NSIZE=1			NSIZE=24			NSIZE=28800		
	Time	Part	Speed-up	Time	Part	Speed-up	Time	Part	Speed-up
HOR	13606	46.2%	19.2	13515	52.7%	20.1	13374	28.9%	20.1
CHEM	10398	35.3%	8.3	6681	26.0%	8.5	25888	56.0%	8.8
VERT	2830	9.5%	14.4	2802	10.9%	15.1	2709	5.9%	15.2
COMM	2316	7.9%	-	2340	9.1%	-	3925	8.5%	-
Total	29449	100.0%	13.3	25654	100.0%	14.5	46210	100.0%	11.7

Runs on several other computers have been carried out. Some of the obtained results are given in Tables **5** and **6**.

Table 5: Computing times, measured in seconds, when three different values of parameter **NSIZE** are used on three different computers. One processor only is used in this run. Discretization parameters $\mathbf{N_x = N_y = 96}$ and $\mathbf{N_z = 1}$ were specified for this experiment. The chemical scheme with $\mathbf{N_q = 35}$ was chosen.

NSIZE	Fujitsu	SGI ORIGIN 2000	IBM SMP
1	76964	14847	10313
48	2611	12114	5225
9216	494	18549	19432

Fujitsu is a vector computer. On this computer the best performance is obtained when the largest chunk is used (in fact when no division of the large arrays to smaller arrays is performed in this case).

SGI ORIGIN 2000 is a shared memory computer. On this computer the best result was obtained when $\mathbf{NSIZE = 48}$ was applied. The improvement is not very significant (no more than 20%).

IBM SMP is a computer with a rather complicated structure but for this run (where only one processor is used) this is not very important. Also here the best

result was obtained when $\mathbf{NSIZE = 48}$ was applied, but now the improvement is rather large (about 50%).

Some results run on an IBM Blue Gene computer are presented in Table **6**. It is seen that best results are obtained when $\mathbf{NSIZE = 16}$ is selected. The achieved reduction is about **34%** for the chemical sub-model and about **30%** for the total time.

Table 6: Results from an experiment carried out on the IBM Blue Gene computer in Sofia (Bulgaria). The disctretization parameters used are $\mathbf{N_x = N_y = 480}$ and $\mathbf{N_z = 1}$ were applied and the chemical scheme with $\mathbf{N_q = 35}$ was chosen. The number of processors used was 120. The computing times are given in hours.

NSIZE	Chemistry Time	Total Time
1	34.82	43.20
2	29.62	37.69
4	25.77	33.40
8	22.74	30.41
16	**22.27**	**30.10**
32	23.67	31.06
64	29.45	36.98

Using Computers with more Complex Architectures

The IBM SMP computer (which was already mentioned in the previous sub-section) has a complicated structure that allows us to achieve parallelization on two levels. It consists of several nodes (the particular computer used in our experiments has only two nodes, but this still allows us to perform many meaningful tests). Each node contains eight processors, which have shared memory. This means that each node works in a shared memory mode.

Since each node has its own memory, several nodes work together in a distributed memory mode and, thus, message passing (carried out by using MPI tools) is needed in the communications between the nodes.

It is clear from this short description of the major features of the IBM SMP computer that efforts on two levels are needed in order to obtain an efficient parallel code:

(a) it is worthwhile to apply of parallelization rules within each node (the shared memory mode has to be utilized during this phase), and

(b) it will be more efficient to develop additionally and to exploit in the runs of very big tasks some rules for parallelization across the nodes (in this part the features of message passing machines have to be exploited).

This information (about the major tasks, which have to be resolved when the IBM SMP computer has to be exploited) is by no means complete, but it is quite sufficient to explain why the requirement for achieving high performance becomes more difficult. There are at least two reasons for this:

(a) one should exploit simultaneously the features of both the shared memory machines and the distributed memory machines, and

(b) one should be very careful in the efforts to avoid conflicts between the devices used in the shared memory code and the devices used in the distributed memory code.

While the task of achieving high performance becomes more difficult when such a complicated architecture as the IBM SMP computer is used, it is also true that if the work is carefully done, then very good results can be achieved.

It is intuitively clear that the first task, which has to be solved, is to optimize the code when **it is run on one processor only** (*i.e.*, when it is run in a sequential mode). This is why we first tried to increase the performance of the code when it is run on one processor. Results, which were obtained by the code before the attempts to optimize it for runs on the IBM SMP computer, are given in Table **7**.

The results presented in Table **7** show clearly that the chemistry module was the most time-consuming part of the code before the optimization. More than **80%** of the computing time was spent in this part. Therefore, a lot of efforts (the major rules which have to be applied in this phase of the preparation of the code have been described in the previous section) were spent in the attempts to improve the performance of this part of the code. Some results, which are obtained when the

optimized code was run, were given in the fourth column of Table **5**. Comparing the total computing time in Table **7** with the computing time achieved for $\mathbf{NSIZE = 48}$ in the fourth column of Table **5** indicates that the optimized code is about **four times faster** than the first version of UNI-DEM.

Table 7: Results obtained by the code before optimizing it for runs on IBM SMP. Only one processor is used. Discretization parameters $\mathbf{N_x = N_y = 96}$ and $\mathbf{N_z = 1}$ were specified for this experiment. The chemical scheme with $\mathbf{N_q = 35}$ was chosen. No chunks are used (the length of **NSIZE** was **9216** in this experiment).

Physical Process and Total Time	Computing Time	Part of the Total Time
Horizontal Transport	3013	15.51%
Chemical sub-Model	16147	83.09%
Initialization	2	0.00%
Input Operations	50	0.26%
Output Operations	200	1.13%
Total Computing Time	19432	100.00%

Note that the same numerical algorithms are used in all experiments, the results of which are presented in this chapter. This fact indicates that **it is necessary to apply efficient numerical algorithms, but this might not be sufficient** when large-scale mathematical models are to be handled. It is sometimes even more important to organize the computations properly. Moreover, it is seen from Table **5** that this action (achieving a proper organization of the computational operations) is computer dependent.

When the code was optimized for runs on one processor, it was not very difficult to achieve parallelism both within a node and across the nodes. The rules described in the previous section are used and some results are presented in Tables **8** and **9**.

SOME REMARKS ABOUT THE USE OF COMPUTATIONAL GRIDS

One of the great challenges in the near future will be the utilization of computer grids in the treatment of very large applications involving tasks from meteorology, air pollution and several other related areas of science and engineering. There are many problems in air pollution modeling which we want to solve, even problems

that **must** be solved, but it is impossible to handle these problems on the presently available computers. Such problems must probably be handled on computer grids.

Table 8: Computing times (measured in seconds) obtained by using 1, 2, 4 and 8 processors in one node as well as by using all 16 processors in the two nodes of the available IBM SMP computer. The same parameters as those in Table **7** are used.

Number of Processors	Horizontal Transport	Chemical sub-Model	Total Computing Time
1	933	4185	5225
2	478	1878	2427
4	244	1099	1405
8	144	521	799
16	62	272	424

Table 9: Speed-up and efficiency achieved by using 2, 4 and 8 processors in one node as well as by using all 16 processors in the two nodes of the available IBM SMP computer. The speed-up is measured as (computing time on one processor) / (computing time on **p** processors). The efficiency is measured as **100** (speed-up on **p** processors) / (**p**). The same parameters as those in Table **7** are used.

Number of Processors	Horizontal Transport		Chemical Sub-Model		Total Computing Time	
	Speed-up	Efficiency	Speed-Up	Efficiency	Speed-Up	Efficiency
2	1.95	98%	2.33	112%	2.15	108%
4	3.82	96%	3.81	95%	3.72	93%
8	6.48	81%	8.01	100%	6.54	82%
16	15.01	94%	15.39	96%	12.32	72%

The results in tables presented in this section indicate that it is possible to achieve good results when the rules described in the previous section are properly implemented.

Computer grids open the way to resolving a series of challenging tasks. As an example let us mention the task of running several **loosely connected** large-scale models in order to treat a set of complex problems involving

(a) weather forecasts on different regional scales (starting with results obtained on a global scale),

(b) weather forecasts on an urban scale (perhaps in parallel for several urban areas),

(c) air pollution forecasts on different regional scales,

(d) air pollution forecasts on an urban scale (perhaps in parallel for several urban areas),

(e) treatment of the output results in order to prepare them for the people who will use them (utilizing data mining algorithms and high-speed visualization tools),

(f) sending the relevant data to appropriate media (TV stations, radio stations, relevant Internet sites, different kind of mobile telephones, *etc*.).

It is clear that computer grids will be very powerful tools in the solution of the very challenging tasks related to the possibility to treat efficiently the set of problems described above. Such sets of problems are at present solved only by imposing many simplifying (and very often not physical) assumptions. At the same time, it is also clear that a lot of difficulties must be overcome in the efforts to run such complex tasks efficiently on a computer grid. The greatest difficulties are the tasks of

(a) achieving reliable and robust transition from one scale to another,

(b) communicating in a fast and secure way relevant data from one part of the computational grid to another, and

(c) preparing the final results, which should be easily understandable by the recipients.

The example given above shows that grid computing appears to be a very promising tool both for improving the performance of the existing large-scale air pollution models and for the development and successful treatment of sets of many models, which are different but have to be coupled in order to study interactively some important relations between different physical processes. The sets of models can additionally contain economical models, models controling the

energy development in a given region of the world, models controling the sustainability of the industrial and agricultural development, models controling the critical levels of the air pollution levels, *etc*. It is quite clear that

(a) the air pollution models are only a part of such big sets of mathematical models (and not necessarily the most time-consuming part), and

(b) the computational tasks that are to be resolved, when such big sets of models are to be used (perhaps by running hundreds of carefully chosen scenarios), are enormous.

The great potential power of the computational grids can make it possible to resolve the enormous tasks related to the treatment of big sets of coupled models when this power is efficiently exploited. Several reasons for this expectation are listed below.

(a) Grid computing is a way to handle tasks that cannot be approached without an enormous amount of computing power.

(b) Grid computing suggests that the resources of many computers can be cooperatively and perhaps synergistically managed as collaboration toward a common objective.

(c) Grid computing allows the users to apply in a more efficient way the distributed storage of the data, which are needed as input data in big applications (in the case of large-scale air pollution models the most relevant input data are the enormous sets of meteorological data and emission data) when these data are produced and kept in different sites (which is practically always the case).

(d) A computational grid is potentially able to make more cost-effective use of a given amount of computer resources.

It should be reiterated here that such enormous tasks, as those sketched above, cannot be treated at present without imposing some crude and as a rule non-physical simplifications.

More information about the grid computing could be found in [42, 43].

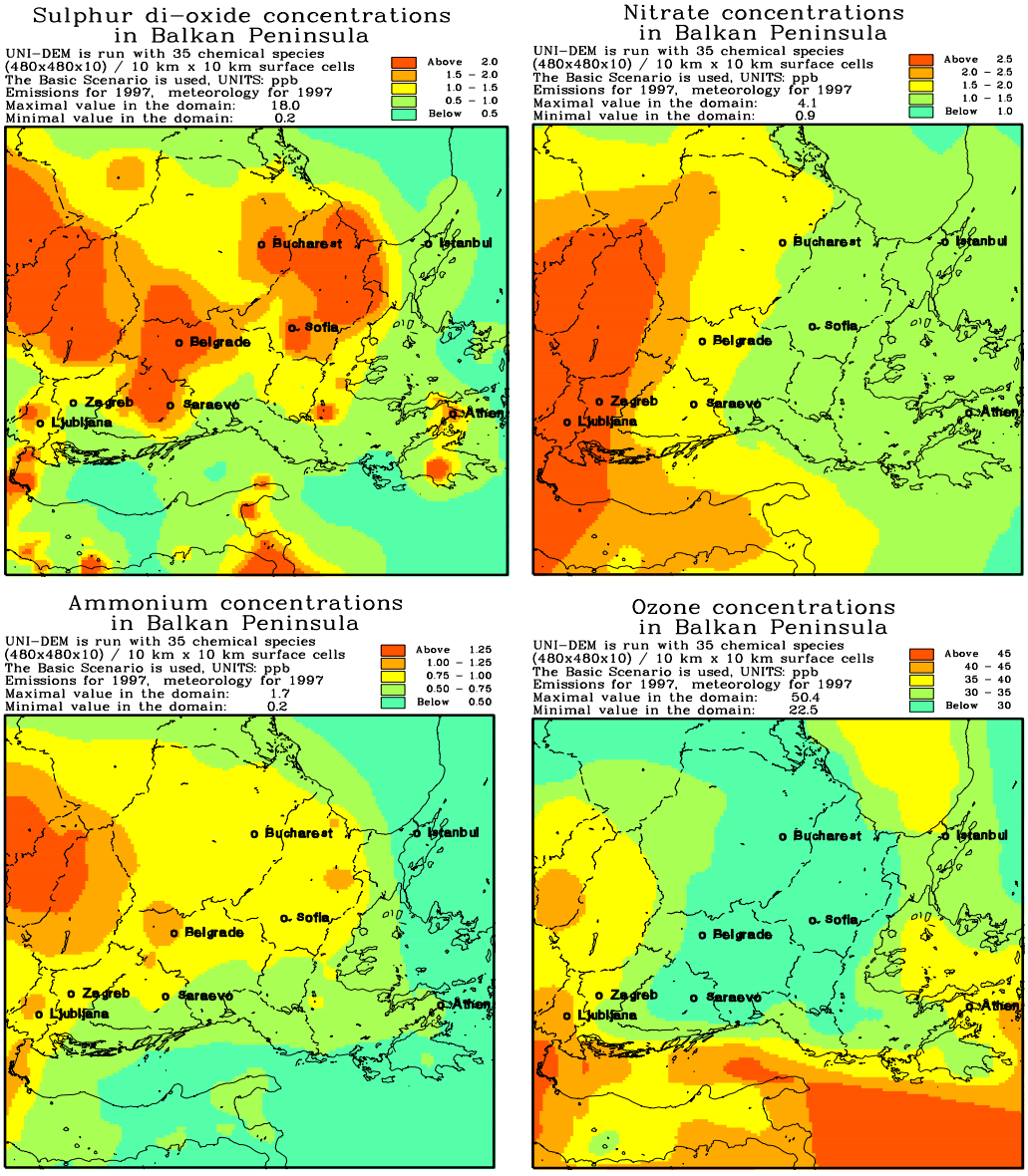

Figure 5: Distribution of the concentrations of sulphur di-oxide (upper left-hand-side plot), nitrate (upper right-hand-side plot), ammonium (lower left-hand-side plot) and ozone (lower right-hand-side plot) in the Balkan Peninsula. The units are ppb.

APPLICATIONS OF UNI-DEM

The development of an extremely efficient code allowed us to apply UNI-DEM in many long-term investigations of air pollution phenomena and trends in Europe.

Many references were quoted in the beginning of this chapter. Two important examples will be given in this section. We shall shortly discuss:

(a) the long-range transport of air pollution to the Balkan Peninsula and

(b) the impact of climatic changes to pollution levels around some of the biggest European cities.

Long-Range Transport of Air Pollutants to the Balkan Peninsula

The long-range transport of air pollutants to the Balkan Peninsula can be studied by performing two runs: a first run where all emission sources in Europe are active and a second run where the emission sources in the countries located in the Balkan Peninsula are set to zero. The results from the first run will show the real pollution levels, while the results from the second run will give the approximate values of the transport of air pollutants from Europe to the Balkan Peninsula. Some examples from this study are given in Figs. **5** and **6**.

The distribution of the concentrations of sulphur di-oxide, nitrate, ammonium and ozone in the studied in this paper area are given in Fig. **5**. It is seen that while the sulphur di-oxide levels are rather high (especially in Bulgaria), the nitrate, ammonium and ozone levels in the Balkan Peninsula are relatively low.

The influence of the European emission sources on the pollution levels in the Balkan countries is shown in Fig. **6**. The changes are given in percent. Denote by **A** the result at a given grid-point in the case where the scenario without emission sources from the Balkan countries is used. Denote by **B** the corresponding result from the Basic Scenario (where all European emission sources are taken into account). Then the quantities **100 A / B** are given in the plots in Fig. **6**.

Three important conclusions can be drawn from the plots in Fig. **6**:

(a) The influence of the European emission sources on the sulphur di-oxide levels is relatively small. This is not a big surprise because the sulphur di-oxide levels in the Balkan Peninsula are rather high.

(b) The influence of the European emission sources on the nitrate and ammonium levels is considerably larger, while the influence on the ozone concentrations is very high.

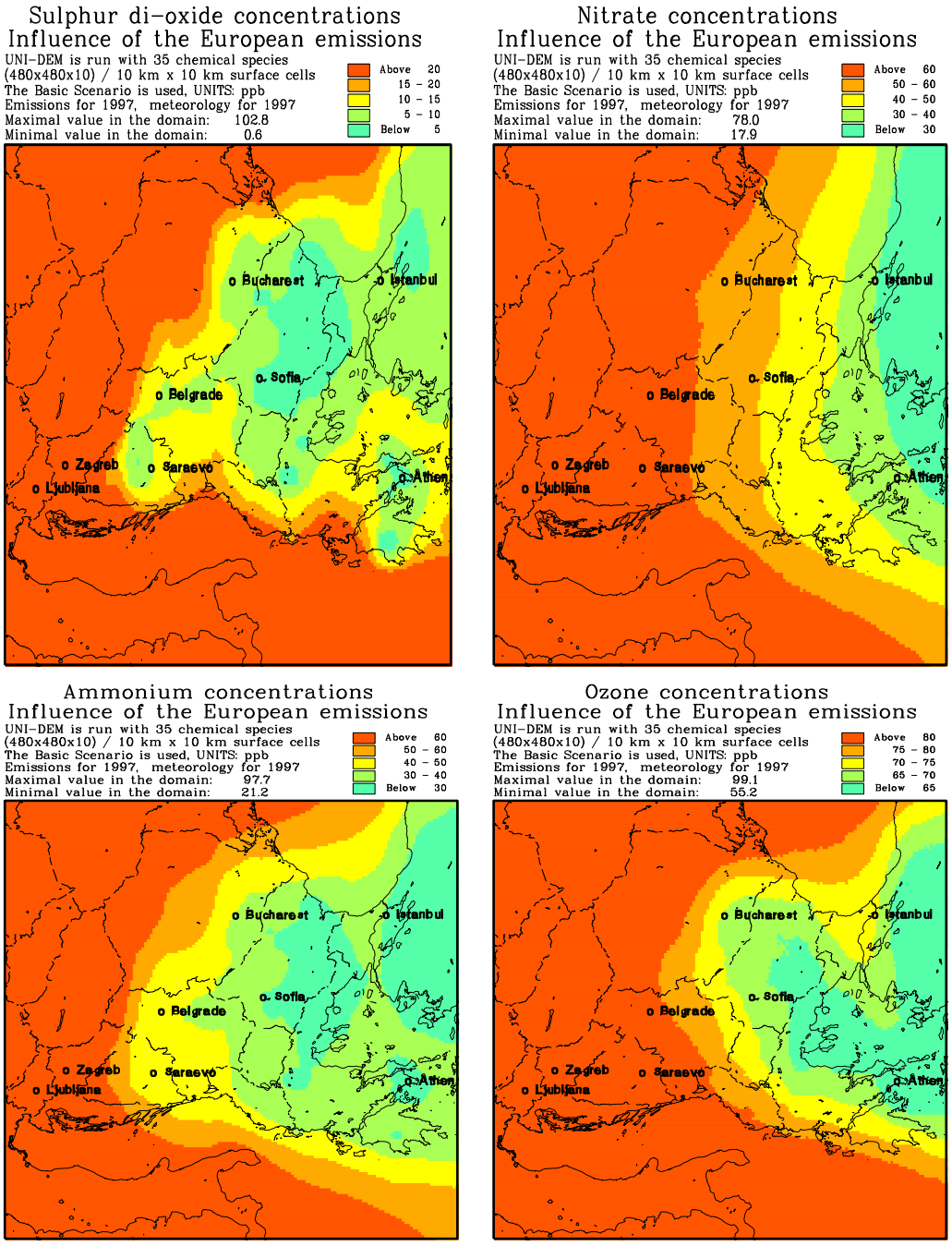

Figure 6: Contributions of the European emission sources to the concentration levels of sulphur di-oxide (upper left-hand-side plot), nitrate (upper right-hand-side plot), ammonium (lower left-hand-side plot) and ozone (lower right-hand-side plot) in the Balkan Peninsula. The units are percent.

(c) The influence of the European emission sources is greater in the Western and Northern parts of the Balkan Peninsula.

The fact that the influence of the European emission sources on the pollution levels is very big for some chemical species indicates that the long-range transport should properly be taken into account. This explains why it is always more preferable to use a large spatial domain also when limited areas of this domain are of interest.

Impact of Climatic Changes on the Pollution Levels Around Big European Cities

High pollution levels might cause some damages on plants, animals and human beings. One of the dangerous for human beings pollutants is ozone. Several critical levels for ozone have been established in the European Union (EU) as well as in other parts of the world. Some of these critical levels are legislated in the EU Ozone Directive [44].

Assume that $\mathbf{c_{max}}$ is the maximum of the eight-hour averages of the calculated by some model or measured ozone concentrations in a given day at site $\mathbf{A}$. If the condition $\mathbf{c_{max} > 60}$ ppb is satisfied at least once in the day under consideration, then the expression a **bad day** will be used for such a day at site $\mathbf{A}$. **Bad days** can have damaging effects on some groups of human beings (first and foremost, on people who suffer from asthmatic diseases). Therefore, the number of such days should be reduced as much as possible. Two important aims are stated in the Ozone Directive issued by the EU Parliament in year 2002 (see again [44]):

(a) **Target Aim.** The number of "bad days" in any site of the European Union should not exceed 25 after year 2010.

(b) **Long-Term Aim.** No "bad day" should occur in the European Union (the year after which the long-term aim has to be satisfied is not specified in the EU Ozone Directive).

Climate changes are causing another challenging problem for the modern society. The quick climate changes have many different consequences. The impact of these changes on the pollution levels is one of the consequences and this

consequence must be carefully investigated by studying the relationship between climatic changes and high pollution levels.

The impact of the climatic changes on numbers of **bad days** in eight sub-urban areas in Europe will be demonstrated in this section. The areas around the cities listed in Table **10** were selected. These areas are densely populated and increases of the numbers of "bad days" may cause damaging effects on large number of human beings in these areas.

Several climatic scenarios were developed (by taking into account the recommendation of the **Intergovernmental Panel on Climate Change**, IPCC, given in [45, 46]; see also [47-51]). These scenarios were run together with the Basic Scenario (where the actual emissions and meteorological conditions are kept unchanged) over a long time-interval of 16 years. Results obtained with the third climatic scenario are presented in Figs. **7** – **9**.

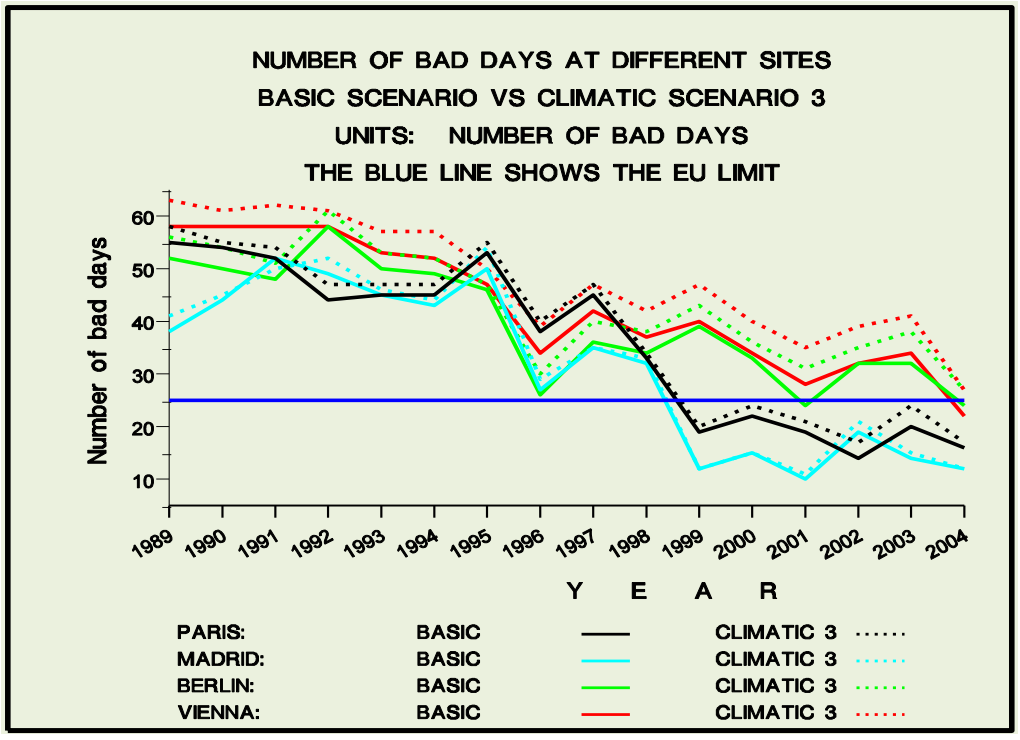

Figure 7: Variation of the bad days in Paris, Madrid, Berlin and Vienna, which are obtained by using the Basic Scenario and one of the climatic scenarios for the time-interval of 16 years.

Table 10: Eight big European cities for which the numbers of bad days per year were calculated by using UNI-DEM are given in this table.

No.	City	Country	Population in City	Population in Metropolitan Area
1	Vienna	Austria	1.7	2.3
2	Berlin	Germany	3.4	5.0
3	Madrid	Spain	3.2	5.1
4	Paris	France	2.2	12.0
5	London	Great Britain	7.5	8.3
6	Rome	Italy	2.7	4.0
7	Milan	Italy	1.3	7.4
8	Moscow	Russia	10.0	

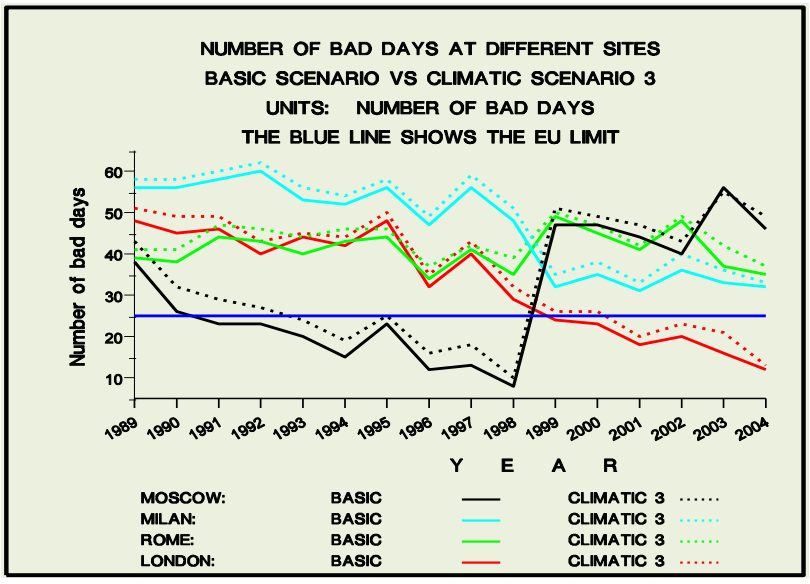

Figure 8: Variation of the bad days in Moscow, Milan, Rome and London, which are obtained by using the Basic Scenario and one of the climatic scenarios for the time-interval of 16 years.

The results presented in Figs. **7-9** show clearly that the numbers of bad days, which in most of the cases exceed the critical levels established in the European Union, are in general further increased when the climatic scenario is used. The size of the increases (see Fig. **9**) is in most of cases about 10%, but could also be higher, up to 30%, which indicates that the number of bad days could be increased rather considerably in the future only because of the global increase of the temperature. This indicates that the climatic changes should seriously be taken

into consideration by the policy makers when measures to keep our environment clean are to be planned and taken.

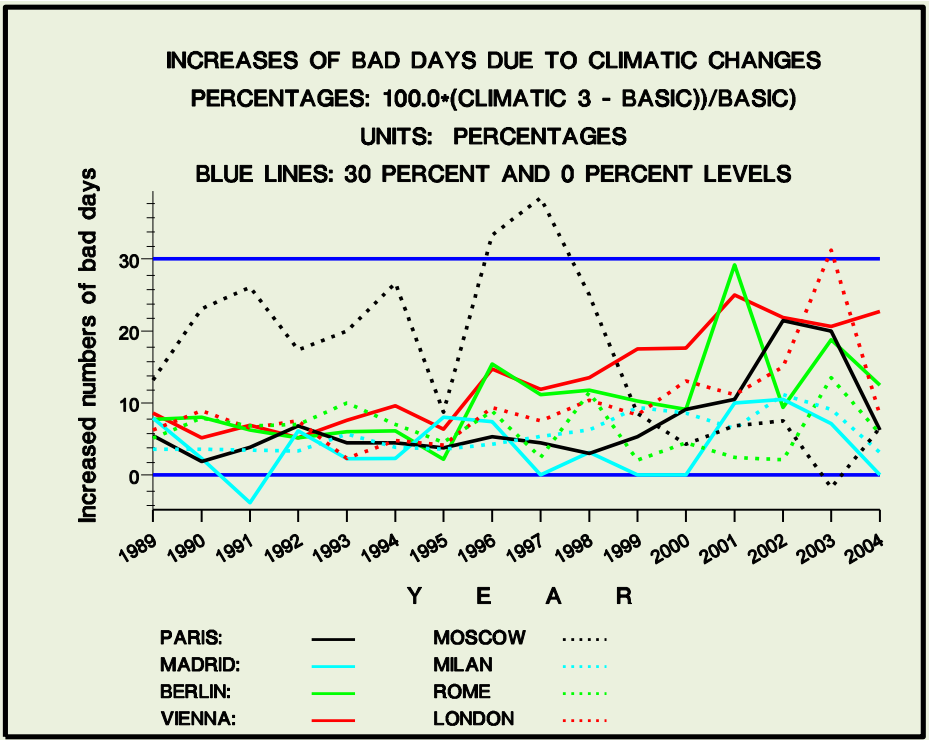

Figure 9: Changes of the numbers of "bad days" in percent (obtained by taking the differences of the numbers of "bad days found by the Climatic Scenario 3 and the Basic Scenario, multiplying this difference by 100 and dividing the results by the numbers of bad days found by the Basic Scenario).

CONCLUDING REMARKS

Large-scale air pollution models must be discretized on fine-resolution grids in order to avoid the appearance of numerical errors which are comparable with or greater than the errors that are caused by other reasons. Therefore, the numerical tasks that are to be handled are enormous and it is absolutely necessary to develop and use efficient methods when long series of simulations are to be run. The achievements obtained during the efforts to run efficiently a particular model, UNI-DEM, on parallel computers were discussed in this paper. The main ideas are pretty general and can also be used in connection with other large-scale models.

In spite of the fact that the obtained results were rather good, much more efforts in this field are still needed. This fact can shortly be explained as follows. Consider the computational problems, which are related to the study of the influence of the climate changes on the pollution levels. Some results from this study were presented and briefly discussed in Sub-section 8.2. The computational complexity of this problem is enormous. Indeed, it was necessary to run **14** scenarios over a time-period of **16** years on a fine grid (**480 x 480 x 10** cells and **35** chemical species resulting in systems of **80 640 000** equations that are to be handled in **209 664** time-steps per year). Therefore it was extremely difficult to complete successfully this study. The task of running so many scenarios over so long time-period can be successfully solved only if at least four requirements are simultaneously satisfied:

(a) fast but also sufficiently accurate numerical methods are to be implemented in the model,

(b) the cache memories of the available computers have to be efficiently utilized,

(c) codes which can be run in parallel have to be developed and used,

(d) reliable and robust splitting procedures have to be implemented.

The solution of sub-tasks (a) – (d) was discussed in this chapter. It must be emphasized here that it is **impossible** to handle the **14** scenarios over a time-period of **16** years on the available super-computers if the sub-tasks (a) – (d) are not efficiently solved. Even when this was done, as described in the previous sections, it took more than two years to compute output data from all **2688** runs (**14** scenarios x **16** years x **12** months) carried out in this study. This fact illustrates the great computational difficulties that are related to the investigation of various impacts of climatic changes on pollution levels.

It should be added here that the storage requirements (the need for huge input and output data files) are also enormous.

This indicates that it is highly desirable to achieve essential improvements in order to be able to handle even more difficult tasks. Such improvements will be a topic of future research.

ACKNOWLEDGEMENTS

We should like to express our gratitude to the specialists in the Danish Centre for Scientific Computing at the Technical University of Denmark [41], which gave us access to their computers and helped us to carry out many runs on their computers and the Bulgarian National Center for Supercomputing Applications (http://www.scc.acad.bg/ncsa/index.php/en/) giving access to the IBM Blue Gene/P computer in Sofia (Bulgaria). K. Georgiev was supported in part by the Grant DCVP-02/1 from the Bulgarian NSF.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

REFERENCES

[1] V. Alexandrov, W. Owczarz, P. G. Thomsen and Z. Zlatev: “Parallel runs of large air pollution models on a grid of SUN computers”, Mathematics and Computers in Simulations, Vol. 65 (2004), pp. 557-577.

[2] Z. Zlatev: “Computer treatment of large air pollution models”, Kluwer Academic Publishers, Dordrecht-Boston-London, 1995.

[3] Z. Zlatev and I. Dimov: “Computational and numerical challenges in environmental modeling”, Elsevier, Amsterdam, 2006.

[4] V. Alexandrov, A. Sameh, Y. Siddique and Z. Zlatev: “Numerical Integration of chemical ODE problems arising in air pollution models”, Environmental Modeling and Assessment, Vol. 2 (1997), pp. 365-377.

[5] Ø. Hov, Z. Zlatev, R. Berkowicz, A. Eliassen and L. P. Prahm, L. P: Comparison of numerical techniques for use in air pollution models with non-linear chemical reactions. Atmospheric Environment, Vol. 23 (1988), pp. 967-983.

[6] Z. Zlatev: “Application of predictor-corrector schemes with several correctors in solving air pollution problems", BIT, Vol. 24 (1984), pp. 700-715.

[7] Z. Zlatev: “Computational for general sparse matrices”, Kluwer Academic Publishers, Dordrecht-Boston-London, 1991.

[8] Z. Zlatev: “Partitioning ODE systems with an application to air pollution models”, Computers and Mathematics with Applications, Vol. 42 (2001), pp. 817-832.

[9] Z. Zlatev: “Massive data sets issues in air pollution modeling”. In “Handbook on Massive Data Sets in Science and Engineering” (J. Abello, P. M. Pardalos and M. G. C. Resende, eds.) pp. 1169-1220, Kluwer Academic Press: Dordrecht-Boston-London, 2002.

[10] Z. Zlatev: "Impact of future climate changes on high ozone levels in European suburban areas", Climatic Change, Vol. 101 (2010), pp. 447-483.

[11] Z. Zlatev, R. Berkowicz and L. P. Prahm: "Stability restrictions of time stepsize for numerical integration of first-order partial differential equations". Journal of Computational Physics, Vol. 51 (1983), pp. 1-27.

[12] Z. Zlatev, I. Dimov, I. Faragó, K. Georgiev, Á. Havasi and Tz. Ostromsky: "Solving advection equations by applying the Crank-Nicolson scheme combined with the Richardson Extrapolation", International Journal of Differential Equations, DOI:10.1155/2011/530840, (2011).

[13] Z. Zlatev, I. Faragó and Á. Havasi: "Stability of the Richardson Extrapolation together with the θ-method", Journal of Computational and Applied Mathematics, Vol. 235 (2010), pp. 507-520.

[14] Z. Zlatev, I. Faragó and Á. Havasi: "Richardson Extrapolation combined with the sequential splitting procedure and the θ-method", Central European Journal of Mathematics, Vol. 10 (2012), pp. 159-172.

[15] Z. Zlatev, K. Georgiev and I. Dimov: "Influence of climatic changes on pollution levels in the Balkan Peninsula". Computers and Mathematics with Applications, published online in August 2012.

[16] Z. Zlatev, Á. Havasi and I. Faragó: "Influence of climatic changes on pollution levels in Hungary and its surrounding countries". Atmosphere, Vol. 2 (2011), pp. 201-221.

[17] G. Georgiev and Z. Zlatev: "Studying air pollution levels in the Balkan Peninsula by using an IBM Blue Gene/P computer". International Journal of Environment and Pollution, Vol. 46, No, 1-2 (2011), pp. 97-114.

[18] Z. Zlatev and D. Syrakov: "A fine resolution modeling study of pollution levels in Bulgaria. Part 1: SOx and NOx pollution", International Journal of Environment and Pollution, Vol. 22, Issues 1-2 (2004), pp. 186-202.

[19] Z. Zlatev and D. Syrakov: "A fine resolution modeling study of pollution levels in Bulgaria. Part 2: High ozone levels", International Journal of Environment and Pollution, Vol. 22, Issues 1-2 (2004), pp. 203-222.

[20] I. Dimov, G. Geernaert and Z. Zlatev: "Impact of future climate changes on high pollution levels", International Journal of Environment and Pollution, Vol. 32, Issue 2 (2008), pp. 200-230.

[21] Z. Zlatev and L. Moseholm: "Impact of climatic changes on pollution levels in Denmark", Environmental Modeling, Vol. 217 (2008), pp. 305-319.

[22] S. Abdalmogith, R. M. Harrison and Z. Zlatev: "Intercomparison of inorganic aerosol concentrations in the UK with predictions of the Danish Eulerian Model", Journal of Atmospheric Chemistry, Vol. 54 (2006), pp. 43-66.

[23] C. Ambelas Skjøth, A. Bastrup-Birk, J. Brandt and Z. Zlatev: "Studying variations of pollution levels in a given region of Europe during a long time-period", Systems Analysis Modeling Simulation Vol. 37 (2000), pp. 297-311.

[24] A. Bastrup-Birk, J. Brandt, I. Uria and Z. Zlatev: "Studying cumulative ozone exposures in Europe during a 7-year period", Journal of Geophysical Research, Vol. 102 (1997), pp. 23917-23035.

[25] G. Geernaert and Z. Zlatev: "Studying the influence of the biogenic emissions on the AOT40 levels in Europe". International Journal of Environment and Pollution, Vol. 23, Issues 1-2, (2004), pp. 29-41.

[26] Á. Havasi, L. Bozó and Z. Zlatev: "Model simulation on transboundary contribution to the atmospheric 201ulphur concentration and deposition in Hungary", Időjárás, Vol. 105 (2001), pp. 135-144.

[27] Á. Havasi and Z. Zlatev: "Trends of Hungarian air pollution levels on a long time-scale", Atmospheric Environment, Vol. 36 2002, pp. 4145-4156.

[28] R. M. Harrison, Z. Zlatev and C. J. Ottley: "A comparison of the predictions of an Eulerian atmospheric transport chemistry model with experimental measurements over the North Sea". Atmospheric Environment, Vol. 28 (1994), pp. 497-516.

[29] I. Dimov, R. Georgieva, S. Ivanovska, Tz. Ostromský and Z. Zlatev: "Studying sensitivity of pollutant concentrations caused by variations of chemical rates", Journal of Computational and Applied Mathematics, Vol. 235, (2010), pp. 391-402.

[30] H. Hass, M. van Loon, C. Kessler, R. Stern, J. Mathijsen, F. Sauter, Z. Zlatev, J. Langner, V. Foltescu and M. Schaap: "Aerosol modeling: Results and intercomparison from European regional-scale modeling systems", GSF–National Research Center for Environment and Health, International Scientific Secretariat (ISS), EUROTRAC-2, Münich (this report is also available at: http://www.trumf.fu-berlin.de/veranstaltungen/events/glream/GLOREAM_PMmodel-comparison.pdf), 2004.

[31] M. Roemer, M. Beekman, R. Bergsröm, G. Boersen, H. Feldmann, F. Flatøy, C. Honore, J. Langner, J. E. Jonson, J. Matthijsen, M. Memmesheimer, D. Simpson, P. Smeets, S. Solberg, D. Stevenson, P. Zandveld and Z. Zlatev: "Ozone trends according to ten dispersion models", GSF – National Research Center for Environment and Health, International Scientific Secretariat (ISS), EUROTRAC-2, Münich, (available also at: http://www.mep.tno.nl/eurotrac/EUROTRAC-trends.pdf), 2004.

[32] P. Csomós, R. Cuciureanu, G. Dimitriu, I. Dimov, A. Doroshenko, I. Faragó, K. Georgiev, Á Havasi, R. Horváth, S. Margenov, Tz. Ostromsky, V. Prusov, D. Syrakov, and Z. Zlatev: "Impact of Climate Changes on Pollution Levels in Europe", Final report for a NATO Linkage Project (Grant 980505),2006: available at http://www.cs.elte.hu/~faragois/NATO.pdf,
http://www2.dmu.dk/atmosphericenvironment/Climate%20and%20Pollution,
http://www.umfiasi.ro/NATO.pdf,
http://www.personal.rdg.ac.uk/~sis04itd/MyPapers/climatic_scenarios_NATO.pdf,
http://www.softasap.net/ips/climatic_scenarios_NATO.pdf,
http://www.meteo.bg/bulair/NATO.pdf.

[33] EMEP: "Emission Data: Status Report", EMEP/MSC-W Report 1/99, July 1999, Meteorological Synthesizing Centre – West, Norwegian Meteorological Institute, P. O. Box 43 – Blindern, N-0313 Oslo 3 Norway, 1999.

[34] EMEP Home Web-page: http://www.emep.int/index_data.html, 2006.

[35] O. Hertel, C. Ambelas Skjøth, L. M. Frohn, E. Vignati, J. Frydendall, G. de Leeuw, U. Schwarz and S. Reis: "Assessment of the atmospheric nitrogen and sulphur inputs into the North Sea using a Lagrangian model". Physics and Chemistry of the Earth, Vol. 27 (2002), pp. 1507-1515.

[36] C. Anastasi, L. Hopkinson, V. J. Simpson: "Natural hydrocarbon emissions in the United Kingdom". Atmospheric Environment, Vol. 25A (1991), pp. 1403-1408.

[37] B. Lübkert and W. Schöpp: "The OECD-map emission inventory for, and in Western Europe", Report No. WP-89-082, International Institute for Applied Systems and Analysis (IIASA), Laxenburg, Austria, 1989.

[38] D. Simpson, A. Guenther, C. N. Hewitt and R. Steinbrecher: "Biogenic emissions in Europe: I. Estimates and uncertainties", Journal of Geophysical Research, Vol. 100 (1995), pp. 22875-22890

[39] WEB-site for OPEN MP tools: http://www.openmp.org, 2012.

[40] W. Gropp, E. Lusk and A. Skjellum: "Using MPI: Portable programming with message passing interface", MIT Press, Cambridge, Massachusetts, 1994.

[41] WEB-site of the Danish Centre for Scientific Computing at the Technical University of Denmark: Sun High Performance Computing Systems. http://www.hpc.dtu.dk, 2002.

[42] I. Foster and C. Kesselman: "Computational Grids". In: "The Grid: Blueprint for a New Computing Infrastructure" (I. Foster and C. Kesselman, eds.), pp. 15-52, Morgan Kaufmann Publishers, San Francisco, California, 1998.

[43] I. Foster, C. Kesselman and S. Tuecke: (2001). "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", International Journal of Supercomputer Applications, Vol. 15, No, 3 (2001), pp. 1-25.

[44] European Parliament: "Directive 2002/3/EC of the European Parliament and the Council of 12 February 2002 relating to ozone in ambient air", Official Journal of the European Communities, L67, 9.3. 2002, pp. 14-30.

[45] Climate Change: "The physical science basis. Contribution of the Working Group I to the Fourth Assessment Report of IPCC (Intergovernmental Panel on Climate Change)", Cambridge University Press, Cambridge-New York-Melbourne-Madrid-Cape Town, 2007.

[46] J. T. Houghton, Y. Ding, D. J. Griggs, M. Noguer, P. J. van der Linden, X. Dai, K. Maskell and C. A. Johnson (eds.): Climate Change 2001: The Scientific Basis, Cambridge University Press, Cambridge-New York-Melbourne-Madrid-Cape Town, 2001.

[47] M. L. Bell, R. Goldberg, C. Hogrefe C, P. L. Kinney, K. Knowlton, B. Lynn, J. Rosenthal, C. Rosenzweig, A. Patz: "Climate change, ambient ozone, and health in 50 US cities", Climatic Change, Vol. 82 (2007), pp. 61-76.

[48] V. S. Bouchet, R. Laprise, E. Torlaschi and J. C. McConnel: "Studying ozone climatology with a regional climate model, 1. Model description and evaluation", Journal of Geophysical Research, Vol. 104 (1999), pp. 30351-30371.

[49] V. S. Bouchet, R. Laprise, E. Torlaschi, J. C. McConnel and D. A. Plummer: "Studying ozone climatology with a regional climate model, 2 Climatology", Journal of Geophysical Research, Vol. 104 (1999), pp. 30373-30385.

[50] A. Carvalho, A. Monteiro, S. Solman, A. I. Miranda, C. Borrego: "Climate-driven changes in air quality over Europe by the end of the 21st century, with special reference to Portugal", Environmental Science and Policy, Vol. 13 (2010), pp. 445-458.

[51] M. Z. Jacobson: "On the causal link between carbon dioxide and air pollution mortality", Geophysical Research Letters L03809, Vol. 35 (2008), doi:10.1029/2007GL031101.