PARALLEL IN TIME ALGORITHMS WITH
REDUCTION METHODS FOR SOLVING
CHEMICAL KINETICS

ADEL BLOUZA, LAURENT BOUDIN
AND SIDI MAHMOUD KABER

PARALLEL IN TIME ALGORITHMS WITH REDUCTION METHODS FOR SOLVING CHEMICAL KINETICS

ADEL BLOUZA, LAURENT BOUDIN AND SIDI MAHMOUD KABER

We design suitable parallel in time algorithms coupled with reduction methods for the stiff differential systems integration arising in chemical kinetics. We consider linear as well as nonlinear systems. The numerical efficiency of our approach is illustrated by a realistic ozone production model.

1. Introduction

Parareal algorithms were first introduced in [18] to solve evolution problems in real time. The principle is the following. One first approximates the solution on a coarse time grid, and then locally solves the equations on fine time subgrids on parallel computers. One can prove that the associated iterative procedure ensures an accuracy which is of same order as a sequential algorithm on a global fine time grid. Mathematical properties of these algorithms have been recently investigated; see [2; 25; 16; 15; 13; 14; 12], for example. They have been applied in various fields, such as financial mathematics [3], fluid mechanics and fluid-structure interaction [9; 11; 10], oceanography [19], chemistry [21] and quantum chemistry [22].

The present work is dedicated to standard chemistry. We study monomolecular chemistry, as in [21], for which we carry out a new modified parareal algorithm preserving stoichiometric invariants. We also investigate the nonlinear chemistry case.

When the reaction scheme is monomolecular, the kinetic equations describing species evolutions are linear but may be stiff. In this context, we consider the thyroid reaction scheme given in [23]. An efficient reduction algorithm is described in [5], and applied to this biochemical model. It is an inductive procedure, based on linear algebraic techniques. The reduction process, applied to the initial kinetic system, eliminates the fastest dynamics, and no change of coordinates is required. This process is systematic and does not rely on conventional chemical assumptions (see [27] for a large survey of these techniques). Applied to chemical kinetic systems with kinetic constants in different scales, the algorithm eliminates reactants

arising in some of the fastest reactions. The reduced system then provides an accurate approximation for the slow dynamics.

The ozone model we study here is a typical nonlinear, realistic model for ozone production in the troposphere. The issue of ozone pollution is one of the most important environmental problems we have faced for the last three decades (see [24] and the references therein). The massive presence in the troposphere, mainly above urban areas, of nitrogen dioxide NO_2, coupled with one of various hydrocarbons, induces a preferential chain of reactions which produces ozone O_3. This chain is really favored by a large amount of ultraviolet rays, basically during sunny summers. The ozone concentration then reaches a level that may be dangerous for both human health and ecosystems.

The ozone model of [1; 4] describes the evolution of the main species concentrations at stake. Numerical simulations of reactive flows can often be really difficult to tackle, mainly because of the intricate chemical mechanisms that must be taken into account. That is the case, for instance, with the air quality issue: we do not focus on a simple description of the chemical kinetics of reactions of nitrogen oxides and ozone. We need to take into account more reactions including pollutants themselves to model more faithfully the pollution in the atmosphere.

We aim to compute numerically the evolution of the chemical species in the atmosphere, including the pollutants, within a reasonable computational time. Some of the phenomena are really stiff and have to be discretized with a very small time step. To get around this major numerical difficulty, we use a suitable parareal algorithm where the associated coarse propagator is applied to the reduced system in order to minimize its cost. We also need an accurate description of both the physics (convection, diffusion, source or well of pollution) and the chemistry (reactions). Nevertheless, it is quite clear that our PDE system is large, and most of the nonlinearity comes from the chemical part. We here assume that the chemistry also governs the coupling between our equations. Therefore, the chemical kinetics naturally stand as the key point of our study of the ozone model in the troposphere.

We first focus on the kinetics of the reactions producing ozone, and drop the dependence on the space variable. Those reactions lead to an ODE system, where several problems have to be taken care of.

- There is a large variety of characteristic time scales for the species involved.

- Dozens or even hundreds of chemical reactions and species may be concerned.

- Most of the ODEs are nonlinear.

Once again, our approach consists in first using a reduced model: when possible, we approximate the full differential system by an algebraic-differential system where transitional fast states are neglected. This allows us to simultaneously lower the size of the system and to avoid or at least weaken its stiffness. In the nonlinear

case, which is the most common situation, reduction algorithms are not so easy to design. However, the quasisteady state (QSS) method seems to be an efficient compromise: some species are put at chemical equilibrium, with high rates of both production and consumption (the species is destroyed as soon as it is produced).

This work is organized as follows. In Section 2, we recall a convenient parareal algorithm for our models, which considers stoichiometric invariants. Section 3 is dedicated to the study of monomolecular chemistry, and Section 4 to the ozone model. In each case, we describe the chemical models, briefly discuss a reduction method fitted to the situations, and show numerical results that indicate the efficiency of the parareal algorithm.

2. Parareal algorithm

Let $m \geq 1$ and consider the following ordinary differential equation, where $y : \mathbb{R}_+ \to \mathbb{R}^m$ is the unknown:

$$y'(t) = f(t, y(t)), \tag{1}$$

with initial Cauchy condition $y(0) = y_0 \in \mathbb{R}^m$ and where $f : \mathbb{R}_+ \times \mathbb{R}^m \to \mathbb{R}^m$ is given. Section 3 corresponds to the linear version of (1), that is, with

$$f(t, y(t)) = J y(t),$$

where J is a time-independent matrix.

We are interested in computing the solution u of (1) on an interval $[0, T]$, with $T > 0$. For any $N \geq 1$, we consider intermediate times $0 = T_0 < T_1 < \cdots < T_N = T$ and, for the sake of simplicity, a constant coarse time step $\Delta T = T_{n+1} - T_n$, where n denotes the coarse time index. Note that ΔT may not be constant, and that the associated algorithm would only be an adjustment of the one presented below.

Let k denote the parareal iteration index. The parareal scheme consists in designing a sequence $(y_n^k)_{k \in \mathbb{N}}$ at each coarse time step $[T_n, T_{n+1}]$ such that, for each n,

$$\lim_{k \to +\infty} y_n^k = \bar{y}_n,$$

where $\bar{y}_n$ is an approximation of $y(T_n)$, and the convergence, which of course depends on the accuracy of the coarse propagator, should be fast. Indeed, in many applications, $k \leq 5$ is enough to get a satisfying approximation.

The parareal algorithm uses two different schemes: a fine one $\mathcal{F}_{\delta t}$, based on a fine time step $\delta t > 0$, and a coarse one $\mathcal{C}_{\Delta T}$, based on coarse time step $\Delta T = s \delta t$, $s \in \mathbb{N}^*$. In most situations, we consider $s \gg 1$.

The coarse solver is applied for the evolution on $[0, T]$. The quantity

$$v_{n+1} = \mathcal{C}_{\Delta T}(T_{n+1}; T_n, v_n)$$

is an approximation, at time T_{n+1}, of the solution of (1) on $[T_n, T_{n+1}]$, with initial value v_n at time T_n. Note that v_{n+1} is the only value computed by the coarse solver on $]T_n, T_{n+1}]$. For simplicity and stability reasons, we use the implicit Euler scheme. In the linear case, we write $v_{n+1} = Cv_n$, where $C = (I - \Delta T\, J)^{-1}$.

The fine solver is applied for the evolution on each subinterval $[T_n, T_{n+1}]$. The quantity

$$w_{n+1} = \mathcal{F}_{\delta t}(T_{n+1}; T_n, w_n)$$

is an approximation, at time T_{n+1}, of the solution of (1) on $[T_n, T_{n+1}]$, with initial value w_n at time T_n. In the linear case, w_{n+1} can be written under the form $w_{n+1} = F^s w_n$, where F is a time-independent matrix. For instance, in the case of an explicit Euler scheme, we have $F = I + \delta t\, J$. We may also use the Runge–Kutta RK4 and the implicit Euler schemes.

The full parareal sequence $(y_n^k)_{n,k}$ is inductively defined, for any k, n, by

$$y_0^k = y_0, \tag{2}$$
$$y_{n+1}^0 = \mathcal{C}_{\Delta T}(T_{n+1}; T_n, y_n^0), \tag{3}$$
$$y_{n+1}^{k+1} = \mathcal{C}_{\Delta T}(T_{n+1}; T_n, y_n^{k+1}) + \mathcal{F}_{\delta t}(T_{n+1}; T_n, y_n^k) - \mathcal{C}_{\Delta T}(T_{n+1}; T_n, y_n^k). \tag{4}$$

Passing to the limit in (4) as k goes to $+\infty$, we get

$$\bar{y}_{n+1} = \mathcal{F}_{\delta t}(T_{n+1}; T_n, \bar{y}_n).$$

This relation means that $\bar{y}_{n+1}$ is obtained from $\bar{y}_n$ by use of the fine scheme $\mathcal{F}_{\delta t}$ on the interval $[T_n, T_{n+1}]$. That ensures that y_n^k may be a good approximation of the fine solution when k is not too small.

In order to reduce the CPU cost, we shall apply the coarse propagator $\mathcal{C}_{\Delta T}$ to a reduced system of ODEs, that is

$$\tilde{y}'(t) = \tilde{f}(t, \tilde{y}(t)) \tag{5}$$

where $\tilde{m} < m$ and $\tilde{f} : \mathbb{R}_+ \times \mathbb{R}^{\tilde{m}} \to \mathbb{R}^{\tilde{m}}$ can be computed from f. It is assumed that (5) is easier to solve than (1) and that $\tilde{f}$ describes a simpler but still faithful physics.

Remark 1. In a chemical kinetics context, when we apply algorithms (2)–(4) to integrate the differential system, we should pay special attention to preserving certain properties of the system, such as stoichiometric invariants and stationary points. We shall see below how to establish from (2)–(4) an ad hoc numerical scheme considering these invariants.

3. Linear chemistry

3.1. *Framework.* Consider N gaseous species $(\mathcal{A}_i)_{1 \le i \le N}$ and denote by $(y_i)_{1 \le i \le N}$ the corresponding concentrations. In the gaseous mixture, there can also be some species $(\mathcal{N}_l)_{1 \le l \le L}$, like O_2 or N_2 in air, whose concentration variations are neglected. Consequently, we only take into account the variations of (y_i) of the limiting species $(\mathcal{A}_i)$. We assume that R chemical reactions take place simultaneously in the mixture, and that these R reactions are monomolecular in the species $(\mathcal{A}_i)$; that is, each reaction r can be written as

$$\mathcal{A}_{i_r} + \mathcal{N}_{i_r} \xrightarrow{k_r} \mathcal{A}_{j_r} + \mathcal{N}_{j_r},$$
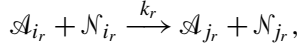
with reaction rate $v_r = k_r y_{i_r}$, where k_r is a given positive kinetic constant. For the sake of simplicity, we assume that a pair $(\mathcal{A}_i, \mathcal{A}_j)$ appears in at most one reaction as reactant-product.

The time evolution of the concentration y_i, $1 \le i \le N$, is governed by the law of mass action

$$\frac{dy_i}{dt} = - \sum_{r,\ i_r = i} k_r y_{i_r} + \sum_{r,\ j_r = i} k_r y_{j_r} = - \sum_{r,\ i_r = i} v_r + \sum_{r,\ j_r = i} v_r.$$

The first sum deals with the reactions where $\mathcal{A}_i$ is a reactant, and the second one deals with the ones where $\mathcal{A}_i$ is a product. In other words, $y = (y_i)_{1 \le i \le N}$ satisfies the stoichiometric system

$$\frac{dy}{dt} = Sv,$$

where $v = (v_r)_{1 \le r \le R}$ is the vector of the reaction rates and $S \in \mathbb{R}^{N \times R}$ is the stoichiometric matrix, defined by $S_{i,r} = -1$ for $i = i_r$, $S_{i,r} = 1$ for $i = j_r$, and $S_{i,r} = 0$ otherwise. Since the chemical reactions are assumed to be monomolecular, v linearly depends on y. Hence, y solves the following differential system, for a given initial datum,

$$\frac{dy}{dt} = Jy, \quad t \ge 0, \tag{6}$$

where $J \in \mathbb{R}^{N \times N}$ is defined by $J_{jj} = - \sum_{r,\ i_r = j} k_r$, $J_{ij} = k_r$ if $i \ne j$ and there exists r such that $(i_r, j_r) = (j, i)$, and $J_{ij} = 0$ otherwise. By assumption on our system, if $i \ne j$ and if there exists a reaction r such that $(i_r, j_r) = (j, i)$, then r is unique. Note that J is a kinetic matrix, that is, it satisfies, for any j, $J_{jj} \le 0$, $J_{ij} \ge 0$ for all $i \ne j$, and $\sum_i J_{ij} = 0$.

The kinetic matrix J is semistable, that is, all its nonzero eigenvalues have negative real parts. Besides, 0 is an eigenvalue of J and its multiplicity indicates the number of stoichiometric invariants. That implies that any solution of (6) has a finite limit when t goes to $+\infty$. The reader may find more details in [7; 8; 26].

3.2. *A reduction method.* In [5], the authors introduced an algorithm to obtain, from (6), a reduced and nonstiff system which only involves a subset of the initial species, coupled with algebraic equations for the remaining species concentrations. This algebraic-differential system accurately approximates the full stiff system, after the exit time from the boundary layer. Let us briefly recall this reduction method and the associated error estimate.

At each reduction step $1 \leq k \leq N - 1$, an index $i_k \in \{1, \ldots, N\}$, a semistable matrix J^k, and real coefficients $(\beta_{i_k, j})_{j \in \mathcal{H}_k}$ are inductively built with

$$\mathcal{H}_k = \{1, \ldots, N\} \setminus \{i_1, \ldots, i_k\}.$$

Let us then fix a step $1 \leq p \leq N - 1$ at which we decide to stop the reduction process. For any $y = (y_i)_{1 \leq i \leq N}$, we denote $\tilde{y} = (y_i)_{i \in \mathcal{H}_p}$. The reduced system associated to (6) up to step p is the following algebraic-differential system of unknown $z^p = (z_i^p)_{1 \leq i \leq N}$, defined for a given initial datum (at $t = T^*$) for the differential part of the system:

$$\frac{d\tilde{z}^p}{dt} = J^p \tilde{z}^p, \quad t \geq T^*, \tag{7}$$

$$z_{i_k}^p(t) = \sum_{j \in \mathcal{H}_k} \beta_{i_k j} z_j^p(t), \quad 1 \leq k \leq p, \quad t \geq T^*. \tag{8}$$

If p is suitably chosen, the reduced matrix J^p only contains the $N - p$ eigenvalues of J which have the lowest real parts. Note that, when the eigenvalues of J^p are small with respect to the p first eigenvalues of J, the differential system (7) is not stiff anymore (this idea was also used in [17; 20]). Here, the algebraic equations (8) approximate the fast species $(z_i)_{i \notin \mathcal{H}_p}$, and T^* is an exit time from the corresponding boundary layer. Real coefficients $\beta_{i_k j}$ are expressed in terms of left eigenvectors of matrices J^k, $k = 0, \ldots, p$.

It is shown in [5] that the nonstiff problem (7)–(8) actually yields a relevant approximation of the solutions of (6) for $t \geq T^*$. More precisely, assume that the initial data for (7) is chosen such that

$$|z_j^p(T^*) - y_j(T^*)| \leq ch, \quad \text{for all } j \in \mathcal{H}_p,$$

where h can be viewed as the numerical error of the underlying scheme at time T^* and c is a nonnegative constant which does not depend on h. Then there exists $\alpha \geq 1$ and $C \geq 0$, depending on $y(0)$, such that, for any $t \geq T^*$, with $T^* \geq \alpha \varepsilon \ln(1/\varepsilon)$,

$$|z_j^p(t) - y_j(t)| \leq C(h + \varepsilon), \quad 1 \leq j \leq N.$$

In other words, provided that the errors due to the prescribing of the values of the slow species at the exit time T^* from the boundary layer are small, the errors with respect to the exact solution remain small at any further time $t \geq T^*$.

3.3. *Numerical tests.* We apply the previous algorithm to the dynamics of the thyroid hormones [23]. It was investigated in [5] for the reduction method, and in [21] for both reduction method and parareal algorithm, including a source term. The chemical network (see Figure 1) involves 8 species and 14 reactions.

The kinetic matrix of the set of reactions reads

$$J = \begin{pmatrix} -5.1 & 0.01 & 0. & 0. & 0.06 & 0. & 0. & 0. \\ 0. & -2.516 & 0. & 0. & 0. & 0.0008 & 0. & 0. \\ 0. & 0. & -1.3 & 0.001 & 0.0003 & 0. & 0. & 0. \\ 0. & 0. & 0. & -1.091 & 0. & 0.00008 & 0. & 0. \\ 5. & 0. & 1. & 0. & -0.0603 & 0. & 0. & 0. \\ 0. & 2.5 & 0. & 1. & 0. & -0.00088 & 0. & 0. \\ 0.1 & 0.006 & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0.3 & 0.09 & 0. & 0. & 0. & 0. \end{pmatrix}.$$

In the following subsections, we compare the computational behavior of reduced and/or parareal algorithms with respect to the fine algorithm. Let us note that for all computations, the speed-up is defined as the ratio

$$\frac{\text{CPU (fine scheme)}}{\text{CPU (current scheme)}},$$

where the CPU of the currently studied scheme takes into account the initialization step and neglects the communications between processors since we only simulate parareal implementation, and not perform actual parareal computations.

3.3.1. *Parareal algorithm vs. fine algorithm.* The computations are first performed up to final time $T = 3$ with the following parameters for the parareal algorithm. The coarse grid has $N = 50$ cells which constitute a regular subdivision of $[0, 3]$, so that the coarse time step is $\Delta T = 0.06$. Then each coarse cell is divided into a regular fine subdivision of $s = 500$ cells, so that the fine time step is $\delta t = 0.00012$. The numerical tolerance is set to 0.01. We use Runge–Kutta RK4 for the fine
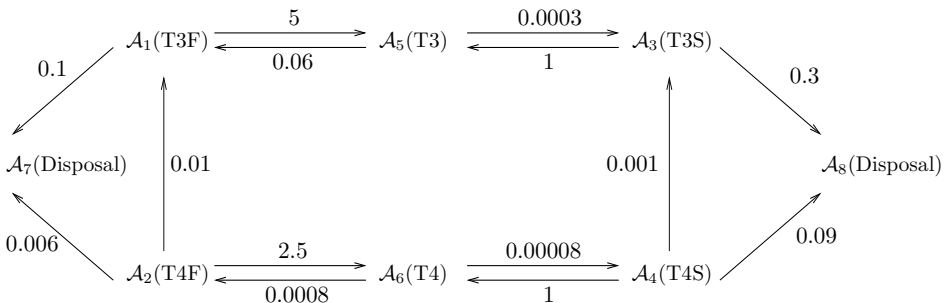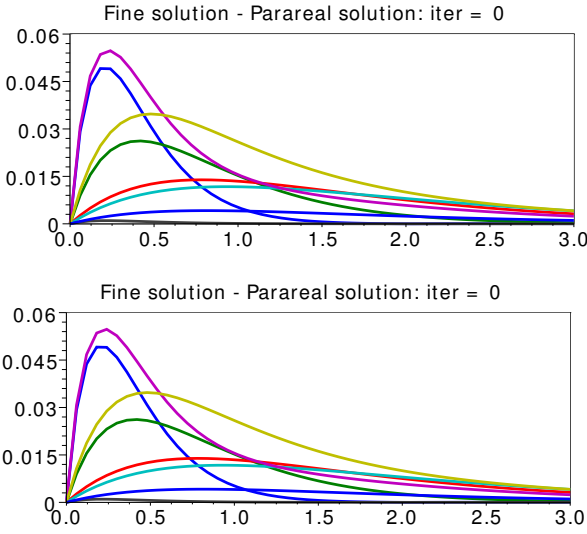

Figure 1. The thyroid reaction scheme.

Figure 2. Thyroid: numerical error between the fine solution and
(top) the coarse initialized solution, and (bottom) the parareal so-
lution after 3 parareal iterations.

scheme and the implicit Euler for the coarse scheme. The speed-up we obtain is
approximately 12.

Figure 2 shows the error at the initialization step (top) and the error between the
fine and parareal solutions (bottom). We observe that the maximal error is divided
by 1000 after three iterations of the parareal algorithm.

Moreover, we check numerically (Figure 3) that the two stoichiometric invari-
ants of this problem still hold. The plots of both invariants are exactly superim-
posed, since both fine and coarse solvers in the parareal algorithm conserve the
stoichiometric invariants.
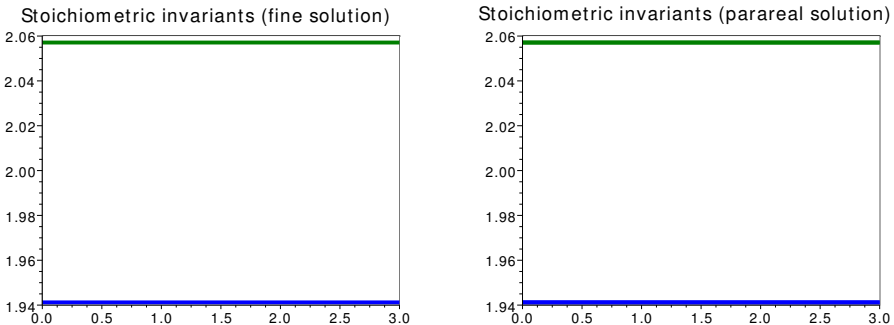
Figure 3. Thyroid: stoichiometric invariants.
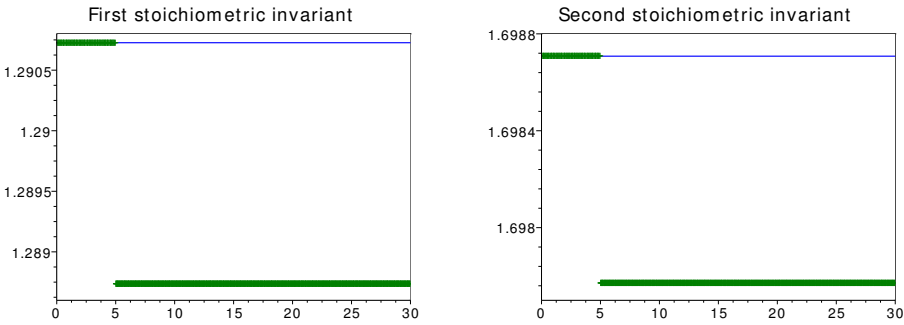
Figure 4. Thyroid: stoichiometric invariants with reduction.

3.3.2. *Reduction algorithm vs. fine algorithm.* We now use the reduction method described in Section 3.2. Matrix J has four dominant eigenvalues of the same magnitude, lying between -6 and -1. The remaining eigenvalues are either 0, with multiplicity 2, or very small with respect to 1. We compare the results obtained with the fine solver and by applying the reduction method. For the reduction, we compute the solution of the full system up to the characteristic time $T^* = 5$, then the solution of the reduced system up to final time $T = 30$, including the algebraic equations.

One can check on Figure 4 that the stoichiometric invariants are not conserved. However, the jumps in the stoichiometric invariants are small. We also note that computations with the reduction method are 2.5 times faster compared to those obtained by the fine solver.

3.3.3. *Parareal algorithm with reduction method.* We use the values $T^* = 10$ and $T = 30$ as in 3.3.2. For the parareal algorithm, the coarse solver is, on $[0, T^*]$ (respectively on $[T^*, T]$), the implicit Euler scheme for the full problem with a coarse time step $\Delta T_1 = 2$ (respectively for the reduced problem with a coarse time step $\Delta T_2 = 4$). The fine solver is still RK4, with a fine time step $\delta t_1 = 0.02$ on $[0, T^*]$, and $\delta t_2 = 0.04$ on $[T^*, T]$. The tolerance is set at 0.01. The method converges in three iterations and we obtain a speed-up of 4.3 for 10 processors, that is, an efficiency of 0.43.

On Figure 5, we can check that the error between the parareal algorithm and the reduction method decreases with the number of parareal iterations. Eventually, we note on Figure 6 that the stoichiometric invariants are not conserved. Hence, a basic parareal algorithm connected with the reduction method does not conserve the stoichiometric invariants.

3.3.4. *A modified parareal algorithm preserving stoichiometric invariants.* Recall that a stoichiometric invariant γ is a linear combination of concentrations (y_i) such that its time derivative is nil.
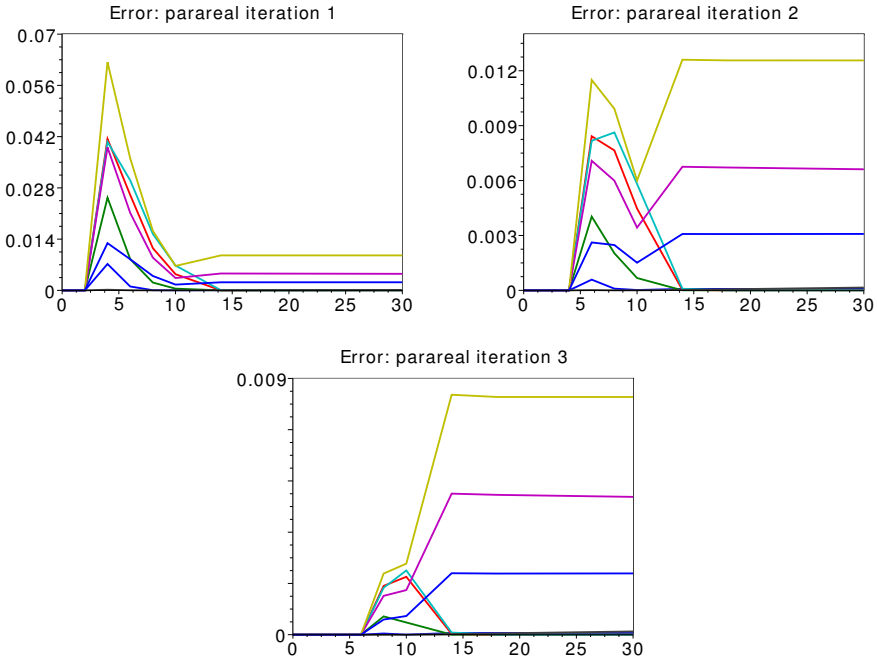
Figure 5. Thyroid: errors during the parareal iterations.

In other words, there exists a nonzero vector $w \in \ker(S^T)$ such that $\gamma = \langle w, y \rangle$ and

$$\frac{d\gamma}{dt} = \langle w, Sy(t) \rangle = \langle S^T w, y(t) \rangle = 0.$$

This shows that $y(t)$ belongs to the affine space $y(0) + \mathrm{Range}(S)$.

In order to conserve the stoichiometric invariants, we consider the modified parareal algorithm

$$y_{n+1}^{k+1} = \mathcal{F}_{\delta t}(y_n^k) + \Pi(\mathcal{C}_{\Delta T}(y_n^{k+1}) - \mathcal{C}_{\Delta T}(y_n^k)) \tag{9}$$
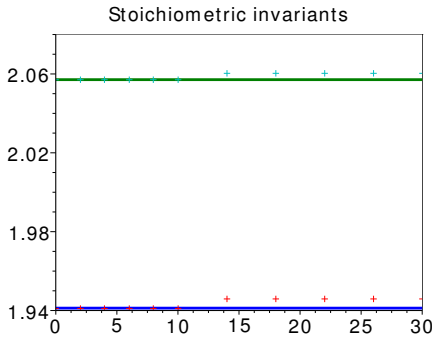
Figure 6. Thyroid: fine (solid) and parareal ($+$) stoichiometric invariants.
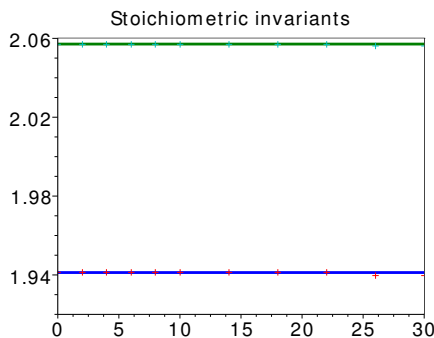
Figure 7. Thyroid: fine (solid) and new parareal $(+)$ stoichiometric invariants with the improved parareal algorithm.

where Π is the orthogonal projection on the range of J. Nonorthogonal projections can be used too.

Of course, one can then check on Figure 7 that the stoichiometric invariants are now conserved. Moreover, the algorithm is still a good approximation of the fine solution with the same number of iterations and the same speed-up.

4. Example of nonlinear chemistry: an ozone model

4.1. *Description of the model.*

4.1.1. *Ozone model.* The ozone model we investigate here is described in detail in [1]. It involves 16 species and 12 reactions, that makes it a rather simple model, though realistic. The kinetic constant of reaction r is denoted k_r, $1 \leq r \leq 12$. The chemical reactions are

$$OD + air + O_2 \xrightarrow{k_1} O_3 + air + O_2, \qquad RCHO + OH \xrightarrow{k_7} RCO_3,$$

$$O_3 + NO \xrightarrow{k_2} NO_2, \qquad RCHO \xrightarrow{k_8} RO_2 + CO + HO_2,$$

$$NO + HO_2 \xrightarrow{k_3} NO_2 + OH, \qquad NO + RO_2 \xrightarrow{k_9} NO_2 + RCHO + HO_2,$$

$$OH + NO_2 \xrightarrow{k_4} NHO_3, \qquad NO + RCO_3 \xrightarrow{k_{10}} NO_2 + RO_2 + CO_2,$$

$$NO_2 \xrightarrow{k_5} NO + OD, \qquad RCO_3 + NO_2 \xrightarrow{k_{11}} RCO_3NO_2,$$

$$RH + OH \xrightarrow{k_6} RO_2, \qquad RCO_3NO_2 \xrightarrow{k_{12}} RCO_3 + NO_2.$$
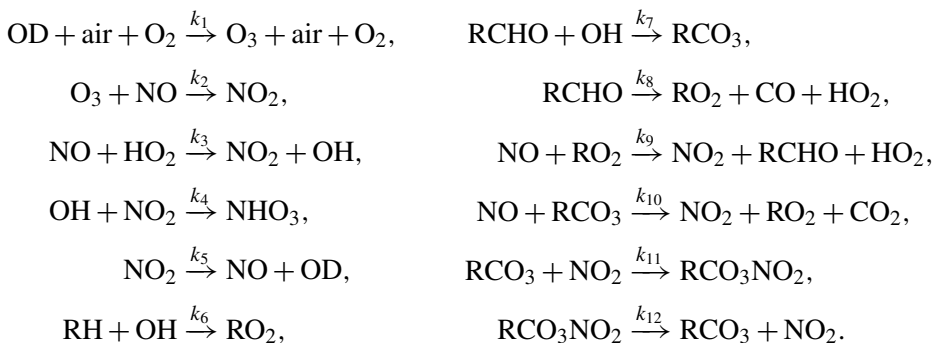
The previous scheme is already somehow reduced, since the concentrations of air, O_2 and H_2O (which does not appear in this system, but is necessary) are considered as very high constants. Therefore, some reactions do not seem to be balanced. That only means that we may not take into account those three species in

the previous reactions. Note also that CO_2, NHO_3 and CO are not reactants. The values of the kinetic constants are the following:

$$
\begin{array}{llllll}
k_1 &=& 10^{-33}, & k_5 &=& 8.9\ 10^{-3}, & k_9 &=& 7.6\ 10^{-12}, \\
k_2 &=& 2\ 10^{-14}, & k_6 &=& 2.6\ 10^{-12}, & k_{10} &=& 7.6\ 10^{-12}, \\
k_3 &=& 8.2\ 10^{-12}, & k_7 &=& 1.6\ 10^{-11}, & k_{11} &=& 4.7\ 10^{-12}, \\
k_4 &=& 1.1\ 10^{-11}, & k_8 &=& 3.2\ 10^{-6}, & k_{12} &=& 4\ 10^{-4}.
\end{array}
$$

Each species is denoted by an integer index, as follows:

index	1	2	3	4	5	6	7	8
species	air	O_2	CO_2	NHO_3	RH	CO	NO	NO_2

index	9	10	11	12	13	14	15	16
species	RCO_3NO_2	RCHO	O_3	OH	HO_2	RCO_3	RO_2	OD

The vector $y = (y_i)_{1 \le i \le 16} \in \mathbb{R}^{16}$, whose coordinates y_i are the concentrations of the species represented in the previous table solves the following differential system

$$
\begin{array}{ll}
y_1' = 0, & y_9' = v_{11} - v_{12}, \\
y_2' = 0, & y_{10}' = -v_7 - v_8 + v_9, \\
y_3' = v_{10}, & y_{11}' = v_1 - v_2, \\
y_4' = v_4, & y_{12}' = v_3 - v_4 - v_6 - v_7, \\
y_5' = -v_6, & y_{13}' = -v_3 + v_8 + v_9, \\
y_6' = v_8, & y_{14}' = v_7 - v_{10} - v_{11} + v_{12}, \\
y_7' = -v_2 - v_3 + v_5 - v_9 - v_{10}, & y_{15}' = v_6 + v_8 - v_9 + v_{10}, \\
y_8' = v_2 + v_3 - v_4 - v_5 + v_9 + v_{10} - v_{11} + v_{12}, & y_{16}' = -v_1 + v_5,
\end{array}
$$

where $v = (v_r)_{1 \le r \le 12}$ denotes the reaction rate vector, depending on y.

More precisely, we have

$$
\begin{array}{ll}
v_1 = k_1 y_1 y_2 y_{16}, & v_2 = k_2 y_7 y_{11}, \\
v_3 = k_3 y_7 y_{13}, & v_4 = k_4 y_8 y_{12}, \\
v_5 = k_5 y_8, & v_6 = k_6 y_5 y_{12}, \\
v_7 = k_7 y_{10} y_{12}, & v_8 = k_8 y_{10}, \\
v_9 = k_9 y_7 y_{15}, & v_{10} = k_{10} y_7 y_{14}, \\
v_{11} = k_{11} y_8 y_{14}, & v_{12} = k_{12} y_9.
\end{array}
$$

The differential system can be rewritten under the form

$$
y' = Sv, \tag{10}
$$

where $S = (S_{i,r})_{1 \leq i \leq 16, 1 \leq r \leq 12}$ is the stoichiometric matrix. Equation (10) is clearly nonlinear, since each v_r nonlinearly depends on y.

Figures 8–9 show the evolution of the concentrations of all the species involved in the model, directly computed with the software Scilab, within two time scales,
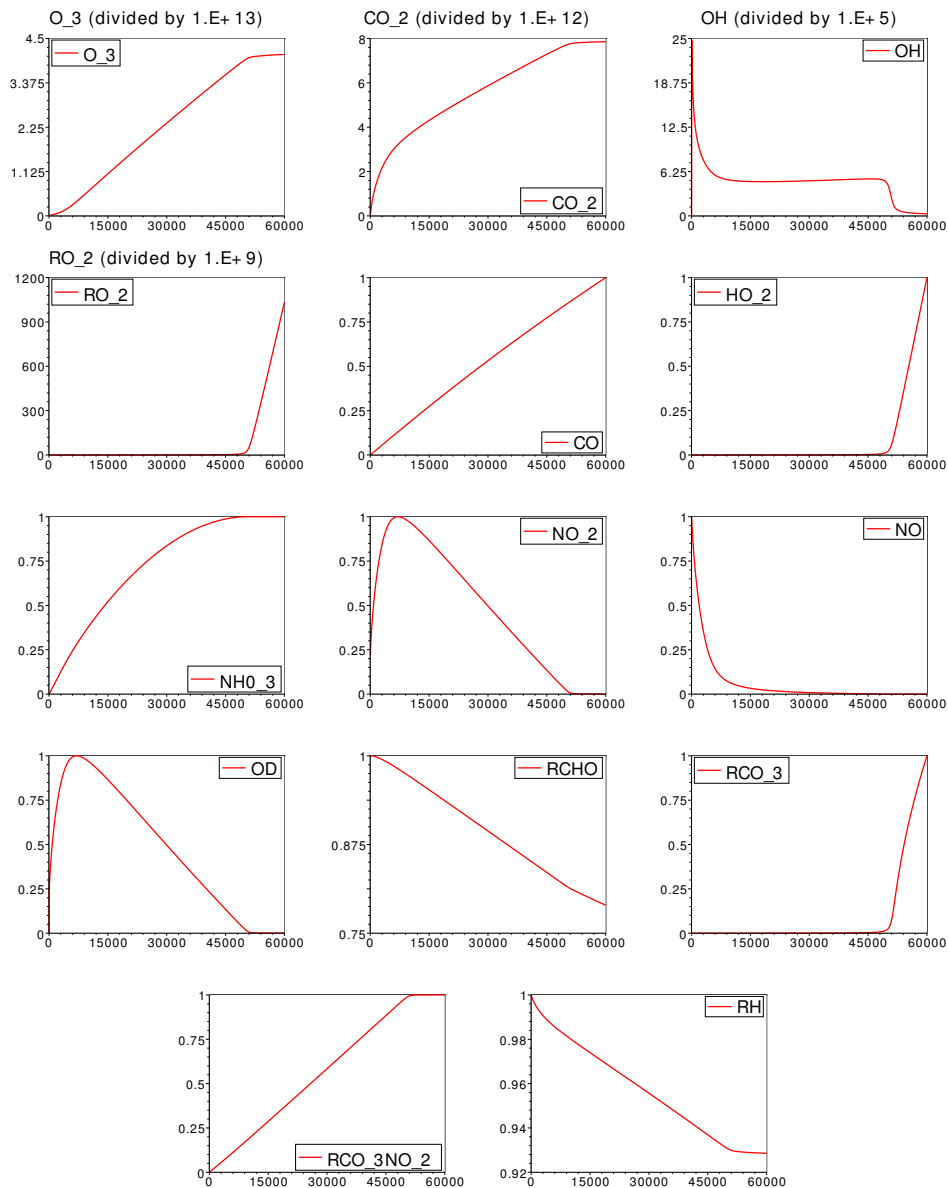

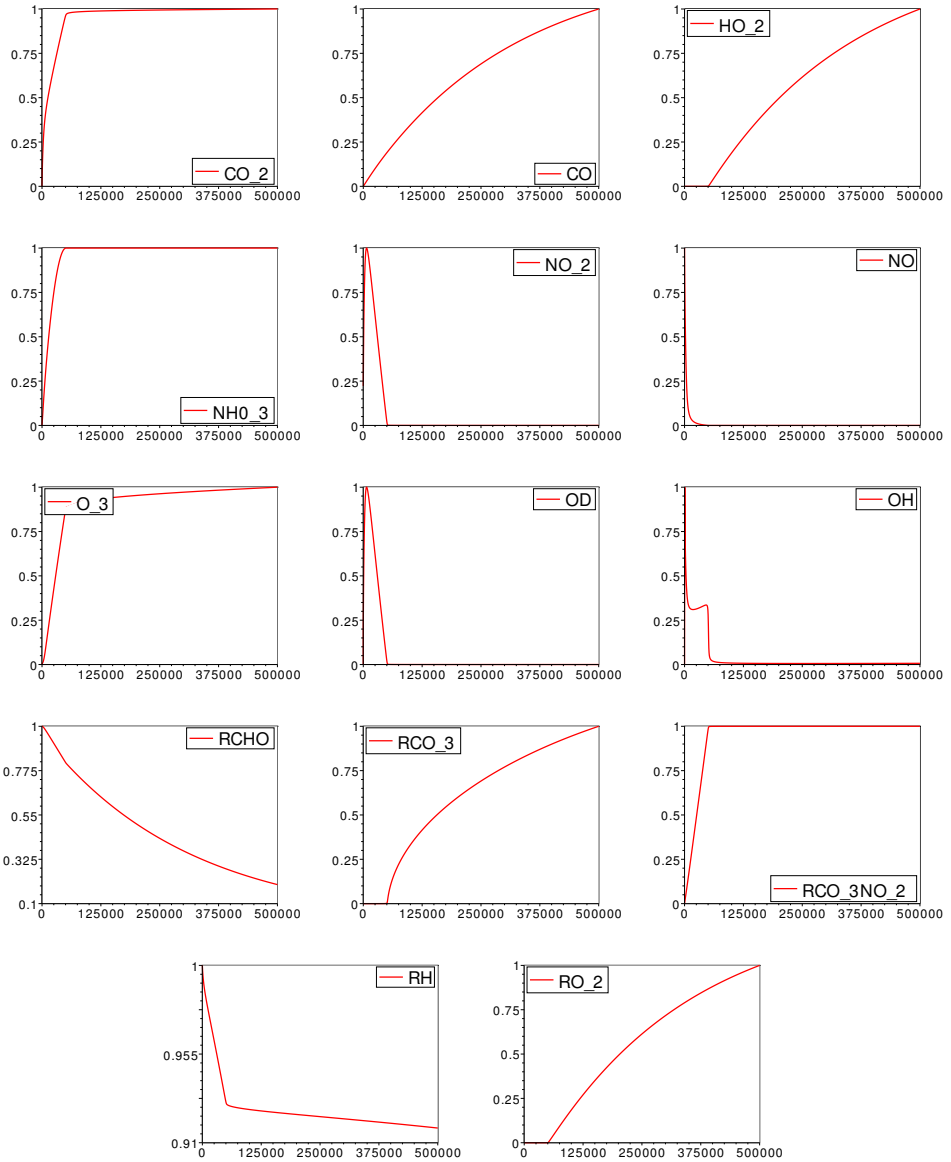**Figure 8.** Ozone: Normalized concentrations of all species (final time 16 h 40 min).

Figure 9. Ozone: Normalized concentrations of all the species (final time 5 days and 19 h).

with respective final times 16 h 40 min, and 5 days and 19 h. (Note that the model may not hold anymore at the second time scale: this is discussed in [1; 4].) The concentrations of air and O_2 are not plotted, since they remain constant. The initial values of the concentrations are:

air	O_2	CO_2	NHO_3	RH	CO	NO	NO_2
$2.45\ 10^{19}$	$4.18\ 10^{18}$	100	100	$5\ 10^{13}$	100	$1.23\ 10^{13}$	$2.5\ 10^{12}$
RCO_3NO_2	RCHO	O_3	OH	HO_2	RCO_3	RO_2	OD
100	$5\ 10^{13}$	100	200	100	300	200	100

We obviously recover the results from [1] with the same set of initial data.

4.1.2. *Stoichiometric invariants.* Since the rank of matrix S is 10, there are six stoichiometric invariants in the model, which can be chosen as follows [1]:

$$\frac{d}{dt} y_1 = 0, \tag{11}$$

$$\frac{d}{dt} y_2 = 0, \tag{12}$$

$$\frac{d}{dt}(y_4 + y_7 + y_8 + y_9) = 0, \tag{13}$$

$$\frac{d}{dt}(y_5 + y_9 + y_{10} + y_{14} + y_{15}) = 0, \tag{14}$$

$$\frac{d}{dt}(y_4 - 2y_6 + y_9 + y_{12} + y_{13} + y_{14} + y_{15}) = 0, \tag{15}$$

$$\frac{d}{dt}(-3y_3 - 3y_6 - y_7 - 2y_9 - 2y_{10} + y_{11} + y_{13} - 2y_{14} + y_{16}) = 0. \tag{16}$$

The first two invariants are immediate, and equations (13)–(14) come from the conservation of species involving N and R radicals. Invariants (15)–(16) are numerically controlled with the same computation. More precisely, we can see in Figure 10 that the invariant (15) is not as well conserved as invariant (16). This is a consequence of the computation. Indeed, some concentrations involved in (15)–(16) are of very high order of magnitude ($\sim 10^{14}$). The numerical variations of a
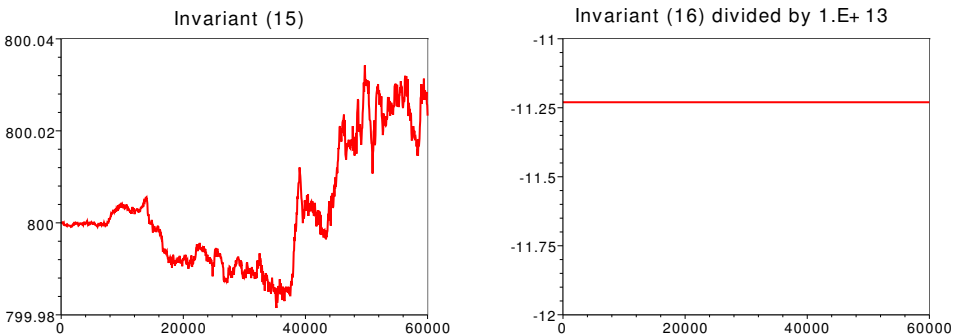
Figure 10. Ozone: conservation of invariants (15) and (16).
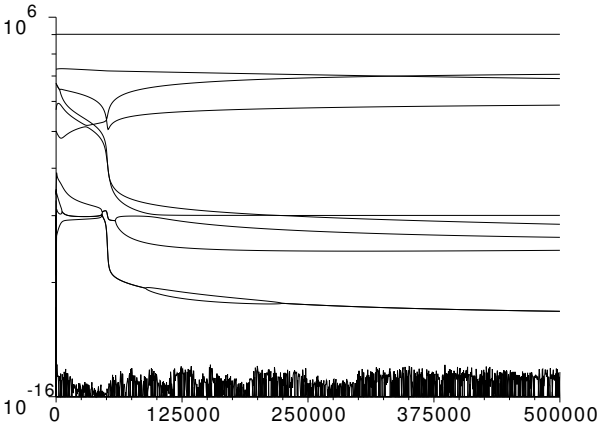
Figure 11. Ozone: evolution of the spectrum of the Jacobian matrix.

linear combination of such quantities, whose value is approximately 10^3, clearly involve unavoidable numerical errors.

4.1.3. *Stiffness.* We recall that the differential system (10) is stiff if there are some eigenvalues of its Jacobian matrix $(\partial_{y_i}(Sv)_j)_{1 \leq i,j \leq 16}$ whose real parts are not of the same order of magnitude with respect to the other eigenvalues. We can check on Figure 11 that there are mainly three orders of magnitude for the eigenvalues. Note that there are a lot of oscillations for the smaller eigenvalues, again due to numerical errors.

4.2. *A reduction method: the quasisteady states.* There is no systematic method to obtain, from a nonlinear stiff differential system, a reduced model giving fine numerical approximations at any time. One solution consists in linearizing the system in a neighborhood of a given stationary point [6]. Other possibilities exist, such as the quasisteady state assumption on some species, or the partial equilibrium assumption for some reactions [27].

In the section, we focus on the quasisteady state method. Before applying it to the ozone model, let us briefly recall its mechanism. Denote by A an intermediary compound in a given chain of reactions. The evolution of its concentration is governed by

$$\frac{dy_A}{dt} = p - c,$$

where p and c are respectively the production and consumption of A. The species A is in a quasisteady state when its production rate is very close to its destruction rate, more precisely, if the quasistationary index, defined by

$$\mathcal{I}_A = \frac{|p - c|}{p + c},$$

is such that $\mathscr{I}_A \ll 1$. This index only gives an a posteriori criterion to select the quasisteady state species. Moreover, the quasisteady state of the species A does not mean that the concentration of A is constant.

In [1], using the a posteriori criterion defined above, one gets five quasistationary species in the ozone model: OH, HO_2, RCO_3, RO_2 and OD, for times smaller than 16 h and 40 min. In that situation, the concentration of OD, which is now denoted z_{16}, can be directly computed in terms of $(y_i)_{1 \leq i \leq 11}$:

$$z_{16} = \frac{k_5 \, y_8}{k_1 \, y_1 \, y_2}.$$

The concentrations of the other quasistationary species, also denoted $(z_i)_{12 \leq i \leq 15}$, depend on each other. In fact, $(z_i)_{13 \leq i \leq 15}$ can be written in terms of z_{12}, more precisely, we have

$$z_{15} = \frac{(k_4 \, y_8 + k_6 \, y_5 + k_7 \, y_{10})z_{12} - k_8 \, y_{10}}{k_9 \, y_7}, \quad z_{14} = \frac{(k_4 \, y_8 + k_7 \, y_{10})z_{12} - 2k_8 \, y_{10}}{k_{10} \, y_7},$$

$$z_{13} = \frac{(k_4 \, y_8 + k_6 \, y_5 + k_7 \, y_{10})z_{12}}{k_3 \, y_7},$$

with

$$z_{12} = \frac{2k_8 \, y_{10}(k_{10} \, y_7 + k_{11} \, y_8) + k_{10} \, k_{12} \, y_7 \, y_9}{(k_4 \, y_8 + k_7 \, y_{10})(k_{10} \, y_7 + k_{11} \, y_8) - k_7 \, k_{10} \, y_7 \, y_{10}}.$$

The five previous equalities come from (10), where we put $(Sv)_i = 0$ for $12 \leq i \leq 16$.

Note that, beyond 16 h and 40 min, the quasistationary species are not the same, and that the model itself does not hold anymore. Hence, in the sequel, we only present computations on times smaller than 16 h and 40 min, when we make the quasisteady state assumption.

4.3. Numerical tests. We apply the parareal algorithm (4) to solve the equations of the ozone model. We use the implicit Euler scheme for the fine and the coarse solvers, with different time steps. The parareal iterations are stopped as soon as the sum of the relative errors on each concentration (between parareal iterations k and $k + 1$) is smaller than the numerical tolerance, which is set to 0.05. In the sequel, we only focus on the concentrations of four species: O_3, CO_2, OH and RO_2, for the sake of simplicity. Of course, the behaviors of the remaining species concentrations have been checked too.

4.3.1. Parareal algorithm vs. fine algorithm: Test 1. We compute the solution in the interval [0, 16 h 40 min]. One processor was used to solve the problem in the tiny interval [0, 0.01 h] to capture the first boundary layers, 9 processors for the computation in [0.01 h, 2 h 45 min], and 10 processors for the computation in [2 h 45 min, 16 h 40 min]. The parareal algorithm converges after only 1 iteration and the parareal computation is about 18 times faster than the fine computation
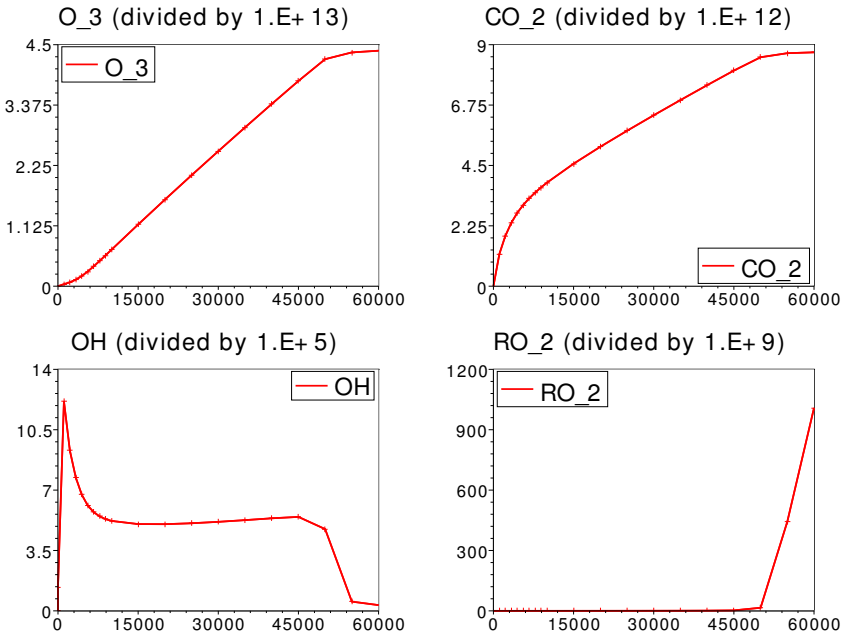
Figure 12. Ozone: fine (solid) and parareal (+) solutions on [0, 16 h 40 min].

obtained with the fine solver on the whole interval [0, 16 h 40 min]. We display the evolution of the species of interest on Figure 12.

4.3.2. *Parareal algorithm vs. fine algorithm: test 2.* We compute the solution over a long period of time [0, 5d19h]. The parareal computation converges after 2 iterations. It is 31 times faster than the fine computation on the whole interval [0, 5d19h]. Let us precise that we have used 170 processors for these computations: only one processor on [0, 0.01], 9 on [0.01, 1], 10 on [1, 10] and 150 processors on [10, 5d19h]. The concentrations of some species are shown on Figure 13.

4.3.3. *Parareal algorithm coupled with reduction vs. fine algorithm.* As already noted in Section 4.2, we focus on the time interval [0, 16 h 40 min]. The computational parameters are the same as in Section 4.3.1. The parareal algorithm is coupled with the QSS reduction, that is, the coarse solver uses the full system up to 2h45min, and the reduced system beyond that time, whereas the fine solver remains the same. The algorithm converges after 3 iterations. The parareal computation is about 4 times faster than the fine computation obtained with the fine scheme on the whole interval [0, 16 h 40 min]. The evolution of some species is shown on Figure 14 and the stoichiometric invariants are quite well conserved, see Figure 15. Note that the relative errors are at most of order 10^{-4}.
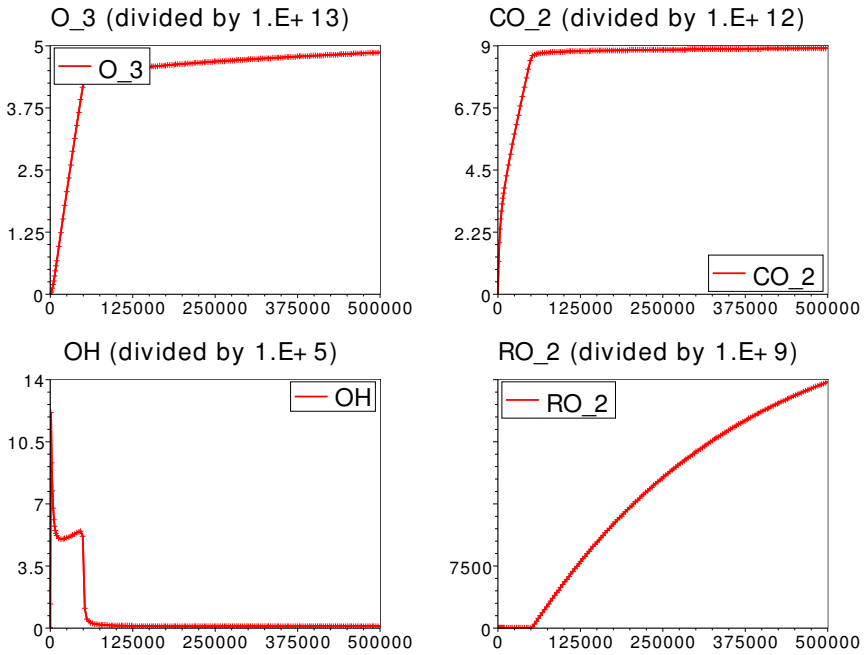
Figure 13. Ozone: fine (solid) and parareal (+) solutions on [0, 5 d 19 h].
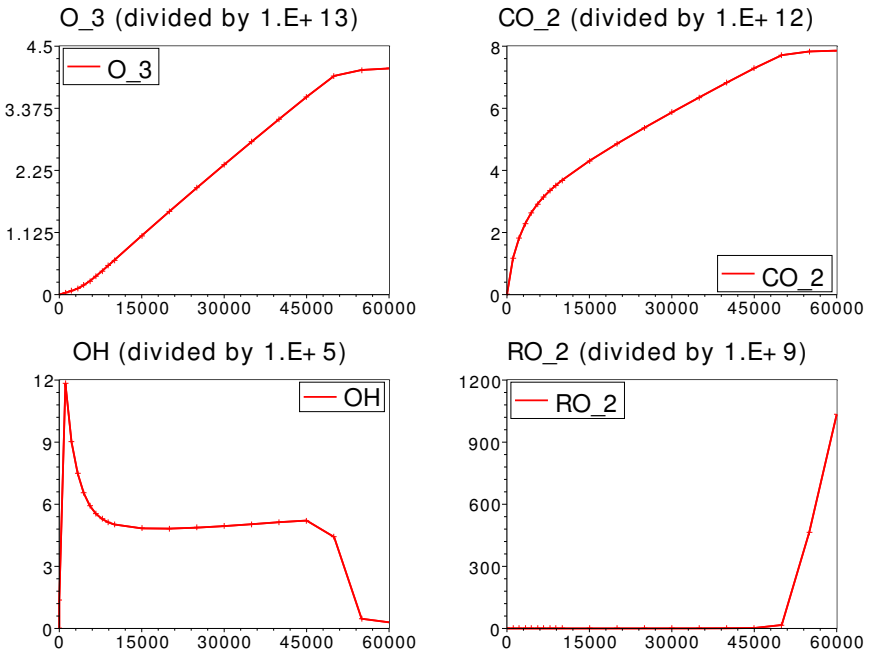
Figure 14. Ozone: fine (solid) and reduced parareal (+) solutions on [0, 16 h 40 min].
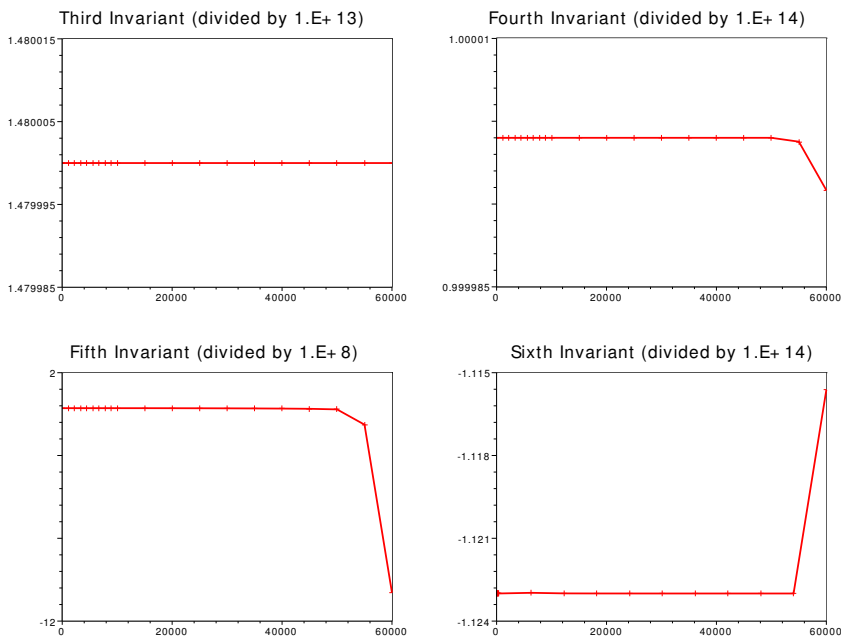
Figure 15. Ozone: fine (solid) and reduced parareal (+) stoichio-metric invariants on [0, 16 h 40 min].

In order to decrease the numerical errors on Figure 15, we proceed in the same way as in Section 3.3.4 to modify the parareal algorithm, that is, we add an orthogonal projection to ensure a better conservation of the stoichiometric invariants; see Figure 16.

5. Concluding remarks

In chemical kinetics, the parareal algorithms coupled with reduction methods provide an essential tool to solve stiff differential systems with accuracy. Numerical tests point out the efficiency of the parareal approach in both linear and nonlinear cases. Let us note that one can ensure a better numerical conservation of the stoichiometric invariants by adding a projection in the standard parareal algorithm.

Acknowledgement
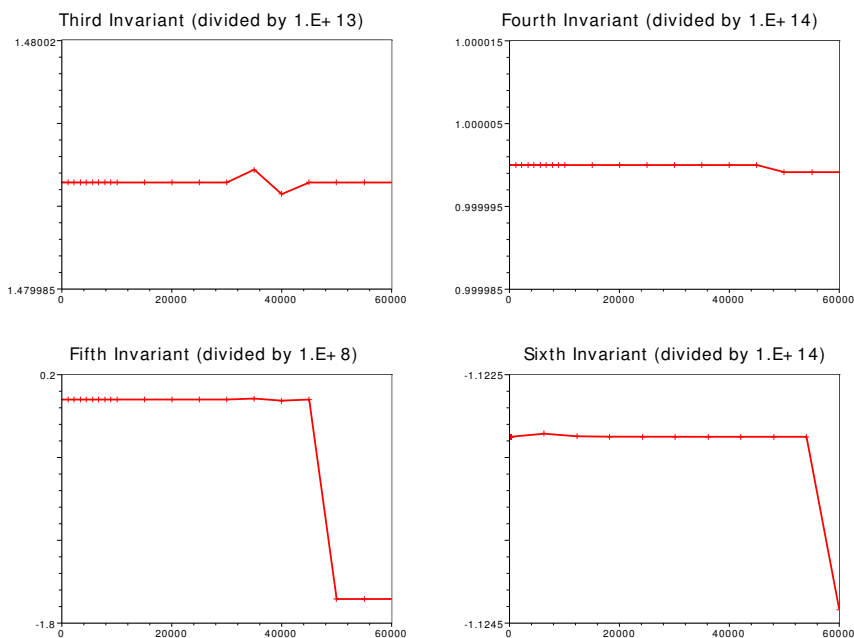
Figure 16. Ozone: fine (solid) and reduced parareal with projection (+) stoichiometric invariants on [0, 16 h 36 min].

References

[1]   P. Ayoub, A. Bamberger, and Z. Benjelloun-Dabaghi, *Étude mathématique et numérique complète pour la réduction d'un schéma cinétique de production d'ozone*, technical report, Institut Français du Pétrole, 1994.

[2]   G. Bal, *On the convergence and the stability of the parareal algorithm to solve partial differential equations*, Domain decomposition methods in science and engineering (R. Kornhuber et al., eds.), Lect. Notes Comput. Sci. Eng., no. 40, Springer, Berlin, 2005, pp. 425–432. MR 2235769

[3]   G. Bal and Y. Maday, *A "parareal" time discretization for non-linear PDE's with application to the pricing of an American put*, Recent developments in domain decomposition methods (L. Pavarino and A. Toselli, eds.), Lect. Notes Comput. Sci. Eng., no. 23, Springer, Berlin, 2002, pp. 189–202. MR 1962689

[4]   A. Bamberger and Z. Benjelloun-Dabaghi, *Étude mathématique d'un schéma cinétique de la production de l'ozone*, technical report, Institut Français du Pétrole, 1994.

[5]   A. Blouza, F. Coquel, and F. Hamel, *Reduction of linear kinetic systems with multiple scales*, Combust. Theory Model. **4** (2000), no. 3, 339–362. MR 2001h:80009

[6]   V. I. Bykov, V. I. Dimitri, and A. N. Gorban, *Marcelin-de Donder kinetics near equilibrium*, React. Kin. Catal. Lett. **12** (1979), no. 1, 19–23.

[7]   S. L. Campbell and N. J. Rose, *Singular perturbation of autonomous linear systems*, SIAM J. Math. Anal. **10** (1979), no. 3, 542–551. MR 80i:34089a

[8]   M. Coderch, A. S. Willsky, S. S. Sastry, and D. A. Castañon, *Hierarchical aggregation of linear systems with multiple time scales*, IEEE Trans. Automat. Control **28** (1983), no. 11, 1017–1030. MR 84k:93007

[9]   C. Farhat and M. Chandesris, *Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications*, Internat. J. Numer. Methods Engrg. **58** (2003), no. 9, 1397–1434. MR 2004h:65154

[10] C. Farhat, J. Cortial, C. Dastillung, and H. Bavestrello, *Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses*, Internat. J. Numer. Methods Engrg. **67** (2006), no. 5, 697–724. MR 2007a:74041

[11] P. F. Fischer, F. Hecht, and Y. Maday, *A parareal in time semi-implicit approximation of the Navier–Stokes equations*, Domain decomposition methods in science and engineering (R. Kornhuber et al., eds.), Lect. Notes Comput. Sci. Eng., no. 40, Springer, Berlin, 2005, pp. 433–440. MR 2235770

[12] M. Gander and M. Petcu, *Analysis of a Krylov subspace enhanced parareal algorithm for linear problems*, Paris–Sud Working Group on Modelling and Scientific Computing 2007–2008 (E. Cancés et al., eds.), ESAIM Proc., no. 25, EDP Sci., Les Ulis, 2008, pp. 114–129. MR 2010i:65119

[13] M. J. Gander, *Analysis of the parareal algorithm applied to hyperbolic problems using characteristics*, Bol. Soc. Esp. Mat. Apl. SēMA (2008), no. 42, 21–35. MR 2009b:65268

[14] M. J. Gander and E. Hairer, *Nonlinear convergence analysis for the parareal algorithm*, Domain decomposition methods in science and engineering XVII (M. Bercovier, ed.), Lect. Notes Comput. Sci. Eng., no. 60, Springer, Berlin, 2008, pp. 45–56. MR 2009j:65165

[15] M. J. Gander and S. Vandewalle, *Analysis of the parareal time-parallel time-integration method*, SIAM J. Sci. Comput. **29** (2007), no. 2, 556–578. MR 2008c:65386

[16] ———, *On the superlinear and linear convergence of the parareal algorithm*, Domain decomposition methods in science and engineering XVI (O. Widlund and D. Keyes, eds.), Lect. Notes Comput. Sci. Eng., no. 55, Springer, Berlin, 2007, pp. 291–298. MR 2008g:65103

[17] S. H. Lam and D. A. Goussis, *The CSP method for simplifying kinetics*, Int. J. Chem. Kinet. **26** (1994), 461–486.

[18] J.-L. Lions, Y. Maday, and G. Turinici, *Résolution d'EDP par un schéma en temps "pararéel"*, C. R. Acad. Sci. Paris Sér. I Math. **332** (2001), no. 7, 661–668. MR 2002c:65140

[19] Y. Liu and J. Hu, *Modified propagators of parareal in time algorithm and application to Princeton ocean model*, Internat. J. Numer. Methods Fluids **57** (2008), no. 12, 1793–1804. MR 2009g:86009

[20] U. Maas and S. B. Pope, *Simplifying chemical kinetics: intrinsic low-dimensional manifolds in composition space*, Combust. flame **88** (1992), 239–264.

[21] Y. Maday, *Parareal in time algorithm for kinetic systems based on model reduction*, High-dimensional partial differential equations in science and engineering (A. Bandrauk et al., eds.), CRM Proc. Lecture Notes, no. 41, Amer. Math. Soc., Providence, RI, 2007, pp. 183–194. MR 2009b:65162

[22] Y. Maday, J. Salomon, and G. Turinici, *Monotonic parareal control for quantum systems*, SIAM J. Numer. Anal. **45** (2007), no. 6, 2468–2482. MR 2008k:81003

[23] H. Mak and J. J. DiStefano, *Optimal control policies for the prescription of thyroid hormones*, Math. Biosciences **42** (1978), 159–186.

[24] B. Sportisse, *Modélisation et simulation de la pollution atmosphérique*, habilitation à diriger des recherches, Université Pierre et Marie Curie, Paris, 2007.

[25] G. A. Staff and E. M. Rønquist, *Stability of the parareal algorithm*, Domain decomposition methods in science and engineering (R. Kornhuber et al., eds.), Lect. Notes Comput. Sci. Eng., no. 40, Springer, Berlin, 2005, pp. 449–456. MR 2235772

[26] R. S. Varga, *Matrix iterative analysis*, expanded ed., Springer Series in Computational Mathematics, no. 27, Springer, Berlin, 2000. MR 2001g:65002

[27] F. A. Williams, *Combustion theory: the fundamental theory of chemically reacting flow systems*, Benjamin-Cummings Publishing Company, 1985.

ADEL BLOUZA: adel.blouza@univ-rouen.fr
Laboratoire de Mathématiques Raphaël Salem, UMR 6085 CNRS–Université de Rouen, 76801 Saint-Etienne-du-Rouvray, France

LAURENT BOUDIN: laurent.boudin@upmc.fr
Laboratoire J.-L. Lions, Université Pierre et Marie Curie, 75252 Paris cedex 05, France

SIDI MAHMOUD KABER: sidi-mahmoud.kaber@upmc.fr
Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie, 75252 Paris cedex 05, France

Communications in Applied Mathematics and Computational Science

pjm.math.berkeley.edu/camcos

See inside back cover or pjm.math.berkeley.edu/camcos for submission instructions.

Communications in Applied Mathematics and Computational Science

vol. 5   no. 2   2010