Some proposals about the vector and parallel implementations of STEM-II

D.E. Singh[1], M. Arenaz[2], F.F. Rivera[1], J.D. Bruguera[1], J. Touriño[2],
R. Doallo[2], M.R. Méndez[3], J.A. Souto[3] & J.J. Casares[3]

[1]*Dept. of Electronics and Computer Science, University of Santiago de Compostela, SPAIN, e-mail: david@dec.usc.es*

[2]*Dept. of Electronics and Systems, University of A Coruña, SPAIN, e-mail: arenaz@des.fi.udc.es*

[3]*Department of Chemical Engineering, University of Santiago de Compostela, SPAIN, e-mail: jasouto@usc.es*

Abstract

STEM-II is an Eulerian air quality model which includes different pollutant emissions, transport by advection, convection and turbulence, with dynamic meteorological conditions; chemical transformation by gas-phase and aqueous-phase, and pollutants removal by dry and wet deposition, complete the main air pollution processes. This model is being applied to simulate local and regional acid deposition (episodic and annual basis) from As Pontes Power Plant, considering both the local power plant environment and the EMEP 17,6 cell (150x150 km^2). In this work, a study of the large computational requirements of STEM-II, both in terms of storage and execution times, is done. First, the possibilities of vectorization of the code (with minor changes) is discussed and tested; secondly, different issues about the parallelization of the STEM-II code over distributed memory machines are considered.

1 Introduction

The computational cost of the air quality models (AQMs) is one of the most important factors to take into account when they are going to be applied in air pollution management. Specially, AQMs that consider all the physical and chemical processes involved in the air pollution transport, i.e. comprehensive air pollution models, are one of the applications of high performance computing.

Typical air pollution chemistry models need to solve tenths to hundreds of gas-phase chemical reactions, coupled to the air pollution transport, usually applying Eulerian solutions. Gas-phase chemistry is a heavy task for personal computers, so vector processors and/or multiprocessor systems should be applied, for a reasonable response time. Lagrangian models are more efficient in single source problems, but only simple chemical mechanisms (as in LADM EMEP model; Tsyro [1]) can be considered.

Aqueous chemistry and scavenging processes add more complexity to the problem; i.e., MADE50 EMEP model (Berge [2]), that consider a lumped chemical mechanism with 9 gas-phase reactions and 4 aqueous-phase solution equilibria (including solution reaction), need 90 minutes on a Cray T3E using 8 processors, for a one month simulation all over Europe.

In this work, the computational cost of one of the most complex air quality models, Sulphur Transport Eulerian Model 2 (STEM-II) (Carmichael et al. [3]), is analysed. Some possibilities of vectorization and parallelization of the STEM-II code are considered.

2 Sulphur Transport Eulerian Model, STEM-II

STEM-II is an Eulerian numerical air quality model that accounts for the transport, chemical transformation, and dry and wet removal of atmospheric pollutants, mainly S and N oxidised species, and VOCs.

The model considers 56 chemical species, 16 long-lived species, 40 short-lived species, and 28 radicals, such as OH and HO_2. The chemical mechanism includes 176 gas-phase reactions, 31 aqueous-phase reactions and 12 aqueous-phase solution equilibria. Model equations are integrated by using the locally one-dimensional finite element method (LOD-FEM) (Carmichael et al. [4]), with the resulting transport equations solved by the Petrov-Crank-Nicolson-Galerkin FEM (Heinrich et al. [5]).

Chemistry and mass transfer terms are integrated by using a semi-implicit Euler's method (Preussner and Brand [6]) and the pseudo-analytic method (Carmichael et al. [4]). Therefore, several integration time steps are used in the model, depending on the different physical and chemical processes solved, and the atmospheric conditions.

2.1 Testing environment

The structure, computational cost and accuracy of the STEM-II code have been analysed, considering two different simulation frameworks, with different spatial resolution, in and around the EMEP 17,6 150x150 km^2 cell (located at Galicia, NW of Spain). Dynamic meteorological conditions and different emissions scenarios were considered.

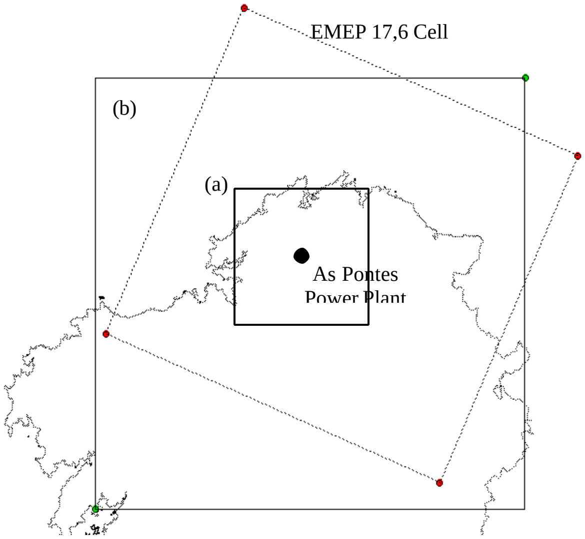

Figure 1. EMEP 17,6 cell and simulation environments: (a) Local environment; (b) Regional environment, covering most of the EMEP 17,6 cell.

The first simulation environment considered is located into the EMEP 17,6 cell, covering 61x61 km^2 and centred in the As Pontes Power Plant. This simulation is solved in a 1x1 km^2 horizontal grid resolution, with 15 vertical levels up to 4200 m. All the emissions considered correspond to As Pontes Power Plant source, and includes SO_2, SO_3, NO, NO_2 and several VOCs. Only background concentration of O_3 is considered, so the transport and chemical simulation is focused in the power plant plume.

The second simulation environment covers 165x165 km^2 around the EMEP 17,6 cell (see Figure 1). In this case, a 5x5 km^2 horizontal grid resolution was applied so, in fact, the number of the 3D grid points is lower than in the first simulation (16335 grid points in the second one, against 55815 grid points in the first one). In addition, this second simulation also considers As Pontes Power Plant emissions, other anthropogenic emissions in the area and a estimated distribution of biogenic emissions, covering a wide range of chemical species.

Simulations were focused in the period 19-26/May/1997. During this period, 19th day was characterised by a significant precipitation (22.4 L/m^2·day), and this day was selected to test the computational cost of both the dry and wet chemical mechanisms of the STEM-II.

2.2 Meteorological data

STEM-II can use meteorological inputs from different sources, measurements and/or numerical models. In this testing work, two different meteorological data sets were applied depending on the spatial resolution and covered area for the simulation,

- In the first local high resolution simulation, results from the PMETEO meteorological model were used, as this hydrostatic model has been widely tested in the As Pontes Power Plant environment (Souto et al. [7]).
- In the second simulation, covering the EMEP 17,6 cell, results of the Advanced Regional Prediction System (ARPS) model (Xue et al. [8]) were applied. ARPS non-hydrostatic model is being tested in the Galician region for an operative daily numerical weather forecast.

Meteorological results of both models were compared with measurements provided by the As Pontes Power Plant meteorological and air quality network (Souto et al. [7]; Souto [9]), and good agreements were achieved. Apart from the meteorological numerical results, precipitation rate measurements observed near As Pontes were considered as input data for the wet removal.

3 Performance analysis of STEM-II

In this section we carry out a study of the efficiency of STEM-II, from a computational point of view. As will be seen, the simulation of all the processes described in the previous section has a high cost, both in terms of memory requirements as well as in execution time. The large amount of resources that are required by this code means that the simulation process is unfeasible on the majority of current computers, given that the calculation time is higher than the simulated one. For these reasons, this model is an excellent candidate for execution on a high performance computer.

In this work we have followed two different lines of analysing the model. On the one hand, a study of the cost and use of resources for the different processes that are simulated is carried out, with the aim of realising optimisation and restructuring processes that will lead to reductions in execution times. On the other hand, an analysis of the data flow was carried out, taking into account the spatial and temporal locality of accesses, as well as the dependences between them. On the basis of the results of this analysis, we will be able to determine which type of high performance computer is the most suitable for an efficient execution of the code. STEM-II simulates different physical phenomena. As all these processes are highly interrelated, prior analysis is not simple. In the following sections we will present the results obtained in greater detail.

3.1 Computational cost

Memory use and memory access method determine the efficiency of a programme. STEM-II uses a large number of variables. It must store physical information from each point of the mesh (concentration of species, temperature, etc.) using large arrays. Furthermore, the chemical phase operates on replicas of these data, which involves copying the information between the original variables and these replicas. The size of these auxiliary arrays does not usually depend on the mesh size, given that, in the majority of cases, they have one dimension which corresponds to that of the number of chemical species.

Table 1. Memory requirements of the model.

Mesh size (*x*,*y*,*z*)	Data memory (Kb)	Total memory (Kb)
(61,61,15)	175974	176184
(33,33,15)	52938	53172

Using global variables instead can eliminate many of these arrays. Thus, small memory savings are achieved.

Table 1 shows the amount of memory used solely for storing the data of the model. The total memory required by the programme is also shown. These measurements appear for two different types of meshes. We have found that the memory required is proportional to the mesh size, and is given by the expression $M = \boldsymbol{a} \times x \times y \times z$ where *M* is the memory used in kilobytes (Kb) and *x*, *y* and *z* are the spatial dimensions of the mesh. We have experimentally shown that $\boldsymbol{a}\ @ 3.15$.

The implementation of STEM-II consists of seven nested loops. The outermost loop determines the simulation time. In our examples, we have used a one-minute time step. Next three loops traverse the spatial dimensions of the 3D mesh. Next, two inner loops are used to refine the time step to simulate the chemical gas and aqueous phases. Finally, the innermost loop iterates over the chemical species.

Figure 2 shows the execution time of each iteration in time loop for a simulation of 8 hours, starting at 4.00pm, on a Sun Enterprise 250. It can be seen how the step of simulating all chemical processes results in a sharp increase in the execution time. The peaks in the figure correspond to periodic intervals of disk I-O (new meteorological data are read and results data are written). The sharp decrease in execution time between iterations 120 and 360 is due to the fact that, because of climatological circumstances, there are less chemical processes in the aqueous phase. The first iterations of the programme are the most costly as they require a large amount of data to be loaded in the memory system. The decrease in execution time during nighttime with respect to daytime is due to the computation of new chemical reactions caused by the sun radiation.

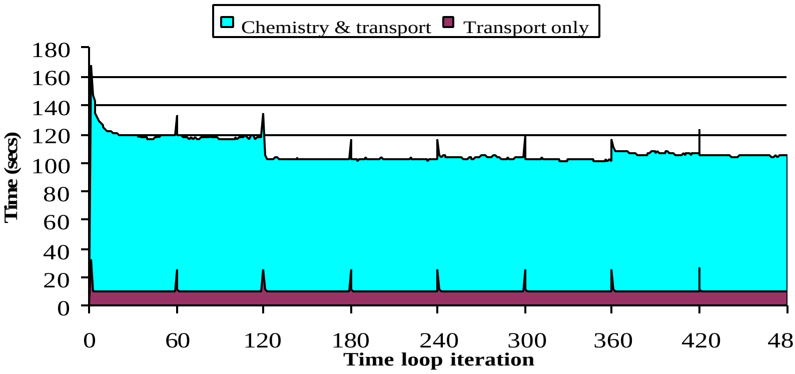

Figure 2: Execution time for each iteration.

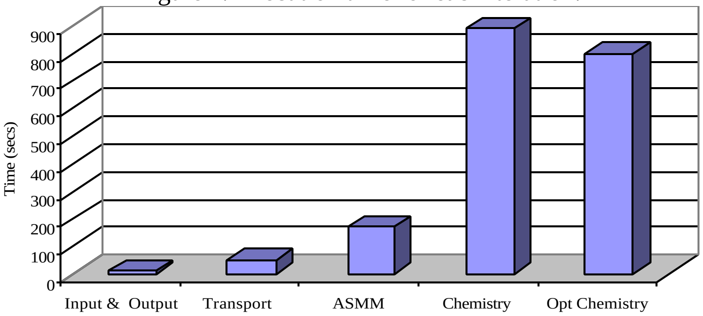

Figure 3: Execution time for each module.

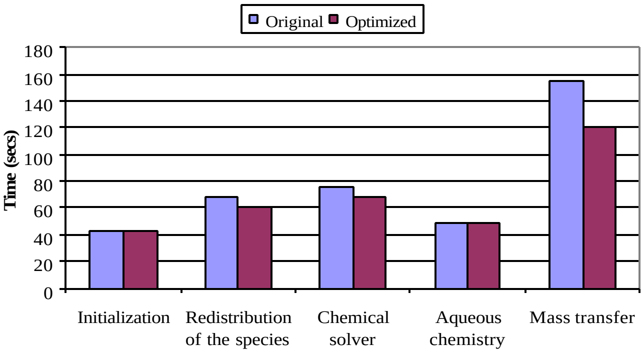

Figure 4: Execution time of the processes in the chemical module.

In order to evaluate the importance of each process in greater detail, we have carried out an analysis of the computational cost of each module of the code. The results for 4 iterations are shown in Figure 3. The chemical module is the most costly part of the programme because it is executed once for each mesh node and for each time step. The time bar labelled ASMM in Figure 3 includes both the

Advanced Scavenging Module (ASMM) and the pre and post chemical processing subroutines (PPCPs).

With the aim of reducing the execution time, sequential optimisations have been applied mainly to the chemical phase. Various types of code restructuring techniques were used: loop-invariant removal, deadcode elimination, elimination of redundant expressions, strength reduction (complex numerical operations are substituted by less costly equivalent ones), and restructuring of memory access. The execution time of the optimized chemical module is also shown in this figure as opt chemistry. Figure 4 shows in detail the results after the application of these optimisations. There is a 13% improvement in the execution time.

3.2 Data flow

In this section we have focused on the analysis of spatial and temporal locality of memory references as well as on data dependences.

Table 2. Types of dependences in each module.

Module	Dependence	Loops that can be parallelized
Chemistry	None	*x*, *y*, *z*
PPCPs	None	*x*, *y*, *z*
ASMM	*z*	*x*, *y*
Vertical transport	*z*	*x*, *y*
Horizontal transport	*x*,*y*	*z*

First, the main arrays of the code were identified. These arrays contain information that completely characterises the simulated physical system: chemical concentration of the species, topographical and meteorological data, etc. We have classified these arrays according to the type of information they store. Each group present the same type of dependences.

3.2.1 Species concentration arrays

The arrays that contain the concentration of the chemical species have four dimensions: three spatial ones (*x*, *y* and *z*) and an additional one to classify each specie. These arrays are accessed in the three main modules of the code: horizontal and vertical transport, and chemical phase.

In the horizontal transport, arrays are accessed by setting the *z* co-ordinate, i.e., on the different planes of the 3D mesh. In the vertical transport, accesses are performed by setting *x* and *y* co-ordinates, i.e. on the different columns of the mesh. In the chemical phase, *x*, *y* and *z* co-ordinates are set, so the simulation of the chemical reactions on the different nodes of the mesh are independent. That is, all the calculations are carried out only with regard to the information contained in the specific point of the mesh, without accessing to adjacent elements in the array.

We have also verified that in the vertical transport and in the advanced scavenging module there are data dependences only in the *z* co-ordinate.

Table 2 shows the main loops of the code that can be parallelized given the data dependences found in each module. Arrays that are only used as input variables were identified; for instance, those containing information on the temperature at mesh nodes and on chemical emissions. Another groups of arrays, such as those that contain information on the levels of environmental humidity, are only modified in horizontal transport; being used as input variables in the chemical and vertical transport modules.

3.2.2 Meteorological and topographical arrays

STEM-II uses arrays to store the meteorological characteristics of the environment, such as wind speed, horizontal/vertical eddy diffusivity or the presence of rain conditions. Those arrays that store wind characteristics are only accessed by the horizontal and vertical transport modules, without any influence on the simulation of the chemical phase. Access to these variables is read-only, so they do not generate any data dependence. Those other variables that contain information on the presence of clouds, rain or snow are also accessed as read-only variables in the chemical part. As expected, the same occurs with topographical and emission variables, as they are read-only variables.

4 Approach to high performance computing of STEM-II

We can consider two main approaches in order to execute the STEM-II code on a high performance computer: vector processors and scalar multiprocessors.

4.1 Vector processors

These systems exploit loop-level parallelism. Vectorization is prevented by the presence of data dependences in the loop body as well as calls to external functions or procedures. As these constraints are present in the main seven nested loops of the application, STEM-II is not suitable for being executed on a vector processor. Nevertheless, we have identified a large number of vectorizable loops in the chemical module, but the performance improvement is negligible due to the short length of the arrays involved. Several measures were realised in the Fujitsu VPP300 system [10]. A one hour simulation lasted out 223 minutes. Only 18 minutes of this time were executed in the vector units.

4.2 Scalar multiprocessor

Coarse-grain parallelism is more appropriate for this kind of systems. It basically consists of dividing the problem into sub-problems as independent as possible. According to the data dependence results shown in Table 2, the most costly loops of the programme could be parallelized. Therefore, we consider that the structure of the STEM-II code is appropriate for an execution on a scalar multiprocessor.

Next we present a preliminary parallel version of STEM-II. We have focused on the chemical module as it is by far the most costly part of the algorithm in terms of execution time (see Figure 3).

Although there are no data dependences inside the chemical module (see Table 2), the dependences with the modules ASMM and vertical transport led us to distribute computations as shown in Figure 5. A block distribution that ensures load balancing was used for each column of the 3D mesh.

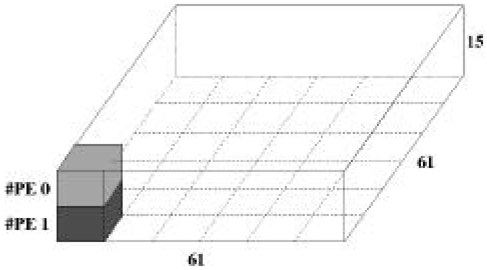

Figure 5: Distribution of the computations.

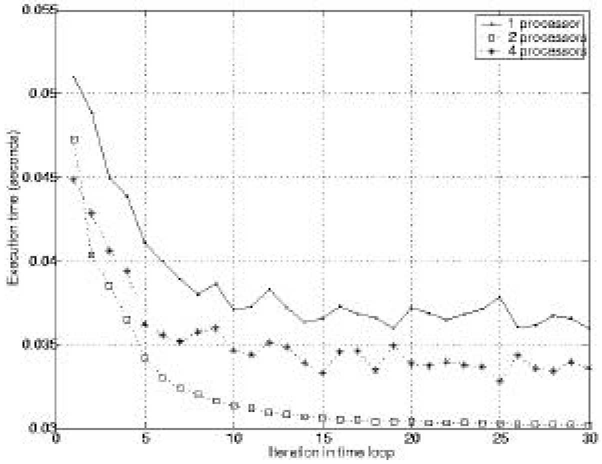

Figure 6: Execution time results.

The shaded regions identify which nodes of the column are assigned to each processor (#PE0 and #PE1 in the figure).

The target machine was the Fujitsu AP3000 distributed-memory multiprocessor (Ishihata et al. [11]). It consists of UltraSparc-II processors at 300MHz interconnected in a two-dimensional torus topology. The parallel programme was written in Fortran77 using MPI (*Message-Passing Interface*) (Pacheco [12]).

Figure 6 shows execution times corresponding to the simulation of the chemical reactions at the nodes belonging to a simple column of the mesh. Measurements for the first 30 iterations in time loop on 1, 2 and 4 processors are presented. Execution time is reduced when using 2 processors, but it increases from 4 processors due to communication overhead.

Note that execution times are higher in the first iterations in time loop. This is due to the fact that the system takes some time to load algorithm data on its memory system efficiently.

5 Conclusions

In this paper we have described the parallelization of the chemical module of the STEM-II. A study of the large computational requirements of STEM-II, both in terms of storage and execution times, was done. The results show that this code is a good candidate to be parallelized in most of their loops. With the aim of reducing the execution time, sequential optimisations have been done. Although the overall computational cost of the algorithm has been reduced, the results have shown that there is still a lot of work to do in order to develop an efficient parallel algorithm.

Acknowledgements

FEDER and CICYT financially support this work, under co-ordinated projects 1FD97-0118-C02-01 and -02, and Endesa Company. The assistance of Prof. Gregory R. Carmichael's team, CGRER, University of Iowa, on providing and explaining the use of STEM-II model, the computational time assigned on the Fujitsu VPP300E and the Fujitsu AP3000 at the CESGA, and its technical support, are gratefully acknowledged.

References

[1] Tsyro, S. G. *Description of the Lagrangian Acid Deposition Model EMEP/MSC-W Status Report 1/98, Part 2: Numerical Addendum. EMEP/MSC-W Status Report 1/98. Appendix A.* Norwegian Meteorological Institute, Oslo, Norway, 1998. http://www.emep.int.

[2] Berge, E. *Transboundary Air Pollution in Europe. MSC-W Status Report 1997. Part 1 and 2 EMEP/MSC-W Report 1/97*. The Norwegian Meteorological Institute, Oslo, Norway, 1997. http://www.emep.int.

[3] Carmichael, G.R., Peters, L.K., Saylor, R.D. The STEM-II regional scale acid deposition and photochemical oxidant model - I. An overview of model development and applications. *Atmospheric Environment*, **25A**, 10, pp. 2077-2090, 1991.

[4] Carmichael, G.R., Peters, L.K., Kitada, T. A second generation model for regional-scale transport/chemistry/deposition. *Atmospheric Environment*, **20**, pp. 173-188, 1986.

[5] Heinrich, J.C., Huyakorn, P.S., Zienkiewicz, O.C., Mitchel, A.R. An "upwind" finite-element scheme for two dimensional convective transport equation. *Int. J. num. meth. Engn.*, **11**, pp. 131-143, 1977.

[6] Preussner, P.R., Brand, K.P. Application of a semi-implicit Euler method to mass action kinetics. *Chem. Engng. Sci*, **10**, pp. 1633-1641, 1981.

[7] Souto, J.A., Pérez-Muñuzuri, V., DeCastro, M., Souto, M.J., Casares, J.J., Lucas, T. Forecasting and diagnostic analysis of plume transport around a power plant. *Journal of Applied Meteorology*, **37**, pp. 1068-1083, 1998.

[8] Xue, M., Droegemeier, K.K., Wong, V.C., Shapiro, A., Brewster, K. *ARPS Version 4.0 User's Guide*. Centre for Analysis and Prediction of Storms, University of Oklahoma, 1995.

[9] Souto, M.J. *Validation of two meteorological models (hydrostatic and non-hydrostatic) coupled to a lagrangian particle dispersion model around As Pontes Power Plant environment* (in spanish). PhD Thesis, University of Santiago de Compostela, 1999.

[10] www.fujitsu.co.jp/hypertext/Products/Info_process/hpc/vpp-e/index.html

[11] H. Ishihata, M. Takahashi & H. Sato. Hardware of AP3000 Scalar Parallel Server. *Fujitsu Sci. Tech. Journal*, **33**(1), pp. 24-30, 1997.

[12] P.S. Pacheco. *Parallel Programming with MPI*. Morgan Kaufmann Publishers, Inc., 1997.