A POSITIVE NUMERICAL INTEGRATION METHOD
FOR CHEMICAL KINETIC SYSTEMS

ADRIAN SANDU

Abstract. Mass action chemical kinetics conserves mass and renders non-negative solutions; a good numerical simulation would ideally produce a mass balanced, positive numerical concentration vector. Many time stepping methods are mass conservative; however, unconditional positivity restricts the order of a traditional method to one. The positive projection method presented in the 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 using a primal-dual optimization routine; this vector is shown to better approximate the true solution. The technique works best when the underlying time-stepping scheme favors positivity. Projected methods are able to use larger integration time steps, being more efficient than traditional methods for systems which are unstable outside the positive quadrant.

Key words. Chemical kinetics, linear invariants, positivity, numerical time integration, primal-dual quadratic optimization.

* Department of Computer Science, 205 Fisher Hall, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931 (asandu@mtu.edu)
1. Introduction. Air quality models [3, 10] solve the convection-diffusion-reaction set of partial differential equations which model 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 (otherwise, non-physical sources and sinks may corrupt the quality of the solution); and keep the concentrations positive. Positivity is needed

- since concentrations are non-negative quantities;
- for stability of the chemical system;
- an operator-split solution of convection-diffusion-reaction atmospheric equations alternates chemical integration steps with advection steps; the results of each process need to be positive; 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, or even unstable solutions.

An extensively used method is fixed time-step integration plus clipping. Clipping means setting to zero the negative concentration values that appear during numerical integration of a chemical kinetics system. It enhances stability, but adds mass to the system, therefore corrupting mass balance.

There are numerical integration methods that preserve both mass and positivity, a well-known example being backward Euler. 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.

In this paper we try to alleviate the order and step size restrictions that come with positivity. The idea is to obtain a mass-balanced solution using a standard integration method; if some com-
ponents are negative, we “project” this solution back onto the reaction simplex. This means that we look for the mass-balanced, non-negative vector that is closest to our computed solution, and make it the next step approximation. Lemma 4.1 shows that, in fact, this is a better approximation of the true solution itself.

We believe the technique developed here is of interest not only for atmospheric chemistry, but also for general kinetic mechanisms, whenever non-linearity makes negative solutions unstable. One usually expects the step control mechanism to compensate for non-positivity and maintain stability; this is most often the case but there is an efficiency penalty: very small time steps may be called for. Projecting the solution back onto the reaction simplex (when needed) stabilizes the integration process and allows larger time steps.

The paper is organized as follows. Section 2 presents the chemical kinetic problem and its mass balance and positivity properties; the preservation of these properties by numerical schemes is discussed in Section 3, with more details given in Appendix A. Section 4 develops the proposed positivity-preserving algorithm. The optimization routine is briefly discussed in Section 5, and implementation details are provided in Appendix B. A test case from stratospheric chemistry is considered in Section 6, where different numerical results are analyzed. Finally, the findings and conclusions of the paper are summarized in Section 7.

2. Mass action kinetics, linear invariants and positivity. Consider a chemical kinetic system with \( s \) species \( y_1, \cdots, y_s \) interacting in \( r \) chemical reactions,

\[
\sum_{j=1}^{s} \ell_{ij} y_j \xrightarrow{k_i} \sum_{j=1}^{s} r_{ij} y_j , \quad i = 1, \cdots, r .
\]

Here \( \ell_{ij} \) is the stoichiometric coefficient of the reactant \( y_j \) in the chemical reaction \( i \). Similarly, \( r_{ik} \) is the stoichiometric coefficient of the product \( y_k \) in reaction \( i \) and on can build the matrices of
stoichiometric coefficients

\[ R = (r_{ij})_{ij}, \quad L = (l_{ij})_{ij}, \quad S = R - L \in \mathbb{R}^{s \times r}. \]

Let \( \omega(y) \in \mathbb{R}^r \) be the vector of reaction velocities,

\[ \omega_i(y) = k_i \prod_{j=1}^{s} (y_j)^{l_{ij}}, \quad i = 1, \ldots, r. \]

The time evolution of the system is governed by the “mass action kinetics” differential law

\[ y' = S \cdot \omega(y), \quad y(t_0) = y^0. \]

If \( e \in \ker S^T \) then \( S^T e = 0 \), or \( e^T S = 0 \), which implies that

\[ e^T y'(t) = 0 \iff e^T y(t) = \text{const}, \]

that is \( e \) is a linear invariant of the system. If \( \text{rank}(S) = s - m \), then the system admits \( m \) linearly independent invariants, \( e_1, \ldots, e_m \); consider the \( s \times m \) matrix \( A \) whose columns form a basis for the null space of \( S^T \), and the \( m \) dimensional vector \( b \) of “invariant values”, \( e_i^T y(t) = \text{const} = b_i \) for all \( i \),

\[ A = [e_1 | e_2 | \cdots | e_m], \quad b = [b_1, \ldots, b_m]^T. \]

Then the solution satisfies

\[ A^T y(t) = b = A^T y^0. \]

Simply stated, the existence of linear invariants ensures that mass is conserved during chemical reactions.

Let us now separate the production reactions from the destruction ones in (3). The mass action kinetic system (3) can be written in terms of the production terms \( P(y) \) and the destruction terms \( D(y) \)

\[ P(y) = R \omega(y), \quad D(y) = \text{diag} \left( \frac{[L \omega(y)]_1}{y_1}, \ldots, \frac{[L \omega(y)]_s}{y_s} \right), \quad y' = P(y) - D(y) y. \]
The special form of the reaction speeds (2) ensures that $D_i(y)$ are polynomials in $y$. If at some time moment $t = \tau$ the concentration of a certain species is zero, $y_i(\tau) = 0$, then its derivative is nonnegative, $y'_i(\tau) = P_i(\tau) \geq 0$, which implies that

$$y(t_0) \geq 0 \implies y(t) \geq 0, \text{ for all } t \geq t_0.$$  

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

Linear invariants and positivity mean that the solution of (3) remains all the time within the reaction simplex

$$y(t) \in \mathcal{S} \text{ for all } t \geq t_0, \quad \mathcal{S} = \left\{ y \in \mathbb{R}^n \text{ s.t. } A^Ty = b \text{ and } y_i \geq 0 \text{ for all } i \right\}.$$  

3. Numerical preservation of the linear invariants and of 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 (3) should therefore

- be unconditionally stable, as the system is usually stiff. This requires implicit integration formulas;
- preserve the linear invariants. One certainly wants to avoid artificial mass creation or destruction during integration;
- preserve solution positivity. Negative concentrations are non-physical; in addition, for negative concentrations the whole kinetic system may become unstable.

A simple example of negative concentrations and instability is provided by Verwer et al. [12]. The chemical reaction

$$C + C \xrightarrow{k} \text{ products}$$
gives the following time evolution of $C$

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

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 $t \to t_0 + 1/(-C_0 k)$.

Fortunately, the most popular integration methods (Runge-Kutta, Rosenbrock and Linear Multistep) preserve exactly\textsuperscript{1} all the linear invariants of the system. Moreover, the (modified) Newton iterations used to solve the nonlinear systems - which define implicit solutions - preserve exactly all linear invariants during each iteration. This is briefly reminded in Appendix A. Therefore the only errors in the linear invariants are given by roundoff. With linear-preserving integration methods the errors in the individual species concentrations are 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, e.g. $10^{-14}$).

Positivity of the numerical solution is much harder to achieve. Bolley and Crouzeix [2] showed that (in the linear case) ensuring positivity limits the order of the numerical method to one. Hundsdorfer [8] proves that the implicit Euler method is positivity preserving. In practice, even implicit Euler may produce negative values; the reason is that the implicit equations are solved with a modified Newton process, which is halted after a finite number of iterations; while the exact solution of implicit Euler equations is guaranteed to be non-negative, the successive approximations computed by (modified) Newton are not.

The most common method to avoid negative concentrations (and possible unstable behavior) is clipping. If the solution vector has negative components, they are simply set to zero. The problem with clipping is that it destroys the preservation of linear invariants; more exactly, the

\textsuperscript{1} If the computations are performed in infinite arithmetic precision.
linear invariants are corrupted by errors comparable in magnitude to the errors in the individual components; instead of roundoff magnitude we have truncation error magnitude. The situation is worsened by the fact that all clipping errors act in the same direction, namely increase mass; therefore they accumulate over time and may lead to significant global errors, especially if the time interval of interest is long. The net result is adding non-physical sources of material.

4. Solution projection method. Suppose the system (3) is solved with a linear-preserving numerical method $\Phi$. Let $t^n$ denote the discrete time value at $n^{th}$ step, $y^n$ the computed solution and $h^n = t^{n+1} - t^n$ the next step size. The numerical solution is represented as

$$y^{n+1} = \Phi (f, y^n \ldots y^{n-k}, t^n, h^n \ldots h^{n-k+1}) .$$

As noted before, we consider only methods which preserve linear invariants, i.e.

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

Positivity is not preserved, however, and at the new moment $t^{n+1}$ some numerical concentrations may become negative

$$y^{n+1}_i < 0 .$$

We want to perform “clipping” without modifying the linear invariants. The idea is to project the numerical solution $y^{n+1}$ back onto the simplex $\{ y_i \geq 0 \text{ for all } i \} \cap \{ A^T y = b \}$. 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 solution $y^{n+1}$.

We reformulate the clipping problem as a linearly constrained, quadratic optimization problem. Given the “computed value” $y^{n+1}$ find the “projected value” $z^{n+1}$ which

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

(6)
The norm used here is
\[ \|y\|_G = \sqrt{y^T G y}, \]
with \( G \) a positive definite matrix to be specified later.

Adjustable step ODE solvers compute the solution \( y^{n+1} \) together with an estimate of the (component-wise) truncation error \( e^{n+1} \). For example, a Runge-Kutta formula is paired with an “embedded formula” that gives a second, lower order approximation to the solution \( \tilde{y}^{n+1} \); the truncation error estimate is then \( e^{n+1} = y^{n+1} - \tilde{y}^{n+1} \). We will denote the error estimator by
\[ e^{n+1} = \Phi_E \left( f, y^n \ldots y^{n-k}, t^n, h^n \ldots h^{n-k+1} \right). \]

The step size control is based on user-prescribed relative (\( rtol \)) and absolute (\( atol \)) error tolerances; the following measure of the truncation error is computed
\[ E^{n+1} = \sqrt{\frac{1}{s} \sum_{i=1}^{s} \left( \frac{e_{i}^{n+1}}{\text{atol} + \text{rtol} |y_{i}^{n+1}|} \right)^2}. \]

The current value \( y^{n+1} \) is accepted if \( E^{n+1} < 1 \) and rejected otherwise. It can be easily seen that
\[ E^{n+1} = \|e^{n+1}\|_{G(y^{n+1})}, \text{ where } G(y) = \text{diag}_{1 \leq i \leq s} \left[ \frac{1}{s (\text{atol} + \text{rtol} |y_{i}|)^2} \right]. \]

Since 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 (6) in terms of the same \( G \)-norm; that is, we try to find \( z \) in the reaction simplex (5) that the projection-error norm \( \|z^{n+1} - y^{n+1}\|_{G(y^{n+1})} \) is minimal.

The computed solution \( y^{n+1} \) approximates the true solution \( y(t^{n+1}) \); 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 that

**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 \( Y = G^{1/2} y^{n+1} \), \( Z = G^{1/2} z^{n+1} \), \( W = G^{1/2} y(t^{n+1}) \). We have that

\[
A^T G^{-1/2} Y = b \quad \text{and} \quad W \in \mathfrak{S}, \quad \text{where} \quad \mathfrak{S} = \left\{ X : A^T G^{-1/2} X = b, \ X \geq 0 \right\}.
\]

The problem (6) is equivalent to

\[
\min \frac{1}{2} \|Z - Y\|_2^2 \quad \text{subject to} \quad Z \in \mathfrak{S}.
\]

Since \( \mathfrak{S} \) is a convex set, \( Y \notin \mathfrak{S} \), and \( Z \) is the point in \( \mathfrak{S} \) that is closest to \( Y \), the hyperplane perpendicular to the direction \( YZ \) and which passes through \( Z \) separates \( \mathfrak{S} \) and \( Y \). For any \( W \in \mathfrak{S} \) the angle \( (YZW) \) is larger than 90\(^\circ\), therefore \( YW \) is the largest edge in the triangle \( YZW \). In particular,

\[
\|Y - W\|_2 \geq \|Z - W\|_2 \quad \implies \quad \|y^{n+1} - y(t^{n+1})\|_G \geq \|y^{n+1} - y(t^{n+1})\|_G.
\]

Schematically, one step of the method reads

\[
\begin{align*}
\overline{y}^{n+1} & = \Phi (y^n, \ldots, y^{n-k}, t^n, h^n, \ldots, h^{n-k+1}) ; \\
G & = G(\overline{y}^{n+1}) ; \\
\text{IF} \{ \overline{y}_i^{n+1} < 0 \text{ for some } i \} \text{ THEN} & \\
\min \|z^{n+1} - \overline{y}^{n+1}\|_G \text{ subject to } A^T z^{n+1} = b, \ z^{n+1} \geq 0 ; \\
y^{n+1} & = z^{n+1} ; \\
\text{ELSE} & \\
y^{n+1} & = \overline{y}^{n+1} ; \\
\text{END IF}
\end{align*}
\]  

(7)

We will call this method the positive-projection method, since at each step the numerical solution is “projected” back onto the reaction simplex (5). A direct consequence of Lemma 4.1 is that the
consistency order of the projection method (7) is the order of the underlying time discretization $\Phi$, since projection step does not increase the truncation error. The same convergence analysis apply. In this particular sense the positive projection method (7) 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 the optimum in (6) is not attained, but rather we are content with a value $z^{n+1} \in S$ that is “reasonably close” to $y^{n+1}$, 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 in the presence of numerical errors, or if $y^{n+1}$ is “very wrong”, for example due to an exagerate step size) then reject the step and continue with a reduced step size.

5. The optimization algorithm. It only remains to find a suitable way of computing $z^{n+1}$, i.e. a way to solve the quadratic minimization problem (6). Note that the reaction simplex (5) is never empty - it contains at least the initial conditions. Therefore, the theoretical minimization problem (6) is always feasible - we compute the minimal distance between a point and a non-empty convex set, plus the closest point in the convex set. In principle, numerical errors may render the problem infeasible; in this case, the whole step is rejected, and the step size reduced.

The optimization algorithm needs to have the following properties:

- Find a solution or indicate that the problem is infeasible in a finite number of steps;
- Since the number of equality constraints is much smaller than the dimension of the system,
\( m \ll s \), we want to work with \( m \)-dimensional problems;

- A good starting point for the optimization routine is offered by \( \bar{y}^{n+1} \); we have \( A^T \bar{y}^{n+1} = b \) and the negative part \( (\bar{y}^{n+1})^{-} \) is small - of the order of truncation error. Therefore, we want to be able to initialize the optimization algorithm at an infeasible point.

The primal-dual algorithm of Goldfarb and Idnani [5] fulfills all the above criteria. This algorithm has been adapted to our particular problem (6), which contains equality constraints, and for which the inequality constraints have the simplest form possible.

Let \( K = \{1, \ldots, s + m\} \) be the set of constraints of problem (6) - \( \{1, \ldots, m\} \) are equality and \( \{m + 1, \ldots, m + s\} \) are inequality constraints. For a vector \( z \) with \( A^T z = b \) the set of active constraints is \( A = \{1, \ldots, m\} \cup \{m + i \mid z_i = 0\} \). The basic approach of the algorithm is [5]

\begin{align*}
\text{Step 0.} & \quad z^{n+1} = \bar{y}^{n+1}, \quad A = \{1, \ldots, m\} \\
\text{Step 1.} & \quad \text{Repeat until all constraints are satisfied} \\
(a) & \quad \text{Choose a violated constraint } p \in K - A \quad (z^{n+1}_{p-m} < 0) \\
(b) & \quad \text{If (6) with constraints } A \cup \{p\} \text{ infeasible STOP;} \\
(c) & \quad \text{Find unnecessary constraints } r \in A \text{ and set } A \leftarrow A \cup \{p\} - \{r\}; \\
(d) & \quad \text{Solve (6) with equality constraints } A \text{ only to get the new } z^{n+1}; \\
\text{Step 2.} & \quad \text{STOP, } z^{n+1} \text{ is the optimal solution.}
\end{align*}

A detailed presentation is given in Appendix B.
6. Numerical Results. Consider the basic stratospheric reaction mechanism (adapted from NASA HSRP/AESA [9])

\[ \begin{align*}
    &r_1 \quad O_2 + h\nu \xrightarrow{k_1} 2O \quad (k_1 = 2.643 \times 10^{-10} \cdot \sigma^3) \\
    &r_2 \quad O + O_2 \xrightarrow{k_2} O_3 \quad (k_2 = 8.018 \times 10^{-17}) \\
    &r_3 \quad O_3 + h\nu \xrightarrow{k_3} O + O_2 \quad (k_3 = 6.120 \times 10^{-04} \cdot \sigma) \\
    &r_4 \quad O + O_3 \xrightarrow{k_4} 2O_2 \quad (k_4 = 1.576 \times 10^{-15}) \\
    &r_5 \quad O_3 + h\nu \xrightarrow{k_5} O^{1D} + O_2 \quad (k_5 = 1.070 \times 10^{-03} \cdot \sigma^2) \\
    &r_6 \quad O^{1D} + M \xrightarrow{k_6} O + M \quad (k_6 = 7.110 \times 10^{-11}) \\
    &r_7 \quad O^{1D} + O_3 \xrightarrow{k_7} 2O_2 \quad (k_7 = 1.200 \times 10^{-10}) \\
    &r_8 \quad NO + O_3 \xrightarrow{k_8} NO_2 + O_2 \quad (k_8 = 6.062 \times 10^{-15}) \\
    &r_9 \quad NO_2 + O \xrightarrow{k_9} NO + O_2 \quad (k_9 = 1.069 \times 10^{-11}) \\
    &r_{10} \quad NO_2 + h\nu \xrightarrow{k_{10}} NO + O \quad (k_{10} = 1.289 \times 10^{-02} \cdot \sigma)
\end{align*} \]

Here \( \sigma(t) \) is the normalized sunlight intensity, equal to 1 at noon at equal to 0 at nighttime.

It is easy to see that along any trajectory of the system (9) the number of oxygen and the number of nitrogen atoms are constant,

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

therefore if we denote the concentration vector

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

the linear equality constraints have the matrix \( A \) below and a right hand side \( b \) given by the initial conditions

\[
    A^T = \begin{bmatrix}
        1 & 1 & 3 & 2 & 1 & 2 \\
        0 & 0 & 0 & 0 & 1 & 1
    \end{bmatrix} , \quad A^T y(t) = A^T y(t_0) = b .
\]
The simulation starts at noon and continues for 72 hours. The initial conditions are given in Table I. Figure 1 shows the evolution of concentrations in time; one notices three distinct diurnal cycles.

We implemented the numerical examples in MATLAB. Throughout the tests the minimal values in the optimization routine were set to $\epsilon_i = 1 \text{molec/cm}^3$ (which is, physically, a zero concentration, but numerically prevents the optimal solution to become negative due to roundoff). Reference solutions were obtained with the MATLAB integration routine ode15s (variable order numerical differentiation formula); the control parameters were $rtol = 10^{-6}$ and $atol = 10^{-2}$, with analytic Jacobian.

The integration is performed first with the BDF-2 method (13) started with a backward Euler first step. The stepsize of the integration is fixed, $h = 15 \text{min}$. If the concentration of species $i$ is small and drops sharply we may have that $Y_i^n < 0$, and also $f_i \leq 0$; Figure 2 shows that this is precisely what happens with the numerical concentrations of $O^{1D}$, $O$ and $NO$, which take negative values right after strong decreases.

For this example the original integration requires 506,691 floating point operations (flops), while the positive version needs 755,623, i.e. 50% more. In Figure 2 we see that the $O^{1D}$ concentration falls frequently below zero and is “projected” back to $1 \text{molec/cm}^3$; small positive concentrations, of the order $10^{-15} \text{molec/cm}^3$ (which are not corrected) alternate with small negative concentrations (which are corrected to $1 \text{molec/cm}^3$); this gives the pattern in the figure. To eliminate this curious behaviour we could correct all the values that fall below $1 \text{molec/cm}^3$, but this will require more calls to the optimization routine, and an increased overhead.

The global evolution of the system is not significantly altered by the projective clipping; Figure 1 presents the time evolution of the concentrations, as well as the drift in the invariants. The positive trajectory is not distinguishable from the uncorrected one; both reproduce very well the reference trajectory. The invariants relative errors $(a_i^T y - b_i)/b_i$, shown in the lower right plot, have
the same order of magnitude as the double precision roundoff errors.

Next, we integrate the system with RODAS-3 (16) with a fixed step \( h = 15 \text{ min} \). We do not give the actual solution plots, but will compare the results with other methods later on.

Finally, we integrate the system with ROS-2 (17), fixed step \( h = 15 \text{ min} \). Verwer et. al. [13, 1] advocated the favorable positivity properties of this method. The concentration evolution is similar to the ones obtained with BDF-2. The negative concentrations obtained are presented in Figure 3. It is clear from the figure that this method seldom produces negative concentrations. Therefore, the optimization routine is called fewer times and the overhead is considerably smaller. Note that, although \( O^{1D} \) is the only negative concentration produced, the optimization routine also adjusts the values of \( NO \) and \( O \) (which are positive but smaller than \( \epsilon_i = 1 \text{ molec/cm}^3 \)).

The minimal concentrations throughout the trajectories are given in Table II. Note that for ROS-2 only \( O^{1D} \) becomes negative its minimal value \(-5E-315\) suggests that this is the result of roundoff rather then algorithmic error. This conclusion is further strengthen by the results obtained with the equivalent formulation (18), which does not produce negative values at all (the minimal concentration of \( O^{1D} \) along the trajectory is \(+5.8E-36\)). In conclusion, ROS-2 favours positivity (more details are presented in Appendix A).

The number of floating point operations needed for the standard and the positive versions of the two codes to complete the integration are given in Table III. The conclusion is that the projective method has to be paired with a numerical scheme that favors positivity in order to reduce the overhead; a suitable integration method is ROS-2.

Note that in the reduced stratospheric system (9) the destruction reactions of the “possibly-negative” species \( O, O^{1D} \) and \( NO \) involve just 1 molecule of them; for example, the destruction term for oxygen is \(-D_{[O]}[O]\), where \( D_{[O]} \) does not depend on any “possibly-negative” concentration

\[
D_{[O]} = k_2[O_2] + k_4[O_3] + k_9[NO_2] ;
\]
this means $D_{[O]} > 0$ throughout the simulation, and for $[O] < 0$ the destruction term $-D_{[O]}[O] > 0$ 
(both production and destruction reactions now produce $O$). Since $y_i < 0 \Rightarrow f_i > 0$ when $y_i$
becomes negative its derivative becomes positive, and the system pulls itself back to the quadrant 
$\{y \geq 0\}$; this auto-correction the trajectory in case of negative values and explains why the standard 
and projected trajectories are undistinguishable.

Not all chemical systems have the auto-correction property; in fact, the projection technique
is useful for systems which do not autocorrect. Consider the kinetic scheme (9) and append the 
extra reaction

(10) \[ r11) \quad NO + O \xrightarrow{k_{11}} NO_2 \quad (k_{11} = 1.0E - 8) \]

In this case there is no auto-correction for negative values; reaction (10) will continue to destroy
$NO$ and $O$ even when their concentrations become negative.

The results of integrating the extended system (9)–(10) with the fixed-step ROS-2, $h = 15$
min, are shown in Figure 4. The standard solution “explodes”. This behavior can be explained
as follows: at some moment the concentrations $NO, O$ become negative reaction (10) continues at
an accelerated pace, which will trigger the accumulation of $NO_2$, etc. In contrast, the positively-
projected solution is bounded, and is very close to the exact solution. This is shown in Figure
5.

Additional experiments (not shown here) revealed that BDF-2 with $h = 15$ min also has
unstable behavior. For $h = 5$ min both BDF-2 and ROS-2 produce stable trajectories. RODAS-3
produces stable, but very distorted trajectories for $h = 15$ min, and we have to go down to $h = 5$
min to obtain an accurate picture. Positively-projected RODAS-3 produces reasonably accurate
trajectories for $h = 15$ min.

An accurate solution is also produced by the variable-step ROS-2 with the modest error toler-
ances $rtol = 0.01$, $atol = 0.01$ molec/cm$^3$; however, the chosen step-sizes are quite small (average
Table IV shows the number of floating point operations needed by each method; in particular, the variable step version is 12 times more expensive than the fixed step \((h = 30 \text{ min})\) positively-projected version. The positive-variable-step ROS-2 gives almost identical results for \(rtol = 0.01\); for the larger \(rtol = 0.1\) the positive-variable-step chooses an average time step of 3.8 min, vs. 3.45 min for the standard version. In many atmospheric simulations it is customary to impose a minimal step size, for example \(h \geq h_{\text{min}} = 1 \text{ min}\). Experiments not reported here showed that if a minimal step size is used positive projection can improve the quality of the solution.

At this point it is natural to ask whether simple clipping can stabilize the solution at a lower cost than the full projection. Figure (6) shows the exact evolution of \(NO_2\) for the extended system (9)–(10), the evolution with clipping and with projection, for step sizes \(h = 15 \text{ min}\) and \(h = 30 \text{ min}\). Indeed, clipping stabilizes the solution, but it also introduces large errors. The clipped \(NO_2\) concentration is larger than the “exact” one and the error is increasing in time, as expected. For the larger time step the \(NO_2\) concentration increases to \(1.2 \times 10^{13} \text{ molec/cm}^3\), orders of magnitude away from the reference solution. Large errors accumulate in the linear invariants also; for the smaller step the accuracy drops from \(10^{-15}\) to \(10^{-1}\) and continues to degrade; for the larger any meaningful information in the invariants is lost. The \(NO_2\) values calculated with the projection method are extremely close to the reference values; in addition the invariant errors are very small, and remain constant in time.

Lumping does not improve the stability problem either. 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 the one of the non-lumped system.

7. Conclusions. We have presented a technique that ensures mass conservation and positivity for the numerical solutions of mass action kinetics initial value problems. The first numerical
approximation is computed with one step of a linear-invariant-preserving numerical method. If there are negative components we find the nearest vector in the reaction simplex (5) using a primal-dual optimization routine. The optimal vector better approximates the true solution (Lemma 4.1) and provides the next-step numerical solution.

Standard numerical methods that guarantee positivity for any stepsize, like backward Euler, are first order at most; higher order methods can be positive only for very small step sizes [2]. The positive projection method overcomes this barrier; its order of consistency is given by the order of the underlying time stepping scheme, while the solutions are guaranteed to be non-negative. In practice the technique alleviates the step size restrictions when higher order integration methods are used, and renders a more efficient integration process.

The projection technique has to be paired with a positivity-favourable method, for example ROS-2 [13, 1]. Although such methods do not guarantee positivity, they seldom produce non-positive results; this minimizes the overhead incurred by the optimization routine.

Variable step can ensure stability even when the solution becomes non-positive; sometimes very small steps are called for. At least for the example considered, projection does not seem to improve significantly the performance of the standard variable-step algorithm; however, improvements are visible when a minimal step is imposed.

In air quality modeling to use fixed step integration and clipping is a popular approach; clipping, however, constitutes a non-physical mass source which may adversely impact the quality of the solution. Fixed step plus simplex projection ensures both positivity and mass balance. In the example presented large values of the step size made the original solution unstable; the clipped solution was corrupted by large errors, while the projected solution remained very similar to the reference one. When the accuracy requirements are modest, projection may be a computationally cheaper alternative to variable step size. In addition, in a parallel implementation of an air quality
model fixed step sizes lead to a better load balance and increased overall efficiency.

Some kinetic systems have the property of self-correcting negative concentrations; for them the projection technique does not seem necessary; for kinetic systems which become unstable if some concentrations are negative the technique is useful and improves the standard solution.

An apparent disadvantage 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. [4].

**Appendix A. Numerical integration methods.** Consider the differential system (3) written in the general form

\[
y'(t) = f(t, y), \quad y(t_0) = y^0.
\]

(11)

The system has the linear invariants

\[
A^T y(t) = A^T y^0 = b = \text{const}, \quad A^T f(t, y) = 0, \quad A^T J = 0 \ (J = \partial f / \partial y).
\]

In what follows we will denote by \( y^n \) the numerical solution at \( n \)-th step (and time \( t^n \)), and \( h = t^{n+1} - t^n \) is the current time step. A linear \( k \)-step method [6, 7] is given by the general formula

\[
y^{n+1} + \sum_{i=0}^{k-2} \alpha_i y^{n-i} = h \sum_{i=-1}^{k-2} \beta_i f\left(t^{n-i}, y^{n-i}\right).
\]

Multiplying the relation by \( A^T \) gives

\[
A^T y^{n+1} + \sum_{i=0}^{k-2} \alpha_i A^T y^{n-i} = 0.
\]

If all previous approximations satisfy the invariant **then** the new solution also does,

\[
A^T y^{n-i} = b, \quad \text{for } i = 0, \cdots, k-2 \quad \implies \quad A^T y^{n+1} = b,
\]

since consistency requires \( \sum_{i=0}^{k-2} \alpha_i = -1 \).
The simplest example is the (first order) backward Euler formula,

\[ y^{n+1} = y^n + h f \left( t^{n+1}, y^{n+1} \right) . \]  

(12)

To solve this nonlinear equation one uses modified Newton iterations,

\[ (I - hJ(t^n, y^n)) \delta^{n+1} = f \left( t^{n+1}, y^{n+1} \right), \quad y^{n+1}_{\text{new}} = y^{n+1}_{\text{old}} - \delta^{n+1} . \]

Left-multiplying by \( A^T \) shows preservation of linear invariants

\[ A^T J(t^n, y^n) = 0, \quad A^T \delta^{n+1} = 0, \quad A^T y^{n+1}_{\text{new}} = A^T y^{n+1}_{\text{old}} . \]

Another example is the second order backward differentiation formula (BDF-2)

\[ y^{n+1} = Y^n + \frac{2}{3} h f \left( t^{n+1}, y^{n+1} \right), \quad Y^n = \frac{4}{3} y^n - \frac{1}{3} y^{n-1} . \]  

(13)

For variable timesteps the coefficients change. The very first step requires both \( y^0 \) and \( y^1 \); the former is given, while the latter is obtained with one backward Euler step.

An \( s \)-stage Runge-Kutta method [6] defines the solution by the formula the solution is given by the formulas

\[ y^{n+1} = y^n + \sum_{i=1}^{s} b_i k_i, \quad k_i = h f \left( t^n + c_i h, y^n + \sum_{j=1}^{s} a_{ij} k_j \right) \]  

(14)

where \( s \) and the formula coefficients \( b_i, c_i, a_{ij} \) are chosen for a desired order of consistency and stability for stiff problems. Multipling the definition by \( A^T \) gives

\[ A^T k_i = 0, \quad A^T y^{n+1} = A^T y^n , \]

which shows that linear invariants are preserved during each step.

An \( s \)-stage Rosenbrock method [7] is usually defined for autonomous systems; the solution is given by the formulas

\[ y^{n+1} = y^n + \sum_{i=1}^{s} b_i k_i, \quad k_i = h f \left( y^n + \sum_{j=1}^{i-1} \alpha_{ij} k_j \right) + hJ \sum_{j=1}^{i} \gamma_{ij} k_j, \]  

(15)
where $s$ and the formula coefficients $b_i, \alpha_{ij}$ and $\gamma_{ij}$ give the order of consistency and the stability properties. Again, multiplying the definition by $A^T$ shows the preservation of the linear invariants,

$$A^T k_i = 0, \quad A^T y^{n+1} = A^T y^n.$$ 

An example Rosenbrock method is Rodas-3 [11] which, in autonomous form, reads

$$(16) \quad y^{n+1} = y^n + 2k_1 + k_3 + k_4,$$

$$
\begin{align*}
(\frac{1}{\gamma h} I - J) k_1 &= f(y^n), \\
(\frac{1}{\gamma h} I - J) k_2 &= f(y^n) + \frac{4}{h} k_1, \\
(\frac{1}{\gamma h} I - J) k_3 &= f(y^n + 2k_1) + \frac{1}{h} k_1 - \frac{1}{h} k_2, \\
(\frac{1}{\gamma h} I - J) k_4 &= f(y^n + 2k_1 - k_3) + \frac{1}{h} k_1 - \frac{1}{h} k_2 - \frac{8}{3h} k_3,
\end{align*}$$

where $\gamma = 1/2$.

Another example of Rosenbrock scheme is ROS-2 [13, 1]

$$(17) \quad y^{n+1} = y^n + \frac{3}{2} k_1 + \frac{1}{2} k_2,$$

$$
\begin{align*}
(\frac{1}{\gamma h} I - J) k_1 &= \frac{1}{\gamma} f(y^n), \\
(\frac{1}{\gamma h} I - J) k_2 &= \frac{1}{\gamma} f(y^n + k_1) - \frac{2}{h} k_1,
\end{align*}$$

with $\gamma = 1 + 1/\sqrt{2}$. An equivalent formulation for ROS-2 can be given,

$$(18) \quad y^{n+1} = y^n + \frac{3\gamma}{2} k_1 + \frac{\gamma}{2} k_2,$$

$$
\begin{align*}
(\frac{1}{\gamma h} I - J) k_1 &= f(y^n), \\
(\frac{1}{\gamma h} I - J) k_2 &= f(y^n + k_1) - \frac{2}{h} k_1.
\end{align*}$$

In [13, 1] it was noted that ROS-2 have favorable positivity properties, and the method is stable for nonlinear problems even with large fixed step sizes. It was also noted that ROS-2 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
the view of the theory developed in [2] this might reduce the the negative values and have a good influence on the positivity of solutions.

Appendix B. The optimization algorithm. Consider the optimization problem (6), reformulated as

\[
\begin{equation}
\min \frac{1}{2} z^T G z - y^T G z \quad \text{subject to} \quad A^T z = b , \ z \geq \epsilon .
\end{equation}
\]

The problem has \( m \) equality constraints \((a_i^T z = b_i, \ i = 1 \cdots m, \text{where} \ a_i \text{is the} \ i^{th} \text{column of} \ A)\) and \( s \) inequality constraints \((z_i \geq \epsilon_i \text{for} \ i = 1 \cdots s)\). The entries \( \epsilon_i > 0 \) are small positive numbers; their role is to keep \( z_i \geq 0 \) even when the computation is corrupted by roundoff. We assume that

- \( A^T \) has full row rank, and that
- \( A^T y = b \) (\( y \) satisfies the equality constraints).

The original algorithm of Goldfarb and Idnani is presented in [5]. We modified this algorithm in order to

- accomodate the equality constraints \( A^T y = b \) at all times, given that the starting point satisfies them;
- take advantage of the diagonal form of \( G \);
- take advantage of the special form of inequality constraints, \( z_i \geq \epsilon_i \);

The algorithm of Goldfarb and Idnani preserves at all times dual feasibility. The algorithm is started from the unconstrained optimum \( z = y \), which violates primal inequality constraints. Let \( \mathcal{K} = \{1, \cdots, s + m\} \) be the set of constraint indices. For a vector \( z \) with \( A^T z = b \) the set of active constraints is \( \mathcal{A} = \{1, \cdots, m\} \cup \{m + i | z_i = \epsilon_i\} \); we will denote by \( q \) the number of active constraints. Let \( N \) be the matrix of normal vectors of active constraints; for example, if \( z_i = \epsilon_i \) and \( z_j = e_j \) \((q = m + 2)\) this matrix is

\[
N = [a_1 | \cdots | a_m | \epsilon_i | e_j] \in \mathbb{R}^{s \times q} .
\]
Here and below $e_i$ is the unit vector with entry $i$ equal to 1 and all other entries equal to 0. The active constraints are expressed as

$$N^Tz = \begin{bmatrix} b \\ 0 \end{bmatrix} \begin{bmatrix} m \\ 2 \end{bmatrix}.$$

Consider $N^*$, the generalized inverse of $N$ in the space of scaled variables $Z = G^{1/2}z$, and the reduced inverse Hessian operator $H$

$$N^* = (N^T G^{-1} N)^{-1} N^T G^{-1}, \quad H = G^{-1} - G^{-1} N (N^T G^{-1} N)^{-1} N^T G^{-1}.$$

At each step the algorithm looks for a violated inequality constraint $z_p < \epsilon_p$. A step of the form

$$z^+ = z + tH e_p$$

will preserve all the active constraints, $N^T H = 0 \implies N^T z^+ = N^T z$. The step (20) can move the current vector toward satisfying the constraint. We have

$$z_p^+ = e_p^T z^+ = e_p^T z + t e_p^T H e_p = z_p + t H_{p,p}$$

and if $H_{p,p} > 0$ we can choose $t$ such that $z_p^+ = \epsilon_p$. This becomes a new active constraint and is added to the set, $N \leftarrow [N|e_p]$.

If $z$ is the optimum solution with respect to the given set of active constraints, the Lagrange multipliers satisfy

$$\lambda(y) = N^* G(z - y).$$

The multipliers corresponding to inequality constraints have to be non-negative, if not, the active constraint corresponding to a negative multiplier is removed (this means that $z_k = \epsilon_k$ is not optimal, and we remove the constraint, allowing $z_k > \epsilon_k$).
The implementation form differs somewhat from this presentation, for efficiency and stability reasons. Define the matrices $R \in \mathbb{R}^{q \times q}$ and $J \in \mathbb{R}^{s \times s}$ via the following QR decomposition:

\[
G^{-1/2} N = QR = \begin{bmatrix}
Q_1 & Q_2
\end{bmatrix}
\begin{bmatrix}
R
0
\end{bmatrix}_{q \times \mathbb{R}}, \quad J = G^{-1/2} Q = \begin{bmatrix}
J_1 & J_2
\end{bmatrix}_{q \times \mathbb{R}^{s-q}}.
\]

Step 0. Initializations.

\[
z \leftarrow y;
\]

\[
\mathcal{A} \leftarrow \{1, \ldots, m\}; q \leftarrow m;
\]

\[
\lambda = [0 \ldots 0] \text{ (Lagrange multipliers)};
\]

\[
N \leftarrow A;
\]

Compute $R, J$ by (21).

Step 1. Choose a violated constraint (if any). Is $z_p < \epsilon_p$ for some $p$?

\[
\text{IF } \{\text{no}\} \text{ THEN}
\]

\[
z \text{ is the optimal solution. STOP}
\]

\[
\text{ELSE } \{\text{if yes}\}
\]

\[
\lambda^+ \leftarrow [\lambda, 0] \text{ (consider the violated constraint } m + p\text{)}
\]

END IF
Step 2.

(a) Determine primal and dual step directions

\[ \delta_{\text{primal}} = J(1: s, q + 1: s) \cdot J(p, q + 1: s)^T \]

\[ R(1: q, 1: q) \cdot \delta_{\text{dual}} = J(p, 1: q)^T \] (triangular system)

(b) Max dual step length (which maintains dual feasibility)

\[
\begin{align*}
\text{IF } & \{ \delta_{\text{dual}} \leq 0 \text{ or } q = m \} \text{ THEN} \\
\tau_{\text{dual}} & = \infty \\
\text{ELSE} \\
\tau_{\text{dual}} & = \min_{j=m+1:q; \delta_{\text{dual}} > 0} \left\{ \lambda_j^+ / \delta_{\text{dual}}^j \right\} = \lambda_k^+ / \delta_k^{\text{dual}} \\
\text{END IF}
\end{align*}
\]

Primal step length (which makes \( z_p = \epsilon_p \))

\[
\begin{align*}
\text{IF } & \{ \delta_{\text{primal}} = 0 \} \text{ THEN} \\
\tau_{\text{primal}} & = \infty \\
\text{ELSE} \\
\tau_{\text{primal}} & = (\epsilon_p - z_p) / \delta_p^{\text{primal}} \\
\text{END IF}
\end{align*}
\]

Step length

\[ t = \min \left( \tau_{\text{primal}}, \tau_{\text{dual}} \right) \]
Step 3.

(a) Check Infeasibility

\[
\text{If } \{ (\tau^{\text{primal}} = \infty) \text{ and } (\tau^{\text{dual}} = \infty) \} \text{ THEN}
\]

The problem is infeasible. STOP.

END IF

(b) Primal Infeasible, step in dual space only

\[
\text{IF } \{ \tau^{\text{primal}} = \infty \text{ (Primal Infeasible) } \} \text{ THEN}
\]

\[
\lambda^+ \leftarrow \lambda^+ + t[\delta_{\text{dual}}, 1]
\]

Drop \( k - m \)th inequality constraint:

\[
\mathcal{A} \leftarrow \mathcal{A}\setminus\{k\}; \ q \leftarrow q - 1
\]

Update \( R, J \) for newly dropped constraint

Go to Step 2(a).

END IF

(c) Step in both primal and dual space

\[
z \leftarrow z + t \delta^{\text{primal}}
\]

\[
\lambda^+ \leftarrow \lambda^+ + t [\delta_{\text{dual}}, 1]
\]

\[
\text{IF } \{ t = \tau^{\text{primal}} \text{ (Full Step) } \} \text{ THEN}
\]

\[
\lambda \leftarrow \lambda^+
\]

Add \( p \)th inequality constraint:

\[
\mathcal{A} \leftarrow \mathcal{A}\cup\{m + p\}, \ q \leftarrow q + 1;
\]

Update \( R, J \) for newly added constraint;

Go to Step 1.
ELSE IF \{ t = \tau^{\text{dual}} \text{ (Partial Step) } \} THEN

Drop $k - m$th inequality constraint:

\[ A \leftarrow A \setminus \{k\}; q \leftarrow q - 1 \]

Update $R$, $J$ for newly dropped constraint;

Go to Step 2(a).

END IF

When a constraint is added, the matrices $R$ and $J$ are modified to accommodate it; for clarity, in what follows the updated versions will be marked by the superscript $(\cdot)$.

Adding the active constraint $\{z_p = e_p\}$ leads to the following changes.

\[
d = (J(p,:))^T = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \begin{bmatrix} q \\ s - q \end{bmatrix}
\]

\[
U = \text{Householder reflector s.t. } Ud = \begin{bmatrix} d_1 \\ \sigma \\ 0 \end{bmatrix} \begin{bmatrix} q \\ 1 \\ s - q - 1 \end{bmatrix}
\]

\[
R^+ \leftarrow \begin{bmatrix} R & d_1 \\ 0 & \sigma \end{bmatrix} \in \mathbb{R}^{(q+1) \times (q+1)}
\]

\[
J^+ \leftarrow \begin{bmatrix} J_1 \\ q \\ J_2U^T \\ s - q \end{bmatrix} = \begin{bmatrix} J_1^+ \\ q+1 \\ J_2^+ \\ s-q-1 \end{bmatrix}
\]

When a constant is dropped, the matrices $R$ and $J$ are updated; for clarity, we marked the updated versions by the superscript $(\cdot).$ Dropping the active constraint $\{z_\ell = e_\ell\}$ means deleting the $k = m + \ell$th constraint.

\[
R = \begin{bmatrix} R_1 & r_1^1 & S \\ 0 & r_2^2 & T \end{bmatrix} \Rightarrow R^+ = \begin{bmatrix} R_1 & S \\ 0 & T \end{bmatrix} \left( \begin{array}{c} \text{Delete the } k^{\text{th}} \text{ column of } R \end{array} \right)
\]
\[ UT = \begin{bmatrix} R_2 \\ 0 \end{bmatrix} \] (\( T \) Hessenberg, compute easily its QR decomposition)

\[ R^+ \leftarrow \begin{bmatrix} R_1 & S \\ 0 & R_2 \end{bmatrix} \in \mathbb{R}^{(q-1) \times (q-1)} \]

\[ J = \begin{bmatrix} J_1^a & J_2^a \\ q & s-q \end{bmatrix} = \begin{bmatrix} J_1^b & J_2^b \\ \ell-1 & q-\ell+1 \end{bmatrix} \]

\[ J^+ \leftarrow \begin{bmatrix} J_1^b U^T & J_2^b U^T \\ q-\ell+1 & s-q \end{bmatrix} \begin{bmatrix} J_1^c & J_2^c \\ s-q & s-q+1 \end{bmatrix} = \begin{bmatrix} J_1^- & J_2^- \\ q-1 & s-q+1 \end{bmatrix} \]

To compute quantities of the form \( J_2 U^T, J_1^b U^T \) note that \( U \) is a composition of Householder reflectors, \( U = U_n \cdots U_1 \); they are applied in succession to the rows of \( J_2 \), \( J_2 U^T = (U_1 \cdots U_n J_2^T)^T \).

**Acknowledgements.** I thank Mihai Anitescu for a fruitful discussion regarding quadratic optimization algorithms and the approximation properties of the solution.

**REFERENCES**


Fig. 1. Concentration evolution for BDF-2 integration, $h = 15$ min; there is virtually no difference between the standard and the positively-projected solutions. The lower right plot shows relative errors in the invariants, $(a_i^T y - b_i)/b_i$.

Fig. 2. Zoom in on concentration evolution with BDF-2, fixed $h = 15$ min. Upper figure shows the original solution, and lower figure shows the positively-projected solution.

Fig. 3. Zoom in on concentration evolution with ROS-2, fixed $h = 15$ min. Upper figure shows the original solution, and lower figure shows the projected solution.

Fig. 4. Concentration evolution for the extended system (9)–(10). Integration with ROS-2, fixed stepsize $h = 15$ min, standard vs. positive methods. Note the unstable behavior of the standard solution.

Fig. 5. Concentration evolution for the extended system (9)–(10). Integration with ROS-2, fixed stepsize $h = 15$ min, positive method.

Fig. 6. NO$_2$ concentration for the extended system (9)–(10) integrated with ROS-2, fixed stepsizes $h = 15$ min and $h = 30$ min. The projection solution is very close to the reference one, while clipping introduces errors which accumulate in time. For the larger step all meaningful information is lost.
Fig. 1. Concentration evolution for BDF-2 integration, $h = 15$ min; there is virtually no difference between the standard and the positively-projected solutions. The lower right plot shows relative errors in the invariants, $(a_i^T y - b_i)/b_i$. 
**Fig. 2.** Zoom in on concentration evolution with BDF-2, fixed $h = 15\text{min}$. Upper figure shows the original solution, and lower figure shows the positively-projected solution.
Fig. 3. Zoom in on concentration evolution with ROS-2, fixed $h = 15$ min. Upper figure shows the original solution, and lower figure shows the projected solution.
Fig. 4. Concentration evolution for the extended system (9)-(10). Integration with ROS-2, fixed stepsize $h = 15\text{ min}$, standard vs. positive methods. Note the unstable behavior of the standard solution.
Fig. 5. Concentration evolution for the extended system (9)-(10). Integration with ROS-2, fixed stepsize $h = 15$ min, positive method.
Fig. 6. NO$_2$ concentration for the extended system (9)-(10) integrated with ROS-2, fixed stepsizes $h = 15$ min and $h = 30$ min. The projection solution is very close to the reference one, while clipping introduces errors which accumulate in time. For the larger step all meaningful information is lost.
<table>
<thead>
<tr>
<th>$O^{1D}$</th>
<th>$O$</th>
<th>$O_3$</th>
<th>$O_2$</th>
<th>$NO$</th>
<th>$NO_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.906E+01</td>
<td>6.624E+08</td>
<td>5.326E+11</td>
<td>1.697E+16</td>
<td>8.725E+08</td>
<td>2.240E+08</td>
</tr>
</tbody>
</table>

**Table I**

*Initial concentrations for the simulation (molec/cm$^3$).*
### Table II

<table>
<thead>
<tr>
<th>Integrator</th>
<th>Min $[O^{1D}]$</th>
<th>Min $[O]$</th>
<th>Min $[NO]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDF-2</td>
<td>$-1.1E - 9$</td>
<td>$-2.2$</td>
<td>$-4.3E + 6$</td>
</tr>
<tr>
<td>RODAS-3</td>
<td>$-2.0E - 7$</td>
<td>$-16.5$</td>
<td>$-9.8E + 5$</td>
</tr>
<tr>
<td>ROS-2</td>
<td>$-5.5E - 315$</td>
<td>$+2.9E - 112$</td>
<td>$+4.2E - 21$</td>
</tr>
</tbody>
</table>

Minimal concentrations (molec/cm$^3$) for the stratospheric test problem with various integration methods. The step size is fixed $h = 15$ min.
The number of flops for integrating the small stratospheric test problem with various algorithms using both the standard and the positive versions. The step size is fixed $h = 15$ min.

<table>
<thead>
<tr>
<th>Integrator</th>
<th>Standard</th>
<th>Positive</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDF-2</td>
<td>506 Kflops</td>
<td>755 Kflops</td>
<td>50%</td>
</tr>
<tr>
<td>RODAS-3</td>
<td>652 Kflops</td>
<td>982 Kflops</td>
<td>50.6%</td>
</tr>
<tr>
<td>ROS-2</td>
<td>485 Kflops</td>
<td>502 Kflops</td>
<td>3.6%</td>
</tr>
</tbody>
</table>
The number of flops for integrating the small stratospheric test problem with ROS-2, fixed step size $h = 15$ min for clipping and projection, and variable step size.

<table>
<thead>
<tr>
<th>Integrator</th>
<th>Clipping</th>
<th>Projection</th>
<th>Variable step</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROS-2</td>
<td>$h = 15$ min</td>
<td>$h = 30$ min</td>
<td>$h = 15$ min</td>
</tr>
<tr>
<td></td>
<td>543 Kflops</td>
<td>272 Kflops</td>
<td>566 Kflops</td>
</tr>
</tbody>
</table>

Table IV