Positive numerical integration methods for chemical kinetic systems

Adrian Sandu

CSTR-9905
December 1999

Michigan Technological University, Houghton, MI 49931

POSITIVE NUMERICAL INTEGRATION METHODS 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 then 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 [4, 11] 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.

A general analysis of conservation laws and the numerical solutions of ordinary differential equations is given by Shampine [13]. The author considers "perturbation methods" where the computed solution is perturbed to satisfy exactly the desired invariants. The methods of this paper belong to this "perturbation" category, but they are specific to systems with linear equality and inequality invariants, e.g. chemical kinetic models.

There are numerical integration methods that automatically preserve mass and positivity, a well-known example being backward Euler. However, as shown by Bolley and Crouzeix [3], 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 components 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 simpler technique to stabilize the reaction simplex is discussed in Section 6. A test case from stratospheric chemistry is considered in Section 7, where different numerical results are analyzed. 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 $y_1, \cdots, y_s$ 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 ℓ_{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

$$R = (r_{ij})_{ij} \ , \quad L = (\ell_{ij})_{ij} \ , \quad S = R - L \in \Re^{s\times r} \ . \tag{1}$$

Let $\omega(y) \in \Re^r$ be the vector of reaction velocities,

$$\omega_i(y) = k_i \prod_{j=1}^{s} (y_j)^{\ell_{ij}} \ , \quad i = 1, \cdots, r \ . \tag{2}$$

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

$$y' = S \cdot \omega(y) \ , \quad y(t_0) = y^0 \ . \tag{3}$$

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

$$e^T y'(t) = 0 \quad \Longleftrightarrow \quad e^T y(t) = \text{const} \ ,$$

that is e is a linear invariant of the system. If rank(S)=$s - m$, then the system admits m linearly independent invariants, e_1, $\cdots$, 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,

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

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 $D(y)$

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

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

$$y(t_0) \geq 0 \quad \Longrightarrow \quad y(t) \geq 0 \ , \quad \text{for all } t \geq t_0 \ . \tag{4}$$

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

$$y(t) \in \mathcal{S} \text{ for all } t \geq t_0 \ , \quad \mathcal{S} = \left\{ y \in \Re^s \text{ s.t. } A^T y = b \text{ and } y_i \geq 0 \text{ for all } i \right\} \ . \tag{5}$$

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 concetrations the whole kinetic system may become unstable.

A simple example of negative concentrations and instability is provided by Verwer et. al. [14]. The chemical reaction

$$C + C \xrightarrow{k} \text{products}$$

gives the following time evolution of C

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

Note that if $C_0 > 0$ the solution is bounded ($0 \le C(t) \le C_0$) 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[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 [3] showed that (in the linear case) ensuring positivity limits the order of the numerical method to one. Hundsdorfer [9] 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 succesive 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 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.

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

4. Solution projection method. Suppose the system (3) is solved with a linear-preserving numerical method Φ. 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

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

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

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

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

$$\min \frac{1}{2} \left\| z^{n+1} - y^{n+1} \right\|_G^2 \quad \text{subject to} \quad A^T z^{n+1} = b \ , \ z^{n+1} \geq 0 \ . \tag{6}$$

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

$$e^{n+1} = \Phi_E \left(f, y^n \cdots y^{n-k}, t^n, h^n \cdots h^{n-k+1} \right) \ .$$

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

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

The current value y^{n+1} is accepted if $E^{n+1} < 1$ and rejected otherwise. It can be easily seen that

$$E^{n+1} = \left\| e^{n+1} \right\|_{G(y^{n+1})} \ , \quad \text{where} \quad G(y) = \text{diag}_{1 \leq i \leq s} \left[\frac{1}{s\ (atol + rtol\, |y_i|)^2} \right] \ .$$

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 $\|z^{n+1} - y(t^{n+1})\|_G \le \|y^{n+1} - y(t^{n+1})\|_G + \|z^{n+1} - y^{n+1}\|_G$. A closer look reveals that

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

$$\left\|z^{n+1} - y(t^{n+1})\right\|_G \le \left\|y^{n+1} - y(t^{n+1})\right\|_G$$

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

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

The problem (6) is equivalent to

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

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

$$\|Y - W\|_2 \ge \|Z - W\|_2 \quad \Longrightarrow \quad \|y^{n+1} - y(t^{n+1})\|_G \ge \|y^{n+1} - y(t^{n+1})\|_G\ .$$

□

Schematically, one step of the method reads

(7)
```
$\overline{y}^{n+1} = y^{n+1} = \Phi_h^f\,(y^n)$ ;
$G = G(\overline{y}^{n+1})$ ;
IF { $\overline{y}_i^{n+1} < 0$ for some $i$ } THEN
    $\min \|z^{n+1} - \overline{y}^{n+1}\|_G$ subject to $A^T z^{n+1} = b$ , $z^{n+1} \ge 0$;
    $y^{n+1} = z^{n+1}$ ;
ELSE
    $y^{n+1} = \overline{y}^{n+1}$ ;
END IF
```

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 Φ, 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 [3].

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 contempt with a value $z^{n+1} \in \mathcal{S}$ that is "reasonably close" to $\overline{y}^{n+1}$, 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 $\overline{y}^{n+1}$ 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 intial 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 $\overline{y}^{n+1}$; we have $A^T\overline{y}^{n+1} = b$ and the negative part $\left(\overline{y}^{n+1}\right)^-$ 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 [6] fullfills 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 $\mathcal{K} = \{1, \cdots, s+m\}$ be the set of constraints of problem (6) - $\{1, \cdots, m\}$ are equality and $\{m+1, \cdots, m+s\}$ are inequality constraints. For a vector z with $A^Tz = b$ the set of active constraints is $\mathcal{A} = \{1, \cdots, m\} \cup \{m+i \,|\, z_i = 0\}$. The basic approach of the algorithm is [6]

(8)
```
Step 0.  $z^{n+1} = \overline{y}^{n+1}$ ,   $\mathcal{A} = \{1, \cdots, m\}$
Step 1.  Repeat until all constraints are satisfied
             (a)  Choose a violated constraint $p \in \mathcal{K} - \mathcal{A}$   $\left(z^{n+1}_{p-m} < 0\right)$
             (b)  If (6) with constraints $\mathcal{A} \cup \{p\}$ infeasible STOP;
             (c)  Find unnecessary constraints $r \in \mathcal{A}$ and set $\mathcal{A} \leftarrow \mathcal{A} \cup \{p\} - \{r\}$;
             (d)  Solve (6) with equality constraints $\mathcal{A}$ only to get the new $z^{n+1}$;
Step 2.  STOP , $z^{n+1}$ is the optimal solution.
```

A detailed presentation is given in Appendix B.

6. A simpler projection method. We try to find a simpler alternative to the optimization routine. Given y with $y_{i1}, \cdots, y_{ip} < 0$ denote

$$B = [A \,|\, e_{i1} \,|\, \cdots \,|\, e_{ip}\,] \ ,$$

where e_j is the j^{th} unit vector. B is a collection of active constraint normals, in the sense that we look for z to satisfy

$$B^T z = \begin{bmatrix} A^T \\ e_{i1}^T \\ \vdots \\ e_{ip}^T \end{bmatrix} z = \begin{bmatrix} b \\ 0 \\ \vdots \\ 0 \end{bmatrix} .$$

The solution z is the orthogonal projection of y onto the manifold $\{B^T z = [b, 0]^T\}$. The proposed method is then

$$(9) \qquad \begin{aligned} \overline{y}^{n+1} &= \Phi_h^f(y^n) \ , \quad B = [A \,|\, e_{i1} \,| \cdots |\, e_{ip}\,] \text{ for } \overline{y}_{i1}^{n+1} < 0 \ , \cdots , \overline{y}_{ip}^{n+1} < 0 \ , \\ y^{n+1} &= \overline{y}^{n+1} - B \left(B^T B\right)^{-1} \begin{bmatrix} A^T\left(\overline{y}^{n+1} - y^0\right) \\ \overline{y}_{i1}^{n+1} \\ \vdots \\ \overline{y}_{ip}^{n+1} \end{bmatrix} = 0 \ . \end{aligned}$$

Since step control mechanism tries to keep the truncation-error norm $\|e^{n+1}\|_{G(y^{n+1})}$ small, it is natural to use a G-orthogonal projection

$$(10) \qquad y^{n+1} = \overline{y}^{n+1} - G^{-1} B \left(B^T G^{-1} B\right)^{-1} \begin{bmatrix} A^T\left(\overline{y}^{n+1} - y^0\right) \\ \overline{y}_{i1}^{n+1} - \epsilon_{i1} \\ \vdots \\ \overline{y}_{ip}^{n+1} - \epsilon_{ip} \end{bmatrix} = 0 \ .$$

The numbers ϵ_i are small and guarantee that $y_{i1}^{n+1} ... y_{ip}^{n+1}$ do not become negative due to roundoff errors. In fact, it is easy to see that the solution satisfies

$$A^T y^{n+1} = A^T y^0 \ , \quad y_{i1}^{n+1} = \epsilon_{i1} \ , \ \cdots \quad y_{ip}^{n+1} = \epsilon_{ip} \ .$$

If Φ_h^f preserves the linear invariants (as it should) we can replace $A^T\left(\overline{y}^{n+1} - y^0\right) = 0$ in (10). The implementation is done in a numerically-stable fashion using a reduced QR decomposition:

$$G^{-1/2} B = QR = Q_1 R_1 \ , \quad \left(B^T G^{-1} B\right)^{-1} = R_1^{-1} R_1^{-T} \ .$$

The method does not guarantee positivity, since the projection step may render other components negative (i.e. $y_j^{n+1} < 0$ for $j \neq i1 ... ip$). To guarantee positivity we can extend B by appending the column e_j, $B \leftarrow [B | e_j]$, and repeat the projection step 10, etc. This is just the Goldfarb and Idnani algorithm, minus the relaxation of unneeded active constraints.

However, even the non-iterative 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

$$y' = f(y) \ , \quad y(t^0) = y^0 \ , \quad h(y) = \begin{bmatrix} A^T\left(y - y^0\right) \\ y^- \end{bmatrix} = 0 \ .$$

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 non-negative components $y_i \geq 0$. In the spirit of [1] the method (9) can be viewed as a discretization of the "stabilized" system

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

(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 terms is such that it "pulls back" the trajectory toward the symplex; for example, if some component becomes negative, the correction term will tend to increase its concentration. Therefore, the simplex becames "attractive"; the solution cannot drift away so occasional negative concentrations do not lead to instability. For these reasons, we will reffer to the above method as the *stabilization* method.

7. Numerical Results.

7. Numerical Results. Consider the basic stratospheric reaction mechanism (adapted from NASA HSRP/AESA [10])

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

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 (11) 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

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

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

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

The simulation starts at noon and continues for 72 hours. The initial conditions are give 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 $\epsilon_i = 1$ 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 ROS-2 (19), fixed step $h = 30\,\text{min}$. Verwer et. al. [15, 2] advocated the favorable positivity properties of this method. The concentrations evolution is shown in Figure 1. ROS-2 seldom produces negative concentrations. Therefore, the optimization routine is called only a few times and the overhead is small. 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 $\epsilon_i = 1$ molec/cm^3).

The integration is also performed with the BDF-2 method (15) started with a backward Euler first step. The stepsize of the integration is fixed, $h = 30\,\text{min}$. If the concentration of species i is small and drops sharply we may have that $Y_i^n < 0$, and also $f_i \le 0$; 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 optimal projection positive version is almost 50% more costly than the original integration, since the optimization routine is caled many times. The positive trajectory is not distinguishible from the uncorrected one; both reproduce very well the reference trajectory. The invariants relative errors $(a_i^T y - b_i)/b_i$ have the same order of magnitude as the double precision roundoff errors.

In Figure 2 we see that the O^{1D} concentration falls frequently below zero and is "projected" back to 1 molec/cm^3; small positive concentrations, of the order 10^{-15} molec/cm^3 (which are not corrected) alternate with small negative concentrations (which are corrected to 1 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 molec/cm^3, but this will require more calls to the optimization routine, and an increased overhead.

Next, we integrate the system with RODAS-3 (18) with a fixed step $h = 30\,\mathrm{min}$. We do not give the actual solution plots, but will compare the results with other methods later on.

The minimal concentrations throughout the standard trajectories are given in Table II. Both optimally projected and the stabilization methods produce positive concentrations only. 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 (20), 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 codes to complete the integration are given in Table III. Stabilization is cheaper than optimal projection and produces similar results. Both positivity-preserving techniques have to be paired with a numerical scheme that favors positivity in order to reduce the overhead; a suitable such integration method is ROS-2.

Note that in the reduced stratospheric system (11) 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 simuation, 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-corrects the trajectory in case of negative values and explains why the standard and projected trajectories are undistinguishible.

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 (11) and append the (hypothetical) reaction

$$r11) \quad NO \ + \ O \ \xrightarrow{k_{11}} \ NO_2 \qquad (k_{11} = 1.0E-8) \tag{12}$$

In this case there is no auto-correction for negative values; reaction (12) will continue to destroy NO and O even when their concentrations become negative.

The results of integrating the extended system (11)–(12) with the fixed-step ROS-2, $h = 30$ min, are shown in Figure 3. The standard solution "explodes". This behavior can be explained as follows: at some moment the concentrations NO, O become negative reaction (12) continues at an accelerated pace, which will trigger the accumulation of NO_2, etc.

In contrast, the positive optimally projected solution is bounded, and is very close to the reference solution. This is shown in Figure 4. Stabilization gives a similar solution to optimal projection. Some errors in NO and NO_2 are present at the night-day transitions, but they do not accumulate in time. In addition the invariant errors are also very small.

Additional experiments (not shown here) revealed that BDF-2 with $h = 30$ 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. Positive optimally 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 tolerances $rtol = 0.01$, $atol = 0.01$ molec/cm^3; however, the chosen step-sizes are quite small (average $h = 3.4$ min). Table IV shows the number of foating point operations needed by each method; in particular, the variable step version is 12 times more expensive then the fixed step ($h = 30$ min) positive version. We tested the positively projected variable step ROS-2 with two different error controller techniques: one, the solution is projected onto the reaction simplex only after the step was accepted, and two, the computed solution is first projected and then the error estimation is computed and the decision to accept/reject is made. For our example the first variant is slightly more efficient. The variable-step ROS-2 and its positive version give almost identical results for $rtol = 0.01$; for the larger $rtol = 0.1$ the positive version 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 $h \geq h_{\min} = 1$ 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 (4) shows the exact evolution of of species for the extended system (11)–(12), the evolution with clipping, optimal projection and stabilization, for a step size $h = 30$ min. Indeed, clipping stabilizes the solution, but it also introduces large errors; these errors accumulate in time until all meaningfull information is lost.

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.

8. 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 [3]. The positive optimal 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.

A less expensive alternative is the stabilization method, which performs only one projection at the end of the step if negative component concentrations appear. Although positivity is not guaranteed, the overall behavior is very good and the solutions comparable to the ones obtained by optimal projection.

Both technique have to be paired with a positivity-favourable numerical integration method, for example ROS-2 [15, 2]. 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 efficincy.

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 of the method 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 [5]).

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

$$y'(t) = f\,(t, y)\ \ ,\quad y(t_0) = y^0\ . \tag{13}$$

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 [7, 8] 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,

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

since consistency requires $\sum_{i=0}^{k-2} \alpha_i = -1$.

The simplest example is the (first order) backward Euler formula,

$$y^{n+1} = y^n + h\, f\left(t^{n+1}, y^{n+1}\right) \ . \tag{14}$$

To solve this nonlinear equation one uses modified Newton iterations,

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

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

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

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

$$y^{n+1} = Y^n + \frac{2}{3}\, h\, f\left(t^{n+1}, y^{n+1}\right) \ , \quad Y^n = \frac{4}{3} y^n - \frac{1}{3} y^{n-1} \ . \tag{15}$$

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 [7] defines the solution by the formula the solution is given by the formulas

$$y^{n+1} = y^n + \sum_{i=1}^{s} b_i k_i, \quad k_i = hf\left(t^n + c_i h, y^n + \sum_{j=1}^{s} a_{ij} k_j\right) \tag{16}$$

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. Multypling 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 [8] is usually defined for autonomous systems; the solution is given by the formulas

$$y^{n+1} = y^n + \sum_{i=1}^{s} b_i k_i, \quad k_i = hf\left(y^n + \sum_{j=1}^{i-1} \alpha_{ij} k_j\right) + hJ \sum_{j=1}^{i} \gamma_{ij} k_j, \tag{17}$$

where s and the formula coefficients b_i, α_{ij} and γ_{ij} give the order of consistency and the stability properties. Again, multypling 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 [12] which, in autonomous form, reads

$$\begin{aligned}
y^{n+1} &= y^n + 2\,k_1 + k_3 + k_4 \ , \\
\left(\frac{1}{\gamma h} I - J\right) k_1 &= f\left(y^n\right) \ , \\
\left(\frac{1}{\gamma h} I - J\right) k_2 &= f\left(y^n\right) + \frac{4}{h} k_1 \ , \\
\left(\frac{1}{\gamma h} I - J\right) k_3 &= f\left(y^n + 2\,k_1\right) + \frac{1}{h} k_1 - \frac{1}{h} k_2 \ , \\
\left(\frac{1}{\gamma h} I - J\right) k_4 &= f\left(y^n + 2k_1 - k_3\right) + \frac{1}{h} k_1 - \frac{1}{h} k_2 - \frac{8}{3h} k_3 \ ,
\end{aligned} \tag{18}$$

where $\gamma = 1/2$.

Another example of Rosenbrock scheme is ROS-2 [15, 2]

$$\begin{aligned} y^{n+1} &= y^n + \frac{3}{2}k_1 + \frac{1}{2}k_2 \ , \\ \left(\frac{1}{\gamma h}I - J\right) k_1 &= \frac{1}{\gamma} f\left(y^n\right) \ , \quad \left(\frac{1}{\gamma h}I - J\right) k_2 = \frac{1}{\gamma}\left[f\left(y^n + k_1\right) - \frac{2}{h}k_1\right] \ , \end{aligned} \tag{19}$$

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

$$\begin{aligned} y^{n+1} &= y^n + \frac{3\gamma}{2}k_1 + \frac{\gamma}{2}k_2 \ , \\ \left(\frac{1}{\gamma h}I - J\right) k_1 &= f\left(y^n\right) \ , \quad \left(\frac{1}{\gamma h}I - J\right) k_2 = f\left(y^n + k_1\right) - \frac{2}{h}k_1 \ . \end{aligned} \tag{20}$$

In [15, 2] 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 $C' = -k\,C^2$, $C(t_0) \geq 0$. 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 [3] 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

$$\min \ \frac{1}{2} z^T G z - y^T G z \quad \text{subject to} \quad A^T z = b \ , \ z \geq \epsilon \ . \tag{21}$$

The problem has m equality constraints ($a_i^T z = b_i$, $i = 1 \cdots m$, where a_i is the i^{th} column of A) and s inequality constraints ($z_i \geq \epsilon_i$ 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 [6]. 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 ne the matrix of normal vectors of active constraints; for example, if $z_i = e_i$ and $z_j = e_j$ ($q = m + 2$) this matrix is

$$N = [a_1 | \cdots | a_m | e_i | e_j] \in \Re^{s \times q} \ .$$

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

$$N^T z = \begin{bmatrix} b \\ 0 \end{bmatrix} \begin{matrix} \}m \\ \}2 \end{matrix} \ .$$

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

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

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

$$z^+ = z + tHe_p \tag{22}$$

will preserve all the active constraints, $N^T H = 0 \implies N^T z^+ = N^T z$. The step (22) can move the current vector toward satisfying the constraint. We have

$$z_p^+ = e_p^T z^+ = e_p^T z + te_p^T He_p = z_p + tH_{p,p}$$

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 $R \in \Re^{q \times q}$ and $J \in \Re^{s \times s}$ via the following QR decomposition:

$$G^{-1/2} N = QR = [\underbrace{Q_1}_{q} \mid \underbrace{Q_2}_{s-q}] \underbrace{\begin{bmatrix} R \\ 0 \end{bmatrix}}_{q} \ , \quad J = G^{-1/2} Q = [\underbrace{J_1}_{q} \mid \underbrace{J_2}_{s-q}] \ . \tag{23}$$

```
Step 0.  Initializations.
           $z \leftarrow y$;
           $\mathcal{A} \leftarrow \{1, \cdots, m\}$; $q \leftarrow m$;
           $\lambda = [0 \cdots 0]$ (Lagrange multipliers);
           $N \leftarrow A$;
           Compute $R$, $J$ by (23).
Step 1.  Choose a violated constraint (if any). Is $z_p < \epsilon_p$ for some $p$ ?
           IF {no} THEN
           |  $z$ is the optimal solution. STOP
           ELSE {if yes}
           |  $\lambda^+ \leftarrow [\lambda, 0]$ (consider the violated constraint $m + p$)
           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)
           IF { $\delta^{\text{dual}} \leq 0$ or $q = m$ } THEN
           |  $\tau^{\text{dual}} = \infty$
           ELSE
           |  $\tau^{\text{dual}} = \min_{j=m+1:q;\ \delta_j^{\text{dual}}>0} \left\{\lambda_j^+ / \delta_j^{\text{dual}}\right\} = \lambda_k^+ / \delta_k^{\text{dual}}$
           END IF

         Primal step length (which makes $z_p = \epsilon_p$)
           IF { $\delta^{\text{primal}} = 0$ } THEN
           |  $\tau^{\text{primal}} = \infty$
           ELSE
           |  $\tau^{\text{primal}} = (\epsilon_p - z_p)/\delta_p^{\text{primal}}$
           END IF

         Step length
           $t = \min\left(\tau^{\text{primal}}, \tau^{\text{dual}}\right)$
Step 3.
    (a)  Check Infeasibility
           If { $(\tau^{\text{primal}} = \infty)$ and $(\tau^{\text{dual}} = \infty)$ } THEN
           |    The problem is infeasible. STOP.
           END IF

    (b)  Primal Infeasible, step in dual space only
           IF { $\tau^{\text{primal}} = \infty$ (Primal Infeasible) } THEN
           |    $\lambda^+ \leftarrow \lambda^+ + t[-\delta^{\text{dual}}, 1]$
           |    Drop $k - m^{\text{th}}$ inequality constraint:
           |          $\mathcal{A} \leftarrow \mathcal{A} \backslash \{k\}$; $q \leftarrow q - 1$
           |          Update $R$, $J$ for newly dropped constraint
           |    Go to Step 2(a).
           END IF

    (c)  Step in both primal and dual space
           $z \leftarrow z + t\,\delta^{\text{primal}}$
           $\lambda^+ \leftarrow \lambda^+ + t\,[-\delta^{\text{dual}}, 1]$
           IF { $t = \tau^{\text{primal}}$ (Full Step) } THEN
           |    $\lambda \leftarrow \lambda^+$
           |    Add $p^{\text{th}}$ 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 { $t = \tau^{\text{dual}}$ (Partial Step) } THEN
    Drop $k - m^{\text{th}}$ inequality constraint:
        $\mathcal{A} \leftarrow \mathcal{A}\backslash\{k\}$; $q \leftarrow q-1$
        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 accomodate it; for clarity, in what follows the updated versions will be marked by the superscript $(+)$. Adding the active constraint $\{z_p = e_p\}$ leads to the following changes.

$$\begin{aligned}
d &= (J(p,:))^T = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \begin{array}{l} \}q \\ \}s-q \end{array} \\
U &= \text{Householder reflector s.t. } Ud = \begin{bmatrix} d_1 \\ \sigma \\ 0 \end{bmatrix} \begin{array}{l} \}q \\ \}1 \\ \}s-q-1 \end{array} \\
R^+ &\leftarrow \begin{bmatrix} R & d_1 \\ 0 & \sigma \end{bmatrix} \in \Re^{(q+1)\times(q+1)} \\
J^+ &\leftarrow [\underbrace{J_1}_{q} \,|\, \underbrace{J_2U^T}_{s-q}] = [\underbrace{J_1^+}_{q+1} \,|\, \underbrace{J_2^+}_{s-q-1}]
\end{aligned}$$

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

$$\begin{aligned}
R &= \begin{bmatrix} R_1 & r_k^1 & S \\ 0 & r_k^2 & T \end{bmatrix} \implies \overline{R} = \begin{bmatrix} R_1 & S \\ 0 & T \end{bmatrix} \left(\begin{array}{c} \text{Delete the } k^{\text{th}} \\ \text{column of } R \end{array} \right) \\
UT &= \begin{bmatrix} R_2 \\ 0 \end{bmatrix} \text{ (T Hessenberg, compute easily its QR decomposition)} \\
R^- &\leftarrow \begin{bmatrix} R_1 & S \\ 0 & R_2 \end{bmatrix} \in \Re^{(q-1)\times(q-1)} \\
J &= [\underbrace{J_1}_{q} \,|\, \underbrace{J_2}_{s-q}] = [\underbrace{J_1^a}_{\ell-1} \,|\, \underbrace{J_1^b}_{q-\ell+1} \,|\, \underbrace{J_2}_{s-q}]\ ; \\
J^- &\leftarrow [\underbrace{J_1^a}_{\ell-1} \,|\, \underbrace{J_1^b U^T}_{q-\ell+1} \,|\, \underbrace{J^c}_{s-q}] = [\underbrace{J_1^-}_{q-1} \,|\, \underbrace{J_2^-}_{s-q+1}]
\end{aligned}$$

To compute quantities of the form J_2U^T, $J_1^bU^T$ 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, $J_2U^T = (U_1 \cdots U_n J_2^T)^T$.

Acknowledgements. I thank Mihai Anitescu for a fruitful discussion regarding quadratic optimization algorithms and the norm approximation property of the solution (Lemma 4.1).

REFERENCES

[1] U. M. Ascher and H. Chin andS. Reich. Stabilization of DAEs and invariant manifolds. 1994.

[2] J. G. Blom and J.G. Verwer. A comparison of integration methods for atmospheric transport-chemistry problems. *Journal of Computational and Applied Mathematics*, 2000.

[3] C. Bolley and M. Crouzeix. Conservation de la positivite lors de la discretization des problemes d'evolution parabolique. *R.A.I.R.O. Numerical Analysis*, 12(3):237–245, 1978.

[4] G.R. Carmichael, L.K. Peters, and T. Kitada. A second generation model for regional-scale transport/ chemistry/ deposition. *Atmospheric environment*, 20:173–188, 1986.

[5] 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.

[6] D. Goldfarb and A. Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27:1–33, 1983.

[7] E. Hairer, S.P. Norsett, and G. Wanner. *Solving Ordinary Differential Equations I. Nonstiff Problems.* Springer-Verlag, Berlin, 1993.

[8] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems.* Springer-Verlag, Berlin, 1991.

[9] W.H. Hundsdorfer. Numerical solution of advection-diffusion-reaction equations. Technical Report NM-N9603, Department of Numerical Mathematics, CWI, Amsterdam, 1996.

[10] D.E. Kinnison. NASA HSRP/AESA stratospheric models intercomparison. *NASA ftp site*, contact kinnison1@llnl.gov.

[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. G. Verwer, F.A. Potra, and G.R. Carmichael. Benchmarking stiff ODE solvers for atmospheric chemistry equations II - Rosenbrock Solvers. *Atmospheric Environment*, 31:3459–3472, 1997.

[13] L.F. Shampine. Conservation laws and the numerical solution of ODEs. *Computers and Mathematics with Applications*, 12B(5/6):1287–1296, 1986.

[14] J. Verwer, W. Hunsdorfer, and J. G. Blom. Numerical time integration of air pollution models. (MAS-R9825), October 1998.

[15] J. Verwer, E.J. Spee, J. G. Blom, and W. Hunsdorfer. A second order Rosenbrock method applied to photochemical dispersion problems. *SIAM Journal on Scientific Computing*, 20:1456–1480, 1999.

Figure captions

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

FIG. *3. Concentration evolution for the extended system (11)–(12). Integration with ROS-2, fixed stepsize* $h = 15$ *min, standard vs. positive methods. Note the unstable behavior of the standard solution.*

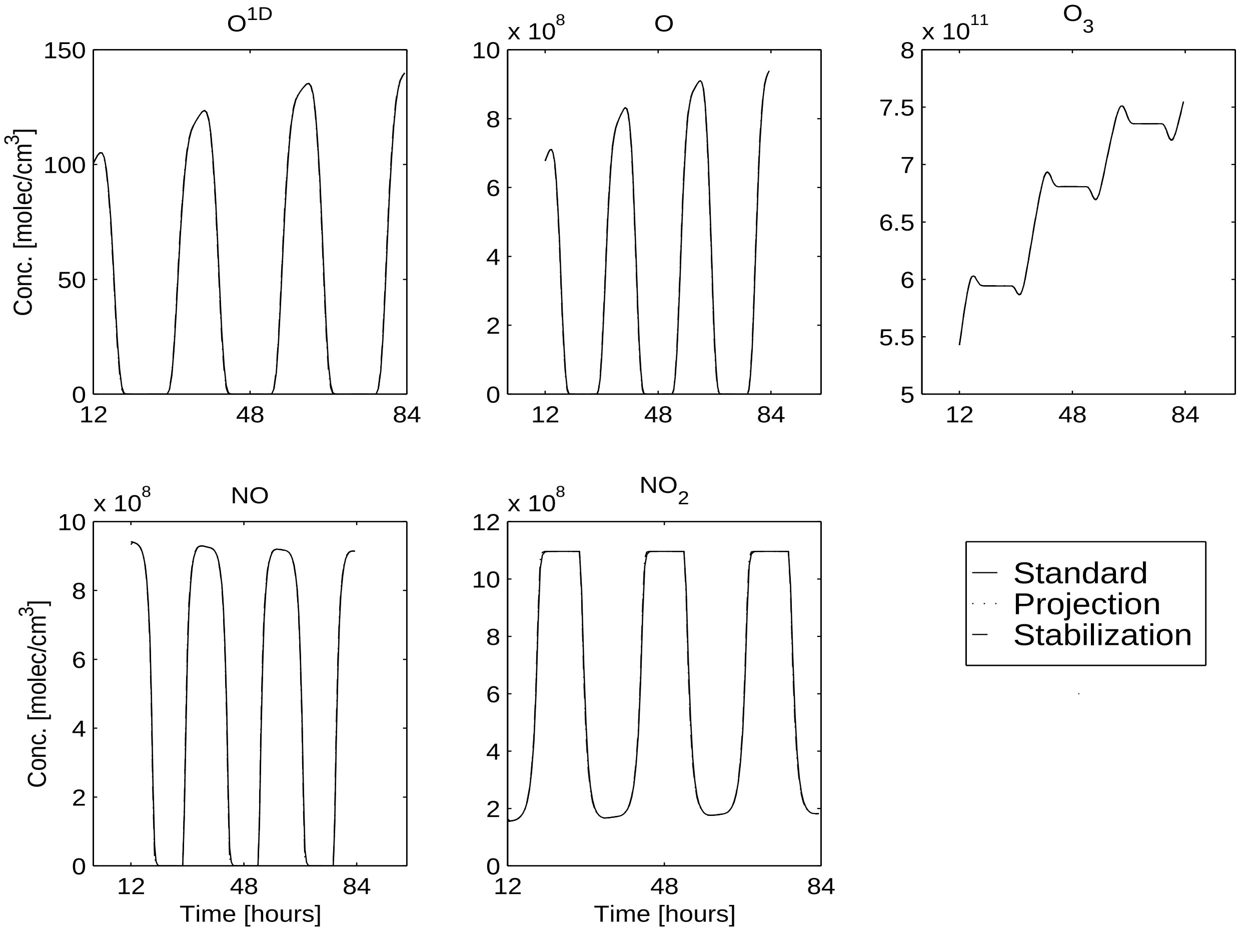

FIG. 1. *Concentration evolution for the system (11). Integration with ROS-2, fixed stepsize $h = 30$ min, standard method vs. optimal projection and stabilization methods. All solutions are similar.*

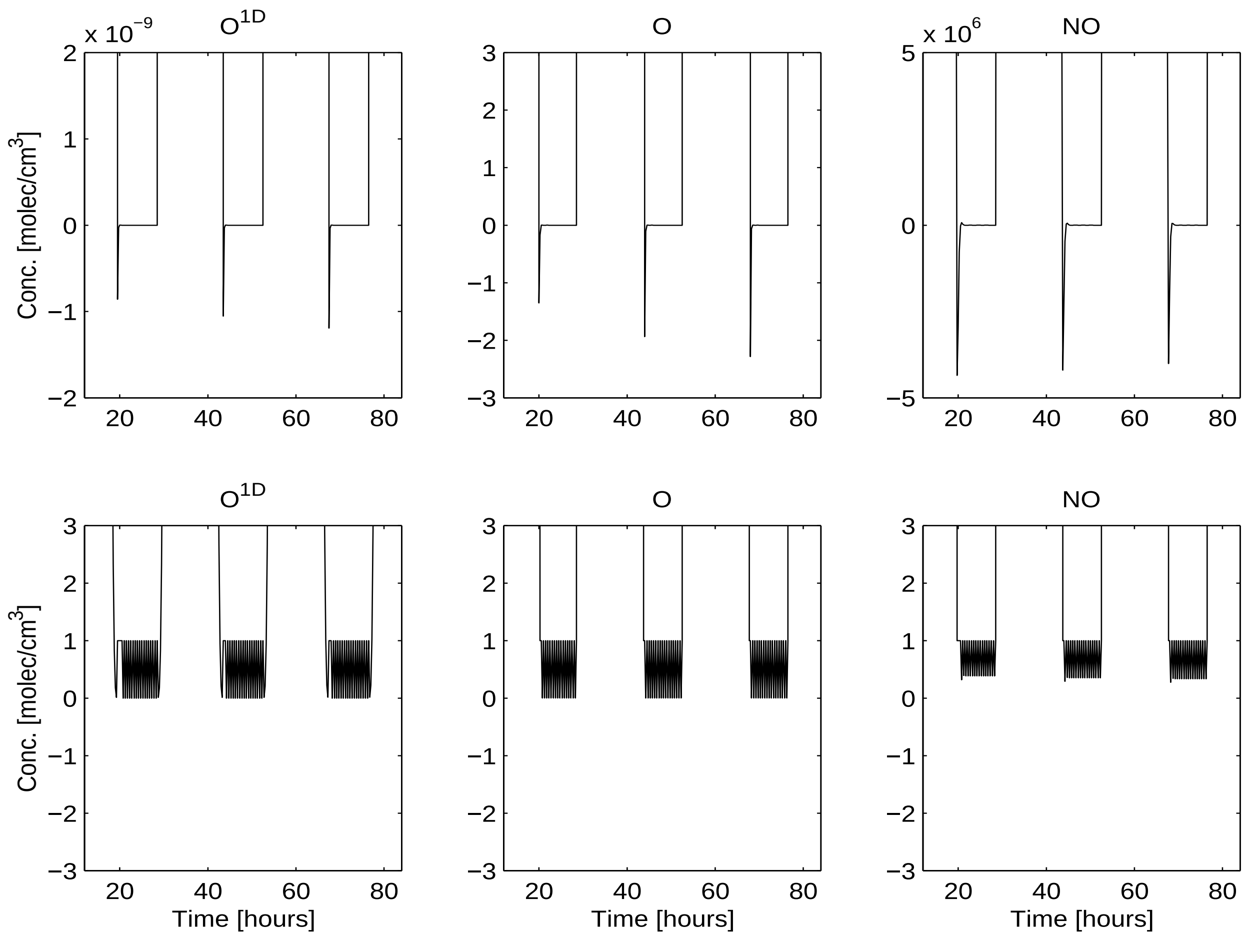

FIG. 2. *Zoom in on concentration evolution with BDF-2, fixed $h = 30min$. Upper figure shows the original solution, and lower figure shows the positive optimally projected solution.*

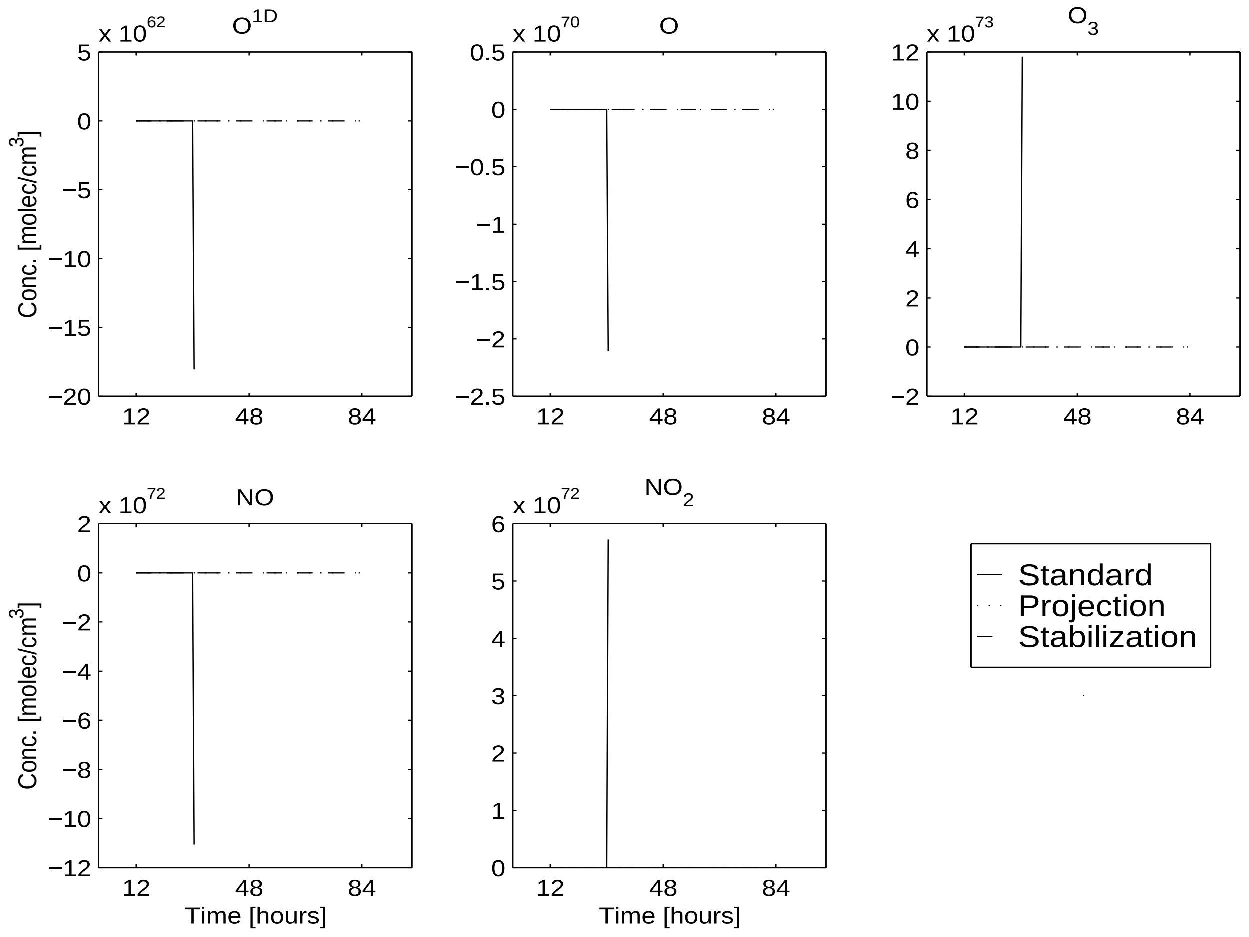

FIG. 3. *Concentration evolution for the extended system (11)–(12). Integration with ROS-2, fixed stepsize* $h = 30$ *min, standard method vs. optimal projection and stabilization methods. Note the unstable behavior of the standard solution.*

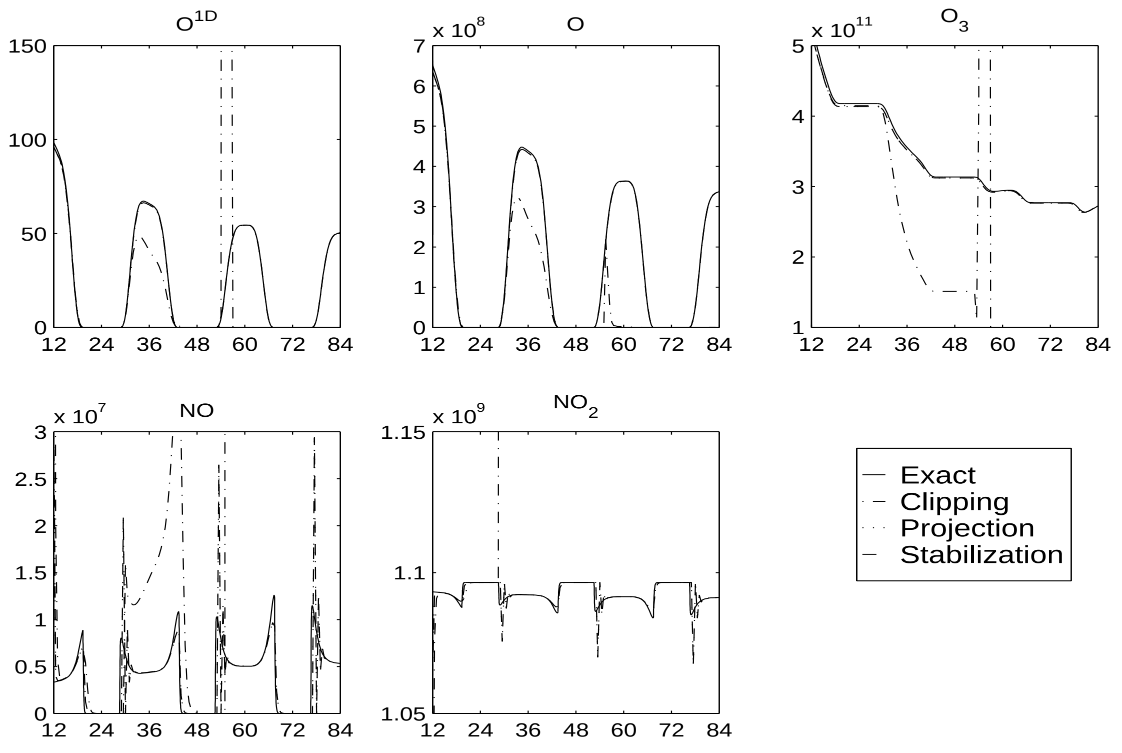

FIG. 4. *Computed concentrations for the extended system (11)–(12) with ROS-2, fixed stepsize $h = 30$ min. The optimally projected and stabilized solutions are indistinguishable and are very close to the reference one; clipping introduces significant errors which accumulate in time and adversely impact the quality of the solution.*

O^{1D}	O	O_3	O_2	NO	NO_2
9.906E+01	6.624E+08	5.326E+11	1.697E+16	8.725E+08	2.240E+08

TABLE I

Initial concentrations for the simulation (molec/cm^3).

Integrator	Min $[O^{1D}]$	Min $[O]$	Min $[NO]$
BDF-2	$-7.0E-8$	-3.3	$-1.7E+7$
RODAS-3	$-1.3E-5$	-17.8	$-4.6E+6$
ROS-2	$-5.5E-315$	$+2.9E-112$	$+4.2E-21$

TABLE II

Minimal concentrations (molec/cm^3) for the stratospheric test problem with various integration methods. The step size is fixed $h = 30$ min.

Integrator	Standard	Optimal Projection	Overhead	Stabilization	Overhead
BDF-2	276 Kflops	407 Kflops	47%	352 Kflops	27%
RODAS-3	326 Kflops	495 Kflops	52%	420 Kflops	29%
ROS-2	241 Kflops	257 Kflops	6.7%	255 Kflops	5.7%

TABLE III

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.

Integrator	Clipping	Optimal Projection	Stabilization	Variable step
ROS-2	272 Kflops	303 Kflops	290 Kflops	3,798 Kflops

Table IV

The number of flops for integrating the extended stratospheric test problem with ROS-2 ($h = 30$ min.) for clipping, optimal projection, stabilization and variable step size ($rtol = 0.01$).