THE CURRENT STATE AND THE FUTURE DIRECTIONS IN AIR QUALITY MODELING

Gregory R. Carmichael, Adrian Sandu, Florian Potra, Valeriu Damian, Mirela Damian

Center for Global & Regional Environmental Research and the Department of Chemical & Biochemical Engineering University of Iowa, Iowa City, Iowa 52240
Tel: 319/335-1399 FAX: 319/335-1415 EMAIL: gcarmich@icaen.uiowa.edu

In this paper the present "state" of air quality modeling (as viewed by the authors) is presented. The focus of the paper will be on the "current" state-of-affairs. Due to limitation of space (and the focus of this Dagstuhl Seminar) the discussion will focus on only a few aspects of air quality modeling: i.e., chemical integration, sensitivity analysis and computational framework.

KEY WORDS: Air quality, atmospheric chemistry, sensitivity analysis

1 INTRODUCTION

A detailed understanding of the relationships between the emissions and the resulting distribution of primary and secondary species in the atmosphere is a requisite to designing actions for the maintenance of a healthy environment. Scientific efforts to understand the atmospheric processes governing these relationships involve a combination of laboratory experiments, field studies, and modelling analysis. Laboratory experiments provide basic data on individual physical and chemical processes. Field studies are designed to investigate a limited number of processes under conditions in which a few processes dominate. Unlike controlled laboratory experiments, field studies cannot be parametrically controlled. Since laboratory experiments and field studies by themselves cannot fully elucidate complex atmospheric phenomena, comprehensive models that allow multiple processes to occur simultaneously are required for data analysis and scientific inquiry.

The models by Carmichael et al. [6], Jacob et al. [10], Dentener and Crutzen [9] are examples of regional and global scale atmospheric chemistry models in use today. These models treat transport, chemical transformations, emissions and deposition processes in an integrated framework, and serve as representations of
our current understanding of the complex atmospheric processes. They provide a means to perform parametric studies for quantitative analysis of the relationships between emissions and the resulting distribution, and can also be used to study the response of the pollutant distributions to system perturbations, and to link pollutant distributions to environmental effects.

As our scientific understanding of atmospheric chemistry and dynamics has expanded in recent years, so has our ability to construct comprehensive models which describe the relevant processes. However, these comprehensive atmospheric chemistry models are computationally intensive because the governing equations are nonlinear, highly coupled, and extremely stiff. As with other computationally intensive problems, the ability to fully utilize these models remain severely limited by today’s computer technology.

Grid resolutions of $0.5^\circ \times 0.5^\circ$ or better in the horizontal, 20 vertical grids, and as many as 40 to 100 species are necessary for adequate analysis of perturbations to atmospheric chemistry on the global and regional scales. With such detail, the finest grid resolution that can be achieved for global-scale analysis on todays computers is about $2^\circ \times 2^\circ \times 10$ vertical levels, even if one is content with a 1 : 1 simulation time to CPU time ratio. This latter ratio should be at least 10 : 1, and preferably 100 : 1 or better, to effectively address many of the important scientific questions.

The scientific issues associated with analysis of our chemically perturbed atmospheres are dominated by a number of underlying considerations. Several of the more important ones are: (a) the anthropogenic sources of trace species are quite localized and occur only over a fraction of the Earth’s land area; (b) natural sources of trace species are, for the most part, very disperse and are not in the same areas as the anthropogenic sources (although this trend may be changing in regions such as tropical rain forests and the savannahs); (c) in virtually no case can an individual species be studied in isolation from other species; (d) many of the mechanisms that effect transformation of the species are non-linear (e.g., chemical reactions and nucleation processes); and (e) species of importance have atmospheric lifetimes that range from milliseconds and shorter to years (e.g., $OH$ radical to $CH_4$). These considerations require: finer grid resolutions than currently existing ones in present-day models; simultaneous treatment of many species; and long simulation times (i.e. months to years) to assess the impacts. These demands present considerable challenges to the air quality modeling community.
2 AIR QUALITY MODELING FRAMEWORK

Air quality models are computer-based models which calculate the distribution of trace gases in the troposphere from specified emissions distributions and meteorological scenarios. The major features consist of:

1. a transport component (or module) to describe the wind speed and direction, the eddy diffusivity and mixing layer height, the temperature, the water vapor, cloud water content, and the radiation intensity of each location as a function of time;

2. a chemical kinetic mechanism to describe the rates of atmospheric reactions, including homogeneous gas-phase, heterogeneous, and liquid phase reactions;

3. removal modules to describe the dry deposition of material, and the in-cloud and below-cloud removal processes.

Each process incorporated into a model is itself a very complex and incompletely understood phenomenon. Therefore, in formulating such models it is necessary to incorporate the processes into the model framework by utilizing chemical, dynamic, and thermodynamic parameterizations. Furthermore, even processes that are quite well understood may require parameterization to maintain some balance of the details among the different processes that are treated in the model.

The theoretical basis is the atmospheric advection-diffusion equations (i.e., the mass balance equations):

\[
\frac{\partial C_i}{\partial t} + \frac{\partial (U_j C_i)}{\partial x_j} = \frac{\partial}{\partial x}[K_{jj} \frac{\partial C_i}{\partial x_j}] + R_i + E_i + G_i \quad i = 1, \cdots, \# \text{ of species} \tag{1}
\]

where \(C_i\) denotes the gas phase concentrations, \(U_j\) are velocity components, \(x_j\) represents the spatially coordinates, most generally three dimensional, \(K_{jj}\) are the eddy diffusivities, \(R_i, G_i\) and \(E_i\) are the rates of chemical reactions, mass transfer and emissions, respectively.

In the above equations term (A) represents the unsteady accumulation of mass, (B) changes in mass due to advective fluxes, (C) changes in mass due to turbulent diffusive fluxes, (D) the rate of production/destruction due to chemical reaction, (E) the source term due to emissions, and (F) the rate of mass transfer between
phases. These equations are nonlinear due to the nonlinear nature of the chemical processes, and are also highly coupled within a given phase, again due to the chemical processes, and coupled between phases through the inter-phase mass transfer processes (e.g., gas absorption, nucleation, and accretion processes).

Comprehensive air quality models presently explicitly treat a wide variety of sulfur compounds (e.g., $SO_2$, sulfate, some reduced sulfur species), nitrogen compounds (e.g., $NO$, $NO_2$, $HNO_3$, $NO_3$, $N_2O_5$, $PAN$), and hydrocarbon species (e.g., $CO$, $CO_2$, $CH_4$, a variety of alkenes, alkanes, aromatics, aldehydes and organic acids), along with the important oxidizing species such a $O_3$, $OH$, and $H_2O_2$.

3 REACTION MECHANISMS AND SOLVERS

The chemistry presents the greatest computational challenge in air quality modeling. The chemical mechanisms needed in air quality models are specified by the intended applications and are designed to address a wide variety of scientific and public policy issues. The applications range from interpretation of observations taken at various locations around the world, to estimating what will happen to important trace species in the future. This range of applications places severe design constraints on the chemical mechanism. Not only must the mechanism be able to handle the very different chemical regimes encountered around the world, but it also must be sufficiently robust to reflect accurately the changes in atmospheric chemistry that would result from changes in meteorological and climate attributes (such as temperature, solar actinic flux, water vapor, cloudiness, etc.) as well as changes resulting from trace gas emissions (both anthropogenic and biogenic).

Since the majority of the computing time is consumed in integration of the chemistry rate equations, it is critically important to avoid introducing unnecessary complexity into the chemical mechanism. In addition, each species and reaction expression included in the chemical mechanism must be supported by a large body of data (reaction rate constants, activation energies, yields, products, ambient observations, emission rates of primary species, etc.). The compromises which must be struck between the need to have an accurate and general chemical mechanism, and the restrictions imposed by computational considerations and the availability and uncertainties associated with the supporting data, remain critical decisions in the formulation of atmospheric chemistry models. In light of recent advances in our knowledge of atmospheric chemistry, several topics warrant further
consideration.

3.1 Mechanism considerations

At the heart of the atmospheric photochemical oxidant cycle are those reactions involving hydroxyl and peroxy radicals. The application, computational resources and state of the science, all help dictate the level of chemical detail included in the model. In addition the user wants the flexibility to ”play with ” the chemistry. Thus the model must make it as easy as possible to change both the chemistry in the model (starting with the species to be treated in detail, and the ”explicitness” of the reaction mechanism), and the numerical integration method. Thus contemporary air quality models are of modular design to enable the relatively easy construction of an appropriate chemical mechanism for each individual application of the model. Such modularity also ensures that as advances in understanding atmospheric chemistry continue to occur, the model’s chemical mechanism can be modified and updated in an easy and straightforward procedure.

There are several ways to deal with the chemical mechanism. One approach is to establish a core mechanism that would serve as the backbone for global tropospheric chemistry calculations. This core mechanism would consist of a basic set of reactions that describe $CH_4/CO/NO_x/O_3$ photochemistry and the generation of hydroxyl ($OH$ and $RO$) and peroxyl ($HO_2$ and $RO_2$) radicals in the troposphere. Once the core mechanism is in place within the model, specific studies may require other modules that describe additional chemical systems. For example, a study designed to assess the role of biogenic sulfur emissions on global climate change requires a chemical module consisting of appropriate reactions of sulfur species in the troposphere; or a study of the role of oxygenated organics requires a more detailed reactive hydrocarbon description. An alternative approach is to make use of a single mechanism that would dynamically change within the model as a function of the local chemical regimes (e.g., remote marine, remote continental, urban, etc.) or with the number of species treated explicitly in the model.

3.2 Improved Algorithms for Solution of Chemistry Equations

3.2.1 Improved algorithms for solution of stiff atmospheric chemistry equations

The large computational requirements in the study of chemically perturbed environments arise from the complexity of the chemistry of the atmosphere. Integra-
tion of the chemistry rate equations typically consumes as much as 90 percent of the total CPU time! Obviously, more efficient integration schemes for the chemistry solvers would result in immediate benefits through the reduction of CPU time needed for each simulation. As more and more chemical species and reactions are added to the chemical scheme for valid scientific reasons, the need for faster yet more accurate chemical integrators becomes even more critical.

It is well known that the chemistry rate equations comprise a system of stiff ordinary differential equations (ODE). This fact precludes the use of explicit integration schemes. Indeed, theoretical considerations and numerical practice show that when an explicit integrator is applied to stiff ODEs, the code will choose prohibitively small integration steps in order to preserve stability. On the other hand, implicit integrators have infinite stability regions, and consequently, the step-size selection is done by monitoring the local truncation errors to meet the accuracy requirements set by the user. For integration of the chemistry rate equations, implicit integrators are likely to work with relatively large step-sizes when the accuracy requirements are not too stringent. However, at each integration step, a nonlinear system of equations has to be solved. This involves the repeated evaluation of Jacobians and the solution of linear algebraic systems of dimension \( n \), the number of species considered in the model. General stiff ODE solvers do not take advantage of the sparsity pattern of the Jacobian, and the number of arithmetic operations required for the numerical solution of the corresponding linear system is proportional to \( n^3 \). This is one of the reasons why general stiff ODE solvers are not very efficient for integration of the chemistry rate equations with a moderate to large number of species. A comparison of the exactness and time efficiency of different integrators can be found in the paper of Shieh, Carmichael et al. [26].

More efficient chemistry integration algorithms for atmospheric chemistry have been obtained by carefully exploiting the particular properties of the model. One of the commonly used methods is the QSSA method of Hesstvedt et al. [18]. There are many other methods in use in atmospheric chemistry models. For example, S. Sillman in [27] developed an integration scheme based upon the analysis of sources and sinks of odd-hydrogen radicals in the troposphere. The basis of the method is the implicit Euler formula. Observing that nearly all tropospheric reactions include \( OH, HO_2, NO, NO_2 \) or \( NO_3 \) as a reactant and tropospheric reactions form sequential chains with a few recursive relationships, the author reorders and decomposes the vector of species so that the resulting Jacobian (used in the nu-
merical solution of the implicit Euler equations) is nearly diagonal, thus enabling an elegant ”decoupling” of the system of equations. Each decoupled part is treated differently. A fully implicit scheme is used for short-lived species (no matrix inversion needed here, because of the triangular upper left corner of the Jacobian) while a semi-implicit method is sufficient for long-lived species. This greatly helps the numerical computations. However, the scheme is difficult to generalize to solve more general problems.

Recently, Hertel, Berkowicz, Christensen, and Hov [17] proposed an algorithm based on the implicit Euler method. The algorithm uses linear operators only and preserves the mass. The nonlinear system is solved using functional iterations. The main idea is to speed up these iterations using explicit solutions for several groups of species. The method seems to work fine for very large step-sizes.

A particularly clear approach is taken by Gong and Cho [14]. They divide the species into slow and fast ones, according to the size of their lifetimes; the slow species are estimated using an explicit Euler scheme; the fast ones are integrated with the implicit Euler scheme (and Newton-Raphson iterations for solving the nonlinear system); as a last step, the slow species are “corrected”, reiterating the explicit Euler step. The hybrid technique allows a significant increases of the time step.

A fancy projection/forward differencing method was proposed by Elliot, Turco, and Jacobson [12]. The species are grouped together in families; the distribution of the constituents inside a family is recalculated before each integration step using an implicit relation and solving the corresponding nonlinear system (this “projection” can be viewed as a “predictor”); then the integration is carried out for families using a significantly improved time step.

QSSA type methods were also explored by Seinfeld [23]. Verwer [21] investigates the extension of QSSA to a second order consistent scheme. It is tested against the “two step method”, which is the second order BDF plus Gauss-Seidel iterations for solving the nonlinear system (according to the author, these iterations perform similarly to the modified Newton method, but with less overhead). The two step method seems to allow very large step-sizes.

A different approach is taken in [19] by Jacobson. The 3-D calculations are vectorized around the grid-cell dimension (very interesting idea) and advantage is taken of the sparse structure of Jacobians and a specific reordering of species (that makes Jacobians close to lower triangular form).
Although the above mentioned methods are more effective than general stiff ODE solvers, they are generally based on specific properties of the particular chemistry model rather than on general mathematical principles, and therefore, they are not easily applied directly to new situations. Furthermore, the main concern has been computational speed, and very little attention has been paid to formal error analysis.

We have looked in detail at the widely used QSSA method and have demonstrated that by combining a solid understanding at the mathematical level of the structure of this efficient ad hoc chemistry integration algorithm, with recent advances in the theory and practice of numerical methods for stiff ODE’s, significant improvements in the efficiency and general applicability of this method are achieved [20]. We have shown that the plain QSSA is an order one method, and that by using evaluations at one more point we can build an order two method. It is also shown that the global error of the plain QSSA has an asymptotic expansion in integer powers of the stepsize $h$ and therefore extrapolation methods of arbitrarily high order can be constructed in the nonstiff case. In particular we obtained an order two method that uses two function evaluations per step which we call the **Extrapolated QSSA Method**. We also constructed a nontrivial modification of the well known GBS extrapolation algorithm based on an appropriate QSSA approximation whose global error has an asymptotic expansion in even powers of $h$ (the proof of this fact is quite involved). In particular we obtained an order two method that uses three function evaluations per step which we call (for good reason) the **Symmetric QSSA Method**. In the stiff case the extrapolated methods are no longer so effective. However, we proved that the Extrapolated QSSA method and the Symmetric QSSA Method have a much smaller error constant which explains their superior performance.

Numerical experiments show that these techniques perform well, compared to the plain or iterated QSSA (especially when precision requirement is 10% or higher). In three dimensional atmospheric models, the optimal value for the relative error in the solution of chemistry kinetic equations is considered to be approximately 1%. More precision is redundant, due to inexactities in the transport scheme, and less precision can have an unpredictable effect on overall accuracy, as it is hard to assess the error propagation through the transport scheme in an operator splitting algorithm. For this level of accuracy (two significant digits) the Extrapolated QSSA and the Symmetric QSSA Methods seem to be more efficient.
than Classical QSSA, CHEMEQ, and VODE [5] algorithms. To test the properties of the new numerical methods we have chosen the chemical mechanism that is presently used in the STEM-II regional-scale transport/chemistry/removal model (Carmichael et al., [6]). This mechanism consists of 86 chemical species and 178 gas phase reactions. The mechanism, based on the work of Lurmann et al. [22] and Atkinson et al. [2] is representative, of those presently being used in the study of chemically perturbed environments. Figure 1 presents a work-precision comparison among different methods. The chemical mechanism was run for 5 days. The rate constants and initial conditions follow the IPCC scenario 3 (“Bio”). At the beginning of each hour, an injection of 0.01 ppb/hour NO and NO₂ and 0.1 ppb/hour of isoprene is carried out, to simulate an operator-splitting environment. The rate constants are updated at the beginning of each hour to their half interval values.

![Figure 1: Work-precision diagram. QSSA (solid), VODE(dotted), Extrapolated QSSA (dashed with x), Symmetric QSSA (dashed with o) and CHEMEQ (dash-dots)](image)

3.2.2 Efficient treatment of sparsity

A different approach to solving stiff ODE’s arising from atmospheric chemistry is to use implicit integrators. Implicit integrators are very useful for solving stiff systems
of ODE’s arising from atmospheric chemistry kinetics. However the performance of implicit methods depends critically on the implementation of linear algebra (which appears in connection with Newton steps) and on their restart overhead (when used in an operator splitting environment). We have begun to explore these ideas [25]. We believe that it is possible to exploit sparsity in the Jacobian systematically and hence to develop more efficient implementations of fully implicit methods for solving atmospheric chemistry problems.

1. We evaluated reordering techniques that lead to minimal fill-in during LU decomposition. Our preliminary results indicate that diagonal Markowitz criterion performs very well (best, among the strategies considered so far).

2. We tested various off-the-shelf as well as tailored linear system solvers. We found that a simple Doolittle code, without pivoting, which relies on the off-line analysis of the sparsity structure performed by the symbolic preprocessor KPP, was the fastest method.

3. Finally we explored the effect of using this information to improve the efficiency of some state-of-the-art stiff ODE solvers. We chose two test problems consisting of comprehensive chemical mechanisms used to study stratospheric and tropospheric chemistry mechanism discussed previously. Information about the speed-ups obtained with sparse versus full linear algebra are presented in Table 1.

In the usual practice, atmospheric modellers sometimes employ Gear type methods (see [19]). Our results indicate that when linear algebra is treated efficiently, implicit methods from different families, like Singly-Diagonal-Implicit-Runge-Kutta or Rosenbrock, [25] can be competitive.

Figure 2 compares results for three sparse codes with the widely used QSSA and CHEMEO. A restart is carried out every hour or every 15 minutes (Figure 2). This restart corresponds to an equal step-size of the transport scheme, in an operator splitting code. Since off-the-shelf integrators, endowed with the above sparse techniques, are quite competitive, it is clear that further work needs to be done for obtaining other implicit algorithms, that better suit the application. Considering the modest accuracy requirements, we expect that low order SDIRK methods and low order generalized linear methods are well suited for 3-dimensional air quality models, when sparsity is exploited to reduce the linear algebra overhead.
<table>
<thead>
<tr>
<th>CODE</th>
<th>STRATO</th>
<th>TROPO</th>
</tr>
</thead>
<tbody>
<tr>
<td>VODE</td>
<td>2.48</td>
<td>4.63</td>
</tr>
<tr>
<td>SDIRK4</td>
<td>2.52</td>
<td>4.48</td>
</tr>
<tr>
<td>RODAS</td>
<td>2.01</td>
<td>3.50</td>
</tr>
</tbody>
</table>

Table 1: Average speed-ups obtained

Figure 2: Work-precision diagram with a restart carried every hour (left) and every 15 minutes (right). Sparse VODE (solid), Sparse Rodas (dash - dots), Sparse Sdirk4 (dashed), QSSA (dots with “x”) and CHEMEQ (dots with “o”)
4 COMPUTATIONAL CONSIDERATIONS

To improve significantly the overall computational performance of comprehensive models the transport calculations must also be improved. As discussed previously, presently most models use the concept of operator splitting to solve the PDE’s arising in atmospheric chemistry models. For example the STEM model solves the transport equations using a locally-one-dimensional implementation of a Petrov-Galerkin finite element method. However, operator splitting can induce numerical problems. For example even though the 3-d flow fields are divergent-free, numerical errors may occur when there are large divergences in one of the individual directions. There is the need for further work on improving the transport algorithms used in air quality models.

4.1 Parallel implementation

Besides speed, one of the main issues when designing a parallel code is portability. One of the best ways to deal with this is to use the standard message-passing library MPI [15], that assures the portability with almost no penalty on efficiency. We also must consider scalability, i.e. to achieve an increasing speedup when the program is run on more processors. This means that in parallel implementation we have to find the most appropriate grain for the parallelization.

There is another important issue to be considered here. The code that runs the air quality model has to adapt easily to using different integration techniques both for the chemistry part and for the transport part. With a modular organization this has already been achieved for our sequential code. For the parallel code use of other integration schemes may introduce other data dependencies, therefore the data splitting used in one case may not work in other cases.

It is desirable to use parallelizing compiler techniques that will automatically generate efficient parallel code. There have been many attempts to write such a compiler, but most of them solved the problem of discovering fine grain of parallelism. While this is reasonably efficient for a vector machine (CRAY, for example), it is very inefficient for multiprocessor systems that use message passing, including network of workstations. Discovering coarse grain parallelism is a much more difficult task. There have been some attempts, but none of them has been completely successful. We are designing a special tool that will generate parallel C code that includes appropriate calls to MPI functions. Besides the sequential program, some
additional data dependencies about the program will be provided. The SUIF compiler [28] designed at Stanford University offers a best way to implement this tool. One of the most important questions for which this tool should provide an answer is to be able to select the best way of partitioning the data among processors.

The kind of code analysis that this tool has to perform includes interprocedural analysis [16] and array region analysis. An optimizing compiler usually performs only intraprocedural analysis, i.e. it does not attempt to understand the data dependencies between procedures. In order to detect coarse grain parallelism in an application one has to do some kind of interprocedural analysis. The array region analysis is useful to identify which portion of each array is read or written by a specific portion of the code that is to be parallelized. In this way only those portions are to be distributed at the beginning and gathered at the end of the parallel part. With an automatic evaluation of the execution time of each code sequence and the communication time for the parallel version the code can decide whether it is better to parallelize the code, replicate it, or just execute it sequentially.

4.2 Preliminary results for the 3D parallel implementation

In describing the parallel algorithm we use a 3D cubic model for the chemical system of dimensions $N_x$, $N_y$ and $N_z$.

We use the following notations for describing the algorithms:

- “$t$” is the length of the time-split interval (the temporal step-size). “$O^t$” means that the operator $O$ acts from $t_0$ to $t_0 + t$; here $O$ may designate $C$ (integration of the chemical rate equations) or $T_{\text{dir}}$ (integration of the convection-diffusion equation, projected on the coordinate direction $\text{dir}$).

- $C'(x_1 \cdots x_2, y_1 \cdots y_2, z_1 \cdots z_2)$ - represents the integration of the chemical rate equations for all boxes identified by the coordinates $(x, y, z)$, with $x_1 \leq x \leq x_2$, $y_1 \leq y \leq y_2$, $z_1 \leq z \leq z_2$.

As a particular case, we define $X = 1 \cdots N_x$, $Y = 1 \cdots N_y$, $Z = 1 \cdots N_z$.

Using these definitions, $C'(X, Y, Z)$ represents the chemistry for all the boxes in our 3D model. Let $P$ be the number of processors available. The initial data are uniformly distributed among processors. We discuss the simplest case in which $N_y$ and $N_z$ are multiples of $P$ and all the processors own the same amount of data. We define:

$$Y_i = (i - 1) * N_y/P \cdots i * N_y/P,$$

$$Z_i = (i - 1) * N_z/P \cdots i * N_z/P.$$
These coordinates identify the set of planes from the original cube that belong to the process \( P_i \) at a time in the program.

Then we may use:

- \( C^l(X,Y,Z_i) \) to denote the chemistry for all boxes in a set of horizontal planes, namely all the \((X,Y)\) horizontal planes at levels \( Z_i \) in the original cube.
- \( C^l(X,Y_i,Z) \) to denote the chemistry for all boxes in a set of vertical planes, namely all the \((X,Z)\) vertical planes at positions \( Y_i \) in the original cube.

- \( T^l_x \) - the horizontal transport in \( X \) direction for all \((Y,Z)\) lines (of size \( N_z \)) in the cube.
- \( T^l_y \) - the horizontal transport in \( Y \) direction for all \((X,Z)\) lines (of size \( N_y \)) in the cube.
- \( T^l_z \) - the vertical transport in \( Z \) direction for all \((X,Y)\) lines (of size \( N_z \)) in the cube.
- \( T^l_x(i) \) - the horizontal transport in \( X \) direction for all \((Y,Z_i)\) lines (of size \( N_z \)) in the cube.
- \( T^l_y(i) \) - the horizontal transport in \( Y \) direction for all \((X,Z_i)\) lines (of size \( N_y \)) in the cube.
- \( T^l_z(i) \) - the vertical transport in \( Z \) direction for all \((X,Y_i)\) lines (of size \( N_z \)) in the cube.
- \( XYtoZ(i) \) - the MPI shuffle from the \( XY \) to \( Z \) data distribution, that is, the processor \( P_i \) sends all the boxes in the range \((X,Y,Z_i)\) to other processors that need this data for the \( T^l_z \) transport part.
- \( ZtoXY(i) \) - the MPI shuffle from \( Z \) to \( XY \) data distribution, that is, the processor \( P_i \) sends all the boxes in the range \((X,Y_i,Z)\) to the other processors to complete the \( T^l_z \) and \( T^l_y \) transport part.

4.2.1 Sequential algorithm vs. parallel algorithm

First let us take a look at the sequential algorithm for the 3D-atmospheric chemistry simulation. The sequence of operations applied to the cubic model in one time step is shown in Table 2 (a).

In the parallel implementation, the cube has to be divided into \( P \) well-balanced parts (\( P \) is the number of processors) such that each piece of the original cube belongs to one processor. At this point of the program we need to find a compromise between minimizing the communication overhead of the program and maximizing the efficiency (the speedup) of the parallel program. Since the transport functions \( T^l_x, T^l_y, T^l_z \) work on vectors of elements of sizes \( N_x, N_y, N_z \), the cube has to be divided into smaller pieces for which at least one dimension is conserved. If only one dimension is conserved, then there must be three points of communication in only one time step (after each transport function). The problem of minimizing
the communication imposes for each piece of the cube to inherit 2 dimensions of the original cube. This leads us to initially “cut” the cube into horizontal “slices” so that each processor owns a parallelepiped of dimensions $N_x$, $N_y$ and $N_z/P$. It causes problems at the Z-transport point in the program, where data has to be redistributed among processors and each processor is assigned a parallelepiped of dimensions $N_x$, $N_y/P$, $N_z$. After the Z-transport, the initial configuration of the system has to be restored in order to go to the next time step of the algorithm. This implies we need two communication points for each time step of the algorithm.

Using the above ideas, the sequential algorithm could be rewritten into the parallel form as shown in Table 2 (b).

<table>
<thead>
<tr>
<th>(a) one step sequential</th>
<th>(b) one step parallel</th>
<th>(c) two steps, symmetric parallel</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_t(X, Y, Z)$</td>
<td>$C_t(X, Y, Z_i)$</td>
<td>$T_x^t(i)$</td>
</tr>
<tr>
<td>$T_x^t$</td>
<td>$T_y^t(i)$</td>
<td>$T_y^t(i)$</td>
</tr>
<tr>
<td>$T_y^t$</td>
<td>$T_z^t(i)$</td>
<td>$T_z^t(i)$</td>
</tr>
<tr>
<td>$T_z^t$</td>
<td>$XY$ to $Z(i)$</td>
<td>$XY$ to $Z(i)$</td>
</tr>
<tr>
<td></td>
<td>$T_z^t(i)$</td>
<td>$Z$ to $XY(i)$</td>
</tr>
<tr>
<td></td>
<td>$T_y^t(i)$</td>
<td>$T_x^t(i)$</td>
</tr>
</tbody>
</table>

Table 2: Possible implementations for each step

4.2.2 Symmetric sequential and parallel algorithm for 2 time steps

The sequential algorithm described above does not depend on the order in which the transport functions $T_x^t$, $T_y^t$ and $T_z^t$ are executed. Moreover, a simple analysis of the symmetric configuration

$$T_x^t \quad T_y^t \quad T_z^t \quad C^{2t}(X, Y, Z) \quad T_z^t \quad T_y^t \quad T_x^t$$

shows that the error in time for a symmetric algorithm applied to our model is less than the error in time for a non-symmetric algorithm. This means that it is better to rewrite the sequential algorithm in a symmetrical form as shown above.
Following the same idea, the symmetric parallel algorithm for 2 time steps is shown in Table 2 (c).

The number of points of communication in the program is reduced by half, since there are now 2 points of communication for 2 time steps, which implies one point of communication per time step. The data layout for the symmetric algorithm described above is presented in Figure 3.

![Figure 3: Data layout for the symmetric algorithm](image)

### 4.2.3 Non-overlapped vs. overlapped parallel algorithm

Each of the $T_x^i$, $T_y^i$, $T_z^i$ nodes in the graph described above corresponds to a function applied to each species in the chemistry system. That is, a node corresponds to $S$ calls of the function, where $S$ is the number of species in the system. One can easily see from the graph that a process starts to execute the following sequence of operations in one time step.

Obviously, the computation and communication parts of this parallel algorithm do not overlap at all. A processor starts sending data to other processors after completely finishing its work (steps 1, 2, 3). It starts the Z-transport (step 6) only after getting all the data that it expects to receive. This approach is inefficient, since the amount of time spent on communication between a Y-transport and a Z-transport (steps 3 and 6) may be very large, depending on the size of the problem. A more efficient version of the parallel algorithm should take advantage
of the fact that the only part of the program that cannot work without having all the concentrations for all species is the chemistry part. This means that in any other part of the program, the computation for some species may overlap the communication for other species already processed, as shown below:

### 4.2.4 3D-results

The MPI implementation of the 3D transport model has been tested on an SGI-Onix multiprocessor system with 8 processors (purchased with NFS support). Each processor is an MIPS R4000 running at 150 MHz.

For the chemistry part four test problems have been selected:

- **ART** - A small chemical mechanism with 10 reactions and 15 species described in [8].

- **SMOG** - Another small chemical mechanism modeling the photochemical smog with 10 reactions, 10 species.
• STRATO - A medium scale chemical mechanism modeling the stratosphere. It has 110 reactions and 38 species. Details can be found from [24].

• TROPO - A large scale chemical mechanism for modeling the troposphere. It has 178 reactions and 88 species. Details can be found from [20].

For the transport part we have used the following two test problems:

• 2D - The grid size is $33 \times 33 \times 1$ that is actually a 2D problem. There is a rotating wind. The initial concentration are cosine hills and rectangular regions as described in [8].

• 3D - The grid size is $33 \times 33 \times 1$. In each horizontal plane we have the same wind profile and the same initial concentrations as in the 2D case. There is no vertical wind.

Combining chemistry models and transport test problems we got 8 test problems that were tested on the SGI-Onyx. The results are presented in figure 4.2.4.

![Figure 4](image)

Figure 4: Results: For each number of processes the results for the test problems are in the order: ART-2D, SMOG-2D, STRATO-2D, TROPO-2D, ART-2D, SMOG-3D, STRATO-3D, TROPO-3D. For TROPO-3D model results for more than 3 processors are not available.

We have noticed almost linear speedups. The communication part kills the speedup only for the small 2D models and more than 4 processes. For larger models, as they occur in real simulations, the communication part requires a very small fraction of total execution time. We believe that this conclusion holds for an medium number of processors (16-32). For massive parallel architectures we plan to develop a tool based on the SUIF compiler as described in the previous section.
5 SENSITIVITY ANALYSIS

Comprehensive sensitivity analysis of air pollution models remains the exception rather than the rule. Most sensitivity analysis applied to air pollution modeling studies has been focused on the calculation of the local sensitivities (first order derivatives of output variables with respect to model parameters) in box model studies of gas phase chemistry (see [7]). The most common form of sensitivity studies with comprehensive atmospheric chemistry/transport models has been the so-called “brute force” method, i.e., a number of input parameters are selected to be varied and the simulation results are then compared. This method becomes less viable as the model becomes more comprehensive. A variety of alternative techniques are available including Green’s function analysis (see [7]), adjoint models and several variations of the direct decoupled methods (see [11]).

A recently developed technique for sensitivity study is automatic differentiation technology. Automatic differentiation is implemented by pre-compilers that analyze the code for evaluating a function of several variables and add instructions needed to compute the required derivatives. The resulting expanded code can simultaneously and efficiently evaluate derivatives and function values. This approach is superior to finite difference approximation of the derivatives because the numerical values of the computed derivatives are much more accurate and the computational effort is significantly lower (see [4], [3]).

A promising new implementation developed at Argonne National Laboratory and Rice University over the last couple of years and which has recently been awarded the 1995 Wilkinson Prize for Numerical Software is the package ADIFOR (Automatic Differentiation in FORTRAN, see [4]). It adopts a hybrid approach to computing derivatives that is generally based on the forward mode, but uses the reverse mode to compute the derivatives of assignment statements containing complicated expressions.

In [24] we analyzed forward mode Automatic Differentiation when applied to chemistry ODE solvers. This analysis shows that applying forward mode Automatic Differentiation is equivalent to using a direct method, i.e. to integrate forward in time the variational equations. This equivalence being established, a form of the direct decoupled method, which uses dedicated integrators, is considered.

We are presently extending the above described algorithms to three dimensional models, especially to computing the sensitivities of concentrations with respect to
the intensities and locations of pollutant emission sources and to explore more fully the capabilities of ADIFOR.

5.1 Preliminaries

If \( c_i(t) \) is the concentration of the \( i^{th} \) species, the kinetics of a chemical system is described as an initial value problem:

\[
\frac{dc_i(t)}{dt} = f(t, c_1, \ldots, c_n, \beta_1, \ldots, \beta_m),
\]

\[
c_i(t_0) = c_i^0, \quad i = 1 \cdots n
\]

where \( \beta_j, \ j = 1, \ldots, m \) are the parameters of the system (for example, reaction rate constants, etc).

The system (2) can be rewritten in matrix production - destruction form as:

\[
\frac{dc(t)}{dt} = P(c(t)) - D(c(t)) \cdot c(t)
\]

where \( P \in \mathbb{R}^n \), \( D \in \mathbb{R}^{n \times n} \), \( D = \text{diag}(D_i) \) are the production and the destruction terms, respectively.

The local sensitivity coefficients are defined as:

\[
s_{i,j}(t) = \frac{\partial c_i}{\partial \alpha_j}
\]

where \( \alpha_j \) represent either the initial values \( c_j^0 \) or the parameter \( \beta_j \). The term local refers to the fact that these sensitivities describe the system around a given set of values for the parameters \( \alpha \). The system being considered to respond linearly for small perturbations, \( s_{i,j} \) measures the ratio between the effect (absolute variation of the output \( \Delta c_i \)) and the cause (absolute variation of the input \( \Delta \alpha_j \)).

5.2 Overview of some computational methods for sensitivity analysis

There are many ways to compute sensitivities. Here we employ three different methods to compute the local sensitivity coefficients\(^1\). Conceptually, all three are equivalent, in the sense that a small perturbation of a certain input is propagated

\(^1\) Other techniques for sensitivity analysis, besides those described here, are available as well; one could mention Green's function method and adjoint models. Their description is beyond the scope of this paper.
forward through the system, and the corresponding deviation of all outputs is estimated. Thus, all the methods described below may be called forward propagation methods. They are effective when the sensitivity of all (or many) outputs with respect to one (or few) entries are desired.

5.2.1 “Brute-Force” Approach

Equation (2) is first solved for parameters $\alpha_1, ..., \alpha_j, ..., \alpha_m$ then for $\alpha_1, ..., \alpha_j, ..., \alpha_m$, and the obtained outputs are $\tilde{c}(t)$ and $\tilde{c}(t)$, respectively using one-sided and central difference approach.

The “Brute-Force” approach requires only one (for one-sided differences) or two (for central differences) extra function evaluations for each independent variable with respect to which sensitivities are desired. The main drawback is that the accuracy of the method is hard to analyze. The smaller the perturbation $\epsilon$, the lower the truncation error resulting from the omission of higher order terms (see the expansion of finite difference formulas in Taylor series), but the higher the loss-of-significance errors, resulting from subtracting two almost equal numbers. At the very best, the brute force approach results in a sensitivity approximation that has half the significant digits of $f$.

5.2.2 Variational Equations

By differentiating (3) with respect to the vector of parameters we obtain the variational equations:

$$\frac{d}{dt} \nabla c_i(t) = \nabla P_i(c) - \nabla D^i(c) \cdot c_i - D^i \cdot \nabla c_i, \quad i = 1, \ldots, n$$  \hspace{1cm} (5)

The fact that the sensitivities satisfy (5) can be proved rigorously, see for example [1]. The notation $\nabla A$ stands for the sensitivity coefficients vector $\frac{\partial A}{\partial \alpha}$, and “ $\hat{\doteq}$ ” represents a vector (element-by-element) assignment.

To obtain $\nabla c_i(t)$, one has to numerically integrate the large system obtained by appending together (2) and (5). This method is usually referred as the direct approach. The initial values $\nabla c_i(0)$ must be set properly \(^2\). There are two main drawbacks of this approach:

- The generation of the variational equations requires significant extra effort;

\(^2\)If $\nabla x = \frac{\partial x}{\partial c_i}$ then $\nabla c_i(0) = 1$, otherwise $\nabla c_i(0) = 0$.  

21
• The integration of the large appended system may be very time consuming.

5.2.3 Automatic Differentiation

Automatic differentiation\(^3\) techniques are based on the fact that any function (regardless of its complexity) is executed on a computer as a well-determined sequence of elementary operations like additions, multiplications and calls to elementary (intrinsic) functions such as sin, cos, etc.

By repeatedly applying the chain rule:

\[
\frac{\partial}{\partial t} f(g(t)) \bigg|_{t=t_0} = \left( \frac{\partial f(s)}{\partial s} \bigg|_{s=g(t_0)} \right) \cdot \left( \frac{\partial g(t)}{\partial t} \bigg|_{t=t_0} \right)
\]  

(6)

to the composition of these elementary operations one can compute, completely automatically, derivatives of \(f\) that are correct up to machine precision.

According to how the chain rule is used to propagate derivatives through the computation, one can distinguish two approaches to AD: the “forward” and the reverse “modes” (see [3], [4] for a detailed discussion).

• **The Forward Mode.** Is similar to the way in which the chain rule of differential calculus is usually taught. At each computational stage derivatives of the intermediate variables with respect to input variables are computed. These derivatives are propagated forward through the computational stages. From now on we will refer to forward automatic differentiation as FAD.

• **The Reverse Mode.** Adjoint quantities - the derivatives of the final result with respect to intermediate variables - are computed at each step. To propagate adjoints, one has to be able to reverse the flow of a program, and remember or recompute any intermediate value that nonlinearly impacts the final result.

5.3 Sensitivity calculation via FAD-generated variational equations

5.3.1 General setting

As shown by our previous considerations, automatic differentiation implicitly generates the linearized equations (5). To take advantage of that without facing any algorithmic problems, we propose the following hybrid approach: generate (5) via

\(^3\)Refered throughout the paper as AD.
automatic differentiation, then solve the variational system using an integrator of choice. This approach should work better than the “blind” automatic differentiation, which is in principle equivalent to the direct approach, with a fixed method for integrating the system; in the hybrid approach we have the extra degree of freedom of choosing the integration method and of doing different optimizations and easier than the usual way of implementing the direct approach where the variational equations are derived either by hand or by using symbolic manipulation.

To be more precise, suppose that the chemistry kinetics equations are described by the subroutine:

\[ \text{subroutine compute (in : } \alpha, \beta, c_0; \text{ out : } \dot{c} \text{)} \text{ where } \alpha, \beta \text{ are some parameters, } c \text{ is the vector of concentrations and } \dot{c} = \frac{dc}{dt} \text{ at given input arguments.} \]

Forward automatic differentiation will generate:

\[ \text{subroutine } g_{\text{compute}} \text{ (in : } \alpha, \beta, \nabla \beta, c_0, \nabla c_0; \text{ out : } \dot{c}, \nabla \dot{c} \text{)} \]

Under the assumptions that sensitivities obey (5) we have that:

\[ \nabla \dot{c} = \frac{\nabla dc}{dt} = \frac{d}{dt} \nabla c. \]  \hspace{1cm} (7)

Hence the subroutine \( g_{\text{compute}} \) describes completely (5). All that we have to do from now on is to properly set the initial conditions and to employ a standard numerical integrator.

5.4 Application of FAD to a comprehensive chemical mechanism

The chemical mechanism used in this study is that presently used in the STEM-II regional scale transport/ chemistry/ removal model (see [6]). This mechanism consists of 86 chemical species and 178 gas phase reactions. The mechanism, based on the work of Lurmann et al. ([22]) and Atkinson et al. ([2]) is representative of those presently being used in the study of chemically perturbed environments. The mechanism represents the major features of the photochemical oxidant cycle in the troposphere and can be used to study the chemistry of both highly polluted (e.g., near urban centers) and remote environments. The photochemical oxidant cycle is driven by solar energy and involves nitrogen oxides, reactive hydrocarbons, sulphur oxides and water vapour. The chemistry also involves naturally occurring species as well as those produced by anthropogenic activities. Many of the chemical reaction
Figure 5: The variations of ozone under different IPCC scenarios (see Table 2 for a detailed description)

rate coefficients vary with the intensity of solar radiation (photolysis rates), and thus follow a strong diurnal cycle. Others vary with temperature.

To test the robustness of the above numerical algorithms, we have employed six different scenarios\(^4\). These conditions represent various chemical environments ranging from: low \(NO_x\) oceanic boundary layer regions (Marine); high \(NO_x\) continental boundary layer regions without (Land) and with isoprene (Bio); dry upper tropospheric regions (Free); biomass burning plumes without (Plume 1) and with (Plume 2) reactive hydrocarbon species. Further details are presented in Chapter 7 of the current WMO Ozone Assessment (see [13]). ADIFOR was used to calculate sensitivities of ozone with respect to initial conditions and reaction rate parameters. In the simulations of these cases the QSSA method with a fixed stepsize of 10 seconds was used. This algorithm is suited for direct automatic differentiation, and is easy to implement when solving (5). Its use leads to results having 1-2 significant digits. The calculated ozone concentrations for the five cases are presented in Figure 5. In the marine boundary layer case, ozone is continuously destroyed throughout the 5 day period. The land and bio conditions show initially ozone production, followed by a net slight destruction of ozone over the simula-

\(^4\)These scenarios follow the IPCC (Intergovernmental Panel on Climate Change) photochemistry intercomparison.
tion period. In the dry free troposphere (Free) ozone values decrease very slowly. Both plume cases show a large net ozone production. The case (Plume-1) without non-methane hydrocarbons (NMHC) shows a much slower net ozone production rate, and a distinct diurnal behavior. The Plume-2 case (with NMHC) shows a very rapid increase in ozone, followed by a period of slow ozone destruction.

The calculated local sensitivities of ozone with respect to the initial conditions of each species for the Marine case are shown in Figure 6. Plotted are the normalized sensitivities at 120 hours of simulation. Also shown are the 8 largest (+) sensitivities (indicating ozone production) and (-) sensitivities. Under these conditions ozone concentrations are most sensitive to the initial concentration of ozone (as expected since this case has a net destruction of ozone). Ozone levels increase with increases in $CH_4$, $NO_x = NO + NO_2$ and $HNO_3$, species which both lead to the production of ozone and also help to modulate the $HO_2$ concentrations which is the principal lose mechanism for ozone under these conditions. Note also that

<table>
<thead>
<tr>
<th>LUMPED</th>
<th>OZONE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Species</td>
<td>Sensitivity</td>
</tr>
<tr>
<td>$HNO_3$</td>
<td>3.0591E+00</td>
</tr>
<tr>
<td>$O_3$</td>
<td>2.9348E+00</td>
</tr>
<tr>
<td>$CH_4$</td>
<td>2.1060E+00</td>
</tr>
<tr>
<td>RAO2</td>
<td>1.8158E+00</td>
</tr>
<tr>
<td>$R_3O_2$</td>
<td>1.6063E+00</td>
</tr>
<tr>
<td>PRN1</td>
<td>1.4141E+00</td>
</tr>
<tr>
<td>CRO2</td>
<td>1.2698E+00</td>
</tr>
<tr>
<td>MCRG</td>
<td>1.1467E+00</td>
</tr>
</tbody>
</table>

Figure 6: Marine case, lumped sensitivities w.r.t. initial values (left) and sensitivities of ozone w.r.t. initial values (right).
HNO$_3$ is the principal source of NO$_x$ in this case since its initial condition is an order of magnitude higher than NO$_x$. The largest negative sensitivity is that with respect to $H_2O_2$, which is the dominant source of HO$_2$ radicals.

Also shown are the lumped sensitivities. Lumped sensitivities can help describe the overall effect of a given perturbation. The lumped sensitivity of the system with respect to parameter $\alpha_j$ is defined as:

$$L(\alpha_j) = \sqrt{\sum_{i=1}^{N} \left( \frac{\partial c_i(t)}{\partial (\alpha_j)} \right)^2}$$ (8)

where $c_i$, $i = 1, ..., N$ are the concentrations of the component species. Since $L(\alpha_j)$, $j = 1, ..., m$ are functions of time, and since we are interested in the global effect of $\alpha_j$ over the system, we employ the mean values of the lumped sensitivity coefficients over the selected time horizon:

$$\bar{L}(\alpha_j) = \frac{1}{t_2 - t_1} \cdot \int_{t_1}^{t_2} L(\alpha_j) \cdot dt$$ (9)

If the mean sensitivities:

$$S_j = \frac{\bar{L}(\alpha_j)}{N}$$

are far less than one, the system is considered stable with respect to the initial conditions (see [7]). A powerful aspect of the use of ADIFOR is that it is readily applied to combined transport/chemistry problems

6 SUMMARY

Air quality models will continue to play a central role in our management of the environment. There remain important and challenging problems, related to the continuous improvement in the scientific foundation and numerical accuracy and efficiency of air quality models. In this paper we have addressed a few of these topics. The future directions of air quality models, while highly varied, will no doubtable be interesting.
7 REFERENCES

References


