Journal of Computational Physics **170,** 589–602 (2001)
doi:10.1006/jcph.2001.6750, available online at http://www.idealibrary.com on IDEAL®

Positive Numerical Integration Methods for Chemical Kinetic Systems

Adrian Sandu

Department of Computer Science, 205 Fisher Hall, Michigan Technological University, 1400 Townsend Drive, Houghton, Michigan 49931
E-mail: asandu@mtu.edu

Received December 14, 1999; revised August 21, 2000

Chemical kinetics conserves mass and renders nonnegative solutions; a good numerical simulation would ideally produce mass-balanced, positive concentration vectors. Many time-stepping methods are mass conservative; however, unconditional positivity restricts the order of a traditional method to one. The projection method presented in this paper ensures mass conservation and positivity. First, a numerical approximation is computed with one step of a mass-preserving traditional scheme. If there are negative components, the nearest vector in the reaction simplex is found by solving a quadratic optimization problem; this vector is shown to better approximate the true solution. A simpler version involves just one projection step and stabilizes the reaction simplex. This technique works best when the underlying time-stepping scheme favors positivity. Projected methods are more accurate than clipping and allow larger time steps for kinetic systems which are unstable outside the positive quadrant.

Key Words: chemical kinetics; linear invariants; positivity; numerical time integration; quadratic optimization.

1. INTRODUCTION

Air-quality models [3, 11] solve the convection–diffusion reaction set of partial differential equations which describe the atmospheric physical and chemical processes. Usually an operator-split approach is taken: chemical equations and convection–diffusion equations are solved in alternative steps. In this setting the integration of chemical kinetic equations is a demanding computational task. The chemical integration algorithm should be stable in the presence of stiffness; ensure a modest level of accuracy, typically 1%; preserve mass; and keep the concentrations positive.

Most popular ODE integrators (multistep, Runge–Kutta, Rosenbrock) preserve mass, but positivity is more difficult to achieve. Clipping (setting the negative concentrations to zero)

0021-9991/01 $35.00

enhances stability but artificially adds mass to the system. There are numerical integration methods that automatically preserve both mass and positivity, e.g., backward Euler [8]. However, as shown by Bolley and Crouzeix [2], positivity either restricts the order of the method to one or restricts the step size to impractically small values.

A general analysis of conservation laws and the numerical solutions of ordinary differential equations are given by Shampine [13]. The author considers "perturbation methods" where the computed solution is modified to satisfy exactly the desired invariants. The methods presented here belong to this "perturbation" category, but they are specific to systems with linear equality and inequality invariants, e.g., chemical kinetic models. The main idea can be directly extended to systems whose solutions remain within a convex set.

In this paper we try to alleviate the order and step-size restrictions that come with positivity. The solutions computed at each step by a standard integration method are "projected" back onto the reaction simplex. The resulting vectors better approximate the true solution itself (Lemma 4.1). The paper also presents a simpler, noniterative stabilization method. The techniques developed are of theoretical interest as they produce mass-balanced and positive solutions with high-order schemes and large step sizes. These techniques are of practical interest for general chemical kinetic mechanisms, whenever nonlinearity makes negative solutions unstable: projection stabilizes the integration and reduces numerical errors at large time steps.

The paper is organized as follows. Section 2 reviews the chemical kinetic problem and its mass-balanced and positivity properties; the preservation of these properties by numerical schemes is discussed in Section 3. Section 4 develops the positivity-preserving projection algorithm, while the optimization process is highlighted in Section 5. A simpler technique for stabilizing the reaction simplex is discussed in Section 6. A test case from stratospheric chemistry is considered in Section 7, where different numerical results are presented. Finally, the findings and conclusions of the paper are summarized in Section 8.

2. MASS-ACTION KINETICS, LINEAR INVARIANTS, AND POSITIVITY

Consider a chemical kinetic system with s species $y_1, \ldots, y_s$ interacting in r chemical reactions:

$$\sum_{j=1}^{s} \ell_{ij} y_j \xrightarrow{k_i} \sum_{j=1}^{s} r_{ij} y_j, \qquad i = 1, \ldots, r.$$

To describe the system one builds the matrices of stoichiometric coefficients

$$R = (r_{ij})_{ij}, \qquad L = (\ell_{ij})_{ij}, \qquad S = R - L \in \Re^{s \times r}, \tag{2.1}$$

and the vector of reaction velocities $\omega \in \Re^r$

$$\omega_i(y) = k_i \prod_{j=1}^{s} (y_j)^{\ell_{ij}}, \qquad i = 1, \ldots, r. \tag{2.2}$$

The time evolution of the chemical concentrations is governed by the "mass-action kinetics" differential law[1]

$$y' = S \cdot \omega(y), \qquad y(t_0) = y^0. \tag{2.3}$$

Each vector $e \in \ker(S^T)$ is a linear invariant of the system (2.3) since

$$e^T S = 0 \Rightarrow e^T y'(t) = 0 \Rightarrow e^T y(t) = \text{const.}$$

If $\text{rank}(S) = s - m$ the system admits m linearly independent invariants; let $A \in \Re^{s \times m}$ be a matrix whose columns form a basis for the null space of S^T. Any solution of (2.3) satisfies

$$A^T y(t) = A^T y^0 = b = \text{const.} \qquad \text{for all } t \geq t_0, \tag{2.4}$$

where $b \in \Re^m$ is the vector of invariant values. Simply stated, the existence of linear invariants ensures that mass is conserved during chemical reactions.

Let us now separate the production terms $P(y)$ from the destruction terms $D(y)$ in (2.3):

$$P(y) = R\omega(y), \qquad D(y) = \text{diag}\left(\frac{[L\omega(y)]_1}{y_1} \cdots \frac{[L\omega(y)]_s}{y_s}\right), \qquad y' = P(y) - D(y)y.$$

The special form of the reaction velocities (2.2) ensures that $D_{ii}(y)$ are polynomials in y. Recall that $R \geq 0$, $L \geq 0$, and $k \geq 0$ (stoichiometric coefficients and reaction rates are positive). If at time moment τ all concentrations are nonnegative, $y(\tau) \geq 0$, and the concentration of species i is zero, $y_i(\tau) = 0$, then the corresponding derivative is nonnegative, $y_i'(\tau) = P_i(\tau) \geq 0$, which implies that

$$y(t_0) \geq 0 \Rightarrow y(t) \geq 0 \qquad \text{for all } t \geq t_0. \tag{2.5}$$

In short, the concentrations cannot become negative during chemical reactions.

Linear invariants (2.4) and positivity (2.5) imply that the solution of (2.3) remains within the reaction simplex all the time,

$$y(t) \in \mathcal{S} \qquad \text{for all } t \geq t_0, \qquad \mathcal{S} = \{y \in \Re^s \text{ s.t. } A^T y = b \text{ and } y \geq 0\}. \tag{2.6}$$

3. NUMERICAL PRESERVATION OF THE LINEAR INVARIANTS AND POSITIVITY

A general principle in scientific computing says that the numerical solution must capture (as much as possible) the qualitative behavior of the true solution. Good numerical methods for integrating chemical reaction models (2.3) should therefore *be unconditionally stable*, as the system is usually stiff (this requires implicit integration formulas); should *preserve the linear invariants*—otherwise artificial mass sources (or sinks) are introduced; and should *preserve solution positivity*.

Negative concentrations are nonphysical. In addition, the kinetic system may become unstable for negative concentrations. An operator-split solution of convection–diffusion

[1] We denote by y_i both the chemical species i and its mass concentration.

reaction atmospheric equations alternates chemical integration steps with advection steps; negative concentrations from chemical integration can hurt the positivity of the following advection step, which will perturb the next chemical step, etc., leading to poor-quality results.

A simple example of negative concentrations and instability is provided by Verwer *et al*. [14]. The chemical reaction ($C + C \xrightarrow{k} \cdots$) gives the following time evolution of C:

$$C' = -kC^2, \qquad C(t_0) = C_0 \Rightarrow C(t) = \begin{cases} 0 & \text{if } C_0 = 0, \\ (k(t - t_0) + 1/C_0)^{-1} & \text{if } C_0 \neq 0. \end{cases}$$

Note that if $C_0 > 0$ the solution is bounded ($0 \leq C(t) \leq C_0$) and decreases monotonically, but if $C_0 < 0$ the solution "explodes" in finite time, $C(t) \to -\infty$ as $t \to t_0 + 1/(-C_0 k)$.

It is well known that the most popular integration methods (Runge–Kutta, Rosenbrock, and linear multistep) preserve exactly[2] all the linear invariants of the system [14]. Moreover, the (modified) Newton iterations used to solve for implicit solutions also preserve the linear invariants at each iteration. With linear-preserving integration methods the accuracy of the individual components is given by the truncation errors (e.g., having a magnitude 10^{-4}), while the accuracy of the linear invariants is only affected by the roundoff errors (having a much smaller magnitude, 10^{-14}).

Positivity of the numerical solution is more difficult to achieve. Bolley and Crouzeix [2] showed that (in the linear case) unconditional positivity limits the order of the numerical method to one; conditional positivity imposes tight upper bounds on the step size (similar to the bounds required for the stability of an explicit integration scheme). Hundsdorfer [8] proved that the implicit Euler method is unconditionally positive. In practice, even the implicit Euler method may produce negative values since the iterative solution process is halted after a finite number of steps; while the exact solution is nonnegative, the successive approximations computed by (modified) Newton method are not.

The most common method for avoiding negative concentrations (and possible unstable behavior) is *clipping*. If the solution vector has negative components, they are simply set to zero. Clipping destroys the preservation of linear invariants. Moreover, all clipping errors act in the same direction, namely, increase mass (artificially); therefore they accumulate over time and may lead to significant global errors over longer simulation intervals.

4. SOLUTION PROJECTION METHOD

Consider the numerical solution of the kinetic system (2.3) by a linear-preserving one-step integration method Φ (e.g., Runge–Kutta or Rosenbrock):

$$y^{n+1} = \Phi_h^f(y^n).$$

Here t^n denotes the discrete time value at nth step, y^n is the computed solution, $h = t^{n+1} - t^n$ is the step size, and $f(t, y) = S\omega(t, y)$. The method preserves the linear invariants,

$$A^T y^{n+1} = A^T y^n = b,$$

[2] If the computations are performed in infinite arithmetic precision.

but not the positivity, and some of the computed concentrations may be negative:

$$y_{i1}^{n+1} < 0 \cdots y_{ip}^{n+1} < 0.$$

We perform "clipping"while preserving the linear invariants; i.e.,we project the numerical solution y^{n+1} back onto the reaction simplex $\mathcal{S} = \{A^T z = b, z \geq 0\}$. The projected value should approximate the true solution $y(t^{n+1})$; therefore it has to be chosen as close as possible to the calculated y^{n+1}.

These considerations lead to a reformulation of the clipping problem as a linearly constrained, quadratic optimization problem. Given the "computed value" y^{n+1}, we can find the "projected value" $z^{n+1} \in \mathcal{S}$ which solves

$$\min \frac{1}{2}\|z^{n+1} - y^{n+1}\|_G^2 \qquad \text{subject to } A^T z^{n+1} = b, z^{n+1} \geq 0. \tag{4.1}$$

The norm is $\|y\|_G = \sqrt{y^T G y}$, where G is a positive definite matrix specified below.

Adjustable step ODE solvers compute the solution y^{n+1} together with an estimate of the (component-wise) truncation error e^{n+1}. The step-size control is based on user-prescribed relative (*rtol*) and absolute (*atol*) error tolerances; the following scalar measure of the truncation error is computed:

$$E^{n+1} = \sqrt{\frac{1}{s}\sum_{i=1}^{s}\left(\frac{e_i^{n+1}}{atol + rtol\,|y_i^{n+1}|}\right)^2}.$$

The current value y^{n+1} is accepted if $E^{n+1} < 1$ and rejected otherwise. Clearly

$$E^{n+1} = \|e^{n+1}\|_{G(y^{n+1})} \qquad \text{with } G(y) = \operatorname*{diag}_{l \leq i \leq s}\left[\frac{1}{s(atol + rtol|y_i|)^2}\right]. \tag{4.2}$$

Since a step-control mechanism tries to keep the truncation-error norm $\|e^{n+1}\|_{G(y^{n+1})}$ small, it is natural to formulate the projection problem (4.1) in terms of the same G-norm, that is, to find z in the reaction simplex (2.6) which minimizes the projection-error norm $\|z^{n+1} - y^{n+1}\|_{G(y^{n+1})}$.

Apparently, projection introduces an extra error such that $\|z^{n+1} - y(t^{n+1})\|_G \leq \|y^{n+1} - y(t^{n+1})\|_G + \|z^{n+1} - y^{n+1}\|_G$. A closer look reveals the following.

LEMMA 4.1. *The projected vector is a better* (G-norm) *approximation to the true solution than is the computed vector*,

$$\|z^{n+1} - y(t^{n+1})\|_G \leq \|y^{n+1} - y(t^{n+1})\|_G.$$

Proof. If $y^{n+1} \geq 0$ then $z^{n+1} = y^{n+1}$ and we have norm equality. If $y_i^{n+1} < 0$ for some i, consider the vectors $Y = G^{1/2} y^{n+1}$, $Z = G^{1/2} z^{n+1}$, and $W = G^{1/2} y(t^{n+1})$. We have

$$A^T G^{-1/2} Y = b \quad \text{and} \quad W \in \bar{\mathcal{S}}, \qquad \text{where } \bar{\mathcal{S}} = \{X : A^T G^{-1/2} X = b, X \geq 0\}.$$

The problem (4.1) is equivalent to

$$\min \frac{1}{2}\|Z - Y\|_2^2 \qquad \text{subject to } Z \in \bar{\mathcal{S}}.$$

Since $\bar{\mathcal{S}}$ is a convex set, $Y \notin \bar{\mathcal{S}}$, and Z is the point in $\bar{\mathcal{S}}$ that is closest to Y, the hyper-plane perpendicular to the direction YZ and which passes through Z *separates* $\bar{\mathcal{S}}$ and Y. For any $W \in \bar{\mathcal{S}}$ the angle (YZW) is larger than 90°; therefore YW is the largest edge in the triangle YZW. In particular,

$$\|Y - W\|_2 \geq \|Z - W\|_2 \Rightarrow \|y^{n+1} - y(t^{n+1})\|_G \geq \|z^{n+1} - y(t^{n+1})\|_G. \qquad \blacksquare$$

Note that the above proof only uses the convexity of the reaction simplex $\mathcal{S}$; consequently, the results of this paper extend to any differential equation whose exact solution stays within a convex set.

Schematically, one step of the method reads

$$\begin{array}{l} \bar{y}^{n+1} = \Phi_h^f(y^n); \quad G = G(\bar{y}^{n+1}); \\ \text{IF } \{\bar{y}_i^{n+1} < 0 \text{ for some } i\} \text{ THEN} \\ \quad \min \tfrac{1}{2}\|z^{n+1} - \bar{y}^{n+1}\|_G^2 \text{ s.t. } A^T z^{n+1} = b,\, z^{n+1} \geq 0; \\ \quad y^{n+1} = z^{n+1}; \\ \text{ELSE} \\ \quad y^{n+1} = \bar{y}^{n+1}. \end{array} \tag{4.3}$$

We will call this method the *positive-projection method*, since at each step the numerical solution is "projected" back onto the reaction simplex (2.6). A direct consequence of Lemma 4.1 is that the consistency order of the projection method (4.3) is the order of the underlying time discretization Φ, since the projection step does not increase the truncation error. The same convergence analysis applies. In this particular sense the positive-projection method (4.3) overcomes the order one barrier of Bolley and Crouzeix [2].

The idea can be combined with a variable time-step strategy. If e^{n+1} is the truncation error estimate and $E^{n+1} = \|e^{n+1}\|_G$, the step is accepted for $E^{n+1} < 1$ and rejected otherwise. We check positivity only for accepted solutions, and if negative we project them. Since projection does not increase the error norm (Lemma 4.1) the optimal value can be accepted as the new approximation. If we stop the optimization procedure before the solution is attained, e.g., if at the current iteration $z^{n+1} \in \mathcal{S}$ is "reasonably close" to $\bar{y}^{n+1}$ and we accept it, then a conservative approach makes sense; estimate $F^{n+1} = \|z^{n+1} - y^{n+1}\|_G$ and check that $E^{n+1} + F^{n+1} < 1$; if not, reject the step and continue with a reduced step size. This ensures that $\|z^{n+1} - y(t^{n+1})\|_G < 1$. Also, if the optimization algorithm does not converge (which is possible only in the presence of large numerical errors), then reject the step and continue with a reduced step size.

5. THE OPTIMIZATION ALGORITHM

It remains to find a suitable way of computing z^{n+1}; i.e., a way to solve the quadratic minimization problem (4.1). Note that the reaction simplex (2.6) is never empty—it contains

at least the initial conditions. Therefore, the theoretical minimization problem (4.1) is always feasible since we compute the minimal distance between a point and a nonempty convex set.

For the practical implementation we reformulate (4.1) as

$$\min \frac{1}{2}(z^{n+1})^T G z^{n+1} - (G\bar{y}^{n+1})^T z^{n+1} \qquad \text{subject to } A^T z^{n+1} = b,\ z^{n+1} \geq \epsilon. \tag{5.1}$$

The entries $\epsilon_i > 0$ are small positive numbers; their role is to keep $z_i^{n+1} \geq 0$ even when the computation is corrupted by roundoff. We assume that the mass invariants are independent (A^T has full row rank) and that $\bar{y}^{n+1}$ satisfies the equality constraints ($A^T\bar{y}^{n+1} = b$).

Any algorithm for quadratic programming can be employed to solve (5.1). We found the primal-dual algorithm of Goldfarb and Idnani [5] to be a suitable solution method. This algorithm finds a solution or detects infeasibility in a finite number of steps. The linear algebra involves m-dimensional systems, as opposed to the integration step which solves s-dimensional systems; since the number of invariants m is much smaller than the number of chemical species s, the optimization process is relatively inexpensive. Last, but not least, the Goldfarb and Idnani algorithm can be initialized with the infeasible vector $\bar{y}^{n+1}$; this is a good starting point since $A^T\bar{y}^{n+1} = b$ and the negative part $(\bar{y}^{n+1})^-$ is small (of the order of truncation error).

For a complete description of the algorithm, the reader is referred to the original paper [5]. The algorithm has been adapted to our particular problem (5.1) to accommodate the equality constraints $A^T z = b$ at all iterations, to take advantage of the diagonal form of G, and to exploit the special form of the inequality constraints $z_i \geq \epsilon_i$.

6. SOLUTION STABILIZATION METHOD

We now present a simpler alternative to the projection algorithm. Given $\bar{y}^{n+1} = \Phi_h^f(y^n)$ with $\bar{y}_{i1}^{n+1}, \ldots, \bar{y}_{ip}^{n+1} < 0$, we denote $B = [A|e_{i1}|\cdots|e_{ip}]$, where e_j is the jth unit vector. B is a collection of active constraint normals. We look for z^{n+1}, "the nearest" point to $\bar{y}^{n+1}$ which satisfies $A^T z^{n+1} = b, z_{i1}^{n+1} = 0, \ldots, z_{ip}^{n+1} = 0$. The solution z^{n+1} is given by the orthogonal projection of $\bar{y}^{n+1}$ onto the manifold $\{z : B^T z = [b, 0]^T\}$.

Following the discussion in Section 5, it is advantageous to project onto the perturbed manifold $\{z : B^T z = [b, \epsilon]^T\}, \epsilon = [\epsilon_{i1}, \ldots, \epsilon_{ip}]^T$, positive and small, as a guard against roundoff errors. Since for our application the "nearest distance" is measured in the G-norm (4.2), the corresponding G-orthogonal projection onto the perturbed manifold is employed. The proposed algorithm reads

$$\bar{y}^{n+1} = \Phi_h^f(y^n) \qquad \text{with } \bar{y}_{i1}^{n+1}, \ldots, \bar{y}_{ip}^{n+1} < 0, \qquad B = [A|e_{i1}|\cdots|e_{ip}],$$

$$y^{n+1} = \bar{y}^{n+1} - G^{-1}B(B^T G^{-1} B)^{-1} \begin{bmatrix} A^T\bar{y}^{n+1} - b \\ \bar{y}_{i1}^{n+1} - \epsilon_{i1} \\ \vdots \\ \bar{y}_{ip}^{n+1} - \epsilon_{ip} \end{bmatrix}. \tag{6.1}$$

It can be directly verified that the solution satisfies $A^T y^{n+1} = b, y_{i1}^{n+1} = \epsilon_{i1}, \ldots, y_{ip}^{n+1} = \epsilon_{ip}$. If Φ_h^f preserves the linear invariants (as it should), we can replace $A^T\bar{y}^{n+1} - b = 0$ in (6.1). The implementation is done in a numerically stable fashion using a reduced QR

decomposition:

$$G^{-1/2}B = QR = Q_1R_1, \qquad (B^TG^{-1}B)^{-1} = R_1^{-1}R_1^{-T}.$$

The method does not guarantee positivity, since the projection step may render other components negative (i.e., $y_j^{n+1} < 0$ for $j \neq i1 \dots ip$). To guarantee positivity we can extend B by appending the column e_j, $B \leftarrow [B|e_j]$, and repeat the projection step, etc. This is just the Goldfarb and Idnani algorithm with full steps at each iteration and without the relaxation of unneeded active constraints. Several steps are also needed if the number of negative components p is large, $m + p > s$.

This noniterative version can have a beneficial effect on maintaining positivity. We justify this informally by noting the relationship with the invariant stabilization method of Ascher *et al*. [1]. The kinetic system together with the invariants in explicit form is

$$y' = f(y), \qquad y(t^0) = y^0, \qquad h(y) = \begin{bmatrix} A^Ty - b \\ y^- \end{bmatrix} = 0.$$

B^T is a reduced form of the Jacobian matrix $H = \partial h/\partial y$, obtained by removing the rows and columns which correspond to nonnegative components $y_i \geq 0$. In the spirit of [1] the method (6.1) can be viewed as a discretization of the "stabilized" system

$$y' = f(y) - H^T(HH^T)^{-1}h(y)$$

(with reduced h, H). The correction term is zero if the solution lies within the simplex; if the solution is outside the reaction simplex, the correction term "pulls back" the trajectory toward the simplex; for example, if some component becomes negative, the correction term will increase its concentration. Therefore, the simplex becomes "attractive"; the solution cannot drift away so occasional negative concentrations do not lead to instability. For these reasons, we will refer to (6.1) as the *stabilization* method.

7. NUMERICAL RESULTS

Consider the basic stratospheric reaction mechanism (adapted from NASA HSRP/AESA [9])

$$\begin{aligned}
&(r1)\ \mathrm{O_2} + h\nu \xrightarrow{k_1} 2\mathrm{O} && (k_1 = 2.643 \times 10^{-10} \cdot \sigma^3)\\
&(r2)\ \mathrm{O} + \mathrm{O_2} \xrightarrow{k_2} \mathrm{O_3} && (k_2 = 8.018 \times 10^{-17})\\
&(r3)\ \mathrm{O_3} + h\nu \xrightarrow{k_3} \mathrm{O} + \mathrm{O_2} && (k_3 = 6.120 \times 10^{-04} \cdot \sigma)\\
&(r4)\ \mathrm{O} + \mathrm{O_3} \xrightarrow{k_4} 2\mathrm{O_2} && (k_4 = 1.576 \times 10^{-15})\\
&(r5)\ \mathrm{O_3} + h\nu \xrightarrow{k_5} \mathrm{O}^{1D} + \mathrm{O_2} && (k_5 = 1.070 \times 10^{-03} \cdot \sigma^2)\\
&(r6)\ \mathrm{O}^{1D} + M \xrightarrow{k_6} \mathrm{O} + M && (k_6 = 7.110 \times 10^{-11})\\
&(r7)\ \mathrm{O}^{1D} + \mathrm{O_3} \xrightarrow{k_7} 2\mathrm{O_2} && (k_7 = 1.200 \times 10^{-10})\\
&(r8)\ \mathrm{NO} + \mathrm{O_3} \xrightarrow{k_8} \mathrm{NO_2} + \mathrm{O_2} && (k_8 = 6.062 \times 10^{-15})\\
&(r9)\ \mathrm{NO_2} + \mathrm{O} \xrightarrow{k_9} \mathrm{NO} + \mathrm{O_2} && (k_9 = 1.069 \times 10^{-11})\\
&(r10)\ \mathrm{NO_2} + h\nu \xrightarrow{k_{10}} \mathrm{NO} + \mathrm{O} && (k_{10} = 1.289 \times 10^{-02} \cdot \sigma).
\end{aligned} \tag{7.1}$$

Here $M = 8.120E + 16$ molec/cm^3 is the atmospheric number density, the rate coefficients are scaled for time t in seconds, and $\sigma(t)$ represents the normalized sunlight intensity,

$$T_L = \left(\frac{t}{3600}\right) \mod 24; \quad T_R = 4.5(\text{sunrise}); \quad T_S = 19.5\,(\text{sunset});$$

$$\sigma(t) = \begin{cases} \frac{1}{2} + \frac{1}{2}\cos\left(\pi \left| \frac{2T_L - T_R - T_S}{T_S - T_R} \right| \left[\frac{2T_L - T_R - T_S}{T_S - T_R} \right]\right) & \text{if } T_R \le T_L \le T_S. \\ 0 & \text{otherwise} \end{cases}$$

It is easy to see that along any trajectory of the system (7.1) the number of oxygen atoms and the number of nitrogen atoms are constant,

$$[O^{1D}] + [O] + 3[O_3] + 2[O_2] + [NO] + 2[NO_2] = \text{const.}, [NO] + [NO_2] = \text{const.};$$

therefore if we denote the concentration vector

$$y = [[O^{1D}], [O], [O_3], [O_2], [NO], [NO_2]]^T,$$

the linear equality constraints have the form

$$A^T = \begin{bmatrix} 1 & 1 & 3 & 2 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \qquad A^T y(t) = A^T y(t_0) = b.$$

We implemented the numerical examples in MATLAB. The simulation starts at noon with the initial concentrations shown in Table I and continues for 72 h. The computation of G-norms was done with $rtol = 10^{-5}$ and $atol = 10^{-3}$. Throughout the tests the minimal values were set to $\epsilon_i = 1$ molec/cm^3 . Reference solutions were obtained with the MATLAB integration routine ODE15S (variable order numerical differentiation formula); the control parameters were $RelTol = 10^{-8}$, $AbsTol = 10^{-8}$, with analytic Jacobian.

The integration algorithms used are BDF2 (8.1), Ros2 (8.4), Rodas3 (8.3), and RK2 (8.2). Verwer *et al.* [15, 16] advocated the favorable positivity properties of Ros2.

For the BDF2 and Rodas3 standard solutions $[O^{1D}]$, [NO], and $[NO_2]$ concentrations fall frequently below zero, while Ros2 only seldom gives negative concentrations; this observation is in agreement with [15, 16]. Clipped, projected, and stabilized versions of all methods produce nonnegative concentrations. Table II shows the computational work of each algorithm (in Kflops—thousands of floating point operations) for a step size $h =$ 24 min. Projected Rodas3 is almost 50% more costly than standard Rodas3 since the optimization routine is called many times. The overheads for BDF2 and RK2 are about 28%. With Ros2 the optimization routine is called only a few times and the overhead is small (8%). For all methods stabilization produces similar results at half the overhead. We

TABLE I
Initial Concentrations for the Simulation (molec/cm^3)

System	O^{1D}	O	O_3	O_2	NO	NO_2
(7.1)	9.906*E*+01	6.624*E*+08	5.326*E*+11	1.697*E*+16	8.725*E*+08	2.240*E*+08
(7.1)–(7.3)	9.906*E*+01	6.624*E*+08	5.326*E*+11	1.697*E*+16	4.000*E*+06	1.093*E*+09

TABLE II
Computational Work (in Kflops) for Integrating the Stratospheric Problem (7.1) with Various Algorithms Using a Step Size h = 24 min

Integrator	Standard	w/Projection	Overhead	w/Stabilization	Overhead
Ros2	335 Kflops	362 Kflops	8%	348 Kflops	4%
Rodas3	441 Kflops	655 Kflops	49%	561 Kflops	27%
BDF2	482 Kflops	622 Kflops	29%	557 Kflops	16%
RK2	971 Kflops	1240 Kflops	28%	1126 Kflops	16%.

conclude that projection and stabilization work best when paired with an integration formula that favors positivity (e.g., Ros2) and that stabilization is the more erective technique.

To compare the performance of different methods we measured the solution accuracy at the end of the integration interval ($t = T_F$). With y^R the reference solution and y the computed solution, the error measure reads

$$E = \sqrt{\frac{1}{s}\sum_{i=1}^{s}\left(\frac{y_i(T_F) - y_i^R(T_F)}{y_i^R(T_F)}\right)^2}. \tag{7.2}$$

Figure 1 shows the solution accuracies (7.2) versus computational work (Kflops) for different methods. At large step sizes clipping introduces significant component errors, while projection and stabilization show good accuracies. Recall that errors in the linear invariants

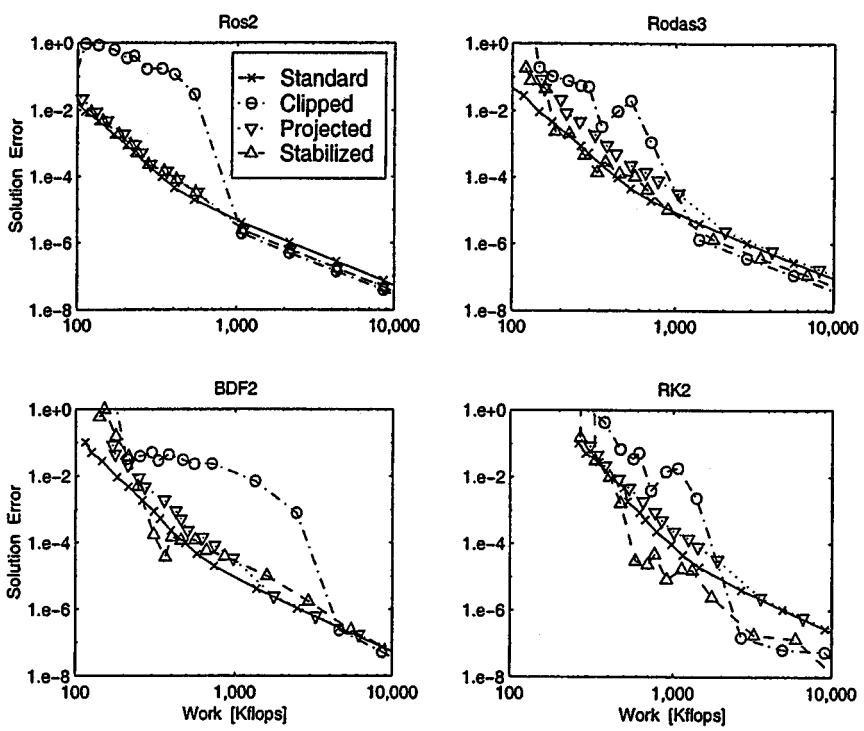

FIG. 1. Work-precision diagrams for the system (7.1). Integration with Ros2, Rodas3, BDF2, and RK2 (standard, clipped, optimal projection, and stabilization methods).

are also very large with clipping and very small with projection and stabilization. For very small step sizes all versions perform similarly. The slopes suggest that for all methods the standard, projected, and stabilized versions have the same orders of accuracy.

The reduced stratospheric system (7.1) autocorrects the negative values of O, O^{1D}, and NO, which explains the good accuracy of the standard methods. For example, the atomic oxygen destruction term is $-[O](k_2[O_2] + k_4[O_3] + k_9[NO_2])$; since the parentheses do not depend on any "possibly negative" concentration, whenever $[O] < 0$ this destruction term is positive (produces O!) and the oxygen concentration increases toward positive values. Not all chemical systems have the autocorrection property. Appending the extra reaction

$$(r11)\ \mathrm{NO} + \mathrm{O} \xrightarrow{k_{11}} \mathrm{NO_2} \quad (k_{11} = 1.0E - 8) \tag{7.3}$$

leads to a noncorrecting kinetic scheme, as (7.3) will continue to destroy NO and O even when their concentrations become negative.

The extended system (7.1)–(7.3) with initial values of Table I was integrated with the positivity-favoring method Ros2. Table III shows the solution errors (7.2) versus the step size, the computational work (in Kflops), and overheads. Clearly, this noncorrecting system is a more challenging computational problem: for fixed step sizes larger than 15 min standard Ros2 is unstable, while clipping introduces significant errors. The projected and the stabilized solutions show good accuracies even at very large time steps. Errors in NO and NO_2 are present at the night-to-day transitions, but they do not accumulate in time. The invariant errors are also very small. For step sizes of 6 min or less the methods give similar results. As expected, the overheads are larger in the large-step regime.

Additional experiments (not shown here) were performed with a variable-step version of Ros2. It produced accurate solutions but the computational costs were much higher than the fixed-step costs for medium-to-modest accuracy. Projection and stabilization did not seem to improve significantly the performance of the standard variable-step algorithm; however, some improvements were noticed when a minimal step $h = 1$ min was imposed.

Lumping does not affect positivity or stability. We integrated the lower dimensional equivalent system obtained by substituting $[O^{1D}] = b_1 - [O] - 3[O_3] - 2[O_2] - [NO] - 2[NO_2]$ and $[NO] = b_2 - [NO_2]$. There still are negative concentrations produced, and the behavior is similar to that of the nonlumped system.

TABLE III

The Work (Kflops) and Overhead versus Solution Accuracy for Integrating the Extended Stratospheric Problem (7.1)–(7.3) with Ros2 for Standard, Clipped, Projected, and Stabilized Versions

Ros2	Standard		w/Clipping		w/Projection		w/Stabilization	
h (min)	Work	Error	Work	Error	Work	Error	Work	Error
48	—	—	189	5.22E+0	222 (17%)	1.48E−3	204 (8%)	1.48E−3
24	—	—	378	5.27E−1	412 (9%)	2.14E−4	393 (4%)	2.14E−4
12	756	1.62E−4	756	4.03E−2	791 (5%)	1.57E−4	773 (2%)	1.57E−4
6	1513	4.43E−5	1513	4.52E−5	1553 (3%)	4.50E−5	1525 (1%)	4.50E−5
3	3027	1.10E−5	3027	1.13E−5	3067 (1%)	1.12E−5	3043 (0.5%)	1.12E−5

8. CONCLUSIONS

We presented two techniques that ensure mass conservation and positivity for the numerical solutions of chemical kinetic problems. The techniques are based on postprocessing the next-step approximations given by linear-preserving methods, should negative concentrations develop.

Projection finds the nearest vector in the reaction simplex (2.6) by solving a quadratic optimization problem. This optimal vector better approximates the true solution (Lemma 4.1). Consequently, projection is an unconditionally positive integration method with the same order of consistency as the underlying time-stepping scheme; in this sense it overcomes the barriers of [2]. In practice, the technique alleviates the step-size restrictions when higher order integration methods are used.

A less expensive alternative is the stabilization method. Although positivity is not guaranteed, the overall behavior is very good and the solutions are similar to the ones obtained by optimal projection.

Both techniques have to be paired with a positivity-favorable numerical integration method, for example, Ros2 [15, 16]. Although such methods do not guarantee positivity, they seldom produce nonpositive results, which minimizes the overheads incurred by projection or stabilization.

Projection and stabilization yield better accuracies than clipping for large time steps. If the kinetic system self-corrects the negative concentrations, neither technique seems necessary. For systems that are unstable at negative concentrations, projection and stabilization allow larger time steps than the standard integration.

In air-quality modeling, fixed-step integration plus clipping is a popular approach; however, clipping adds nonphysical mass. Fixed-step plus simplex projection ensures positivity and mass balance; for medium-to-modest accuracy requirements this is more effective than variable step sizes. In addition, in a parallel implementation of an air-quality model fixed step sizes lead to a better load balance and increased overall efficiency.

An apparent disadvantage of the methods is that one has to compute the linear invariants explicitly. The linear invariants, however, can be automatically generated by specialized software that translates kinetic reactions into differential equations (e.g., KPP [4]).

APPENDIX: NUMERICAL INTEGRATION METHODS

The second-order backward differentiation formula BDF2 [6, Section III.1] is

$$y^{n+1} = Y^n + \frac{2}{3} h f(t^{n+1}, y^{n+1}), \qquad Y^n = \frac{4}{3} y^n - \frac{1}{3} y^{n-1}. \tag{8.1}$$

Here $t^{n+1} = t^n + h = t^{n-1} + 2h$; for variable time steps the coefficients change. The very first step requires both y^1 and y^0; the former is given, while the latter is obtained with one backward Euler step.

The second-order Runge–Kutta method RK2 [10] is

$$\begin{aligned} y^{n+1} &= y^n + (1-\gamma) k_1 + \gamma k_2 \\ k_1 &= h\, f(t^n + \gamma h, y^n + \gamma k_1), \\ k_2 &= h\, f(t^n + h, y^n + (1-\gamma) k_1 + \gamma k_2), \end{aligned} \tag{8.2}$$

with $\gamma = 1 - \sqrt{2}/2$.

For the following Rosenbrock methods the Jacobian matrix $J = \partial f(t, y)/\partial y$ and the time partial derivative $f_t = \partial f(t, y)/\partial t$ are evaluated at $t = t^n$. The Rodas3 method [12] is third-order accurate and reads

$$\begin{aligned} y^{n+1} &= y^n + 2k_1 + k_3 + k_4, \\ \left(\frac{2}{h}I - J\right)k_1 &= f(t^n, y^n) + \frac{h}{2} f_t, \\ \left(\frac{2}{h}I - J\right)k_2 &= f(t^n, y^n) + \frac{4}{h}k_1 + \frac{3h}{2} f_t, \\ \left(\frac{2}{h}I - J\right)k_3 &= f(t^n + h, y^n + 2k_1) + \frac{1}{h}k_1 - \frac{1}{h}k_2, \\ \left(\frac{2}{h}I - J\right)k_4 &= f(t^n + h, y^n + 2k_1 - k_3) + \frac{1}{h}k_1 - \frac{1}{h}k_2 - \frac{8}{h}k_3. \end{aligned} \tag{8.3}$$

The second-order Rosenbrock scheme Ros2 [15, 16] is defined as

$$\begin{aligned} y^{n+1} &= y^n + \frac{3}{2\gamma}k_1 + \frac{1}{2\gamma}k_2, \\ \left(\frac{1}{\gamma h}I - J\right)k_1 &= f(t^n, y^n) + \gamma h f_t, \\ \left(\frac{1}{\gamma h}I - J\right)k_2 &= f\left(t^n + h, y^n + \frac{1}{\gamma}k_1\right) - \frac{2}{\gamma h}k_1 - \gamma h f_t, \end{aligned} \tag{8.4}$$

with $\gamma = 1 + 1/\sqrt{2}$. The vector $y^n + (1/\gamma)k_1$ is a consistent approximation at t^{n+1} and was used to implement the error estimator in the variable step formulation. In [15, 16] it was noted that Ros2 have favorable positivity properties, and the method is stable for nonlinear problems even with large fixed step sizes. It was also noted that Ros2 provides positive solutions for the scalar problems $C' = -kC$ and $C' = -kC^2$, $C(t_0) \geq 0$. A possible explanation for the good observed behavior is that the transfer function of this method and its first two derivatives are all nonnegative for any real, negative argument (i.e., $R(z)$, $R'(z)$, $R''(z) \geq 0$ for any $z \leq 0$). In view of the theory developed in [2] this might reduce the negative values and have a good influence on the positivity of solutions.

ACKNOWLEDGMENT

I thank Mihai Anitescu for a fruitful discussion regarding quadratic optimization algorithms and the norm-approximation property of the solution.

REFERENCES

1. U. M. Ascher, H. Chin, and S. Reich, Stabilization of DAEs and invariant manifolds, *Numer. Math.* **67**, 131 (1994).
2. Bolley C. and M. Crouzeix, Conservation de la positivite lors de la discretization des problemes d'evolution parabolique, *R.A.I.R.O. Numer. Anal.* **12**(3), 237 (1978).
3. G. R. Carmichael, L. K. Peters, and T. Kitada, A second generation model for regional-scale transport/chemistry/deposition, *Atmos. Environ.* **20**, 173 (1986).

4. V. Damian-Iordache, A. Sandu, M. Damian-Iordache, G. R. Carmichael, and F. A. Potra, *KPP—A Symbolic Preprocessor for Chemistry Kinetics—User's Guide*, Technical report, The University of Iowa, Iowa City, IA 52246 (1995).
5. D. Goldfarb and A. Idnani, A numerically stable dual method for solving strictly convex quadratic programs, *Math. Progr.* **27**, 1 (1983).
6. E. Hairer S. P. Norsett, and G. Wanner, *Solving Ordinary Differential Equations I. Nonstiff Problems* (Springer-Verlag, Berlin, 1993).
7. E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff, and Differential-Algebraic Problems* (Springer-Verlag, Berlin, 1991).
8. W. Hundsdorfer, *Numerical Solution of Advection–Diffusion–Reaction equations*, Technical report NM-N9603, Department of Numerical Mathematics, CWI, Amsterdam (1996).
9. D. E. Kinnison, *NASA HSRP/AESA Stratospheric Models Intercomparison*, for NASA ftp site, contact kinnison1@llnl.gov.
10. B. Owren and H. H. Simonsen, Alternative integration methods for problems in structural dynamics,*Comput. Meth. Appl. Mech. Eng.* **122**(1/2), 1 (1995).
11. A. Sandu, *Numerical Aspects of Air Quality Modeling*, Ph.D. thesis (Applied Mathematical and Computational Sciences, The University of Iowa, 1997).
12. A. Sandu, J. G. Blom, E. Spee, J. Verwer, F. A. Potra, and G. R. Carmichael, Benchmarking stiff ODE solvers for atmospheric chemistry equations II—Rosenbrock Solvers, *Atmos. Environ.* **31**, 3459 (1997).
13. L. F. Shampine, Conservation laws and the numerical solution of ODEs, *Comput. Math. Appl.* **12B**(5/6), 1287 (1986).
14. J. G. Verwer, W. Hunsdorfer, and J. G. Blom, *Numerical Time Integration of Air Pollution Models, Modeling, Analysis and Simulations Report*, MAS-R9825, CWI, Amsterdam (1998).
15. J. Verwer, E. J. Spee, J. G. Blom, and W. Hunsdorfer, A second order Rosenbrock method applied to photochemical dispersion problems, *SIAM J. Sci. Comput.* **20**, 1456 (1999).
16. J. G. Blom and J. Verwer, A comparison of integration methods for atmospheric transport-chemistry problems, *J. Comput. Appl. Math.*, to appear.