Centrum voor Wiskunde en Informatica
REPORT*RAPPORT*

A comparison of stiff ODE solvers for atmospheric chemistry problems

J.G. Verwer, J.G. Blom, M. van Loon and E.J. Spee

Department of Numerical Mathematics

NM-R9505 1995

Report NM-R9505
ISSN 0169-0388

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

A Comparison of Stiff ODE Solvers for Atmospheric Chemistry Problems

J.G. Verwer, J.G. Blom, M. van Loon and E.J. Spee

CWI

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

Abstract

In the operator splitting solution of atmospheric transport-chemistry problems modeling air pollution, a major task is the numerical integration of the stiff ODE systems describing the chemical transformations. In this note a numerical comparison is presented between two special purpose solvers developed for this task.

AMS Subject Classification (1991): Primary: 65L05. Secondary: 65F05, 65F10, 80A30.
CR Subject Classification (1991): G.1.7, G.1.3
Keywords & Phrases: atmospheric chemistry, air pollution modeling, numerical stiff ODEs.
Note: This report is one of a series on the development of algorithms for long range transport air pollution models.
We gratefully acknowledge support from the RIVM for the projects EUSMOG and CIRK.

1. Introduction

This note deals with the numerical integration of stiff ODE initial value problems from atmospheric chemistry. Although the numerical stiff ODE field is well developed and an interesting variety of efficient and quite reliable stiff ODE solvers is available [4], the general experience is that for three-space-dimensional transport-chemistry problems, where stiff ODE integrations are carried out at thousands of grid points, still faster tailor-made solvers are needed. In this note we compare two such solvers on a set of three atmospheric chemistry ODE problems from practice.

The first solver is TWOSTEP, a simple code based on the 2nd-order, two-step, variable step size BDF formula. The code has been designed as a special purpose solver for atmospheric chemistry problems. The solver is special purpose in the sense that Gauss-Seidel iteration is used for approximately solving the implicitly defined BDF solution [15, 16], rather than the more commonly used modified Newton technique. The Gauss-Seidel iteration renders the integration explicit which implies low start up costs and a low memory demand. This is of advantage in an operator splitting application of a stiff solver. The Gauss-Seidel iteration is related to the Jacobi iteration as used in the quasi-steady-state-approximation (QSSA) approach, to which it compares very favorably [16].

The second solver is the general, state-of-the-art BDF code VODE [1, 3], provided with sparse matrix techniques to reduce the numerical algebra overhead. Exploiting sparsity has been shown before to be advantageous for atmospheric chemistry problems [8]. Since this code is known as an efficient solver for chemical kinetics problems, optimal use of sparsity should make it one of the best candidates from the numerical stiff ODE field for tailor-made solution of atmospheric transport-chemistry problems. Note that VODE is comparable with LSODE [5] which is often used in the field.

Our main purpose thus is to test TWOSTEP against VODE and to check whether the claims made for this explicit code ([15, 16]) are confirmed if we largely economize on the numerical algebra overhead of the modified Newton process used in a standard solver.

Section 2.1 is devoted to the test methodology used to compare the two solvers, which are discussed briefly in Sections 2.2 and 2.3. Sections 3, 4 and 5 each contain the results for a selected test problem. Concluding remarks are given in the final Section 6.

2. Test methodology

2.1 Set up of experiments

The solvers are tested as if they were used in an operator splitting environment. This means that we split up the total integration interval in a lot of subintervals, representing the length of advection steps taken in the operator splitting. For each subinterval we then restart the integration of the stiff solver. Frequently restarting an implicit solver is not attractive since this involves an unusual amount of small steps in the start phase, which enlarges the overhead in the numerical linear algebra. All three test examples are based on chemical transformations of which part are photochemical. This means that part of the reaction constants are determined by time of the day dependent photolysis rates which undergo a near discontinuity at sunrise and sunset. In all three examples we take the same integration interval covering 112 hours. This interval starts at 04.00 hour at day one and ends at 20.00 hour at day five. Thus the time interval is sufficiently long to include a number of diurnal cycles for the important photochemical transformations and to include a large set of different initial conditions due to the restarts.

The total integration interval of 112 hours is split up in 56 2-hour subintervals which involves 56 restarts. Our measure of accuracy is based on the relative root mean square error $RRMS_k$ for each species k, taken over the endpoints of all 2-hour intervals over the 112 hours. Hence,

$$RRMS_k = \sqrt{\frac{\sum_{n=1}^{N} (y_k^n - y_k(t_n))^2}{\sum_{n=1}^{N} (y_k(t_n))^2}}, \tag{2.1}$$

where $N = 56$, $t_n = 14400 + 7200n$ sec. and $y_k(t_n)$ represents a sufficiently accurate reference solution. We then calculate the number of significant digits for the average of $RRMS_k$, defined by

$$SDA = -\log_{10}\left(\frac{1}{m}\sum_{k=1}^{m} RRMS_k\right). \tag{2.2}$$

Our comparison focuses on modest accuracy, that is relative accuracies near 1%, since higher accuracy levels are redundant for the actual practice of three-dimensional air pollution modeling. For all three test problems we will use the same set of error tolerances for the variable step size control, viz.

$$rtol = 10^{-l},\ atol = 1.0,\ l = 1, 2, 3, 4, 5, \tag{2.3}$$

for all species. In all three test problems the unit of concentration is number of molecules per cm^3. We therefore effectively invoke a relative error control. For some species (radicals) the concentration can be smaller than 1, but these values are insignificant for the overall solution and require no local error control. Since the two solvers use quite different solution techniques, and are therefore difficult to compare, efficiency is measured by CPU time (SGI-Indy workstation, f77 -O option, double precision).

2.2 VODE

To illustrate the wide efficiency range for this general solver [1], it has been used in three different ways. These will be indicated by VODE1, VODE2 and VODE3.

VODE1 is the standard black box use, i.e., no optional input is used and the method parameters $ITASK$ (not essential for our comparison) and MF are set to 4 and 22, respectively. The choice $MF = 22$ implies that the code generates the Jacobian automatically by numerical differencing, while the standard full matrix linear algebra routines DGEFA (factoring) and DGESL (backsolves) from LINPACK are used. $MF = 22$ also implies extra storage because both the Jacobian and its LU-decomposed form are stored. This saves Jacobian updates, on the other hand additional storage may be a disadvantage for higher-space dimensional problems. Because no optional input is used, there is no constraint on the step size selection. For example, the code selects its own starting step size.

In the second manner of calling VODE, special problem properties are exploited so as to obtain a more efficient numerical solution process. VODE2 means use of $ITASK = 4$ and $MF = 21$. The choice $MF = 21$ is important since this implies that Jacobians have to be provided by the user in analytic form. We emphasize that this already can save CPU time because of sparsity. However, VODE2 still uses the same (full matrix) linear algebra routines DGEFA and DGESL as VODE1 does. Hence the sparsity is here not yet exploited in the solution of the linear systems. Like $MF = 22$, the choice $MF = 21$ implies that extra storage is needed. A second important difference with VODE1 is that we also prescribe a starting, a minimal and maximal step size. These are, in seconds,

$$\tau_{start} = 1,\ \tau_{min} = 1,\ \tau_{max} = 900. \tag{2.4}$$

The lower bound of 1 sec. serves to reduce the numerical algebra overhead stemming from the frequent restarts (matrix factorizations caused by step size changes). Step sizes below 1 sec. are redundant in this application. The minimal time constant values of importance for the actual practice are about 1 min. in size and species with a time constant smaller than 1 sec. almost instantaneously get in their (solution dependent) steady state when they are perturbed. Hence the choice of 1 sec. is reasonable and safe compared to 1 min. Note that without the 1 sec. lower bound extremely small steps could be taken since atmospheric chemistry problems containing photochemical reactions can possess time constants as small as 10^{-9} sec. In this connection we should remark that the automatic local error control of VODE does signal time constants smaller than 1 sec. Consequently, we had to overrule the rejection strategy to enable (2.4). Without this overruling the code returns with an error message due to the constraint $\tau_{min} = 1$ and interrupts the integration. In general this is perfectly all-right, of course, and VODE should not be blamed for it. Finally, the maximal step size of 900 secs. is also quite reasonable on chemical grounds, although VODE, and also TWOSTEP, perform equally satisfactorily without this constraint. VODE2 thus has been prepared to solve the atmospheric chemistry problems more efficiently than VODE1.

VODE3 is completely identical to VODE2, except that now the sparsity of the Jacobian is exploited to reduce the costs of solving the linear systems in the modified Newton iteration. We emphasize that this can be very profitable for large systems, as will be illustrated below. To keep the fill-in of the LU-factorization small, the components in the ODE system are reordered in a way that the most dense rows reside in the bottom of the matrix and the most sparse rows at the top. For our test examples we have done this using facilities offered by MAPLE [2]. We note in passing that one and the same sparsity structure is used for the whole time interval. At night, when photochemical reactions are switched off, the sparsity is somewhat larger, but for simplicity we have not used this (small) advantage. After having determined the fill-in arising from the standard LU-factorization, also with MAPLE, the linear systems then can be solved quite efficiently with the ILU (Incomplete LU) routines DSILUS (factorization) and DSLUI2 (backsolves) from the Sparse Linear Algebra Package (SLAP). SLAP is a public domain code written by Greenbaum and Seager (with contributions of several other authors) that is available from Netlib[3]. We should remark, however, that we have slightly modified the original SLAP versions to increase their efficiency. Hence these routines replace the LINPACK routines DGEFA and DGESL, respectively. Like the LINPACK routines, they factorize and backsolve, but omit all redundant calculations in which a zero occurs. It should be remarked, though, that now no longer pivoting occurs in the matrix factorization. This could give rise to errors in the linear system solution which otherwise would have been avoided. We have not experienced problems of this sort. Of course, if the factorization fails, then the step size control of VODE will detect this and a change in the step size will improve matters. It seems that for solving stiff ODEs pivoting is only rarely required (cf. [8, 10]). VODE3 thus has been prepared to solve the atmospheric chemistry problems with higher efficiency than VODE2 and VODE1.

Finally, Table 1 illustrates the sparsity for the three test examples treated in Section 3. The table obviously predicts a large reduction in CPU time for Problem III. For I and II the gain will be less since for these two the dimension is modest. Note that for other atmospheric chemical kinetics problems with a large dimension, a similar level of sparsity exists as for our Problem III. For example, [8] discusses a smog chemistry problem of dimension 92 with only 695 nonzero entries in the Jacobian. The fill-in, minimized in a similar way as in our approach, only increases this to 839.

	Dimension	Nonzero entries	Fill-in	% Nonzeros LU
Problem I	15	57	0	25
Problem II	18	111	12	38
Problem III	66	500	107	14

Table 1: Sparsity data.

2.3 TWOSTEP

The code, as described in [16], has been applied in two different ways, in the remainder indicated by TWOSTEP1 and TWOSTEP2. By TWOSTEP1 we mean the standard black box use, where at any time step two Gauss-Seidel iterations are used and, in this case, the step sizes are constrained by (2.4). It should be emphasized that two iterations are by far not enough to let the Gauss-Seidel iteration fully converge. Our experience is that the overall accuracy of the code improves with the number of iterations, but the efficiency generally not. We therefore prefer a small number of iterations, which is attractive anyhow after a restart with a small step size. TWOSTEP2 refers to the same way of application, but in addition certain ad-hoc rules are used to exploit special problem properties. This means that special techniques like lumping or group iteration are combined with the Gauss-Seidel iteration. The ad-hoc rules used will be discussed with the test problem and of course serve to obtain a more efficient numerical solution process.

3. Example problem I: The EUSMOG chemistry

The chemical model is identical to the one described in [9]. This model is a highly parameterized version of the EMEP MSC-W model [11, 12] that will be used in Section 5. It consists of 15 reactions between 15 species:

1.	$NO + O_3$	$\longrightarrow$	NO_2	k_1	$= 2.0\ {}_{10}{}^{-12} \exp(-1400/T)$
2.	$NO_2 + h\nu$	$\longrightarrow$	$NO + O_3$	k_2	$= 1.45\ {}_{10}{}^{-2} \exp(-0.4/\cos Z)$
3.	$NO_2 + OH$	$\longrightarrow$	NO_3^a	k_3	$= 1.68\ {}_{10}{}^{-12} \exp(560/T)$
4.	$2NO_2 + O_3$	$\longrightarrow$	$2NO_3^a$	k_4	$=$ see appendix
5.	$O_3 + h\nu$	$\longrightarrow$	$b_1 OH + b_2 O_3$	k_5	$= 2.0\ {}_{10}{}^{-4} \exp(-1.4/\cos Z)$
6.	$C_2H_6 + OH$	$\longrightarrow$	$a_1 O_3$	k_6	$= 8.7\ {}_{10}{}^{-12} \exp(-1070/T)$
7.	$C_4H_{10} + OH$	$\longrightarrow$	$a_2 O_3$	k_7	$= 1.4\ {}_{10}{}^{-11} \exp(-559/T)$
8.	$C_2H_4 + OH$	$\longrightarrow$	$a_3 O_3$	k_8	$= 1.66\ {}_{10}{}^{-12} \exp(474/T)$
9.	$C_3H_6 + OH$	$\longrightarrow$	$a_4 O_3$	k_9	$= 4.1\ {}_{10}{}^{-12} \exp(545/T)$
10.	XYL $+ OH$	$\longrightarrow$	$a_5 O_3$	k_{10}	$= 1.4\ {}_{10}{}^{-11}$
11.	ISO $+ OH$	$\longrightarrow$	$a_6 O_3$	k_{11}	$= 2.55\ {}_{10}{}^{-11} \exp(410/T)$
12.	$CO + OH$	$\longrightarrow$	$a_7 O_3$	k_{12}	$= 2.4\ {}_{10}{}^{-13}$
13.	$CH_4 + OH$	$\longrightarrow$	$a_8 O_3$	k_{13}	$= 2.9\ {}_{10}{}^{-12} \exp(-1820/T)$
14.	$SO_2 + OH$	$\longrightarrow$	SO_4	k_{14}	$= 2.32\ {}_{10}{}^{-10} \exp(-457/T)$
15.	SO_2	$\longrightarrow$	SO_4	k_{15}	$= 1.39\ {}_{10}{}^{-6}$

The parameter Z denotes the solar zenith angle and T is the temperature in Kelvin. In [9] the above set of reactions is part of a smog prediction model, consisting, apart from chemistry, of advection, horizontal and vertical diffusion, emission and deposition. Each of these processes is treated in an operator splitting context. This means that per split step for each grid cell one ODE describing the chemical reactions has to be solved. Here, however, we only carry out box calculations with the chemical model. In order to get more realistic concentration profiles, emission terms Q_i and deposition terms vg_i have been added. For NO_2, O_3 and SO_2 deposition is specified and for NO, the VOCs and SO_2, emission. All time-dependent coefficients are updated at the beginning of the 2-hour intervals and then frozen for the rest of the time. A specification of all parameters and input data defining the complete ODE system is given in Appendix A.

Evaluation of the eigenvalues of the Jacobian matrix for various conditions has shown that the eigenvalues range from -10^1 sec^{-1} to -10^{-10} sec^{-1}, approximately, indicating that the system is (moderately) stiff. Recall that time steps of a few minutes size should be achievable for an efficient code.

First we mention the order in which the components are treated in the Gauss-Seidel process (equal for TWOSTEP1/2): NO_2, NO, O_3, OH, NO_3^a, the VOCs, SO_2 and SO_4. Obviously, the order plays a role when only a few iterations are used. As TWOSTEP1 is the standard way to use TWOSTEP, we now only describe how we exploited special problem characteristics in TWOSTEP2, in order to improve the convergence of the Gauss-Seidel iteration. In the chemistry literature the approach we follow is called 'lumping' [6, 7]. In our case, the lumping involves the introduction of the two 'new' species $NO_x = NO_2 + NO$ and $O_x = NO_2 + O_3$. For both a differential equation can be specified with positive production and loss terms. The differential equation for NO_x, for example, takes the form

$$\begin{aligned}\frac{d}{dt}NO_x &= -k_3 \cdot NO_2 \cdot OH - 2k_4 \cdot (NO_2)^2 \cdot O_3 - vg_1 \cdot NO_2 + Q_{NO} \\ &= -k_3 \cdot (NO_x - NO) \cdot OH - 2k_4 \cdot NO_2 \cdot O_3 \cdot (NO_x - NO) \\ &\quad -vg_1 \cdot (NO_x - NO) + Q_{NO} \\ &= P_{NO_x} - L_{NO_x} NO_x, \end{aligned} \tag{3.1}$$

where

$$P_{NO_x} = k_3 \cdot NO \cdot OH + 2k_4 NO_2 \cdot O3 \cdot NO + vg_1 \cdot NO + Q_{NO} \tag{3.2}$$

and

$$L_{NO_x} = k_3 \cdot OH + 2k_4 \cdot NO_2 \cdot O_3 + vg_1. \tag{3.3}$$

When we would compute the implicit BDF2 solution for the original ODE system augmented with the lump species exactly, then the lumping relations also hold for this exact implicit solution. That is, at any n-th time step we have

$$NO_x^n = NO_2^n + NO^n, \quad O_x^n = NO_2^n + O_3^n. \tag{3.4}$$

This, however, is not true for the approximate BDF2 solution obtained with Gauss-Seidel iteration. The idea behind the lumping technique is to impose the lumping relations (3.4) on the solution obtained after each Gauss-Seidel iteration, thus hoping that this will improve the convergence to the exact implicit solution. In our case, the lumping of NO_2 and NO into NO_x, etc. underlies the assumption that the first two reactions from above are in some sense dominant in the whole set of chemical transformations. Because, if we consider only reaction 1 and 2, then we have

$$\frac{d}{dt}NO_x = 0, \quad \frac{d}{dt}O_x = 0, \tag{3.5}$$

showing that for these two reactions NO_x and O_x are conserved. Consequently, if the first two reactions are truly dominant in the whole system, then NO_x and O_x are expected to vary slowly. This, in turn, implies that the integration of the differential equation for NO_x and O_x can be done accurately, so that imposing relation (3.4) by correcting one of the grouped species will make sense,

We perform the integration of the new species as follows. At the end of a Gauss-Seidel iteration, we first compute NO_x from the BDF2 formula, using the production-loss form. We thus get

$$NO_x = (Y + \gamma\tau P)/(1.0 + \gamma\tau L), \tag{3.6}$$

where P and L are evaluated at the solution generated by the last Gauss-Seidel iteration and Y denotes the history term of the BDF2 formula for NO_x. Next, if $NO_2 > NO$, then NO_2 is recomputed from NO_x, otherwise NO is recomputed. In the same way O_x is computed and O_3 or NO_2 is recalculated from O_x. Consequently, relation (3.4) now holds after any Gauss-Seidel iteration. In Appendix C we will show that lumping can be interpreted as a simple form of preconditioning.

For VODE1 and VODE2 we use the same ordering of components as TWOSTEP in the Gauss-Seidel process. This ordering results in a large amount of fill-in elements in the LU decomposition of the matrix $P = I - \gamma\tau J$, J denoting the Jacobian. Whereas the original matrix P has 57 nonzero elements, the sum of the nonzero elements in L and U is 148. For the present chemical system it is possible to reorder the components in such a way that no fill-in elements arise at all, i.e. the total number of nonzero elements in L and U is equal to 57, which number should be compared with 225 for the standard LU-decomposition. For completeness we give the new order of the components: first the VOCs, followed by SO_2, SO_4, NO, NO_3^a, NO_2, O_3 and OH. This new ordering is used by VODE3.

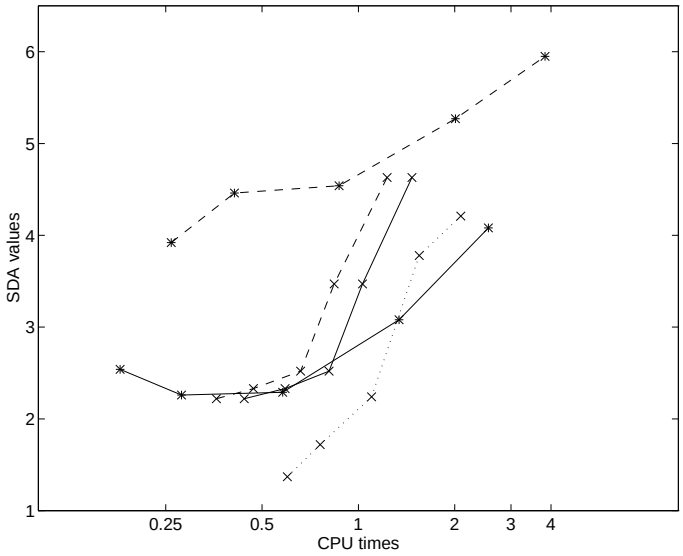

Figure 1: Results for Problem I. TWOSTEP1 (*, solid), TWOSTEP2 (*, dashed), VODE1 (x, dotted), VODE2 (x, solid), VODE3 (x, dashed).

Figure 1 shows all accuracy-efficiency plots for Problem I. The marks on the plots correspond with the five tolerances (2.3). Interestingly, lumping improves the TWOSTEP solution more than expected and in fact brings it very close to the true implicit BDF2 solution. This is shown in Figure 2 where again the plots for TWOSTEP1 and TWOSTEP2 are given, together with the plot for the implicit BDF2 solution. For the three different cases the same step sizes were used. We see that the plots for TWOSTEP2 and the BDF2 solution practically coincide, showing that the lumping indeed has improved the convergence of the Gauss-Seidel iteration as used in TWOSTEP1. Recall that only two iterations have been carried out. Hence for this chemistry the lumping works out very well and because the additional costs are negligible, it is attractive to use. We should note, however, that lumping is problem dependent and in general improves accuracy certainly not as much as here.

The fact that there is hardly any difference in the TWOSTEP1/2 accuracies for the larger tolerances, which also occurs for VODE2 and VODE3, is due to the upper step size bound of 900 sec. Yet, for TWOSTEP the CPU becomes significantly larger when the tolerance is decreased. This is caused by coefficient updates at the beginning of every 2-hour interval, which introduces small, but fast initial transients. These initial transients have disappeared near the end of the 2-hour intervals and have no influence on the accuracies there, but their presence obviously reduces the step size in the initial phase of the integrations. This behavior is characteristic for operator splitting applications, indicating that for TWOSTEP the choice of a minimal step size is of practical importance.

As expected, VODE2 outperforms VODE1. We found that this is due to the step size restriction (2.4) and not a result of using the exact sparse Jacobian instead of a full numerical approximation. For the present model, with only 15 components, the overhead of this numerical approximation is too small to

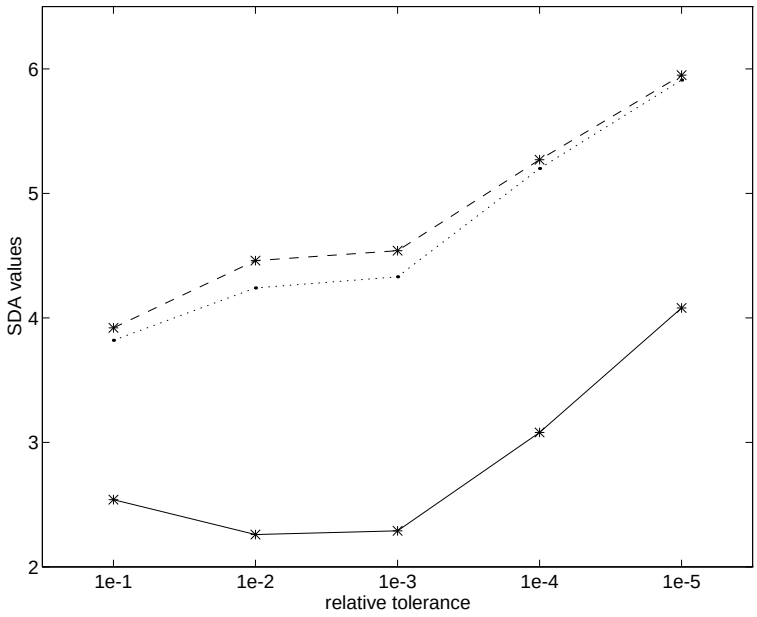

Figure 2: Results for Problem I: TWOSTEP1 (*,solid), TWOSTEP2 (*,dashed), the implicit BDF2 solution (dotted).

become visible in the results. Restriction (2.4) prevents VODE2 from taking very large step sizes which will reduce the accuracy at the end of the 2-hour intervals, but also prevents VODE2 from taking very small step sizes lower than 1 sec. in the initial phase. As noted before, these small step sizes are of no relevance for the accuracies we measure. VODE2 spends only about 30% in routines that handle the LU decomposition and the backsolves, which of course is too small to get much gain in CPU by replacing VODE2 by VODE3. The latter needs approximately 20% less CPU time than VODE2. These numbers reveal that by using the sparse matrix routines, the linear algebra costs have been reduced by a factor 3. Finally, when we compare with the most efficient VODE version, which is VODE3, we can conclude that TWOSTEP2 outperforms VODE3 convincingly. Also TWOSTEP1 is faster in the 1% error range, although the step size selection needs some more attention for this test example.

4. Example problem II: The methane CIRK chemistry

We obtained our second chemical model from [13]. This model is used in long term, global studies and describes a methane oxidation cycle. It consists of 46 reactions between 19 species. Thirteen reactions depend on the solar zenith angle which, different from Problem I, is taken continuous and hence calculated in each time step. The problem is very stiff. Eigenvalues of the Jacobian lie between -10^9 sec^{-1} and -10^{-20} sec^{-1}, approximately. There are two extremely large eigenvalues which originate from the free radicals O^1D and O^3P. These are absent in Problem I, which explains the modest stiffness of that problem. A complete description of the model defining the ODE system used in the experiments can be found in Appendix B.

The order of the components used in the Gauss-Seidel process is (equal for TWOSTEP1/2): O^1D, O^3P, OH, NO_3, HO_2, N_2O_5, NO, NO_2, O_3, HNO_3, HO_2NO_2, HNO_2, H_2O_2, $HCHO$, CH_3OOH, CH_3O_2, CH_4, NO_x, O_x. TWOSTEP2 uses the same NO_x and O_x lumping as for Problem I. VODE1 and VODE2 use a slightly different order with O_x omitted (cf. Table 2 of Appendix B). For the modified Newton process as used by VODE1/2 the order is to a great extent irrelevant, while also the lumped species play no role here. The sequence used by VODE1/2 results in 31 fill-in elements in the LU

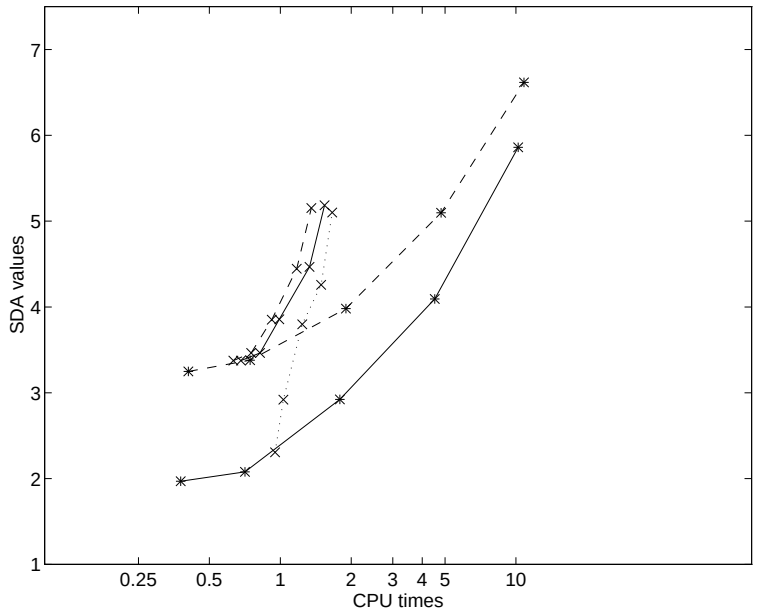

Figure 3: Results for Problem II. TWOSTEP1 (*, solid), TWOSTEP2(*,dashed) VODE1 (x, dotted), VODE2 (x, solid), VODE3 (x, dashed).

decomposition. Reordering leads to a fill-in of 12 elements. Thus the total number of nonzeroes after reordering is $111 + 12 = 123$. The new sequence used by VODE3 reads CH_4, O^1D, HNO_2, H_2O_2, N_2O_5, HNO_3, HO_2NO_2, CH_3OOH, O_3, $HCHO$, CH_3O2, NO_3, O^3P, NO, NO_2, HO_2, NO_x, OH.

Figure 3 shows all results obtained for Problem II. First we notice that, similar as for Problem I, the simple lumping trick improves the TWOSTEP accuracy considerably and for minor costs. The VODE results compare well with those for Problem I. Supplying VODE with an analytical Jacobian and a minimal and maximal step size improves the performance significantly (VODE2). However, here also the gain in CPU from using the sparsity of the Jacobian in the Jacobian evaluation is low, only 10%. Similar as for Problem I, this also holds for the change to VODE3 where the sparsity is exploited in the solution of the linear systems. In the accuracy region of greatest practical interest, both solvers perform well although TWOSTEP is again the most efficient one.

5. Example problem III: The EMEP chemistry

The third example problem is identical to the urban test case reported in [16] for the EMEP MSC-W ozone chemistry model. This chemistry model consists of about 140 reactions between 66 species. The model is state-of-the-art in the field of regional air pollution modeling. Rate coefficients are often variable, depending on temperature and, for some, humidity. The model takes into account emission inputs and dry and wet removal processes. Photolysis rates obviously depend on solar elevation and cloudiness. These rates vary continuously in time, but undergo a discontinuity at sunset and sunrise. As regards to stiffness Problem III is comparable with II. Because the model is too large to describe here, we refer to [11, 12] for more details.

Figure 4 shows all accuracy-efficiency plots we obtained for Problem III. TWOSTEP2 now differs from the version used before. For TWOSTEP2 also two GS-iterations were used, but within each such iteration five group iterations on the $NO_y + O_3$ group are added (cf. [16]). The species in this group are strongly coupled, so it makes sense to perform this group iteration. We emphasize that this group iteration involves a minor change in the code and hence is very simply applicable. Because the group consists of only 7 species, the additional work is minor and it obviously improves the Gauss-Seidel iteration. The TWOSTEP2 result should be compared with the best result obtained for VODE, which clearly is the

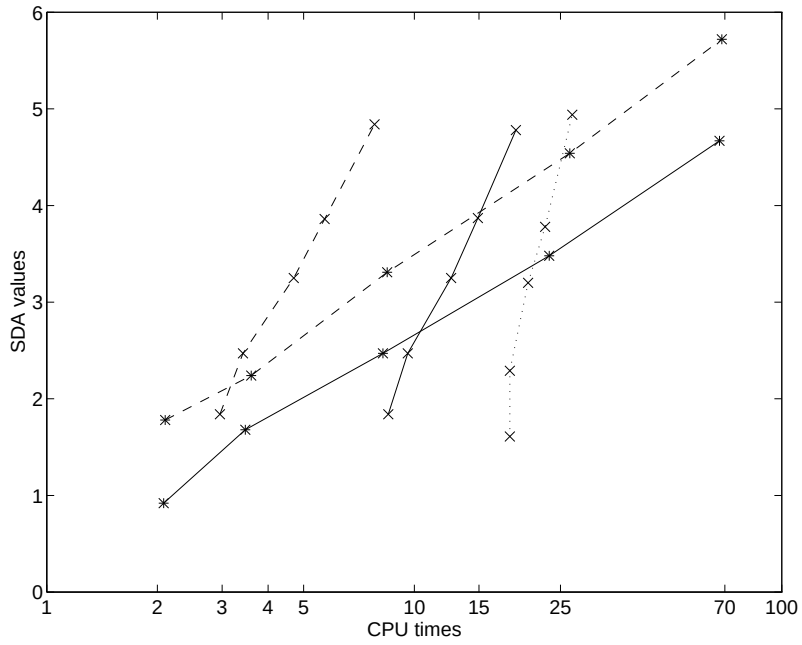

Figure 4: Results for Problem III. TWOSTEP1 (*, solid), TWOSTEP2 (*, dashed), VODE1 (x, dotted), VODE2 (x, solid), VODE3 (x, dashed).

VODE3 case. We see that for the accuracy range of greatest practical interest, TWOSTEP2 and VODE3 are comparable. For higher accuracies the variable order VODE3 is more efficient because it then uses the higher order BDF formulas. The figure also nicely illustrates that by an intelligent use, standard stiff ODE codes like VODE can be improved dramatically. In the low accuracy range VODE3 is about six times more efficient than VODE1. We emphasize that the difference between VODE2 and VODE3 is only due to the use of the sparse matrix techniques, which works out very well for this test problem due to its large number of components. The difference between VODE1 and VODE2 is due to using the analytical sparse Jacobian and the step size constraints (2.4). Both reduce part of the CPU time needed by the black box version VODE1.

6. Concluding remarks

The MAPLE tools for automatically computing the analytical Jacobian and for deriving the data structure for the ILU routines are easy to use. The sparse matrix technique based on the ILU routines from the SLAP library handles the solution of the linear systems well. We have encountered no difficulties in using VODE3, which solves the linear systems without pivoting. Similar experiences were reported by [8] and [10].

For large problems from atmospheric chemistry, like the EMEP MSC-W model, the sparse matrix technique can lead to significant savings in CPU time for codes like VODE. This experience corresponds with the results reported by [8]. For atmospheric chemistry models of a more moderate size, like the EUSMOG and CIRK model, the gain by exploiting sparsity does hardly pay. For such models, with about 20 species say, the solution costs of the linear systems in VODE are simply too low compared to the costs of all other calculations.

There is room for both TWOSTEP and VODE. When used in an intelligent way, both solve our test examples efficiently. In the low accuracy region TWOSTEP always seems to be somewhat faster. Obviously, the (problem dependent) lumping technique and/or the group iteration are recommendable

for TWOSTEP when only a few Gauss-Seidel iterations are used.

An advantage of Gauss-Seidel iteration is that it works matrix free and hence the memory demand is low, which is of interest when grid vectorization is employed. As shown in [17], Gauss-Seidel iteration can be nearly optimally vectorized over the grid, in a similar way as modified Newton combined with sparse solution techniques in the code SMVGEAR [8].

A further attractive feature of Gauss-Seidel iteration is that it can be efficiently extended to solve chemistry and vertical turbulent diffusion in a coupled way [17]. This is not true for the modified Newton process as regards the exploitation of sparsity. If diffusion is coupled with chemistry, then the sparsity of the chemistry Jacobian is almost completely lost in the factorization of the banded linear system.

References

1. P.N. Brown, G.D. Byrne, A.C. Hindmarsh. *VODE: A variable coefficient ODE solver.* SIAM J. Sci. Stat. Comput. 10, 1038 - 1051, 1989.

2. B.W. Char, K.O. Geddes, G.H. Gonnet, M.B. Monagan and S.M. Watt. *Maple V Language Reference Manual.* Springer-Verlag, New York, 1991.

3. J.J. Dongarra, E. Grosse. *Distribution of software via electronic mail.* Commun. ACM 30, 1987, 403 - 407 (netlib@research.att.com).

4. E. Hairer, G. Wanner. *Solving ordinary differential equations II. Stiff and differential algebraic problems.* Springer-Verlag, Berlin, 1991.

5. A.C. Hindmarsh. *LSODE and LSODI, two new initial value ordinary differential equation solvers.* ACM-SIGNUM Newsletter 15, 10-11, 1980.

6. E. Hesstvedt, Ø. Hov, I.S.A. Isaksen. *Quasi-steady state approximation in air pollution modeling: Comparison of two numerical schemes for oxidant prediction.* Int. J. Chem. Kinet. 10, 1978, 971 - 994.

7. Ø. Hov, I.S.A. Isaksen, E. Hesstvedt. *A numerical method to predict secondary air pollutants with an application on oxidant generation in an urban atmosphere.* Proceedings WMO Symposium on Boundary Layer Physics Applied to Specific Problems of Air Pollution, Norrkoping, Sweden, 19-23 June, 1978, WMO Publication No.510, Geneva, Switzerland, pp. 219-226, 1978.

8. M.Z. Jacobson, R.P. Turco, *SMVGEAR: A Sparse-Matrix, Vectorized Gear Code for Atmospheric Models.* Atmospheric Environment 28, 273 - 284, 1994.

9. M. van Loon. *Numerical smog prediction I: the physical and chemical model.* CWI Report NM-R9411

10. A.H. Sherman, A.C. Hindmarsh. *GEARS: a package for the solution of sparse, stiff ordinary differential equations.* Lawrence Livermore Laboratory Report UCRL-84102, 1080.

11. D. Simpson, Y. Andersson-Sköld, M.E. Jenkin. *Updating the chemical scheme for the EMEP MSC-W oxidant model: current status.* Technical Report 2/93, EMEP MSC-W, The Norwegian Meteorological Institute, Oslo, 1993.

12. D. Simpson. *Biogenic VOC in Europe. Part II: implications for ozone control strategies.* 1994 (submitted).

13. H. The, National Institute for Environmental Protection and Hygiene (RIVM), Bilthoven, The Netherlands, Private Communication, 1994.

14. J.G. Verwer, M. van Loon. *An evaluation of explicit pseudo-steady state approximation schemes for stiff ODE systems from chemical kinetics.* J. Comput. Phys. 113, 1994, 347 - 352.

15. J.G. Verwer. *Gauss-Seidel iteration for stiff ODEs from chemical kinetics.* SIAM J. Sci. Comput. 15, 1243 - 1250, 1994.

16. J.G. Verwer, D. Simpson. *Explicit methods for stiff ODEs from atmospheric chemistry.* CWI Report NM-R9409 (to appear in Appl. Numer. Math.)

17. J.G. Verwer, J.G. Blom, W.H. Hundsdorfer. *An implicit-explicit approach for atmospheric transport-chemistry problems*, CWI Report NM-R9501, 1995.

A. The EUSMOG model

We solve the system of equations

$$\dot{y_i} \;=\; f_i(t,y) \;\equiv\; P_i(t,y) - L_i(t,y)y_i, \quad 1 \le i \le n, \tag{A.1}$$

with in our case n=15. The unit of concentration is $molec \cdot cm^{-3}$. The function f is of course not only depending on the concentration vector y but on all kinds of parameters, including meteorological conditions. The latter are specified by the solar zenith angle γ, the relative humidity rh and the temperature T in Kelvin. These parameters are contained in the reaction rates rk_i. Other parameters, like the stoichiometry factors a_i, are concentration dependent. Below first all species are listed together with their initial values (Table 2), next the function f is given in Fortran form, followed by a description of the involved parameters and reaction rates.

	Name	Species	Concentration
1	Nitrogen dioxide	NO_2	$4.92 \cdot 10^{11}$
2	Nitrogen oxide	NO	0.0
3	Ozone	O_3	$4.92 \cdot 10^{11}$
4	Hydroxyl radical	OH	0.0
5	Nitrate aerosol	NO_3^a	$2.46 \cdot 10^{11}$
6	Ethane	C_4H_6	$2.46 \cdot 10^{11}$
7	Butane	C_4H_{10}	$2.46 \cdot 10^{11}$
8	Ethene	C_4H_4	$2.46 \cdot 10^{10}$
9	Propene	C_3H_6	$2.46 \cdot 10^{10}$
10	Xylene	XYL	$2.46 \cdot 10^{10}$
11	Isoprene	ISO	$2.46 \cdot 10^{10}$
12	Carbon monoxide	CO	$3.69 \cdot 10^{12}$
13	Methane	CH_4	$4.182 \cdot 10^{13}$
14	Sulphur dioxide	SO_2	0.0
15	Sulphate aerosol	SO_4	0.0

Table 2: Initial concentrations in [mol.cm^{-3}] for the EUSMOG chemistry.

```
c calculate reaction constant 4 (composite reaction)
      nox=y(1)+y(2)
      IF (nox.ne.0.) THEN
       no3   = rka/(rkb*y(2)+rkc+rkd*rkf*y(1)*ch2o/(rke+rkf*ch2o))
       rk(4) = rkf*rkd*ch2o*no3/(rke+rkf*ch2o)
      ELSE
       rk(4) = 0.
      ENDIF

c calculate stoichiometry factors
      nox=-2.46E10/max(1.e-10,y(1)+y(2))
      do i=1,8
         a(i)=a1(i)*exp(b(i)*nox)+a2(i)
      enddo

      f(1) = rk(1)*y(2)*y(3)-rk(2)*y(1)-rk(3)*y(1)*y(4)
     +        -2.*rk(4)*y(1)**2*y(3)-vg(1)*y(1)
      f(2) = rk(2)*y(1)-rk(1)*y(2)*y(3)+Q(2)
      f(3) = rk(2)*y(1)-rk(1)*y(2)*y(3)-rk(4)*y(1)**2*y(3)
```

```
+       -(1.-b2)*rk(5)*y(3)-vg(3)*y(3)
 f(4) = b1*rk(5)*y(3)-rk(3)*y(1)*y(4)-rk(14)*y(14)*y(4)
 f(5) = 2.*rk(4)*y(1)**2*y(3) + rk(3)*y(1)*y(4)
 do i=6,13
   f(i) = -rk(i)*y(i)*y(4)
   f(4) = f(4)+f(i)
   f(3) = f(3)-a(i-5)*f(i)
   f(i) = f(i) + Q(i)
 enddo
 f(14) = -rk(14)*y(14)*y(4)-rk(15)*y(14)-vg(14)*y(14)+Q(14)
 f(15) = rk(14)*y(14)*y(4)+rk(15)*y(14)+Q(15)
```

Below we give a (Fortran) description of the parameters necessary to compute the function f which are not already specified in the text:

```
 b1=4.6E-10*ch2o/(2.3E-10*ch2o+4.93E8*exp(-100/tk))
 b2=1.-b1/2.
 rka=1.2E-13*exp(-2450./tk)
 rkb=1.5E-11*exp(170./tk)
 rkc=0.192*exp(-0.059*sec)+2.43E-2*exp(-0.081*sec)
 rkd=1.47E-12*exp(-60./tk)
 rke=8.5E14*exp(-11080./tk)
 rkf=1.3E-21
```

where `sec` is $1/\cos Z$. The emission and deposition terms, Q and vg as well as the parameters $a1$, $a2$ and b are given by the following Fortran statements

```
 data vg/2.e-6,0.,5.e-6,0.,0.,0.,0.,0.,0.,0.,0.,0.,3.e-6,0./
 data Q/0.0,1.25e6,0.0,0.0,0.0,3.7e5,1.e6,2.9e5,1.3e5,3.3e5,
+   1.e5,2.5e7,0.,1.0e6,3.0e4/
 data a1/4.4,5.8,5.5,7.0,7.0,0.0,0.9,4.0/
 data a2/1.7,2.2,0.3,0.4,3.0,1.0,0.0,0.0/
 data b /0.35,0.35,5.e-4,5.e-4,0.35,0.35,0.25,0.25/
```

The reaction rates as well as the parameters $b1$ and $b2$ depend on temperature T (in the code `tk`) and the relative humidity rh and the water concentration $[H_2O]$ (in the code `ch2o`). For the period selected, the temperature and relative humidity are modeled by

$$\begin{aligned} T &= 293.1 - 1.91 * sin\left(\frac{\pi}{12}tod\right) - 2.78 * cos\left(\frac{\pi}{12}tod\right) \\ rh &= 0.6 + 0.0764 * sin\left(\frac{\pi}{12}tod\right) + 0.114 * cos\left(\frac{\pi}{12}tod\right) \end{aligned}$$

where tod denotes the time-of-day in hours. The water concentration is modeled as a function of T and rh according to

$$[H_2O] = \frac{4.357521\,_{10}{}^{19}\, rh}{T} \exp\left(-(753.0 - 0.57 * T) * (\frac{1}{T} - \frac{1}{273.16}) * 18./1.986\right).$$

The water concentration is an important quantity for the parameters $b1$ and $b2$ in reaction 5. Finally, the cosine of the solar zenith angle Z is calculated from

$$\cos Z = \max(\sin\Delta \sin\phi + \cos\Delta \cos\phi \cos(\frac{\pi}{12}(tod - 12.67)), 0), \tag{A.2}$$

where latitude $\phi = 52^o$ and declination Δ is given by

$$\begin{aligned}\Delta &= 0.006918 - 0.399912\cos\tilde{d} + 0.070257\sin\tilde{d} - 0.006758\cos(2\tilde{d}) + \\ &\quad +0.000907\sin(2\tilde{d}) - 0.002697\cos(3\tilde{d}) + 0.00148\sin(3\tilde{d})\end{aligned}$$

with

$$\tilde{d} = \frac{2\pi d}{365}, \tag{A.3}$$

and d the day in the year ($1 \leq d \leq 365$). In our test $d = 181$. Note that at night photolysis is switched off by putting $\cos Z = 0$.

B. The CIRK model

The chemical scheme consists of 46 reactions between 19 species, two of which are the lump species NO_x and O_x. These species with their initial concentrations are listed in Table 3 together with H_2O, CO, O_2 and M for which the concentrations are taken constant. The values for temperature and pressure are also kept constant. In the reactions below $T = 288.15\,K$ and $p = 1.013 * 10^5\,Pa$:

0.	$OH + CO + O_2$	$\rightarrow$	$CO_2 + HO_2$	$rk_0 = 1.5 * 10^{-13}(1 + 0.60 * 10^{-5}p)$
1.	$OH + CH_4 + O_2$	$\rightarrow$	$H_2O + CH_3O_2$	$rk_1 = 2.36 * 10^{-12}\,e^{(-1710\,/T)}$
3.	$CH_3O_2 + HO_2$	$\rightarrow$	$CH_3OOH + O_2$	$rk_3 = 7.7 * 10^{-14}\,e^{(1.3*10^3\,/T)}$
4.	$2\,CH_3O_2$	$\rightarrow$	$2\,HCHO + 2\,HO_2$	$rk_4 = 3.1 * 10^{-13}\,e^{(220\,/T)}$
5.	$HCHO + O^3P + O_2$	$\rightarrow$	$CO + HO_2 + OH$	$rk_5 = 1.7 * 10^{-13}\,e^{(-1550\,/T)}$
6.	$O^3P + O_2$	$\rightarrow$	O_3	$rk_6 = F3(6 * 10^{-34},\ 2.3,\ 0,\ 0)$
7.	$NO + O_3$	$\rightarrow$	$NO_2 + O_2$	$rk_7 = 1.8 * 10^{-12}\,e^{(-1370\,/T)}$
8.	$NO + HO_2$	$\rightarrow$	$NO_2 + OH$	$rk_8 = 3.7 * 10^{-12}\,e^{(240\,/T)}$
9.	$NO_2 + O_3$	$\rightarrow$	NO_3	$rk_9 = 1.2 * 10^{-13}\,e^{(-2450\,/T)}$
10.	$NO_2 + NO_3$	$\rightarrow$	$NO + NO_2 + O_2$	$rk_{10} = 2.3 * 10^{-13}\,e^{(-1.0*10^3\,/T)}$
11.	$NO + NO_3$	$\rightarrow$	$2\,NO_2$	$rk_{11} = 1.3 * 10^{-11}\,e^{(250\,/T)}$
12.	$OH + NO_2$	$\rightarrow$	HNO_3	$rk_{12} = F3(2.6 * 10^{-30},\ 3.2,\ 2.4 * 10^{-11},\ 1.3)$
13.	$OH + NO$	$\rightarrow$	HNO_2	$rk_{13} = F3(7 * 10^{-31},\ 2.6,\ 1.5 * 10^{-11},\ 0.5)$
14.	$OH + HNO_3$	$\rightarrow$	NO_3	$rk_{14} = k_0 + (k_3/\,(1 + (k_3/k_2)))$
15.	$NO_2 + NO_3$	$\rightarrow$	N_2O_5	$rk_{15} = F3(2.2 * 10^{-30},\ 4.3,\ 1.5 * 10^{-12},\ 0.5)$
16.	N_2O_5	$\rightarrow$	$NO_2 + NO_3$	$rk_{16} = 5.7 * 10^{14}\,e^{(-1.06*10^4\,/T)}$
17.	$NO_2 + HO_2$	$\rightarrow$	HO_2NO_2	$rk_{17} = F3(2 * 10^{-31},\ 2.7,\ 4.2 * 10^{-12},\ 2)$
18.	HO_2NO_2	$\rightarrow$	$NO_2 + HO_2$	$rk_{18} = 2 * 10^{16}\,e^{(-1.19*10^4\,/T)}$
19.	$OH + HO_2NO_2$	$\rightarrow$	$NO_2 + H_2O + O_2$	$rk_{19} = 1.3 * 10^{-12}\,e^{(380\,/T)}$
20.	$OH + O_3$	$\rightarrow$	$HO_2 + O_2$	$rk_{20} = 1.6 * 10^{-12}\,e^{(-960\,/T)}$
21.	$O_3 + HO_2$	$\rightarrow$	$OH + 2\,O_2$	$rk_{21} = 1.4 * 10^{-14}\,e^{(-580\,/T)}$
22.	$OH + H_2O_2$	$\rightarrow$	$HO_2 + H_2O$	$rk_{22} = 3.1 * 10^{-12}\,e^{(-187\,/T)}$
23.	$2\,HO_2$	$\rightarrow$	$H_2O_2 + O_2$	$rk_{23} =$ see below
24.	$OH + HO_2$	$\rightarrow$	$H_2O + O_2$	$rk_{24} =$ see below
25.	$2\,OH$	$\rightarrow$	$O^3P + H_2O$	$rk_{25} = 4.2 * 10^{-12}\,e^{(-242\,/T)}$
26.	$2\,OH$	$\rightarrow$	H_2O_2	$rk_{26} = F3(6.9 * 10^{-31},\ 0.8,\ 1 * 10^{-11},\ 1)$
27.	O^1D	$\rightarrow$	O^3P	$rk_{27} = [M]\,2.88 * 10^{-11}$

0.	$CH_3OOH + OH$	$\rightarrow$	$CH_3O_2 + H_2O$	$k1(0) = 1.0 * 10^{-11}$
1.	$HCHO + OH + O_2$	$\rightarrow$	$HO_2 + CO + H_2O$	$k1(1) = 1.0 * 10^{-11}$
2.	$O^1D + H_2O$	$\rightarrow$	$2\,OH$	$k1(2) = 2.2 * 10^{-10}$
3.	$N_2O_5 + H_2O$	$\rightarrow$	$2\,HNO_3$	$k1(3) = 1.3 * 10^{-21}$
4.	$NO_2 + OH$	$\rightarrow$	HNO_3	$k1(4) = 1.5 * 10^{-11}$

0.	$HCHO + 2\,O_2 + h\nu$	$\rightarrow$	$CO + 2\,HO_2$	$jc_0 = 5.4 * 10^{-5}\, e^{(-0.79\ \sec(Z))}$
1.	$HCHO + h\nu$	$\rightarrow$	$CO + H_2$	$jc_1 = 6.65 * 10^{-5}\, e^{(-0.6\ \sec(Z))}$
2.	$CH_3OOH + O_2 + h\nu$	$\rightarrow$	$HCHO + HO_2 + OH$	$jc_2 = 2.27 * 10^{-5}\, e^{(-0.6208\ \sec(Z))}$
3.	$O_3 + h\nu$	$\rightarrow$	$O^3P + O_2$	$jc_3 = 1.23 * 10^{-3}\, e^{(-0.6\ \sec(Z))}$
4.	$O_3 + h\nu$	$\rightarrow$	$O^1D + O_2$	$jc_4 = 2 * 10^{-4}\, e^{(-1.4\ \sec(Z))}$
5.	$NO_2 + h\nu$	$\rightarrow$	$NO + O^3P$	$jc_5 = 1.45 * 10^{-2}\, e^{(-0.4\ \sec(Z))}$
6.	$NO_3 + h\nu$	$\rightarrow$	$NO + O_2$	$jc_6 = 3.53 * 10^{-2}\, e^{(-8.10*10^{-2}\ \sec(Z))}$
7.	$NO_3 + h\nu$	$\rightarrow$	$NO_2 + O^3P$	$jc_7 = 8.94 * 10^{-2}\, e^{(-5.92*10^{-2}\ \sec(Z))}$
8.	$HNO_3 + h\nu$	$\rightarrow$	$NO_2 + OH$	$jc_8 = 3 * 10^{-6}\, e^{(-1.25\ \sec(Z))}$
9.	$HNO_2 + h\nu$	$\rightarrow$	$NO + OH$	$jc_9 = 2.8 * 10^{-3}\, e^{(-0.45\ \sec(Z))}$
10.	$H_2O_2 + h\nu$	$\rightarrow$	$2\,OH$	$jc_{10} = 2.2 * 10^{-5}\, e^{(-0.75\ \sec(Z))}$
11.	$HO_2NO_2 + h\nu$	$\rightarrow$	$HO_2 + NO_2$	$jc_{11} = 1.35 * 10^{-4}\, e^{(-0.6024\ \sec(Z))}$
12.	$N_2O_5 + h\nu$	$\rightarrow$	$NO_2 + NO_3$	$jc_{12} = 3.32 * 10^{-5}\, e^{(-0.5670\ \sec(Z))}$

The parameters occurring in reaction 14 take the values

$$k_0 = 7.2 * 10^{-15} e^{785/T}, \quad k_2 = 4.1 * 10^{-16} e^{1440/T}, \quad k_3 = 1.9 * 10^{-33} [M] e^{725/T}. \tag{B.1}$$

The reaction constants rk_{23} and rk_{24} are given by

$$\begin{aligned} rk_{23} &= (2.3 * 10^{-13} e^{590/T)} + [M] 1.7 * 10^{-33} e^{10^4/T}), \\ &* \quad (1 + [H_2O] 1.4 * 10^{-21} e^{2.2*10^3/T}) \end{aligned} \tag{B.2}$$

$$rk_{24} = 1.7 * 10^{-11} e^{416/T} + [M] 3.0 * 10^{-31} e^{5.0*10^2/T}. \tag{B.3}$$

The function $\sec(Z)$ of the solar zenith angle Z is defined by $\sec(Z) = 1/\cos Z$ with

$$\cos Z = \max\left(\sin\Delta\, \sin\phi + \cos\Delta\, \cos\phi\, \cos(\frac{12 - tod}{12}\,\pi), 0\right), \tag{B.4}$$

where tod is the time of the day in hours and Δ denotes the declination. At night the photochemical reaction constants are set to zero since then $\cos Z = 0$. In the current test,

$$\Delta = 0.3696032703233542, \quad \phi = 0.9116029216091582, \tag{B.5}$$

so that we have sunrise at 04.00 h. and sunset at 20.00 h. Finally, the function $F3$ associated to the three-body reactions reads

$$F3(k_0, nn, k_{00}, mm, fc) = \frac{kq_1}{1 + kq_1}\,(fc)^{\dfrac{1}{1 + \log_{10}^2(kq)}}, \tag{B.6}$$

where

$$\begin{aligned} kq_1 &= [M]\, k_0 \left(\frac{T}{300}\right)^{-nn}, \\ \\ kq &= kq_1 \text{ if } k_{00} = 0, \\ \\ kq &= \frac{kq_1}{k_{00}} \left(\frac{T}{300\,K}\right)^{mm} \text{ otherwise.} \end{aligned} \tag{B.7}$$

	Name	Species	Concentration
	-	M	$2.55 * 10^{19}$
	Water	H_2O	$2.55 * 10^{17}$
	Carbon monoxide	CO	$2.55 * 10^{12}$
	Molecular oxigen	O_2	$5.3295 * 10^{18}$
1	Nitric acid	HNO_3	$2.55 * 10^{9}$
2	–	HO_2NO_2	$1.0 * 10^{2}$
3	Nitrous acid	HNO_2	$1.0 * 10^{2}$
4	Hydroperoxide	H_2O_2	$1.0 * 10^{2}$
5	Ozone	O_3	$7.65 * 10^{11}$
6	Formaldehyde	$HCHO$	$1.0 * 10^{2}$
7	Methylhydroperoxide	CH_3OOH	$1.0 * 10^{2}$
8	Methylperoxy radical	CH_3O_2	$1.0 * 10^{2}$
9	Methane	CH_4	$4.335 * 10^{13}$
10	Nitrogen oxide	NO	$1.0 * 10^{2}$
11	Nitrogen dioxide	NO_2	$5.1 * 10^{9}$
12	Nitrate radical	NO_3	$1.0 * 10^{2}$
13	Hydroxyl radical	OH	$1.0 * 10^{2}$
14	Hydroperoxyde radical	HO_2	$1.0 * 10^{2}$
15	Dinitrogen pentoxide	N_2O_5	$1.0 * 10^{2}$
16	Atomic oxigen	O^1D	0.0
17	Atomic oxigen (g.s.)	O^3P	0.0
18	$NO + NO_2$	NO_x	$[NO] + [NO_2]$
19	$NO_2 + O_3$	O_x	$[NO_2] + [O_3]$

Table 3: Initial concentrations in [mol.cm^{-3}] for the CIRK chemistry.

C. Error relations for the Gauss-Seidel iteration

For solving the $m-$component chemical kinetics system

$$\dot{y} \;=\; f(t,y) \;\equiv\; P(t,y) - L(t,y)y, \tag{C.1}$$

TWOSTEP uses the variable step, second order BDF formula [16]

$$y^{n+1} = Y^n + \gamma\tau f(t_{n+1}, y^{n+1}), \qquad \tau = t_{n+1} - t_n, \tag{C.2}$$

where $\gamma = (c+1)/(c+2)$, $c = (t_n - t_{n-1})/(t_{n+1} - t_n)$ and

$$Y^n = \left((c+1)^2 y^n - y^{n-1}\right)/(c^2+2c). \tag{C.3}$$

We have chosen the second order formula in view of the modest accuracy requirement. The approach can of course be examined also for higher order BDF formulas. Our use of the Gauss-Seidel technique exploits the chemical kinetics form (C.1), by which (C.2) can be written as

$$y^{n+1} = F(y^{n+1}) \;\equiv\; \left(I + \gamma\tau L(t_{n+1}, y^{n+1})\right)^{-1} \left(Y^n + \gamma\tau P(t_{n+1}, y^{n+1})\right). \tag{C.4}$$

The Gauss-Seidel technique is now applied to the nonlinear system of equations $y = F(y)$ as follows. Write the fixed-point form $y = F(y)$ in the componentwise form

$$y_k = F_k\left(y_1, \ldots, y_{k-1}, y_k, \ldots, y_m\right), \qquad k = 1, ..., m. \tag{C.5}$$

Let $y^{(i)}$ denote the i-th iterate vector for y^{n+1}. From a given initial choice $y^{(0)}$ we compute, for $i = 0, 1, ...,$

$$y_k^{(i+1)} = F_k\left(y_1^{(i+1)}, \ldots, y_{k-1}^{(i+1)}, y_k^{(i)}, \ldots, y_m^{(i)}\right), \qquad k = 1, ..., m. \tag{C.6}$$

Formula (C.6) defines a genuinely explicit process since L is a diagonal matrix (only division by scalars). The classical nonlinear Gauss-Seidel method applied to (C.2) would render the computation scalarly implicit. The two are identical if for all k we have

$$\frac{\partial P_k}{\partial y_k} = 0, \quad \frac{\partial L_k}{\partial y_k} = 0.$$

For our problem class this always holds for the production term P, but generally not for L.

We note in passing that for components for which both P_k and L_k are constant in y, the solution is obtained in one iteration. Consequently, components for which P_k and L_k will slowly vary are handled efficiently. In this respect the current iterative approach bears a resemblance with the explicit QSSA approach (see e.g. [16, 14]). A difference between the two approaches is that we start from the integration formula (C.2), rather than from the exponential form used for QSSA, and we use Gauss-Seidel iteration instead of Jacobi or Picard iteration as in QSSA. Would we use Picard iteration, then iteration formula (C.6) would be replaced by

$$y_k^{(i+1)} = F_k\left(y_1^{(i)}, \ldots, y_{k-1}^{(i)}, y_k^{(i)}, \ldots, y_m^{(i)}\right), \qquad k = 1, ..., m. \tag{C.7}$$

Introduce the iteration errors

$$e_k^{(i)} = y_k^{n+1} - y_k^{(i)}. \tag{C.8}$$

Next subtract (C.6) from (C.5) and linearize at the fixed point $y = y^{n+1}$. This yields the linearized error equation

$$e_k^{(i+1)} = \frac{\partial F_k}{\partial y_1}\, e_1^{(i+1)} + \ldots + \frac{\partial F_k}{\partial y_{k-1}}\, e_{k-1}^{(i+1)} + \frac{\partial F_k}{\partial y_k}\, e_k^{(i)} + \ldots + \frac{\partial F_k}{\partial y_m}\, e_m^{(i)}. \tag{C.9}$$

The Jacobian matrix $F' \equiv \partial F/\partial y$, computed at the fixed point $y = y^{n+1}$ reads

$$F^{'} = \gamma\tau(I + \gamma\tau L)^{-1}(f^{'} + L), \tag{C.10}$$

where $f^{'} = P^{'} - \mathrm{diag}(y)L^{'} - L$. Hence the entries $\partial F_k/\partial y_j$ occurring in (C.9) are given by

$$\frac{\partial F_k}{\partial y_j} = \gamma\tau\,(1 + \gamma\tau L_k)^{-1}\left(\frac{\partial P_k}{\partial y_j} - y_k\frac{\partial L_k}{\partial y_j}\right). \tag{C.11}$$

In matrix form the error relation (C.9) reads

$$e^{(i+1)} = Ge^{(i)}, \tag{C.12}$$

with the amplification operator G given by

$$G = (I - F_L^{'})^{-1}(F_U^{'} + F_D^{'}), \tag{C.13}$$

where $F_L^{'}$ is the lower triangular matrix of the Jacobian $F^{'}$, $F_U^{'}$ the upper triangular part and $F_D^{'}$ the diagonal. Recall that for the Picard iteration (C.7) the same error relation holds with $G = F^{'}$.

Assuming that the linearization is meaningful, (C.12) determines the convergence of the Gauss-Seidel process in first approximation. Sufficient for convergence then is that, for some appropriate norm $\|\cdot\|$,

$$\|G\| < 1. \tag{C.14}$$

Necessary for convergence is that powers of G vanish, i.e.,

$$\|G^i\| \;\rightarrow\; 0 \quad \text{for} \quad i \;\rightarrow\; \infty. \tag{C.15}$$

For stiff ODE problems from atmospheric chemistry, condition (C.14) appears to be too restrictive, at least for standard norms. It often occurs that the process does converge while (C.14) is violated. This is due to the scaling of the unknown concentrations. Large Jacobian entries (contained in certain columns) are not necessarily harmful if they multiply with error components orders of magnitude smaller than other error components which are multiplied by small Jacobian entries. Hence, imposing (C.14) for testing convergence in actual application would generally lead to too restrictive conditions on the step size τ. Moreover, the computation of the Jacobian entries (C.11) is something we do not recommend for a cheap process like Gauss-Seidel iteration as it leads to a relatively large overhead. This of course also rules out the use of (C.15).

For practical purposes, testing convergence can be based on estimating the decay of differences of successive iterates in the standard manner used in [15] (see also the 'Stopping criterion' paragraph in Section IV.8 of [4]). We have experienced, though, that working with a small fixed number of a priori prescribed iterations often works as good as iterating to convergence. In the latter situation generally more accuracy is obtained, but for higher costs per time step. Alternatively, one may choose to work with a smaller time step and only a small a priori given number of iterations per time step. The latter approach is followed in the experiments in this report.

Although (C.14) should not be used for testing convergence, the formula for the Jacobian entries (C.11) provides some insight into the question why Gauss-Seidel iteration is successful for atmospheric chemistry problems, specifically for coping with the most stiff components. Suppose that for a certain species y_k, all its entries (C.11) are much smaller than one. It then trivially follows from the lower-upper block structure in (C.13) that for y_k the convergence is fast and one or at most a few iterations are sufficient. This is the case for the very short living species (radicals like O^3P, O^1D and OH), since for these the loss factor L_k is sufficiently large to always achieve this for practical values of the step size τ. This means that the stiffest components are dealt with efficiently and will not limit the step size. For other species such a simple estimation of the Jacobian entries cannot be made without making additional assumptions. This, unfortunately, seems out of reach for complicated problems from practice like the test examples used in this report.

We conclude this appendix with explaining the observation made in Section 3 that the lumping trick described there can be interpreted as a form of preconditioning, something which was suggested by our

colleague Willem Hundsdorfer. Let, for simplicity, (C.1) represent a system augmented with only one lumped species, say y_m. Also suppose, again for simplicity, that the correction is made after any complete Gauss-Seidel iteration and let y_k be the corrected species. Instead of (C.6) we then get the two-stage iterative process ($y^{(i)} \rightarrow z^{(i+1)} \rightarrow y^{(i+1)}$) given by

$$z_k^{(i+1)} = F_k \left(z_1^{(i+1)}, \ldots, z_{k-1}^{(i+1)}, y_k^{(i)}, \ldots, y_m^{(i)} \right), \qquad k = 1, \ldots, m, \tag{C.16}$$

$$y^{(i+1)} = P z^{(i+1)}, \tag{C.17}$$

where P is a matrix of which the $k-$th row defines the lumping relation and all other rows are as in the unit matrix. The $k-$th row has a zero at the main diagonal position and nonzero entries at the column positions associated to the species taking part in the lumping relation. For this two-stage process we easily find, instead of (C.12), the error relation

$$e^{(i+1)} = PGe^{(i)}, \tag{C.18}$$

where G is defined in the same way as above. The premultiplication of G by P represents a form of preconditioning and if the lumping is successful, then powers of PG will converge to the zero matrix more rapidly than powers of G. Note that since the $k-$th row of P has a zero at the main diagonal position, the complete $k-$th row of G is eliminated and replaced by a linear combination of other rows. Although the lumping procedure applied in Section 3 is more complicated than the (most simple) one described here, its success can be explained along these lines.