An efficient chemical systems modelling approach

K.-Y. Wang a,∗, D.E. Shallcross b, P. Hadjinicolaou c, C. Giannakopoulos d

a Department of Atmospheric Sciences, National Central University, Chung-Li, Taiwan
b Centre for Biogeochemistry, School of Chemistry, Bristol University, BS8 1TS, UK
c Centre for Atmospheric Science, Cambridge University, CB2 1EW, UK
d Institute of Environmental Research and Sustainable Development, National Observatory of Athens, Athens, Greece

Received 13 September 2001; received in revised form 19 February 2002; accepted 26 March 2002

Abstract

Systems of stiff chemical reactions are often associated with atmospheric chemistry modelling, which plays a very important role in the studies of stratospheric ozone depletion, tropospheric air pollution problems, and future chemistry-climate feedbacks and interactions. This paper revisits an open-source stiff system solver SVODE and presents its efficient use in modelling different levels of complexity of a range of chemical systems. The chemical systems discussed here are the Lotka–Volterra (predator–prey) model, the Brusselator model, the Oregonator model, and the Lorenz model. The first two models consist of two variables, while the remaining two models consist of three variables. Finally, an application of this modelling approach to a generalised organic/NOx mechanism for characterising air pollution development is presented. Since the SVODE is an open-source code, and the simulations were run on a Linux PC (with g77 compiler), all results discussed in this paper can be easily reproduced. Most importantly, the approach shown here can be readily extended to other larger scale applications such as the three-dimensional air pollution modelling.

Keywords: SVODE; Lotka–Volterra; Brusselator; Oregonator; Lorenz Model; Air Pollution

1. Introduction

With the increasing power of computers and the availability of open-source software in the public domain, it is a very fruitful time for those who are interested in using mathematical modelling and computer simulation in the exploration of scientific understanding or to solve problems by means of a desktop or laptop computer. It is certainly true that modelling of large-scale problems such as the numerical simulation of weather and climate requires huge resources. However, to obtain the insight and the potential benefit of numerical simulation it is not necessary to use such very complicated models. It is the purpose of this paper to present examples for setting up basic numerical modelling systems and to solve these numerically by using a open-source numerical solver SVODE.

Though the examples presented here are simple, considerable experience with numerical modelling can still be obtained. The basic modelling procedures outlined here can be summarized in the following steps: (1) formulating mathematical equations, i.e. a system of ordinary differential equations, and its Jacobian; (2) solving the model numerically; (3) visualization and analysis of the time-dependent model variables. It is our hope that the usefulness of computer-aided problem solving procedures can be revealed using these typical models, and the potential benefit of the procedure presented here can be applicable to other areas of interest.

In this paper we revisit an open-source numerical solver called SVODE (Brown et al., 1989) in modelling the time-evolving behavior of chemical reaction systems. There are three advantages in using the SVODE: (1) The SVODE solver is a general purpose numerical solver for solving stiff and non-stiff first order ordinary differential equation (2) The SVODE solver is easy to incorporate into more complicated models. The only requirement is
to provide a suitable matrix of rate equations and its Jacobian. (3) The SVODE solver is open to the scientific community and is easily accessed via the Internet (http://math.nist.gov/). Besides, the SVODE itself is also well documented within the code.

2. Variable-coefficient ODEs

A system of nonlinear ODEs for the change of chemical concentration \( x_i \) \((i=1,\ldots,N)\) is written as

\[ \frac{dx_i}{dt} = f_i(x_1,x_2,\ldots,x_N), \]  

where \( N \) is the total number of species, and \( f_i \) represents the overall chemical reactions responsible for the change of species \( x_i \).

\[ f_i = f_i(x_1,x_2,\ldots,x_N). \]  

We can write the above dynamical system in a more convenient matrix form,

\[ \frac{dX}{dt} = \mathcal{F}, \]  

\( \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \]

where \( X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \). (4)

and

\[ \mathcal{F} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}. \]  

The implicit time-differencing form of Eq. (3) in the time interval \([t,t+\Delta t]\) is written as

\[ X^{t+\Delta t} - X^t = \mathcal{F} \Delta t. \]  

(6)

If \( \mathcal{F} + \Delta t \) is developed using a Taylor expansion near \((t, X^t)\), from which it follows that

\[ \mathcal{F} + \Delta t = \mathcal{F} + f \Delta X + O(\Delta X^2) . \]  

(7)

where \( \Delta X = X^{t+\Delta t} - X^t \). The Jacobian \( J \) of this nonlinear dynamical system is written as

\[ J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_N} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \frac{\partial f_N}{\partial x_2} & \cdots & \frac{\partial f_N}{\partial x_N} \end{pmatrix} \]  

(8)

After re-arrangements of Eq. (6), using Eq. (7), and neglecting the second order term, leads to

\[ X^{t+\Delta t} = X^t + \Delta t \mathcal{F}^{-1} \mathcal{F}, \]  

(9)

where

\[ \mathcal{F} = I - \Delta t J, \]  

(10)

and \( I \) is the unitary matrix. Hence the inversion of the Jacobin matrix \( J \) is the kernel for the implicit numerical time-stepping. Notice that both \( \mathcal{F} \) and \( X \) are single column matrices, while \( J \) is a square matrix. Given known matrices of \( \mathcal{F}, X, \) and \( J \) at time \( t \), we can then calculate the evolution of chemical concentrations \( X \) at time \( t+\Delta t \) by inversion of the Jacobian matrix \( J \), followed by the straightforward matrix operation.

3. Simulations and results

3.1. The Lotka–Volterra model

The Lotka–Volterra mechanism is as follows (Atkins, 1994):

(1a) \( A + X \rightarrow 2X \),

(1b) \( X + Y \rightarrow 2Y \),

(1c) \( Y \rightarrow P \).

Net \( A \rightarrow P \).

Steps (1a) and (1b) are autocatalytic, and the concentration of \( A \) is held constant by supplying it to the reaction vessel as needed. The chemical reactions (1a)–(1c) generate sustained temporal oscillations in the concentrations of intermediates \( X \) and \( Y \) during the net overall reaction \( A \rightarrow P \) (Field and Noyes, 1974). The rate equations for the two intermediates are

\[ \frac{dX}{dt} = k_{1a} AX - k_{1b} XY, \]  

(11)

\[ \frac{dY}{dt} = k_{1b} XY - k_{1c} Y, \]  

(12)
where $A$, $X$, $Y$ represent concentration of the appropriate chemicals, and $k_{1a}, k_{1b}, k_{1c}$ are rate constants. Since both $A$ and the rate constants are all held constant, we can rewrite the Lotka–Volterra model as following:

$$\frac{dx}{dt} = (A - By)x, \quad (13)$$

$$\frac{dy}{dt} = (Cx - D)y, \quad (14)$$

where $A$, $B$, $C$, and $D$ are model parameters. If we assume that $x$ and $y$ represent the population of rabbits and foxes, respectively, then the Lotka–Volterra model is the familiar predator–prey model.

Results from model integration of the predator–prey model are shown in Fig. 1. Model parameters are set as follows; $A$ is the birth frequency of rabbits ($A = 30$ rabbits week$^{-1}$), $B$ is the death probability of a rabbit when it meets a fox ($B = 0.5$ day$^{-1}$), $C$ is the survival probability of a fox when it meets a rabbit and catches it ($C = 0.5$ day$^{-1}$), and $D$ is the death rate of foxes due to starvation ($D = 20$ foxes day$^{-1}$). $B$ means that if a rabbit $x$ meets a fox $y$ once per day, then there is a 50% chance that the rabbit will be caught by the fox, and the fox will have a 50% chance to survive. Initially, the rabbit number is 20, while the fox number is 2.

The model shows that in the first 25 000 s, the rabbit number increases rapidly, while the fox number gradually decreases. However, as the rabbit number quickly reaches their peak population of about 82 at 37 000 s, the fox number picks up significantly in the time interval between 25 000 and 45 000 s. The growing population of foxes, which reaches a peak population of about 34 at 45 000 s, consumes more and more rabbits than their...
natural reproduction rates. This results in a very rapid reduction of the rabbit population in the time interval between 37,000 and 57,000 s, and hence triggers the later collapse of the fox number. When the fox number is significantly reduced due to the death from starvation, the rabbits have the opportunity to regain their population and surge forward again. The overall cycle based on our initial conditions and model parameters is about 68,300 s. For a model simulation of 500,000 s, we see six complete fox–rabbit cycles.

The time evolution of the fox and rabbit population defines a two-dimensional phase space through which the oscillating population traces a closed trajectory. The phase diagram, Fig. 1(c), clearly shows the dynamical interactions between the fox and rabbit population. At the time when the rabbit number gradually picks up from 20 to 80, no obvious increase of the fox number is perceived (the number remains below five foxes). However, a catastrophic increase in the fox number is triggered when the rabbit number becomes greater than 80. Hence, before the rabbit number reaches 80, the foxes are in a quasi-dormant period. Two stages of the fox population are shown during the reduction stage of the rabbit population. In the first stage, a catastrophic increase in the fox number is accompanied by a gradual reduction in the rabbit population. For the second stage, a catastrophic decrease of the fox number is accompanied by a further reduction in the the rabbit population. It is worthwhile noting that the solution for each specified initial condition traces its own unique track. Hence, tracks of solutions exist according to the different initial conditions used (Fig. 2). These set of tracks represents different

periodic cycles, in a striking contrast to the Brusselator model (described next), which shows only a limit cycle of track regardless to the initial conditions used.

In summary, three stages for the dynamics of the fox population can be drawn: (1) a quasi-dormant period; (2) a catastrophic increase period, and (3) a catastrophic decrease period. The dominant dynamics of this fox–rabbit model is provided by the rabbit number, which is maintained by the natural reproduction rates. The other very important characteristic illustrated by this model is the exact repetitive behavior between the fox and rabbit populations as shown in the phase diagram. At each periodic cycle, the corresponding change of these two variables (e.g. fox and rabbit, or X and Y chemicals) traces exactly the same track as they had left in the previous cycle. Note that the periodic behavior revealed by the Lotka–Volterra model is not associated with an external periodic forcing function. In the atmosphere, the phenomenon that comes closest to exhibiting periodic behavior not associated with periodic forcing function is the quasi-biennial oscillation (QBO) in the zonal winds of the equatorial stratosphere (Holton, 1992).

3.2. The Brusselator model

The second test set of a two-variable nonlinear dynamical system is the Brusselator model. Tyson (1973) showed a simple mechanism of chemical reactions that exhibits properties of nonlinear oscillations. The chemical mechanism considered is

\[
\begin{align*}
(2a) & \quad A \rightarrow X, \\
(2b) & \quad B + X \rightarrow Y + D, \\
(2c) & \quad 2X + Y \rightarrow 3X, \\
(2d) & \quad X \rightarrow E,
\end{align*}
\]

where A and B are input chemicals maintained at constant concentrations, D and E are output chemicals, and X and Y are intermediates. The Brusselator is the only known chemical mechanism which exhibits a limit cycle oscillation and yet contains only two dependent variables (X and Y). The rate equations for these two intermediates are

\[
\begin{align*}
\frac{dx}{dt} &= k_{1a}A - k_{1b}BX + k_{1c}x^2y - k_{1d}x, \\
\frac{dy}{dt} &= k_{1b}BX - k_{1c}x^2y,
\end{align*}
\]

where \(A, B, k_{1a}, k_{1b}, k_{1c}, k_{1d}\) are model parameters. Concentrations \(A\) and \(B\) are set to 0.3 and 0.2, respectively. Rate constants for reactions (2a)–(2d) are set to 0.1, 0.3, 0.26, and 0.1, respectively.

Figs. 3–5 show results from time integration of the
Brusselator model with three different initial conditions \((x,y)\) of \((0.5,0.5)\), \((0.8,0.6)\), and \((1.8,0.3)\), respectively. The only differences for the results shown here are the initial conditions. It is clearly seen from the time series plots of \(x\) and \(y\), as seen in Figs. 3a, 4a and 5a and Figs. 3b, 4b and 5b, that the change of concentration of these two species is gradually attracted to the same periodic oscillation. The phase diagrams, as seen in Figs. 3c, 4c and 5c, eventually show the same track of a limit cycle for variables \(x\) and \(y\). Integrated with three different initial conditions. If we transpose those three different phase diagrams into a single plot, Fig. 6, we can see the convergence of initially three different tracks gradually moving toward the same track of a common limit cycle.

In summary, the Brusselator model exhibits track of a common limit cycle on the \(x\)–\(y\) phase diagram regardless of different initial conditions used. In a sharp contrast, the Lokta–Volterra model exhibits many tracks of periodic cycles on the \(x\)–\(y\) phase diagram, and each track corresponding to each individual initial conditions.

3.3. The Oregonator model

The best understood chemical oscillating reaction is the Belousov–Zhabotinskii (BZ) reaction, which is driven by the oxidation of an organic material by the bromate ion \((\text{BrO}_3^-)\) in a strongly aqueous acid media, and the overall reaction is usually catalyzed by a metal...
Fig. 4. Solution of the Brusselator model using SVODE solver. The initial conditions of \((X, Y)\) are \((0.8, 0.6)\). (a) Time variation of concentration \(X\) against time. (b) Time variation of concentration \(Y\) against time. (c) Phase diagram which shows time variation of concentrations \(Y\) against \(X\).

ion with two oxidation states separated by a single electron (e.g. Ce(IV) and Ce(III)) (Field and Schneider, 1989). The overall mechanism involves so many species and reactions that the resulting differential equations are too complex to be readily discernible. A skeleton model of the BZ reaction with a reduced mechanism, usually referred to as the Oregonator model, has been proposed and successfully used in interpreting the behavior of the BZ reaction in understanding its dynamic structure.

The Oregonator model is written as (Field and Schneider, 1989)

\[
\begin{align*}
(3a) & \quad A + Y \rightarrow X + P, \\
(3b) & \quad X + Y \rightarrow P + P, \\
(3c) & \quad A + X \rightarrow 2X + Z, \\
(3d) & \quad X + X \rightarrow A + P, \\
(3e) & \quad Z \rightarrow fY
\end{align*}
\]

where \(A = \text{BrO}_3^{-}\), \(P = \text{HOBr}\), \(X = \text{HBrO}_2\), \(Y = \text{Br}^{-}\), and \(C = \text{Ce(IV)}\). Hypobromous acid (HOBr) is the product, \(A\) is the input reactant, and \(X\), \(Y\), and \(Z\) are intermediates. The major dynamical variables are \(\text{HBrO}_2\), \(\text{Br}^{-}\), and...
Fig. 5. Solution of the Brusselator model using SVODE solver. The initial conditions of \((X,Y)\) are \((1.8,0.3)\). (a) Time variation of concentration \(X\) against time. (b) Time variation of concentration \(Y\) against time. (c) Phase diagram which shows time variation of concentrations \(Y\) against \(X\).

Ce(IV), respectively. The rate equations which describe the Oregonator model are written as

\[
\begin{align*}
\frac{dx}{dt} &= k_{3a}Ay - k_{3b}xy + k_{3c}Ax - 2k_{3d}x^2, \\
\frac{dy}{dt} &= -k_{3a}Ay - k_{3b}xy + f_k z, \\
\frac{dz}{dt} &= k_{3e}Ax - k_{3e}z.
\end{align*}
\]

Following Field and Noyes (1974), the rate constants employed and the initial concentrations of \(\text{BrO}_3^-\), \(\text{HBrO}_2\), \(\text{Br}^-\), Ce(IV) were

\[
\begin{align*}
k_{3a} &= 1.34 \text{ M}^{-1} \text{s}^{-1}, \\
k_{3b} &= 1.6 \times 10^9 \text{ M}^{-1} \text{s}^{-1}, \\
k_{3c} &= 8 \times 10^3 \text{ M}^{-1} \text{s}^{-1}, \\
k_{3d} &= 4 \times 10^7 \text{ M}^{-1} \text{s}^{-1}, \\
k_{3e} &= 1 \text{ s}^{-1}, \\
A &= [\text{BrO}_3^-] = 6 \times 10^{-2} \text{ M}.
\end{align*}
\]
Fig. 6. Solution tracks of the Brusselator model for three different initial conditions.

\[
x = [\text{HBrO}_2] = 5.025 \times 10^{-11} \text{ M},
\]
\[
y = [\text{Br}^-] = 3 \times 10^{-2} \text{ M},
\]
\[
z = [\text{Ce(IV)}] = 4.8 \times 10^{-8} \text{ M},
\]
\[
f = 1.
\] (29)

The numerical results from the time integration of the Oregonator model are shown in Fig. 7. The period of the oscillation is about 48.67 s, which is very similar to that obtained experimentally for the same concentrations of bromate and hydrogen ion reported by Field and Noyes (1974). The time evolution of the concentrations of [HBrO\(_2\)], [Br\(^-\)], and [Ce(IV)] define a three-dimensional phase space through which the oscillating reaction traces a closed trajectory. The trajectory is illustrated by plots of two-dimensional projections as shown in Fig. 8(a), which relates \(\log[\text{Ce(IV)}]\) to \(\log[\text{Br}^-]\), and in Fig. 8(b), which relates \(\log[\text{HBrO}_2]\) to \(\log[\text{Br}^-]\), respectively. There are basically four complete periodic oscillating cycles shown here. Each oscillating cycle traces exactly the same trajectory as the previous cycle had traced. Note that the phase diagrams from tests with different initial conditions (not shown here) also indicate a common limit cycle for the three-variable Oregonator model. A limit cycle system was also seen in the previous test case of the two-variable Brusselator model.

During an oscillating chemical reaction, the concentrations of the intermediate species exhibit oscillations with ranges over several orders of magnitude. For example, from \(~10^{-11}\) M to \(~10^{-6}\) M for [HBrO\(_2\)], from \(~10^{-10}\) M to \(~10^{-4}\) M for [Br\(^-\)], and from \(~10^{-8}\) M to \(~10^{-4}\) M for [Ce(IV)]. Notice that the very abrupt change of [HBrO\(_2\)] concentration happens only in a extremely short period of time compared with its quasi steady state concentration which prevails over most of the time integration period (200 s). The slightly longer period of abrupt change in concentration happened in [Ce(IV)] compared with [HBrO\(_2\)]. For [Br\(^-\)], the increase of its concentration only occurs in a very short period of time, followed by a gradually decrease and then quick reduction in concentrations. Those noticeable features also indicate that it is necessary for us to use a reasonable fine scale resolution (in time) to resolve very delicate dynamical structures of a system like the Oregonator model.

3.4. The Lorenz model

Until now we have been able to resolve dynamic structures of the Lotka–Volterra model, the Brusselator model, and the Oregonator model using SVODE solver. A common feature observed in those experiments is that they all oscillate in a regular periodic fashion. The trajectory of variables all trace their previous history. However, lack of periodicity is very common in a natural
 Fig. 8. Phase diagrams for the solution of the Oregonator model. (a) LOG[Ce(IV)]+7.6167 against LOG[Br⁻]1+6.5228. (b) LOG[HBrO₂]+10.2988 against LOG[Br⁻]1+6.5228.

Fig. 9 shows results from time integration of the Lorenz model. The initial conditions used are \(x=0\), \(y=1\), and \(z=1\). Hence initially the fluid is motionless (\(x=0\) fluid system, and is one of the distinguishing features of turbulent flow (Lorenz, 1963).

The Lorenz model is based on a (gross) simplification of the fundamental Navier–Stokes equations for fluids (Hilborn, 1994). The model describes the motion of a fluid (say, the atmosphere) under the conditions of Rayleigh–Bénard flow: an incompressible fluid is contained in a cell which has a higher temperature \(T_w\) at the bottom (e.g. due to the sun’s heating at the earth’s surface) and a lower temperature \(T_c\) at the top. Both \(T_w\) and \(T_c\) are maintained at constant values, and hence the temperature difference \(\delta T = T_w - T_c\) is held fixed (\(\delta T\) is taken to be the control parameter for the system). When the fluid is motionless, the temperature varies linearly from bottom to top. The governing equations for the Lorenz model may be written as

\[
\frac{dx}{dt} = -\sigma x + \sigma y, \quad (30)
\]

\[
\frac{dy}{dt} = -xz + rx - y, \quad (31)
\]

\[
\frac{dz}{dt} = xy - bz, \quad (32)
\]

where \(x\) is proportional to the fluid stream function, \(y\) is proportional to the temperature difference between the upward and downward moving parts of a convective cell at a given height, \(z\) is proportional to the deviation from the linear temperature variation as a function of vertical position, and \(\sigma\), \(r\), and \(b\) are model parameters. \(\sigma\) is the Prandtl number,

\[
\sigma = \frac{\text{kinematic viscosity}}{\text{thermal diffusion coefficient}} \quad (33)
\]

which measures the rate of energy loss due to viscosity (i.e. dissipation of energy due to the shearing of the fluid flow) to the rate of energy loss due to thermal conduction for a parcel of fluid. \(r\) is proportional to the Rayleigh number which measures the temperature difference between the bottom and the top of the fluid layer (the temperature difference increases, the Rayleigh number increases). \(b\) is related to the ratio of the vertical height of the fluid to the horizontal size of the convection rolls. Following Lorenz (1963), the following values are used in this study: \(\sigma=10\), \(b=8/3\), \(r=470/19\).

Fig. 9 shows results from time integration of the Lorenz model. The initial conditions used are \(x=0\), \(y=1\), and \(z=1\). Hence initially the fluid is motionless (\(x=0\) fluid
stream function). Notice that the externally controlled forcing of the model, which is the imposed temperature difference \( \delta T = T_w - T_c \), is independent of time. If we first examine the graphs of \( x(t) \) and \( y(t) \) show in Fig. 9, we see that \( x \) and \( y \) oscillate symmetrically around values of \( x=0 \) and \( y=0 \), respectively. Since relative vorticity \( \xi \) and stream function \( \psi \) of a convective cell are given by

\[
\xi = \nabla^2 \psi - \psi, \tag{34}
\]

and

\[
x = \psi - \xi, \tag{35}
\]

negative and positive signs of \( x \), indicate that the direction of the convective cell is counterclockwise and clockwise, respectively. This tells us that the fluid is convecting (represented by \( x \)) first in the clockwise direction, then counterclockwise, continually reversing as time goes on. The temperature difference between the upward and downward moving parts of a convective cell (represented by \( y \)) also oscillates symmetrically around \( y=0 \). A close examination of the time series plots between \( x(t) \) and \( y(t) \) shows that the signs of both variables are almost the same through out the whole integration period. Exceptions are found only at times close to \( t=36 \) s, \( t=45 \) s, \( t=57 \) s.

Since the convection (represented by \( x \)) is characterized by the upward and downward moving parts of a convective cell, similar signs of \( y \) and \( x \) at time \( t \) indicate that the upward moving part of convection is accompanied by the warm fluid and the downward moving part of convection is accompanied by the cold fluid. The strength of the rising branch of the warm fluid is stronger than the descending branch of the cold fluid. This result strongly suggests that the convective cell is directly driven by the rising warm fluid which starts close to the bottom boundary (close to the \( T_w \) environment). The \( z \) variable oscillates around the non-zero value of \( z=25 \). A positive value of \( z \) indicates that the strongest gradients of the vertical temperature profiles which are distorted from linearity occur near the boundaries.

The time evolution of \( x \), \( y \), and \( z \) defines a three-dimensional phase space through which the oscillating variables trace a continuously reversing trajectory. The trajectory is illustrated by plots of two-dimensional projections as shown in Fig. 10, which relates \( z \) to \( y \), and in Fig. 11, which relates \( x \) to \( y \), respectively. Since similar signs of \( x \) and \( y \) denote that warm fluid is rising and cold fluid is descending, we see that most of the tracks in the \( x-y \) phase space concentrate on these two regions (Fig. 11), hence the dominant fluid motions are governed by the thermally direct driven convective rolls.

### 3.5. A summary on the of SVODE in Fortran codes

An example demonstrating the use of SVODE in a Fortran code is shown in Fig. 12. The kernel of the code is the time loop, which calculates time varying numerical solutions using the SVODE method. Before going into this time loop, the user has to set up appropriate subroutine arguments to be passed (used) in SVODE. We advice strongly that the user refer to the original SVODE code for a detail description on the set up of individual argument.

In addition to the above mentioned arguments
required by the SVODE solver, the user has to provide two additional codes for computing the forcing (FEX) and the Jacobian (JEX) terms. Fig. 13 shows segments of both FEX and JEX for the examples discussed in this section.

Fig. 12. A code segment showing the call to SVODE in a Fortran code.

Fig. 13. Codes of the model (FEX) and the Jacobians (JEX) for examples discussed in Section 3.

3.6. Comparison with other numerical and analytical solutions

The use of SVODE as an integrator for solving stiff differential equations is further tested against analytical solutions and an explicit numerical solver using the Runge–Kutta–Fehlberg (RKF) method (Wheatley, 1999). Table 1 shows the comparison of the SVODE with the RKF method, respectively, against the analytical solutions of a set of differential equations. The track of the analytical solutions are shown in Fig. 14. It is shown that the RKF method is more accurate than the SVODE method on these tests when running with three different time steps. The computational times required by both methods are similar. In this case, both variables have very similar time scale of variations.

Table 2 shows another test of both methods on a set of stiff differential equations. For this stiff system, the SVODE method shows a much better performance over the RKF method, especially when one considers using larger time steps. For example, the SVODE method can still maintain a relatively high accuracy at a larger time step of 0.01 s, while the RKF method can only be used with a very small time step (less than 0.001 s for this example). Both methods achieve a similar accuracy at a time step of 0.0001 s but with a sharp increase in computational times. Hence it is shown from these tests that the SVODE solver excels at integrating a stiff system, while the RKF method is more suitable for integrating a less stiff system.

4. Application to air pollution modelling

Based on previous simple applications of chemical systems that contain two or three variables, we are now ready to model more complicated situations involving many variables. Table 3 shows a generalised reaction mechanism that illustrates the basic features of air pollution chemistry (Seinfeld and Pandis, 1998). For example, formation of peroxy radicals (R4, R5, R7, R8), conversion of NO to NO2 (R9, R10, R12), formation of O3 (R1, R2), loss of O3 (R3, R13, R14, R15, R20), loss of organic radicals (R16, R17), formation of HNO3 (R15), PAN (R8, R18) and other products (e.g., ROOH, H2O2, CO, CO2). We integrate Table 3 using SVODE with the following initial conditions,

- Temperature=298 K
- Total pressure=760 Torr
- (H2O)=3.81×10^{27} molecules cm^{-3} (Relativity humidity=50%)
- (RH)=100 ppbv
- (HCHO)=100 ppbv
- (RCHO)=100 ppbv
- (NO)=100 ppbv
Table 1
Experimental results of running ana.Fa Using SVODE/RKF method
<table>
<thead>
<tr>
<th>Time Step (s)</th>
<th>0.001</th>
<th>0.01</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS(x)</td>
<td>SVODE</td>
<td>4.805639E-05</td>
<td>8.825901E-05</td>
</tr>
<tr>
<td>RMS(y)</td>
<td>SVODE</td>
<td>5.135581E-05</td>
<td>9.050101E-05</td>
</tr>
<tr>
<td>Total Steps</td>
<td></td>
<td>100000</td>
<td>100000</td>
</tr>
<tr>
<td>End Time (s)</td>
<td></td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>CPU Time (s)</td>
<td></td>
<td>4.62</td>
<td>0.46</td>
</tr>
<tr>
<td>RKF</td>
<td></td>
<td>4.148147E-06</td>
<td>2.006474E-06</td>
</tr>
</tbody>
</table>

a The code using SVODE and RKF methods, respectively, for solving the following set of differential equations (Simmons, 1972): 

\[
\frac{dx}{dt} = -y + x(1-x^2-y^2) 
\]

\[
\frac{dy}{dt} = x + y(1-x^2-y^2), 
\]

The analytical solution for this system is given by 

\[
x = \frac{\cos(t + t_0)}{\sqrt{1 + ce^{-2t}}} 
\]

\[
y = \frac{\sin(t + t_0)}{\sqrt{1 + ce^{-2t}}} 
\]

where c and t_0 are determined according to 

\[
r = \frac{1}{\sqrt{1 + ce^{-2t}}} 
\]

\[
\theta = t + t_0 
\]

Here \( r = \sqrt{x^2 + y^2} \). Hence given a pair of initials conditions, an analytical solution for the above system can be found. A solution is shown in Fig. 14, which is used for comparing SVODE with RKF methods.

b The root mean squared error between analytical and numerical solutions (calculated by SVODE and RKF methods, respectively) for x.

c The same as previous one but for y.

\[(NO_2) = 10 \text{ ppbv}\]

Fig. 15 shows the time evolution of some selected species when the generalised chemical mechanism is integrated from \( t=0 \) to \( t=300 \text{ min} \). We see that the initial hydrocarbon RH is steadily consumed. The aldehydes RCHO increase initially, due to the conversion of some RH into RCHO, followed by a steady decline in concentration as levels of its precursor RH decrease. A distinctive conversion of NO to NO_2 is clearly seen during the first 2 h of integration. This period also shows the most significant O_3 increase to more than 200 ppbv by the end of 2 h. This period also exhibits the pronounced formation of HNO_3 and PAN, indicating that part of the NOx has been converted to these two species. Since HNO_3 is susceptible to atmospheric wet scavenging processes by cloud water and rain water, the formation of HNO_3 provides an effective sink for NOx (and loss to O_3). On the other hand, PAN is more longer lived in cold environments, indicating that the bounding of NOx to PAN provides an effective way for transporting NOx from polluted areas to remote regions.

The inherent nonlinear behaviours of the air-pollution system is shown in Fig. 16 for the nonlinearity of ozone production in the troposphere. It has been shown (e.g. Lin et al. (1988) that ozone production in the polluted troposphere (e.g. in a megacity) depends highly on the ambient levels of nitrogen oxides and hydrocarbons. Our calculations shown here, based on a simplified chemical reaction mechanism, concur with reported characteristics of ozone variations under different NO and RH conditions.

5. Concluding remarks

In this paper we revisit some classic chemical reaction systems using an effective open-source general simulator SVODE. The procedure outlined here is applied to the study of a generalised organic/NOx reaction mechanism.
for air pollution modelling. The SVODE solver has also been employed in other modelling studies of atmospheric chemistry (e.g., Carver et al., 1997). The chemical systems described here are the Lotka–Volterra model, the Brusselator model, and the Oregonator model, together with a dynamical system of the Lorenz model. The first three models have direct connection to the chemical reactions and hence may provide further information for models of atmospheric chemistry.

Although the fourth model is more related to the hydrodynamical than chemical problems, the phenomenon shown here could have direct implication for the study of chemical reaction mechanisms.

Typical phenomena revealed from the model results are characterized by the following words: nonlinearity, oscillation, and periodicity. The importance of the study of nonlinear dynamics is the realization that a great deal of the complexity observed in nature does not necessarily result from great complexity in fundamental structure. It instead may result from rather simple structure if the governing dynamics are nonlinear (Field and Schneider, 1989). During an oscillating chemical reaction the concentrations of reactants constantly decrease, the concentrations of products constantly increase, and the concentrations of intermediate species execute oscillations as the conversion of reactants to products rushes toward equilibrium (Field and Schneider, 1989). The minimum requirements for chemical oscillation are nonlinearities, autocatalysis, self-inhibition, or a delayed feedback loop. Generally, a complex chemical reaction mechanism which involves reactants, products, intermediates, and catalyst species is built up by the sequence of simple chemical reactions. Once the mechanism is known, the differential equations (i.e. rate equations) governing the dynamic behavior of a chemical system can be written and solved numerically. Hence the connection between laboratory experiment and mathematical modelling can be made. Finally, all materials shown in this work are made available to those interested in experimenting themselves on their Linux PC platforms.

![Fig. 14. Phase diagram which shows time variation of Y against X for a set of equations described in Table 1.](image_url)

Table 2

<table>
<thead>
<tr>
<th>Time step (s)</th>
<th>0.0001</th>
<th>0.001</th>
<th>0.01</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMS(x)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVODE</td>
<td>7.86E-08</td>
<td>7.98E-08</td>
<td>9.07E-08</td>
</tr>
<tr>
<td>RKF</td>
<td>8.20E-08</td>
<td>1.04E-05</td>
<td>–</td>
</tr>
<tr>
<td>RMS(y)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVODE</td>
<td>3.94E-08</td>
<td>4.03E-08</td>
<td>4.52E-08</td>
</tr>
<tr>
<td>RKF</td>
<td>4.00E-08</td>
<td>1.04E-05</td>
<td>–</td>
</tr>
<tr>
<td>Total Steps</td>
<td>1000000</td>
<td>100000</td>
<td>10000</td>
</tr>
<tr>
<td>End Time (s)</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>CPU Time (s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVODE</td>
<td>52.06</td>
<td>5.09</td>
<td>0.51</td>
</tr>
<tr>
<td>RKF</td>
<td>54.75</td>
<td>5.26</td>
<td>–</td>
</tr>
</tbody>
</table>

* The code using SVODE and RKF methods, respectively, for solving a stiff set of equations (Press et al., 1999):

\[
\frac{dx}{dt} = 998x + 1998y
\]

\[
\frac{dy}{dt} = -999x - 1999y.
\]

The analytical solution for this system is given by

\[
x = 2e^{-t} - e^{-1000t}
\]

\[
y = -e^{-t} + e^{-1000t}.
\]
Fig. 15. Time evolution of the concentrations of selected species according to the generalised mechanism shown in Table 3.

Fig. 16. Isopleths of O₃ concentrations at 140th min (close to the model noon time) and its dependence on the initial NO and hydrocarbon concentrations calculated according to the generalised mechanism shown in Table 3.

Acknowledgements

We are very grateful to the original creators of the SVODE, Peter N. Brown, Alan C. Hindmarsh, and George D. Byrne, for making their code available for computational research. The authors thank Mathew J. Evans for the helpful discussions regarding SVODE, and the help and support from J.A. Pyle, D.J. Lary, and S.M. Hall. We also thank two anonymous reviewers for their insightful comments that greatly improved the clarity of this paper. The Centre for Atmospheric Science is a joint initiative of the Department of Chemistry and the Department of Applied Mathematics and Theoretical Physics. This work forms part of the NERC UK Universities Global Atmospheric Modelling Programme.
References


