Centrum voor Wiskunde en Informatica
REPORT*RAPPORT*

On the coupled solution of diffusion and chemistry in air pollution models

J.G. Verwer and J.G. Blom

Department of Numerical Mathematics

NM-R9509 1995

Report NM-R9509
ISSN 0169-0388

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

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

On the Coupled Solution of Diffusion and Chemistry in Air Pollution Models *

J.G. Verwer and J.G. Blom

CWI

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

Abstract

A numerical comparison is presented between different techniques considered for the coupled, implicit solution of vertical turbulent diffusion and nonlinear chemical transformations in air pollution models.

AMS Subject Classification (1991): Primary:65M06, 65M20. Secondary: 65Y05, 65Y20.
CR Subject Classification (1991): G.1.8, G.1.1,J.2
Keywords & Phrases: long range transport air pollution models, time-dependent advection-diffusion-reaction, stiff ODEs, numerical methods.
Note: We gratefully acknowledge support from the RIVM (projects EUSMOG and CIRK)

1. INTRODUCTION

Important processes in air pollution models are advective transport, diffusion and nonlinear chemical transformations. While advection and horizontal diffusion usually can be dealt with explicitly, vertical turbulent diffusion and the chemical transformations give rise to stiffness and require an implicit approach. The two latter processes are described by the one-space dimensional parabolic system

$$\frac{\partial \rho}{\partial t} = \frac{\partial}{\partial \sigma}(K\frac{\partial \rho}{\partial \sigma}) + P(t,\rho) - L(t,\rho)\rho, \quad t > t_0, \quad 0 < \sigma < \sigma_H, \tag{1.1}$$

supplemented with the initial condition $\rho(\sigma, t_0) = \rho^0(\sigma)$ and for $t > t_0$ appropriate boundary conditions at the bottom ($\sigma = 0$) and the top of the air column ($\sigma = \sigma_H$). The dependent variable $\rho = \rho(\sigma, t)$ is a vector of m concentration values. The scalar coefficient $K = K(\sigma, t)$ parametrizes the vertical turbulent diffusion and depends on the so-called mixing height. The vector function $P(t, \rho)$ defines production terms from the chemical transformations and emissions. Losses and depositions are defined by the term $L(t, \rho)\rho$, where L is a diagonal matrix.

In air pollution research one commonly applies operator splitting. Because during the day important chemical species have time constants of the same magnitude as vertical turbulent diffusion, the error introduced in treating vertical turbulent diffusion and stiff chemistry decoupled, as in operator splitting, is difficult to estimate and is avoided by the standard implicit approach which treats the two coupled. However, the coupled implicit approach requires the solution of large sets of nonlinear algebraic equations, which in the case of many species leads to an expensive procedure when the standard modified Newton approach is followed. In this note we discuss a novel Gauss-Seidel technique and a novel modified Newton - multigrid technique, both implemented in an implicit backward differentiation (BDF) solver. The Gauss-Seidel technique, which was proposed in [1,2], performs very well and brings the coupled implicit approach within reach, even for very large chemical models.

---

*Submitted to a special issue of the *Zeitschrift für Angewandte Mathematik und Mechanik* (ZAMM), devoted to the proceedings of ICIAM95 (International Congress on Industrial and Applied Mathematics, Hamburg, 1995).

2. The BDF solver

In the actual application non-uniform grids are used. For this purpose we transform (1.1) to

$$\frac{\partial \phi}{\partial t} = \frac{1}{g'}\frac{\partial}{\partial \xi}(\frac{K}{g'}\frac{\partial \phi}{\partial \xi}) + P(t,\phi) - L(t,\phi)\phi, \quad t > t_0, \quad 0 < \xi < 1, \tag{2.1}$$

where $\phi(\xi,t) = \rho(\sigma,t)$ and $\sigma = g(\xi), 0 \leq \xi \leq 1$, for a given function g satisfying $g(0) = 0, g(1) = \sigma_H$. The diffusion term is approximated in 2nd-order and vertex centered at internal, uniform grid points $\xi_k = kh, 1 \leq k \leq N-1$, where $h = 1/N$. Let $c_k(t) \approx \phi(\xi_k,t)$. The approximation then reads

$$\frac{1}{h^2 g'_k}\left(\frac{\kappa_k + \kappa_{k+1}}{2}c_{k+1} - \frac{\kappa_{k-1} + 2\kappa_k + \kappa_{k+1}}{2}c_k + \frac{\kappa_{k-1} + \kappa_k}{2}c_{k-1}\right), \tag{2.2}$$

where $\kappa = K/g'$. The semi-discrete unknowns c_0, c_N at the boundary points are defined by the boundary conditions which for the test model in this paper are of Neumann type. We discretize the boundary conditions here 1st-order by imposing $c_0 = c_1$ and $c_N = c_{N-1}$. This 1st-order boundary approximation will not lead to a deterioration of accuracy because at the boundaries K is very small. Let $N_\sigma = N - 1$ and let the ODE system

$$\frac{d}{dt}c_k = f_k(t,c) \doteq d_k(t,c) + P(t,c_k) - L(t,c_k)c_k, \quad t > t_0, \quad 1 \leq k \leq N_\sigma, \tag{2.3}$$

represent the semi-discrete system, where $c(t)$ is the complete grid function on the space grid and $d_k(t,c)$ represents (2.2). Notice that $c_k(t)$ is a vector in $\Re^m$ and that the diffusion operator introduces no coupling between different species. For each of the species the semi-discrete diffusion operator is equal and constituted by the same tridiagonal matrix. The species are coupled through the chemistry term $P(t,c_k) - L(t,c_k)c_k$, but only per grid point.

For the numerical integration of (2.3) we use the 2nd-order, two-step implicit BDF formula in variable step form,

$$c_k^{n+1} = C_k^n + \gamma\tau f_k(t_{n+1}, c^{n+1}), \quad 1 \leq k \leq N_\sigma, \quad n \geq 1, \tag{2.4}$$

where $c_k^n \approx c_k(t_n)$ and τ denotes the step size. The other variables have the usual meaning (see [1,2]). As starting method the 1st-order implicit Euler rule is used. This combination yields 2nd-order accurate time stepping which for atmospheric transport applications is sufficient in view of the modest accuracy requirement. Generally, relative accuracy larger than 1% is superfluous. The BDF formula has been implemented in an automatic code provided with step size control, similar as in [1,2]. To save space details are omitted.

3. Modified Newton and Gauss-Seidel iteration

In view of the stiffness, the usual solution technique for the approximate solution of (2.4) is modified Newton (MN) iteration. Each iteration then involves the solution of a block-tridiagonal, linear system of the form

$$M\Delta c = b, \quad M = I - \gamma\tau(A + J), \quad A = A_h \otimes I_m, \quad J = \mathrm{diag}\,(J_1, \ldots, J_{N_\sigma}), \tag{3.1}$$

where Δc is the complete MN correction vector $[(\Delta c_1)^T, \ldots, (\Delta c_{N_\sigma})^T]^T$, A_h is the tridiagonal matrix constituted by (2.2) and J_k denotes the chemical kinetics Jacobian matrix. M is computed at times selected by the MN strategy of the BDF solver. This computation can be realized for low costs, since J_k is sparse. For sufficiently large atmospheric chemistry schemes, the number of nonzeros even amounts to only 10%, approximately. So it pays to evaluate M analytically. Suppose that system (3.1) is solved with a standard, non-pivoting band solver. Since the dimension of M equals mN_σ and the bandwith $2m+1$, the operations count (divisions and multiplications) for the backsolve and factorization amount to $\approx 2m^2N_\sigma$ and $\approx m^3N_\sigma$, respectively. However, in current air pollution models from practice, m can be quite large, up to 100, approximately. The numbers of operations thus reveal that even with a band solver the direct solution of (3.1) can be very costly. While the sparsity which we exploit in the computation of M can also be used to advantage in the linear system solution in the ODE case [3], in the PDE case this advantage is lost, as the fill-in in the factorization of M is almost 100%.

These considerations demonstrate that for bringing the coupled implicit integration of (1.1) within practical reach, a solution technique significantly more efficient than MN iteration is necessary. An excellent candidate to replace MN iteration is the Gauss-Seidel (GS) iteration method from [2]. We will show that this method not only is much more efficient, but also avoids the storage of the Jacobian matrix M. Storage considerations are of minor importance for 1D problems, but for 3D problems the demand of storage of a numerical method readily becomes the main bottleneck, even on large super computers (see [2]).

We will outline the GS method. Suppressing the index $n+1$ and t_{n+1} for convenience, we first write (2.4) as

$$c_k = C_k^n + \gamma\tau d_k(c) + \gamma\tau P(c_k) - \gamma\tau L(c_k)c_k, \quad 1 \le k \le N_\sigma. \tag{3.2}$$

Next, let $c_k^{(j)}$ be the j-th component of c_k and introduce, for each of the m species, the following grid vectors,

$$c^{(j)} = [c_1^{(j)}, \ldots, c_{N_\sigma}^{(j)}]^T, \quad P^{(j)}(c) = [P^{(j)}(c_1), \ldots, P^{(j)}(c_{N_\sigma})]^T, \quad j = 1, \ldots, m. \tag{3.3}$$

Introduce the diagonal matrices $L^{(j)}(c) = \mathrm{diag}(L^{(j)}(c_1), \ldots, L^{(j)}(c_{N_\sigma}))$. Then we may write

$$c^{(j)} = C^{(j)} + \gamma\tau A_h c^{(j)} + \gamma\tau P^{(j)}(c) - \gamma\tau L^{(j)}(c)c^{(j)}, \quad j = 1, \ldots, m, \tag{3.4}$$

or

$$c^{(j)} = \left(I - \gamma\tau A_h + \gamma\tau L^{(j)}(c)\right)^{-1} \left(C^{(j)} + \gamma\tau P^{(j)}(c)\right), \quad j = 1, \ldots, m. \tag{3.5}$$

The GS iteration for approximating $c^{(j)}, 1 \le j \le m$, is carried out on equation (3.5) and consists of the following calculations. Let $c_{[i]}$ denote the i-th iterate for c and q the ratio of successive step sizes. Then we apply,

$$\begin{aligned}
&1.\ \text{Initial estimation: } i = 0,\ c_{[i]} = \max(0, c^n + q(c^n - c^{n-1})).\\
&\quad 2.\ \text{Compute, in the order } j = 1, \ldots, m:\\
&\qquad 2a.\ L^{(j)}(c_{[i]}), P^{(j)}(c_{[i]}).\\
&\qquad 2b.\ \text{LU-decompose } I - \gamma\tau A_h + \gamma\tau L^{(j)}(c_{[i]}).\\
&\qquad 2c.\ \text{Backsolve } c_{[i+1]}^{(j)}.\\
&\qquad 2d.\ \text{Update } c_{[i]} = (c_{[i+1]}^{(1)}, \ldots, c_{[i+1]}^{(j)}, c_{[i]}^{(j+1)}, \ldots, c_{[i]}^{(m)}).\\
&3.\ \text{Define } c_{[i+1]} = c_{[i]}.\ \text{If more iterations are required, then go to 2.}
\end{aligned} \tag{3.6}$$

The initial estimate is obtained via linear extrapolation and limited to zero to avoid negative values. The approximations then are corrected specieswise (in the order $j = 1, ..., m$) and simultaneously over the grid since the diffusion term is treated implicitly. This requires the tridiagonal matrix calculations $2b, 2c$ any time a species is corrected. Thus, except for the tridiagonal matrix calculations, the GS process is truly explicit and no Jacobian matrices are computed or stored. The update stage 2d reveals the Gauss-Seidel nature. Without chemistry the diffusion is treated implicitly in the usual way and without diffusion the explicit GS process from [1] is recovered. Notice that (3.6) differs from the classical nonlinear GS method, since this classical method does not distinguish between diffusion and reaction terms and would be applied directly to (2.4), resulting in a scalarly implicit process.

We prefer to apply (3.6) with a fixed number of iterations per time step. The final iterate is then accepted as the new approximation, provided the step size control accepts the solution. This implies that the implicit solution is computed only approximately. In [2] the use of a fixed number of iterations has been shown to work well and competitive with an iteration strategy. A number of 4 iterations per time step was chosen experimentally. One time step with the GS method then roughly costs 4 function evaluations and in total $20mN_\sigma$ operations (divisions and multiplications) for the tridiagonal computations 2b and 2c. Obviously, if with this low number of iterations the GS method leads to approximately the same step size sequences as the MN method, with comparable accuracy, then we may expect a huge increase in the efficiency/accuracy performance of the BDF solver.

4. NUMERICAL ILLUSTRATION

This section presents a numerical comparison between the two iteration techniques to illustrate the above expectation. For this purpose we use the same 1D test problem as in [2]. This problem is based on the EMEP MSC-W ozone chemistry which models 140 reactions between 66 species ($m = 66$) and is provided with a time-space dependent (mixing height) turbulent diffusion coefficient $K(t, \sigma)$. Photolysis rates for the chemistry undergo a discontinuity at sunset and sunrise. As a result, large local concentration gradients exist which require variable step sizes in time and a non-uniform space grid. We put $N = 32$ and $g(\xi) = \sigma_H \xi^3$ with $\sigma_H = 2000$m. At the two lowest grid levels emissions and depositions are specified. A time interval of 112 hours is used which starts at sunrise (04.00 hours) at day one ($t_0 = 14400$ sec.) and ends at sunset (20.00 hours) at day five ($t = 417600$ sec.). The 112 hour interval is divided into 56 two hour intervals, at each of which the integration is restarted using a tenfold smaller step size. This division into 56 subintegrations is in accordance with regular changes in model coefficients and input. Such changes can introduce a discontinuity (like at sunset and sunrise), motivating the many restarts. Observe that when the current procedures would be used in an operator splitting scheme, for dealing with advection in 3D, frequent restarts will be made too. It should be emphasized that restarts increase the overhead in the MN iteration, due to the step size changes. Efficiency is measured by CPU time and accuracy by the number of correct digits

$$SDA = -log_{10}\left(\frac{1}{66}\sum_{j=1}^{66}\left[\sum_{n=8}^{56}\sum_{k=1}^{40}(sol_{kj}^n - app_{kj}^n)^2\right]^{\frac{1}{2}} / \left[\sum_{n=8}^{56}\sum_{k=1}^{40}(sol_{kj}^n)^2\right]^{\frac{1}{2}}\right), \tag{4.1}$$

where sol_{kj}^n denotes an ODE reference solution and app_{kj}^n the numerical solution. Hence we use the (l_2/l_1) ODE error in the comparisons and pay no attention to spatial accuracy. The times t_n are restricted to $t_n = 14400 + 7200n$ with $8 \leq n \leq 56$, j runs over all species, and k runs over the grid. This means we sample at the end of each 2 hour interval, but for the first time at sunset at the first day because we start with an arbitrary initial condition. We thus assume to have get rid of the initial transients at the first sunset.

Table 1 shows for the four local error tolerances $TOL = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$ the results of the comparison. These results are self-evident and fully confirm that (3.6) indeed leads to a huge reduction in CPU time and only a minor loss in accuracy compared to the standard use of MN iteration and a direct linear band solver.

	$TOL = 10^{-1}$			$TOL = 10^{-2}$			$TOL = 10^{-3}$			$TOL = 10^{-4}$		
	SDA	Steps	CPUs	SDA	Steps	CPUs	SDA	Steps	CPUs	SDA	Steps	CPUs
GS	2.0	1255	22	2.5	2642	41	3.3	6979	101	4.4	19292	271
MN	2.7	1259	1020	3.4	2648	1186	4.0	6973	1758	5.0	19292	3191
		Iters	Jacs		Iters	Jacs		Iters	Jacs		Iters	Jacs
		4126	669		7164	667		16201	731		41251	781

Table 1: Comparison between MN and GS iteration. Steps is the number of accepted integration steps. Iters is the number of MN-iterations or backsolves. Jacs is the number of updates of the complete Jacobian M which is close to the number of LU's for M.

	$TOL = 10^{-1}$			$TOL = 10^{-2}$			$TOL = 10^{-3}$			$TOL = 10^{-4}$		
	SDA	Steps	CPUs	SDA	Steps	CPUs	SDA	Steps	CPUs	SDA	Steps	CPUs
MN	2.8	1670	727	3.3	2726	781	3.9	6973	1327	4.7	19292	3011
		Iters	Jacs		Iters	Jacs		Iters	Jacs		Iters	Jacs
		6850	1440		8599	1096		17255	967		42751	994

Table 2: The performance of the block-Jacobi multigrid solver.

We conclude with an attempt to exploit the sparsity of the chemistry Jacobians J_k for the solution of (3.1) by a damped block-Jacobi multigrid technique. Let $A = D + L + U$, with D the diagonal, L the lower and U the upper triangular part. Consider the smoother

$$M_\omega \Delta c_{[i+1]} = \gamma\tau \left((1 - \frac{1}{\omega})D + L + U \right) \Delta c_{[i]} + b, \quad M_\omega = I - \gamma\tau\,(\frac{1}{\omega}D + J), \quad \omega = \frac{2}{3}, \tag{4.2}$$

where the damping only affects the diffusion part. This smoother enables to exploit the sparsity in the treatment of the N_ω decoupled, m-dimensional blocks of the block-diagonal matrix M_ω, in a similar way as in [3]. Because the main diagonal of J_k is nearly full, the sparsity pattern of these blocks hardly differs from that of J_k. For stiff components the convergence will be excellent because J_k is treated implicitly. For nonstiff components and normal step sizes the usual convergence behavior of Jacobi iteration does hold, which suggests to use a multigrid technique to accelerate convergence. A V-cycle was implemented with one pre-smoothing step, one post-smoothing step and the discretization-coarse grid-approximation. The coarsest grid in the cycle (N = 2) contains only one internal grid point and was solved exactly. The prolongation was based on full weighting ($\frac{1}{2}\ \frac{2}{4}\ \frac{1}{2}$) and linear interpolation was used for the projection. We use a single V-cycle per MN iteration and hence do not iterate till convergence. This means that the linear system solution is computed only approximately which leaves the error detection to the MN iteration strategy and the step size control.

The choice for the block-Jacobi multigrid solver is based on two arguments. First, each J_k is treated implicitly as in MN iteration in the ODE case [3]. Second, the sparsity enables to economize on the numerical algebra. If we suppose that the costs are determined by the LU-decompositions and backsolves for the N_ω sparse blocks of M_ω, then the number of operations for all backsolves for our V-cycle amounts to 4 times the labour on the finest grid (1D). Hence the backsolve costs add up to $4 f_{BS} N_\omega m^2$, where f_{BS} is the sparsity fraction of the operations needed for a full matrix. The labour for the LU-decompositions for our V-cycle amounts to 2 times the labour on the finest grid, which yields $2/3 f_{LU} N_\omega m^3$. Hence if $f_{BS} < 1/2$ and $f_{LU} < 1$ (see Section 3), the costs are reduced. Because for large chemical models both f_{BS} and f_{LU} are expected to be significantly smaller, a test with the multigrid technique has been carried out. The results are summarized in Table 2.

We see that the accuracy is maintained and that CPU times have decreased. However, the gain is not very significant, which is partly due to the fact that more Iters and Jacs are spent. Like the direct solver, the multigrid technique also requires a considerable amount of storage compared to the matrix-free GS technique. Because the Gauss-Seidel method also is much faster, this method is proposed for the coupled, implicit treatment of vertical turbulent diffusion and chemistry.

REFERENCES

1. J.G. Verwer. Gauss-Seidel iteration for stiff ODEs from chemical kinetics. SIAM J. Sci. Comput. 15, 1243 - 1250, 1994.
2. J.G. Verwer, J.G. Blom and W. Hundsdorfer. An implicit-explicit approach for atmospheric transport-chemistry problems. CWI Report NM-R9501, 1995.
3. J.G. Verwer, J.G. Blom, M. Van Loon and E.J. Spee. A comparison of stiff ODE solvers for atmospheric chemistry problems. CWI Report NM-R9505, 1995 (to appear in Atmospheric Environment).