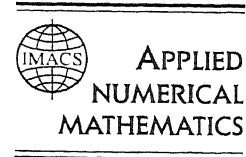




ELSEVIER

Applied Numerical Mathematics 18 (1995) 413–430



Explicit methods for stiff ODEs from atmospheric chemistry [☆]

J.G. Verwer ^{a,*}, D. Simpson ^b

^a *CWI, P.O. Box 94079, 1090 GB Amsterdam, Netherlands*

^b *EMEP MSC-W, Norwegian Meteorological Institute, P.O. Box 43 – Blindern, N-0313 Oslo 3, Norway*

Received 19 September 1994; revised 4 November 1994; accepted 16 November 1994

Abstract

The subject of research is the numerical integration of atmospheric chemical kinetics systems. The application lies in the study of air pollution, modelled by atmospheric chemistry-transport problems. This application puts high demands on the efficiency of the stiff solver. Three explicit methods are discussed and compared for a selected chemical kinetics system which is representative for the state of the art. The first and the second method are of the explicit QSSA type and the third is based on the two-step backward differentiation formula, combined with Gauss–Seidel iteration to approximately solve the implicitly defined solution. This also renders the method explicit. In the comparison, the two-step method comes out best.

Keywords: Atmospheric chemistry; Transport-chemistry problems; Numerical stiff ODEs

1. Introduction

The subject of this paper is the numerical integration of ODE systems from atmospheric chemical kinetics. Such systems can be cast in the nonlinear form

$$\frac{d}{dt}y = f(t, y) := P(t, y) - L(t, y)y, \quad y(t) = (y_1(t), \dots, y_m(t))^T, \quad (1.1)$$

where $P(t, y)$ is a vector and $L(t, y)$ a diagonal matrix. The components $P_k(t, y)$ and $L_k(t, y)y_k$ are nonnegative and represent, respectively, production and loss terms for com-

[☆] *Note:* This paper is one of a series on the development of algorithms for long-range transport air pollution models (CWI projects EUSMOG and CIRK). J.G. Verwer wishes to acknowledge the RIVM—the Dutch National Institute of Public Health and Environmental Protection—for financial support.

* Corresponding author.

pound y_k . The reciprocal of L_k is the physical time constant or characteristic reaction time for y_k . Generally the range of time constants is large, which implies that in most applications the ODE system is stiff.

Our application lies in the study of air pollution, modelled by atmospheric chemistry-transport problems. In practice, the transport nature of air pollution involves the numerical integration of the associated atmospheric chemical kinetics equation (1.1) at thousands of gridpoints or trajectories. Because (1.1) is stiff and can have a large number of components, ranging from 20 to 100 say, this numerical integration can be extremely time-consuming and therefore very fast methods are required. The accuracy demand is modest, however. For atmospheric chemistry-transport problems a relative accuracy higher than 1% is generally thought to be redundant due to the presence of model uncertainties. This low level of accuracy can of course be exploited in the selection of a suitable method.

Nowadays popular in many atmospheric chemistry-transport studies (see e.g. [3,6,10,11,16]) are the so-called QSSA (quasi-steady-state-approximation) methods, in the field introduced in [7,9]. QSSA methods are based on the simple rule

$$y(t + \tau) \approx e^{-\tau L} y(t) + (I - e^{-\tau L}) L^{-1} P, \quad (1.2)$$

which is exact if P and L are constant. QSSA integration rules thus are designed under the assumption that in reality P and L are slowly varying functions. QSSA integration rules are always similar to (1.2) and hence are explicit, since L is diagonal. This means that they possess a workload per time step comparable to that of a low-stage, explicit Runge–Kutta method. Hence per time step a QSSA solver is cheap in comparison with general-purpose stiff ODE solvers, which mostly are implicit and therefore carry a considerable overhead. QSSA methods also enjoy a remarkable good stability behaviour and, as far as stability is concerned, often allow stepsizes as large as an implicit method would use.

A rather limiting disadvantage, however, is that in this range of stepsizes straightforward solution of the chemical equations usually produces errors significantly larger than the above mentioned 1%. This accuracy limitation can be removed to a great extent by tuning the method for the chemical kinetics scheme in use. This tuning is mainly based on lumping sets of related chemical reactions into new ones which are less troublesome to solve numerically, and are designed to conserve the mass of the most important chemical species. When the tuning is carried out carefully, then a QSSA solver can be made more efficient than implicit general-purpose solvers, like those based on the celebrated backward differentiation (BDF/Gear) formulas. Without this tuning the reverse can hold true, as is illustrated, for example, in [16]. A great disadvantage is that tuning and lumping often appears to be a laborious task when the chemistry has to be adapted.

Continuing our previous work on the two-step BDF/Gauss–Seidel method [15], in the present paper we will compare this method with two QSSA schemes. For that purpose we have chosen the EMEP MSC-W ozone model chemistry from [11–13], which is representative for the state of the art in the field of regional air pollution modelling. The first QSSA scheme is the first-order consistent EMEP scheme developed jointly with the chemical model, an updated version of [7,9]. The second QSSA scheme is the second-order, two-stage scheme proposed in [16]. Our comparison serves to show that the equally simple two-step method can be made more efficient than these two QSSA schemes, in particular near the 1% error range.

2. The two-step method

To begin with we present the two-step method. This presentation is entirely based on [15]. There is only one notable difference to mention which has to do with the use of the Gauss-Seidel iteration technique. In [15] we iterate till convergence, subject to an imposed tolerance. Here we prescribe the number of iterations beforehand, which simplifies the implementation of stepsize control. Various experiments have revealed that this works equally well. In fact, it is often possible to integrate efficiently using only a few iterations. Note that then the order of the components processed by the Gauss-Seidel technique can have some influence on the stability and accuracy.

The two-step method is based on the variable-step, second-order BDF formula

$$y^{n+1} = Y^n + \gamma\tau f(t_{n+1}, y^{n+1}), \quad \tau = t_{n+1} - t_n, \quad (2.1)$$

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

$$Y^n = ((c + 1)^2 y^n - y^{n-1}) / (c^2 + 2c). \quad (2.2)$$

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 (1.1), by which (2.1) can be written as

$$y^{n+1} = F(y^{n+1}) := (I + \gamma\tau L(t_{n+1}, y^{n+1}))^{-1} (Y^n + \gamma\tau P(t_{n+1}, y^{n+1})). \quad (2.3)$$

The Gauss-Seidel technique is now applied to the nonlinear system of equations $y = F(y)$ in the standard way. Hence, given the iterate $y^{(i)}$, we have the componentwise formula

$$y_k^{(i+1)} = F_k(y_1^{(i+1)}, \dots, y_{k-1}^{(i+1)}, y_k^{(i)}, \dots, y_m^{(i)}), \quad k = 1, \dots, m, \quad (2.4)$$

which defines a genuinely explicit process, in contrast with the classical nonlinear Gauss-Seidel method applied to (2.1) since the latter is 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 .

Observe that for components for which both P_k and L_k are constant in y , the solution is obtained in one iteration. Consequently, when individual components rapidly approach their steady-state value P_k/L_k , they are handled efficiently and accurately by (2.4). In this respect the current iterative approach bears a resemblance with the explicit QSSA approach. The difference is that we start from the numerical integration formula (2.1), which does not a priori assume that P_k and L_k slowly vary, in contrast with the QSSA formulas. This is a better starting point for components for which P_k and L_k are really varying, supposing that the Gauss-Seidel technique then still yields a sufficiently good approximation within one or a few iterations. Our experience is that this is indeed the case. In addition, the technique also appears to be very stable. In this respect we have not observed much difference with the more

commonly used iterative modified Newton technique. Needless to say that the latter technique introduces much overhead, especially when frequent stepsize changes have to be made.

We now discuss the actual solver based on (2.3)–(2.4). As initial iterate we use the extrapolation

$$y^{(0)} = y^n + \frac{1}{c}(y^n - y^{n-1}). \quad (2.5)$$

This means that even with one iteration, the actually applied integration scheme retains the second order of consistency. The number of Gauss–Seidel iterations is prescribed beforehand and the final iterate is assigned to be the new approximation y^{n+1} at the new time point t_{n+1} . Both variable and constant stepsizes can be used. For the variable-stepsize selection we use the local error indicator

$$E^{n+1} = \frac{2}{c(c+1)}(cy^{n+1} - (1+c)y^n + y^{n-1}), \quad (2.6)$$

which yields $\tau^2 y''(t^n) + O(\tau^3)$ upon substitution of the exact solution. Note that the local error of (2.1) is $O(\tau^3)$. Hence our local error indicator may work out a bit conservative (see also [15]). Now consider the weighted error norm

$$\|E^{n+1}\|_w = \max(|E_k^{n+1}|/W_k^n), \quad W_k^n = atol_k + rtol_k |y_k^n|, \quad (2.7)$$

where $atol_k$ and $rtol_k$ are the componentwise defined absolute and relative error tolerance. If $\|E^{n+1}\|_w \leq 1.0$, then the integration step is accepted and otherwise rejected. The new stepsize τ_{new} is estimated by the common formula

$$\tau_{new} = \max\left(0.5, \min\left(2.0, 0.8/\sqrt{\|E^{n+1}\|_w}\right)\right)\tau_{old}. \quad (2.8)$$

The stepsize is further constrained by a prescribed minimal and maximal value. If two successive rejections occur we simply restart the process. The missing starting value after a restart, or at the true start at the initial point of time, is computed with the implicit Euler method which is treated with the Gauss–Seidel method in the same way as the two-step method. When a variable stepsize is used, then a safe guess for the initial stepsize is computed by replacing E^{n+1} in (2.7) by $\tau f(t_0, y^0)$. Hence we then define τ such that the weighted error norm is equal to one, i.e.,

$$\tau = \min(W_k^0/|f_k(t_0, y^0)|). \quad (2.9)$$

In other words, the initial step is chosen so that the first Taylor series term $\tau f(t_0, y^0)$ satisfies the absolute/relative tolerance requirement. The two-step scheme is then applied with the same stepsize and after that the variable-stepsize mechanism is activated. Normally (2.9) will lead to a rather small initial guess, which will be accepted and subsequently rapidly increased according to (2.8).

We conclude this section with some results obtained by the two-step solver for the model from [3,15], so as to illustrate that fixing the number of Gauss–Seidel iterations is competitive with the “iterating to convergence” approach followed in [15]. Note that in [15] its predecessor has been favorably compared with the stiff ODE solver DASSL [1]. The test model contains 20

Table 1
Values SD, S

TOL
10^{-1}
10^{-2}
10^{-3}

component
the Lipschi
the integra
reaction co
sudden larg
the solution
Variable
was prescri

$$atol_k =$$

For these t
These smal
small, whic
remain sm
specified ti

$$SD =$$

and STEPS
Note that
value of T
Only the a
formula (2.

3. The EM

In this s
Because th
especially
details will
compare th
atmospher

Table 1
Values *SD*, *STEPS* for the test model from [15]

<i>TOL</i>	<i>it</i> = 1	<i>it</i> = 2	<i>it</i> = 3	<i>it</i> = 4	<i>it</i> = 5
10^{-1}	1.34, 059	1.82, 057	1.80, 056	2.01, 056	2.24, 056
10^{-2}	1.96, 132	2.91, 132	3.11, 132	2.91, 132	3.25, 132
10^{-3}	3.32, 362	3.83, 362	4.01, 362	4.19, 362	4.10, 362

components and 25 reactions. The initial data is such that an initial transient is present, while the Lipschitz constant is about $1.5 \cdot 10^7$ (see [16]). Hence the ODE system is very stiff, provided the integration interval is sufficiently large which is true here. It should be noted that the reaction constants in the model example are constant, so that outside the initial transient no sudden large changes in the concentrations occur. We will give results at $t = 60$, at which time the solution gets close to its steady state.

Variable stepizes were used with one to five iterations. No minimal and maximal stepsize was prescribed. The following three values for $atol_k$ and $rtol_k$ were used for all species,

$$atol_k = 10^{-6}TOL, \quad rtol_k = TOL, \quad TOL = 10^{-1}, 10^{-2}, 10^{-3}. \quad (2.10)$$

For these tolerances, the initial stepsize τ_1 determined by (2.9) equals $4.7atol_k$, approximately. These small initial stepizes reveal the initial transient and arise because we take $atol_k$ rather small, which is desirable since some of the concentrations are zero at the initial time and remain small for evolving time. Table 1, the counterpart of Table 1 from [15], yields at the specified time $t = 60$ the number of significant digits for the maximum relative error, defined by

$$SD = -\log_{10} \left(\max_k \frac{|y_k^n - y_k(t)|}{|y_k(t)|} \right), \quad (2.11)$$

and *STEPS* = the number of integration steps. The results compare well with those from [15]. Note that even with a single iteration the solver works satisfactorily. Also note that for a given value of *TOL* and increasing number of iterations, the number of time steps remains equal. Only the accuracy increases until the accuracy is reached which the original second-order BDF formula (2.1) would yield on the stepsize sequence used.

3. The EMEP ozone model and the QSSA schemes

In this section we will briefly discuss the EMEP ozone model and the two QSSA schemes. Because the model is rather voluminous and the schemes are intertwined with the chemistry, especially the EMEP scheme, they cannot be described in full detail in the present paper. More details will be given in the companion paper [14], the purpose of which is to further test and compare the QSSA schemes and the two-step scheme under a much larger variety of realistic atmospheric conditions.

3.1. The EMEP ozone model

The EMEP MSC-W ozone model is a single-layer trajectory model which calculates concentrations of photochemical oxidants every 6 hours over the whole of Europe. The most recent version of the model is described in [11–13]. The model is a development of the previous Norwegian trajectory models from [5,8]. Columns of air in the atmospheric boundary layer (the lowest 1 km, approximately, of the atmosphere) are followed along specified 96-hour trajectories, picking up emissions of SO_2 , nitrogen oxides NO_x , CO and hydrocarbons (RH) from the underlying grid. The model domain covers the whole of Europe with a grid resolution of ca. $150 \times 150 \text{ km}^2$. The mass-conservation equations are integrated along each trajectory, taking into account emission inputs, chemical reactions, dry and wet removal processes, and the influence of relevant meteorological parameters. Chemistry and transport are solved separately, so that the chemical reaction terms are cast in the form of the ODE (1.1).

The chemical scheme consists of about 140 reactions between ca. 70 species. The mechanism has been described fully in [13] although with 8 additional reactions as described in [12]. Rate coefficients are often variable, depending on temperature and, for some, humidity. Photolysis rates obviously depend on solar elevation and cloudiness, and undergo a discontinuity at sunrise and sunset. The time step for changes in spatial position, emissions, meteorological parameters, etc. is 2 hours, whereas that for chemistry has been 15 minutes with the QSSA scheme used so far. One month of calculation currently requires about one CPU hour on a CRAY Y-MP 4D/464 computer, with solution of the chemistry equations taking ca. 85% of CPU time.

2. The QSSA scheme

The QSSA scheme was originally developed in [7,9]. It is derived from the first-order consistent formula

$$y^{n+1} = e^{-\tau L^{n+1}} y^n + (I - e^{-\tau L^{n+1}})(L^{n+1})^{-1} P^{n+1}, \quad (3.1)$$

which is the implicit numerical counterpart of (1.2). Starting from the initial iterate y^n functional iteration is applied for approximating y^{n+1} . This renders the scheme explicit. The number of iterations may vary per component. In the EMEP scheme two sets of components are selected. To the first set only one iteration is applied. This yields

$$y_k^{n+1} = e^{-\tau L_k^n} y_k^n + (I - e^{-\tau L_k^n})(L_k^n)^{-1} P_k^n, \quad (3.2)$$

where $L_k^n = L_k(t_n, y_k^n)$ and P_k^n is defined similarly. If for a component in this first set $\tau L_k^n < 0.01$, then for this component $e^{-\tau L_k^n}$ is replaced by $1 - \tau L_k^n$, resulting in the explicit Euler formula

$$y_k^{n+1} = y_k^n + \tau f_k^n. \quad (3.3)$$

If for a component in this set $\tau L_k^n > 10.0$, then $e^{-\tau L_k^n}$ is replaced by zero, resulting in the steady-state solution expression

$$y_k^{n+1} = P_k^n / L_k^n. \quad (3.4)$$

To the second set of components up to five such iterations are applied. The selection of these two different sets is based on chemical considerations and on stability. Loosely speaking, components for which a rather strong nonlinear coupling exist belong to the second set. Note that the replacements of the exponential function are not essential, since the exponential function automatically takes care of these replacements with sufficiently large accuracy.

Of more importance is that the scheme is rather heavily tuned for the specific chemistry used. The tuning consists of checks on the magnitude of certain concentrations which are known to be related somehow. If the computed values for these concentrations do not obey these relations, then they are fixed to do so. The scheme also greatly benefits from its lumping procedure, which successfully eliminates strong couplings between certain components. Lumping can be viewed as transforming small subsets of the ODE system to sets of differential-algebraic equations. The lumping is based upon that developed in [7], although modified to some extent. The grouped species are

$$\text{O}_3\text{NO} = \text{O}_3 - \text{NO}, \quad (3.5)$$

as recommended in [7],

$$\text{NO}_z = \text{NO}_3 + \text{N}_2\text{O}_5, \quad (3.6)$$

and

$$\text{NO}_y = \text{NO} + \text{NO}_2 + \text{PAN} + \text{MPAN} + \text{NO}_3 + 2\text{N}_2\text{O}_5, \quad (3.7)$$

as discussed in [14]. In general the scheme has been found to give very good results for normal application of the EMEP model, but can become unstable under some experimental condition [14]. The latter problem and the need for tuning and lumping prompted the present search for a more reliable numerical method.

3.3. An alternative QSSA scheme

The tuning and lumping obviously corrupts the numerical comparison. For example, lumping introduces new components not present in the original system. To illustrate the effect of the tuning and lumping, and to provide a second numerical comparison with the two-step scheme, we therefore also include an alternative QSSA scheme which is based on the second-order two-stage scheme proposed in [16]. It reads

$$(I + Z + \frac{1}{2}Z^2)\zeta^{n+1} = y^n + \tau(I + \frac{1}{2}Z)P^n, \quad Z = \tau L^n, \quad (3.8)$$

$$(I + Z + \frac{1}{2}Z^2)y^{n+1} = y^n + \tau(I + \frac{1}{2}Z)P^{n+1/2}, \quad Z = \tau L^{n+1/2}, \quad (3.9)$$

where

$$L^n = L(t_n, y^n), \quad L^{n+1/2} = \frac{1}{2}(L^n + L(t_{n+1}, \zeta^{n+1})) \quad (3.10)$$

and P^n and $P^{n+1/2}$ are defined likewise. Note that for components for which $L_k(y) = 0$, stage one yields the explicit Euler formula and stage two the explicit trapezoidal rule. Like the explicit trapezoidal rule the method has order of consistency two. Because the matrix Z is

diagonal, we call this two-stage scheme also explicit. Here, for all components the exponential has been replaced by the subdiagonal Padé approximation

$$R(z) = \left(1 - z + \frac{1}{2}z^2\right)^{-1}. \quad (3.11)$$

For more details we refer to [16]. The scheme will be applied in combination with two different lumping procedures, viz. the full lumping as used in the EMEP scheme and the more simple NO_y lumping (3.7). The latter will also be used by the two-step scheme.

4. The numerical comparison

4.1. The example problem

We consider a chemistry computation for two emission scenarios. The first “rural” scenario corresponds to typical mid-European values ($E_{\text{NO}_x} = E_{\text{RH}} = 2.5 \cdot 10^{11} \text{ mol} \cdot \text{cm}^{-2}\text{s}^{-1}$), although with quite high emissions of the reactive biogenic hydrocarbon species isoprene ($E_{\text{iso}} = 1.25 \cdot 10^{11} \text{ mol} \cdot \text{cm}^{-2}\text{s}^{-1}$) in order to stress the numerical methods. The second scenario corresponds to a more “urban” situation, but also with high isoprene emissions, and is expressly designed to stress the methods even further ($E_{\text{NO}_x} = E_{\text{RH}} = 2.5 \cdot 10^{12}$, $E_{\text{iso}} = 3.75 \cdot 10^{11} \text{ mol} \cdot \text{cm}^{-2}\text{s}^{-1}$).

These box-model computations can be interpreted as single trajectory computations in the EMEP trajectory model. In our case the integration covers 112 hours, starting at sunrise at 04.00 hours ($t = 14400$) at the first day and ending at sunset at 20.00 hours at day five ($t = 417600$). The initial values are chosen somewhat arbitrary and hence introduce, for certain components, an initial transient. This initial transient has disappeared after a couple of hours. At sunrise and sunset photolysis rates undergo a discontinuity which also introduces sharp gradients for certain components.

For the numerical integration the 112-hour period is divided into 56 integration intervals of 2-hour length, in accordance with the time step for changes in coefficients and parameters in the EMEP model. This implies 56 restarts for the two-step solver, including those at sunset and sunrise. Note that a restart involves the use of the first-order backward Euler formula and that we also restart if we work with a constant stepsize.

4.2. Stiffness

The degree of stiffness can be estimated by computing characteristic reaction times L_k^{-1} . For the rural test case and a certain time interval during the day we have, approximately, $\text{O}(10^7)$ seconds as the maximum and $\text{O}(10^{-9})$ seconds as the minimum time constant. This reveals excessive stiffness. The slowest reacting species are H_2 and CH_4 and the fastest one is the radical $\text{O}(^1\text{D})$. However, if we would assume a steady state for this radical, and likewise for the other two radicals $\text{O}(^3\text{P})$ and OH , then the smallest reaction time of interest for the numerical integration is sharply increased to approximately $\text{O}(1)$ seconds (for the species NO_3). This steady-state assumption is realistic in practical applications and is imposed in the QSSA schemes. It is not imposed in the two-step scheme since this scheme does not need it, as

explained in Section 2. Thus, with the steady-state assumption for the radical species imposed, the system becomes moderately stiff in view of the fact that the integration interval length in our application is 7200 seconds. On the other hand, for this 2-hour interval the required number of steps for a common explicit method remains a great deal too large for the actual practice. Below we will see that the explicit QSSA schemes and the two-step method combined with Gauss–Seidel iteration require only a fraction of what a common explicit method would need.

4.3. Schemes, stepsizes, iterations and CPU times

The schemes are denoted by:

- QSSA1: The first-order consistent EMEP scheme with the full lumping procedure. A constant stepsize is used, as in the EMEP model.
- QSSA2: The second-order consistent two-stage scheme with the full EMEP lumping or the more simple NO_y lumping. Again a constant stepsize is used.
- TWOSTEP: The second-order consistent two-step solver with the NO_y lumping. Now both constant and variable stepsizes are used. Throughout, for the rural test case only one Gauss–Seidel iteration is applied and for the urban test case two.

Applying more Gauss–Seidel iterations usually improves the accuracy of TWOSTEP, as to be expected. However, when also CPU time is taken into account, not always a clear advantage was observed above using one or two iterations with a correspondingly smaller stepsize. This especially holds true when integrating with a variable stepsize. As a rule, we therefore prefer to work with only a very few iterations and a smaller stepsize, because this is advantageous at restarts after sunset and sunrise since then certain concentrations possess very large gradients. Some trial and error runs revealed that for the rural case a single iteration suffices and also works out more efficiently than two iterations. For the more difficult urban case it is just the other way around. We haven't put much effort in trying also a higher number of iterations.

When using large constant stepsizes the NO_y lumping works out beneficial for TWOSTEP for coping with large solution gradients, as it does for the QSSA schemes. For sufficiently small stepsizes the effect will become negligible. Hence when integrating with a variable stepsize there is less need for lumping, since the method automatically will reduce the stepsize when large solution gradients occur. We note that the NO_y lumping is straightforward, as this group of species defines an ODE with a positive production and loss term fitting in the format (1.1). The lumping thus amounts to integrating this additional scalar ODE, combined with a correction step to satisfy (3.7). The correction step is made after every iteration and redefines NO_2 in terms of the other species in the group. It is the combination of an accurate solution for NO_y and this redefinition of NO_2 that ensures nitrogen conservation.

CPU time is determined by the production, loss rate and rate coefficient computations. For the three different methods we found that per time step TWOSTEP and QSSA2 use nearly the same CPU time (on an SGI workstation and a CRAY), whereas QSSA1 is more expensive. The CPU times for TWOSTEP and QSSA2 are nearly equal since the latter computes rate coefficients only in the first stage and its implementation is simple without any overhead. QSSA1 is significantly more expensive per step due to the additional iterations for the second component set. In our tests, the actual CPU time ratio between QSSA1 and the other two

solvers varied between 2 and 4. This ratio of course depends on the programming style, computer, compiler options, I/O, etc.

4.4. Time evolution of O_3 and NO

The constant stepsize currently used in the EMEP trajectory model is 900 seconds. This stepsize is large compared to stepsizes reported in earlier work (see [6,7]), but has usually been found to be acceptable for rural ozone modelling because the low concentrations found in the rural atmosphere lead to quite long time scales for most chemical processes [14]. It is therefore of interest to first compare the three methods, for the rural emission case, using the constant stepsizes of 300 and 900 seconds. Note that with a 300 second stepsize TWOSTEP and QSSA2 require approximately the same CPU time as QSSA1 with 900 seconds. QSSA2 is applied with the full lumping procedure.

We consider the evolution in time of O_3 and NO, depicted in Figs. 1 and 2, respectively. The figures contain a very accurate reference solution, plus the 300- and 900-second approximate solutions. Note that O_3 varies gradually, whereas NO clearly reveals the effect of photolysis at sunset and sunrise. Other concentrations which also rapidly change at sunset and sunrise are, e.g., NO_2 and NO_3 . Our conclusions drawn below for NO are valid, approximately, for these species too.

We see that with the 900-second stepsize the two QSSA methods compute O_3 with a comparable accuracy, whereas TWOSTEP is slightly behind. For the 300-second stepsize no difference can be seen. No doubt the QSSA methods exploit here the benefits of the full lumping procedure. For NO the situation is a bit different. Comparing TWOSTEP and QSSA1, we see that the latter computes NO much more accurately for the 900-second stepsize, no doubt due to the full lumping. If we also take into account CPU time, then we see that the solutions computed by TWOSTEP (300 seconds) and QSSA1 (900 seconds) are equally accurate (up to the plotting accuracy level). It is obvious that QSSA2 falls behind for the NO computation, despite the full lumping employed.

4.5. Accuracy versus CPU for the two test cases

For a second and more thorough comparison we now introduce 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,

$$RRMS_k = \sqrt{\frac{\sum_{n=1}^N (y_k^n - y_k(t_n))^2}{\sum_{n=1}^N (y_k(t_n))^2}}, \quad (4.1)$$

where $N = 56$ and $t_n = 14400 + 7200n$ seconds. We have calculated the number of significant digits for the maximum of $RRMS_k$ for all species, defined by

$$SDM = -\log_{10}\left(\max_k RRMS_k\right), \quad (4.2)$$

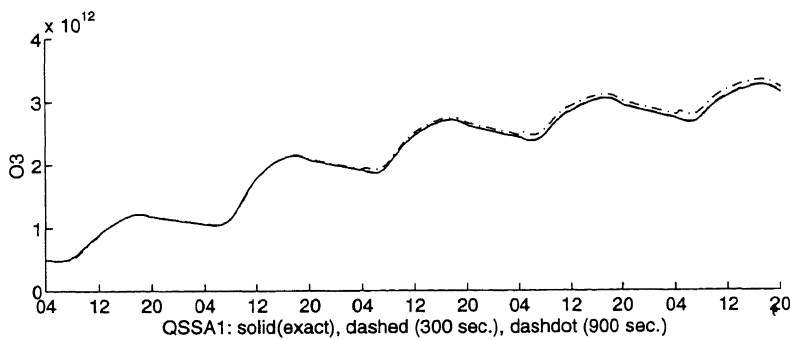
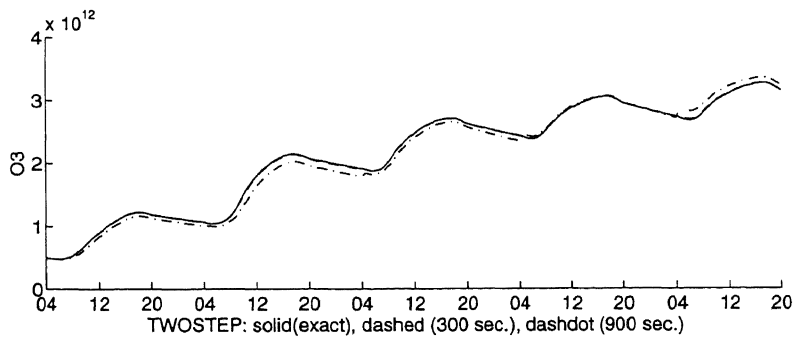
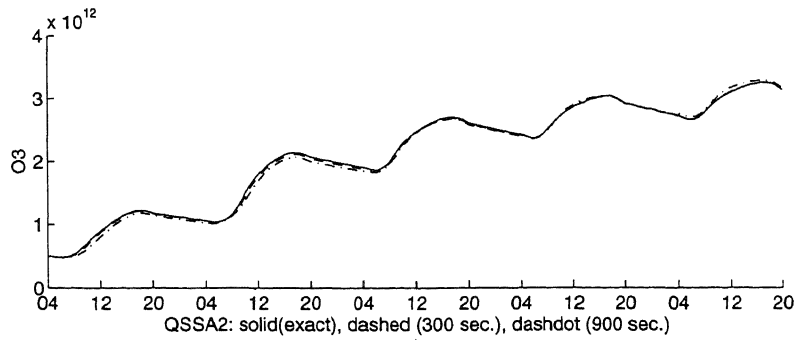


Fig. 1. Time evolution for O_3 .

and the number of significant digits for the average of $RRMS_k$, defined by

$$SDA = -\log_{10} \left(\frac{1}{m} \sum_{k=1}^m RRMS_k \right). \quad (4.3)$$

It is of interest to inspect both (the minimal number of significant digits) SDM and SDA , because componentwise the accuracy varies considerably. Very often NO_3 and N_2O_5 are the two most inaccurate components.

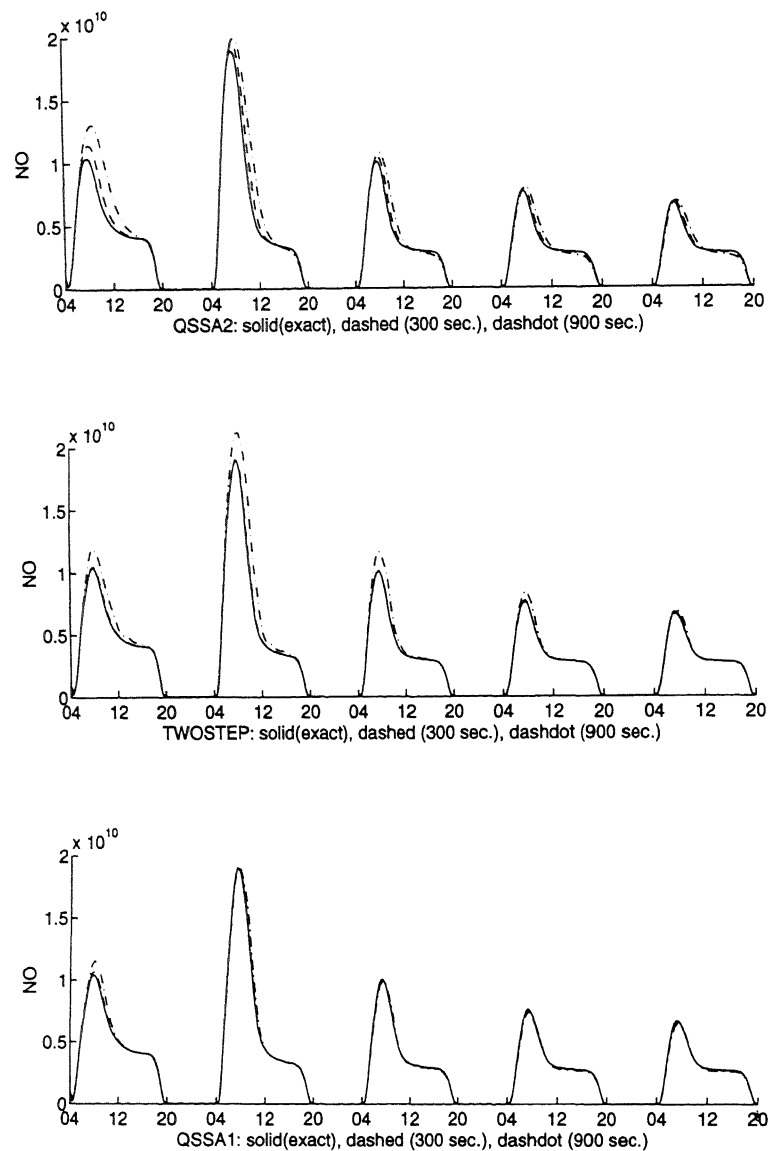


Fig. 2. Time evolution for NO.

For the rural case, Figs. 3 and 4 show values of SDM and SDA , respectively, plotted versus SGI CPU times. The experiments presented in the figures are of two sorts, viz. five fixed-step-size runs with all methods using $\tau = 10, 30, 100, 300, 900$ and twelve variable-step-size runs with TWOSTEP. Note that QSSA2 is applied with the full EMEP lumping and the NO_y lumping. The twelve variable-step-size runs are defined by the twelve combinations made out of

$$\tau_{\min} = 1, 30, 100, \quad \tau_{\max} = 900, \quad (4.4)$$

$$atol_k = 1.0, \quad rtol_k = 10^{-l}, \quad l = 2, 3, 4, 5. \quad (4.5)$$

Rural Case : SGI CPU Times

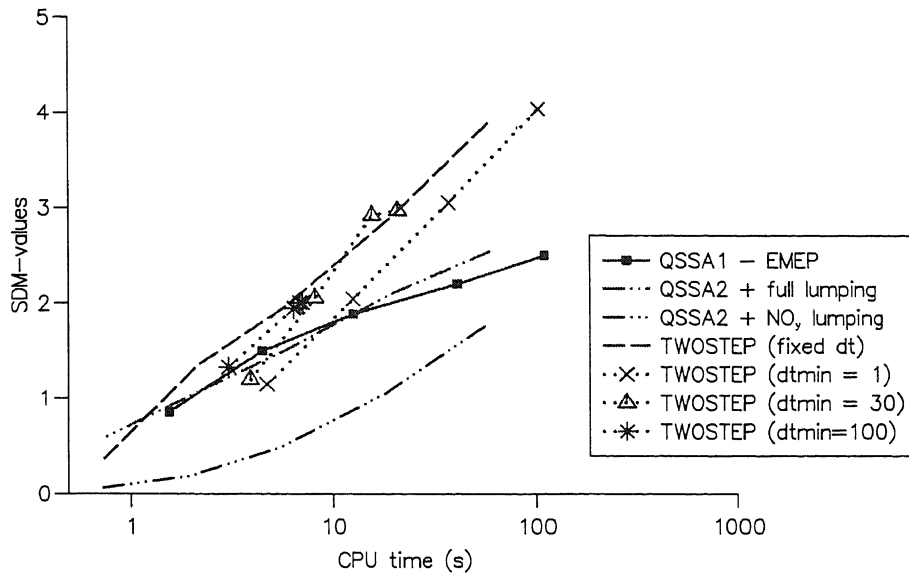


Fig. 3. *SDM*-values (4.2) versus SGI CPU time for the rural case.

Rural Case : SGI CPU Times

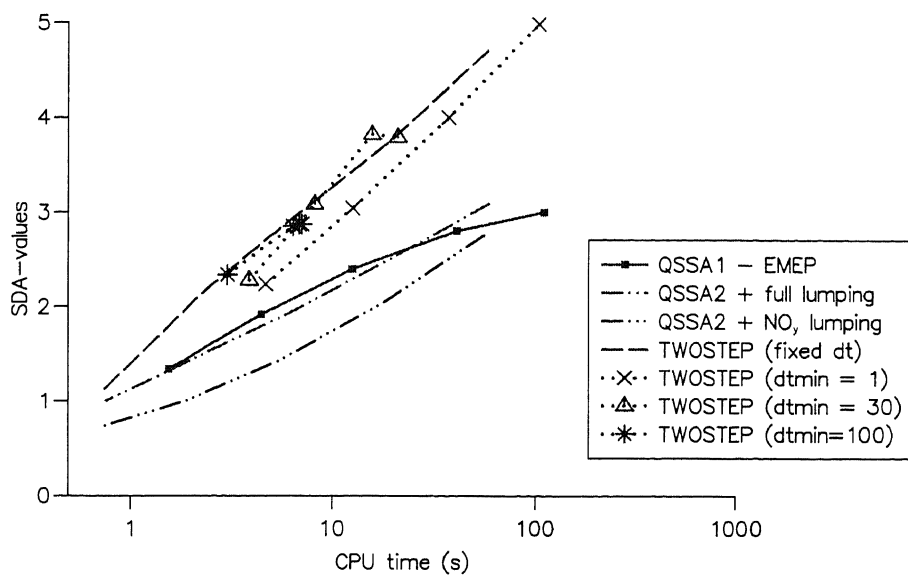


Fig. 4. *SDA*-values (4.3) versus SGI CPU time for the rural case.

Urban Case : SGI CPU Times

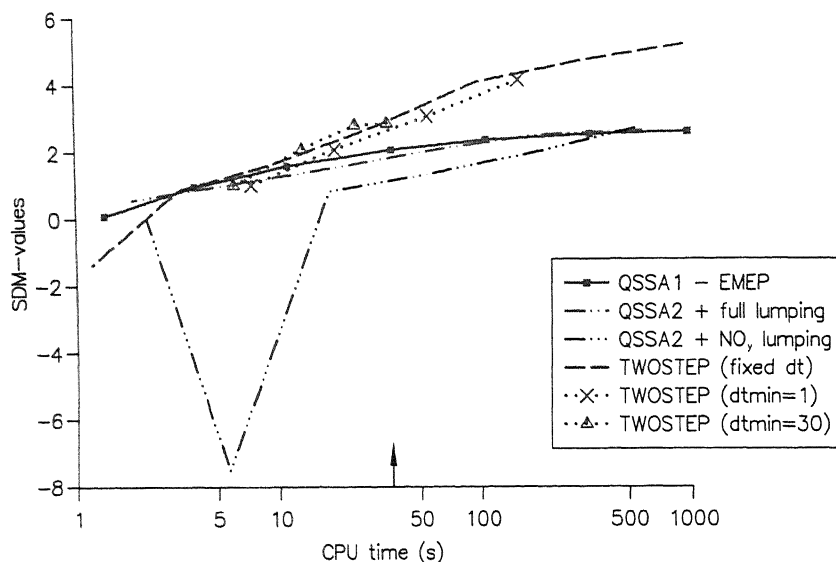


Fig. 5. *SDM*-values (4.2) versus SGI CPU time for the urban case.

As expected, in Fig. 3 we see that for the larger stepsizes the accuracy, displayed by *SDM*, appears to be low for all three fixed-stepsize methods. The crucial importance of full lumping in the QSSA methods is well illustrated for the two QSSA2 versions. Comparing the three methods using fixed stepsize, we see that TWOSTEP shows a much better convergence behaviour than QSSA1 and QSSA2. In fact, its convergence behaviour is in very good accordance with its second-order consistency. The second-order consistency of QSSA2 does not show up clearly, which is in accordance with the occurrence of the local order reduction phenomenon discussed in Section 2 of [16]. For the larger stepsizes the fully lumped QSSA2 offers no improvement over the first-order QSSA1. Noteworthy is that the use of variable stepsizes in TWOSTEP does not offer any performance benefit over the fixed-step version for this rural case. The kink in the curves for $\tau_{\min} = 30, 100$ are due to this prescribed minimal stepsize. An obvious conclusion is that in the current test TWOSTEP performs best in either fixed- or variable-step modes, owing to its good convergence behaviour in accordance with order two. The same conclusions can be drawn from Fig. 4 which displays the average errors through *SDA*.

Equivalent accuracy/efficiency plots are shown in Figs. 5 and 6 for the more difficult urban case. Here seven fixed-stepsize runs are displayed for each method, viz. using $\tau = 1, 3, 10, 30, 100, 300, 900$ and eight variable-stepsize runs with TWOSTEP, which are defined by the combinations made out of

$$\tau_{\min} = 1, 30, \quad \tau_{\max} = 900, \quad (4.6)$$

$$atol_k = 1.0, \quad rtol_k = 10^{-l}, \quad l = 2, 3, 4, 5. \quad (4.7)$$

The
recomm
used fo
lumping
Accord
time st
compar
CPU le
lumping
could c
with va
minima
TWOS
Of cou
interva
note th

5. Con

Wh
the lo

Urban Case : SGI CPU Times

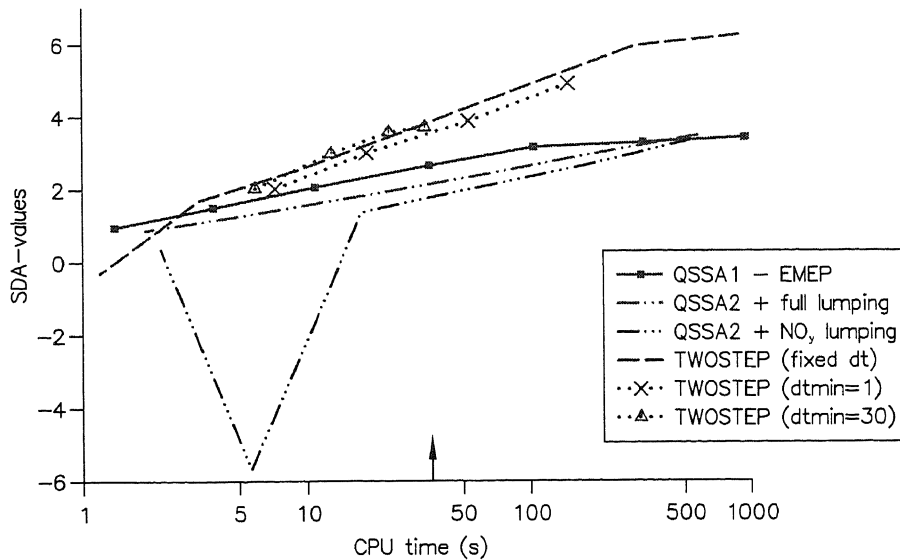


Fig. 6. SDA-values (4.3) versus SGI CPU time for the urban case.

The very poor results found at long stepsizes are expected, and simply underscore the recommendations of [7,9] that with QSSA methods stepsizes of at most 30 seconds should be used for urban simulations. In particular, the collapse of accuracy in QSSA2 with only NO_y lumping shows the instability that can result in QSSA methods without sufficient lumping. Accordingly, the arrow on these figures indicate the CPU usage of QSSA1 for the 30-second time steps, and should serve as a reference for practical usage. Interestingly, also in this comparison TWOSTEP with fixed stepsizes performs notably better than QSSA1, near this CPU level, in spite of the fact that we have implemented only the barest minimum of the lumping procedures in TWOSTEP (only the NO_y group). For fixed stepsizes performance could certainly be improved by additional lumping procedures. We also see that TWOSTEP with variable stepsizes performs again better than QSSA1 at this level, so long as we use minimal stepsizes of approximately 30 seconds or less, although the use of variable stepsizes in TWOSTEP again does not seem to offer much performance benefit over the fixed-step version. Of course, the size of the integration intervals of 2 hours plays a role here. For longer time intervals it is to be expected that variable stepsizes will work out more efficiently. Finally we note that the kink in the curve for $\tau_{\min} = 30$ is due to this prescribed minimal stepsize.

5. Concluding remarks

When used with the appropriate stepsize, QSSA methods can deliver reasonable accuracy at the low CPU usage levels required by air pollution models. However, this can only be achieved

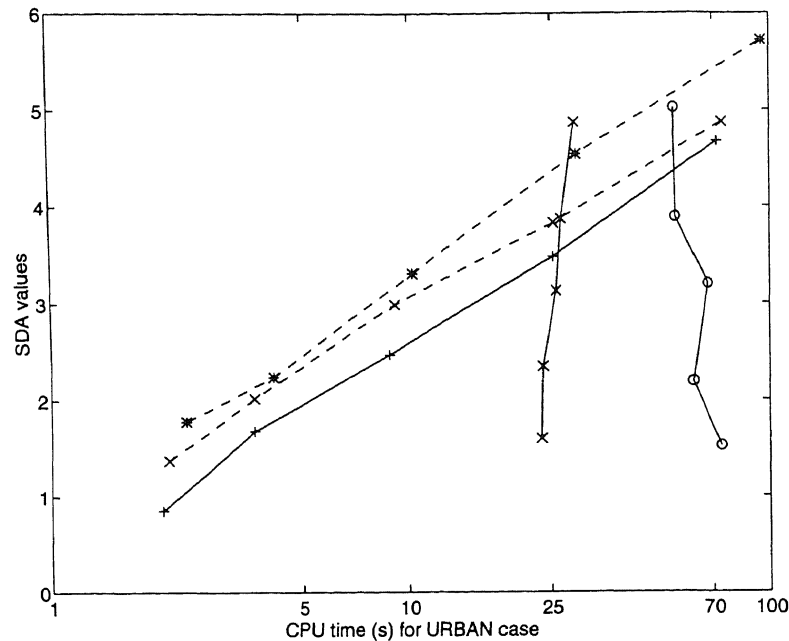


Fig. 7. Results for TWOSTEP applied with NO_y lumping (×, dashed), NO_y + O₃ group iteration (*, dashed) and normal (+, solid). The figure also shows results obtained by DASSL (○, solid) and VODE (×, solid).

by the use of lumping procedures and either iteration as in QSSA1 or a two-stage procedure as used in QSSA2. Without stepsize control, it can also be difficult to specify in advance which stepsize should be used. The two-step method presented here is an attempt to develop a numerical method which is both fast and accurate. We have used only the barest minimum of lumping procedures. Tests with both rural and urban emissions levels have clearly shown the superiority of the two-step method over the QSSA methods near the 1% error level. Undoubtedly, the two-step method could be made even more effective with more tuning (e.g. more grouped species), but even without this the performance level obtained at this stage of development is more than satisfactory. The effect of NO_y lumping is illustrated in Fig. 7, where for the urban test case TWOSTEP is compared, when applied with and without lumping, using

$$\tau_{\min} = 1, \quad \tau_{\max} = 900, \quad (5.1)$$

$$atol_k = 1.0, \quad rtol_k = 10^{-l}, \quad l = 1, 2, 3, 4, 5. \quad (5.2)$$

The effect of lumping is seen to be of help to increase accuracy. We note in passing that CPU times in Fig. 7 differ from CPU times in all previous figures due to working on two different workstations. An alternative to NO_y lumping is to carry out a few additional block Gauss–Seidel iterations on the NO_y + O₃ group since the species in this group are strongly coupled. Fig. 7 shows the results of an experiment with TWOSTEP, using again (5.1)–(5.2), where five such block iterations are carried out for each of the two overall iterations. We see that this leads to a slightly higher performance increase than NO_y lumping. When compared with the normal application of TWOSTEP (solid line, + -marks), the gain is significant. For clarity, note that

here NO_y itself has not been used. Hence only the six individual species listed in (3.7) plus ozone were taken together in the block iteration.

Fig. 7 also shows results for the BDF code DASSL, developed for stiff DAEs [1,4], applied to the ODE problem without lumping and again using the tolerances (5.2). The code was used as a black box integrating with all its default options, except that for each 2-hour interval a starting stepsize $\tau = 1$ was imposed to reduce the start-up costs. DASSL solves the problem without any difficulty, but, as anticipated, near the 1% error range the implicit code is much more expensive than TWOSTEP due to the numerical algebra overhead. For very high accuracy DASSL becomes faster, as it uses a variable-order strategy based on BDF formulas of order one to five. For DASSL the CPU time does not increase for a smaller tolerance, because when the tolerance is decreased less Jacobian updates are used. This obviously saves work and turns out to approximately compensate for the increasing number of integration steps and backsolves which result from decreasing the tolerance.

Because DASSL performed less efficiently than we expected, we also applied the BDF code VODE [2,4] in the same (black box) way as we used DASSL. The method parameters were $\text{ITASK} = 4$, $\text{MF} = 22$. We see that VODE is significantly faster than DASSL, but still much slower than TWOSTEP in the low accuracy range of interest to us. Again a smaller tolerance value does not lead to genuine increase in CPU time. The explanation is the same as for DASSL. However, VODE requires much less Jacobian updates when using $\text{MF} = 22$ (copies of the expensively generated Jacobian are saved) which explains its lower CPU time.

In the near future the current version of the two-step solver will be implemented in the EMEP model. Results will be reported elsewhere. At the same time we will continue our efforts towards the development of a hopefully even faster accurate solver along the ideas used in TWOSTEP.

References

- [1] K.E. Brenan, S.L. Campbell and L.R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations* (North-Holland, Amsterdam, 1989).
- [2] P.N. Brown, G.D. Byrne and A.C. Hindmarsh. VODE: a variable coefficient ODE solver, *SIAM J. Sci. Statist. Comput.* 10 (1989) 1038–1051.
- [3] F.A.A.M. de Leeuw, Numerical solution of ordinary differential equations arising from chemical kinetics, Report 228603005, National Institute of Public Health and Environmental Protection (RIVM), Bilthoven, Netherlands (1988).
- [4] J.J. Dongarra and E. Grosse, Distribution of software via electronic mail, *Comm. ACM* 30 (1987) 403–407 (netlib@research.att.com).
- [5] A. Eliassen, Ø. Hov, I.S.A. Isaksen, J. Saltbones and F. Stordal, A Lagrangian long-range transport model with atmospheric boundary layer chemistry, *J. Appl. Met.* 21 (1982) 1645–1661.
- [6] O. Hertel, R. Bercowicz, J. Christensen and Ø. Hov, Test of two numerical schemes for use in atmospheric transport-chemistry models, *Atm. Env.* 27A (1993) 2591–2611.
- [7] E. Hesstvedt, Ø. Hov and I.S.A. Isaksen, Quasi-steady state approximation in air pollution modelling: comparison of two numerical schemes for oxidant prediction, *Internat. J. Chem. Kinetics* 10 (1978) 971–994.
- [8] Ø. Hov, The abatement of photochemical oxidants in Europe: a 19 days case study using a Lagrangian model with chemistry, Report NILU TR 5/85, Norwegian Institute for Air Research (1987).
- [9] Ø. Hov, I.S.A. Isaksen and E. Hesstvedt, A numerical method to predict secondary air pollutants with an application on oxidant generation in an urban atmosphere, in: *Proceedings WMP Symposium on Boundary Layer*

- Physics Applied to Specific Problems of Air Pollution*, Norrköping, Sweden, WMO Publication No. 510, Geneva, Switzerland (1978) 219–226.
- [10] M.T. Odman, N. Kumar and A.G. Russel, A comparison of fast chemical kinetic solvers for air quality modelling *Atm. Env.* 26A (1992) 1783–1789.
- [11] D. Simpson, Photochemical model calculations over Europe for two extended summer periods: 1985 and 1989. Model results and comparison with observations, *Atm. Env.* 27A (1993) 921–943.
- [12] D. Simpson, Biogenic VOC in Europe, Part II: implications for ozone control strategies (submitted).
- [13] D. Simpson, Y. Andersson-Skold and M.E. Jenkin, Updating the chemical scheme for the EMEP MSC-W model: current status, EMEP MSC-W Note 2/93, The Norwegian Meteorological Institute, Oslo (1993).
- [14] D. Simpson and J.G. Verwer, An evaluation of two numerical methods as applied in a long period atmospheric chemistry model (in preparation).
- [15] J.G. Verwer, Gauss–Seidel iteration for stiff ODEs from chemical kinetics, *SIAM J. Sci. Comput.* 15 (1994) 1243–1250.
- [16] J.G. Verwer and 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.