A MICRO-MACRO PARAREAL ALGORITHM: APPLICATION TO SINGULARLY PERTURBED ORDINARY DIFFERENTIAL EQUATIONS

FRÉDÉRIC LEGOLL*, TONY LELIÈVRE†, AND GIOVANNI SAMAEY‡

Abstract. We introduce a micro-macro parareal algorithm for the time-parallel integration of multiscale-in-time systems. The algorithm first computes a cheap, but inaccurate, solution using a coarse propagator (simulating an approximate slow macroscopic model), which is iteratively corrected using a fine-scale propagator (accurately simulating the full microscopic dynamics). This correction is done in parallel over many subintervals, thereby reducing the wall-clock time needed to obtain the solution, compared to the integration of the full microscopic model over the complete time interval. We provide a numerical analysis of the algorithm for a prototypical example of a micro-macro model, namely singularly perturbed ordinary differential equations. We show that the computed solution are better and better approximations of the full microscopic solution (when the parareal iterations proceed) only if special care is taken during the coupling of the microscopic and macroscopic levels of description. The error bound depends on the modeling error of the approximate macroscopic model. We illustrate these results with numerical experiments.

1. Introduction. In many applications, a system is modeled using a high-dimensional system of differential equations that captures phenomena occurring at multiple time scales. Unfortunately, the computational cost of simulating such fine-scale systems (which we call *microscopic* in this work) on macroscopic time intervals is prohibitive, and one often resorts to low-dimensional, coarse-grained, effective models (which we call *macroscopic*), in which the fast degrees of freedom are eliminated. Many methods have been proposed to obtain such macroscopic models, either analytically (see e.g. [34] for a recent overview) or numerically. We refer, for instance, to the work on equation-free [22, 23] or heterogeneous multiscale methods [8, 9], and references therein. However, by construction, these macroscopic models only capture the original full microscopic dynamics *approximately*.

Here, we present and analyze a numerical multiscale method that aims at efficiently simulating the full microscopic dynamics (and not a macroscopic approximation of it) over long time intervals, using an effective (approximate) macroscopic model as a predictor and the microscopic model as a corrector. To this end, we propose a micro-macro version of the parareal algorithm [26]. The parareal algorithm was originally proposed to solve time-dependent problems using computations in parallel, aiming at exploiting the presence of multiple processors to reduce the real (wall-clock) time needed to obtain a solution on a long time interval. It is based on a decomposition of the time interval into subintervals, and makes use of a predictor-corrector strategy, in which the calculation of the corrections is performed concurrently on the different processors that are available. In what follows, we propose a version of this algorithm well-adapted to our multiscale-in-time context.

For the sake of clarity, and to better describe our aim, we now present the parareal

*Laboratoire Navier, École Nationale des Ponts et Chaussées, Université Paris-Est, 6 et 8 avenue Blaise Pascal, 77455 Marne-La-Vallée Cedex 2, France; INRIA Rocquencourt, MICMAC team-project, Domaine de Voluceau, B.P. 105, 78153 Le Chesnay Cedex, France

†CERMICS, École Nationale des Ponts et Chaussées, Université Paris-Est, 6 et 8 avenue Blaise Pascal, 77455 Marne-La-Vallée Cedex 2, France; INRIA Rocquencourt, MICMAC team-project, Domaine de Voluceau, B.P. 105, 78153 Le Chesnay Cedex, France

‡Scientific Computing, Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium Scientific Computing, Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium

algorithm in some detail. To fix the ideas, assume that the problem at hand is

$$\frac{du}{dt} = f(u), \quad u(0) = u_0, \quad u(t) \in \mathbb{R}^d, \quad t \in [0, T], \tag{1.1}$$

the exact flow of which is denoted $u(t) = \mathcal{E}_t(u_0)$. Suppose that we have at hand two propagators to integrate (1.1), $\mathcal{F}_{\Delta t}$ and $\mathcal{G}_{\Delta t}$. The propagator $\mathcal{F}_{\Delta t}$ is a fine, expensive propagator, which accurately approximates the exact flow $\mathcal{E}_{\Delta t}$ over the time range Δt, whereas the propagator $\mathcal{G}_{\Delta t}$ is a coarse propagator, which is a less accurate approximation of the exact flow. In turn, $\mathcal{G}_{\Delta t}$ is less expensive to simulate than $\mathcal{F}_{\Delta t}$. For example, $\mathcal{F}_{\Delta t}$ and $\mathcal{G}_{\Delta t}$ may correspond to integrating (1.1) over the time range Δt with a given discretization scheme, using either a small time step (for $\mathcal{F}_{\Delta t}$) or a large time step (for $\mathcal{G}_{\Delta t}$). The parareal algorithm iteratively constructs a sequence of N-tuples $\mathbf{u}_k \equiv \{u_k^n\}_{1 \leq n \leq N}$ (with $N = T/\Delta t$), such that, at every iteration $k \geq 0$, u_k^n is an approximation of $u(n\Delta t)$. For $k = 0$, the initial approximation is obtained using the coarse propagator $\mathcal{G}_{\Delta t}$:

$$u_{k=0}^{n+1} = \mathcal{G}_{\Delta t}(u_{k=0}^n), \quad u_{k=0}^0 = u_0.$$

In the subsequent parareal iterations, the approximation is corrected using

$$u_{k+1}^{n+1} = \mathcal{G}_{\Delta t}(u_{k+1}^n) + \mathcal{F}_{\Delta t}(u_k^n) - \mathcal{G}_{\Delta t}(u_k^n), \tag{1.2}$$

with the initial condition $u_{k+1}^0 = u_0$. The solution to (1.2) can be very efficiently computed using the following procedure. Once the solution at parareal iteration k has been computed, we first compute the corrections $\mathcal{F}_{\Delta t}(u_k^n) - \mathcal{G}_{\Delta t}(u_k^n)$ in *parallel* over each subinterval $[n\Delta t, (n+1)\Delta t]$, $0 \leq n \leq N-1$. We then only need to propagate these corrections sequentially, by adding $\mathcal{G}_{\Delta t}(u_{k+1}^n)$ to the stored correction $\mathcal{F}_{\Delta t}(u_k^n) - \mathcal{G}_{\Delta t}(u_k^n)$. This yields the solution at parareal iteration $k+1$.

It has been shown (see e.g. [1, 4, 3, 26, 29]) that, when k goes to infinity, the parareal solution converges to the reference solution, namely the solution given by the fine-scale propagator $\mathcal{F}_{\Delta t}$ used in a sequential fashion from the initial condition:

$$\forall n, \ 0 \leq n \leq T/\Delta t, \quad \lim_{k \to \infty} u_k^n = \mathcal{F}_{\Delta t}^{(n)}(u_0). \tag{1.3}$$

The computational gain of the parareal algorithm stems from the fact that, in (1.2), the accurate simulations (using the fine-scale propagator $\mathcal{F}_{\Delta t}$) are decoupled one from each other, and can therefore be executed in parallel on different processors. Suppose that the cost of a single evaluation of $\mathcal{F}_{\Delta t}$ is much larger than the cost of propagating the system according to $\mathcal{G}_{\Delta t}$ over the complete time range $[0, T]$. Assuming the cost of the fine-scale propagator $\mathcal{F}_{\Delta t}$ to be proportional to Δt, the cost of K iterations of the parareal algorithm is proportional to $K\Delta t$. This cost is to be compared to the cost of computing the reference solution using the fine-scale propagator sequentially, which is proportional to $N\Delta t$. The computational speed-up is thus N/K, which is larger than one if the number K of parareal iterations to obtain convergence in (1.3) is small enough.

In this article, we propose and analyze a micro-macro version of the parareal algorithm. We assume that the variables in the microscopic model can be split into slow and fast components, and that we have at hand an approximate macroscopic model for the slow components under some time scale separation assumption (see Section 2 for the precise model we consider here). In this setting, we will use the

parareal algorithm where the fine-scale propagator $\mathcal{F}_{\Delta t}$ is an integrator for the high-dimensional microscopic model, whereas the coarse propagator, here denoted $\mathcal{C}_{\Delta t}$, is an integrator of the *low-dimensional*, approximate macroscopic model (we use the notation $\mathcal{C}_{\Delta t}$ rather than $\mathcal{G}_{\Delta t}$ to emphasize the fact that our coarse integrator acts on a system of smaller dimension than the reference one). The novelty therefore is to simultaneously use two models at different levels of description, rather than two discretizations of the same model. The cost of the coarse propagator is typically negligible for two reasons: (i) the macroscopic model only contains the slow components of the evolution, and therefore allows for a larger time step; and (ii) the macroscopic model is low-dimensional, and therefore requires less work per time step. Again, the aim of the micro-macro parareal method is to speed up the computations (compared to a full microscopic simulation) by allowing the microscopic simulations starting from the different intermediate time instances $n\Delta t$ to be performed in parallel over each subinterval $[n\Delta t, (n+1)\Delta t]$, with $0 \leq n \leq N-1$.

As a model problem, we take the setting of singularly perturbed systems of ODEs. Such a model problem is a widely accepted first test case when proposing algorithms for problems with time-scale separation, see e.g. [20]. We perform a numerical analysis of the algorithm we propose in a linear setting (see Section 2 for the description of the model problem, and Section 4 for the numerical analysis), and illustrate these results by numerical simulations in Section 5. However, our algorithm is *not* restricted to the linear setting, and we numerically observe in Section 6 that it performs equally well on a nonlinear test-case.

Since its introduction in [26], the parareal strategy has been applied to a wide range of problems, including fluid-structure interaction [11], Navier–Stokes equation simulation [12], reservoir simulation [15], etc. The algorithm has been further analyzed in [30, 31]. Its stability has been investigated in [3, 35]. An alternative formulation of the algorithm has been proposed in [4], or, equivalently, in [1] in a simplified setting. We refer to [14] for a reformulation in a more general setting that relates the parareal strategy to earlier time-parallel algorithms, such as multiple shooting (see e.g. [21, 33]) or multigrid waveform relaxation (see e.g. [27, 36]) approaches. Several variants of the algorithm have been proposed, for instance in [7, 11, 16] (see also [2] in the context of stochastic differential equations). The numerical analysis of the algorithm has been first performed for linear initial-value problems. A numerical analysis in a nonlinear context has been proposed in [13].

A micro-macro version of the parareal algorithm, similar to what is presented in this article, has already been considered in a number of works. The authors of [5, 28] consider a singularly perturbed system of ordinary differential equations (ODEs) at the microscopic level and the limiting differential-algebraic equation (DAE) at the macroscopic level. In these two works, the coarse propagator contains *all* degrees of freedom in the system. The slow degrees of freedom are evolved according to a differential equation, and the fast degrees of freedom are evolved using algebraic constraints (they somehow instantaneously adapt to the values of the slow degrees of freedom). In contrast, our approach completely eliminates the fast variables from the coarse propagator, and only evolves the slow variables. This difference has a number of consequences:

- The coarse propagator in the algorithms proposed here is cheaper than that of [5, 28] (because it contains less degrees of freedom) and more convenient (because the coarse propagator simulates an ODE rather than a DAE);
- The algorithms proposed here require operators to reconstruct microscopic

states from macroscopic ones, while the algorithm in [5, 28] can simply use the parareal iteration (1.2). This also influences the convergence behavior.

A detailed comparison between our algorithms and that proposed in [5, 28] is given in Section 3.3.

Other micro-macro parareal algorithms have also been proposed, in contexts different from ours. In [10], a parareal algorithm for multiscale stochastic chemical kinetics is presented, in which the macroscopic level uses the mean-field limiting ODE. In [32], the parareal algorithm is used with a kinetic Monte Carlo model at the macroscopic level and molecular dynamics at the microscopic level.

Our article is organized as follows. In Section 2, we present the singularly perturbed ODE that is considered here as a model problem, and state some bounds on its solution (The proof of these bounds is postponed until Appendix A). Subsequently, in Section 3, we introduce two micro-macro parareal algorithms. The coupling between the microscopic and macroscopic levels of description is done using a *restriction* operator (to go from the microscopic to the macroscopic level), and either a *lifting* (Algorithm 1) or a *matching* (Algorithm 2) operator (to go from the macroscopic to the microscopic level). This coupling ensures that the numerical solution remains consistent across both levels of description (see Section 3). The two algorithms we introduce in Section 3.2 only differ in how the levels of description are coupled to each other. Algorithm 1 will turn out to be inaccurate, whereas Algorithm 2 is extremely accurate. For the sake of comparison, we discuss in Section 3.3 the scheme proposed in [5, 28], that we denote here Algorithm 3. Section 4 contains a detailed numerical analysis of these three algorithms, when applied to the linear model problem presented in Section 2, and when the dynamics at both microscopic and macroscopic levels of description are exactly integrated. This setting enlightens the effect of how the two levels of description are coupled on the convergence of the algorithms. We show how the modeling error of the approximate macroscopic model affects the accuracy. In particular, the micro-macro parareal algorithm we introduce is a precise approximation of the full microscopic solution only if special care is taken during the coupling of the microscopic and macroscopic levels of description, as is done in Algorithm 2. The analysis is illustrated by numerical experiments in Section 5, where, in addition, we numerically investigate the effect of time discretization. Some numerical results on nonlinear problems are presented in Section 6. We observe there the same good properties of Algorithm 2 as on linear problems. We conclude in Section 7 with some final remarks and a discussion of possible future research.

2. Model problem. In this section, we describe the microscopic model problem considered in this work, as well as its macroscopic limit.

Consider the dynamics

$$\dot{x} = \alpha x + p^T y, \quad \dot{y} = \frac{1}{\epsilon}\left(qx - Ay\right), \tag{2.1}$$

where $x \in \mathbb{R}$ and $y \in \mathbb{R}^{d-1}$ are the state variables, and $\alpha \in \mathbb{R}$, $p \in \mathbb{R}^{d-1}$, $q \in \mathbb{R}^{d-1}$ and $A \in \mathbb{R}^{(d-1)\times(d-1)}$ are parameters. This dynamics models the evolution of a system described by the state variable $u = (x, y) \in \mathbb{R}^d$, where the slow and fast components are x and y, respectively. We denote the initial condition by $u(0) = (x(0), y(0)) = (x_0, y_0) = u_0$. The dynamics can be compactly written as

$$\dot{u} = B^\epsilon u, \tag{2.2}$$

where

$$B^\epsilon = \begin{bmatrix} \alpha & p^T \\ q/\epsilon & -A/\epsilon \end{bmatrix}.$$

In the following, we assume that the fast component of the system has a simple dissipative structure:

$$\begin{gathered} \text{We assume } A \text{ to be a matrix with eigenvalues } \lambda_i \in \mathbb{C} \ (1 \le i \le d-1) \\ \text{satisfying } \mathrm{Re}(\lambda_i) \ge \lambda_- \text{ for any } 1 \le i \le d-1, \text{ for some } \lambda_- > 0. \end{gathered} \tag{2.3}$$

Under this assumption, for each fixed value $x = x^\star$ of the slow component, the dynamics of y, obeying the equation

$$\dot{y} = \frac{1}{\epsilon}\left(qx^\star - Ay\right),$$

satisfies

$$\lim_{t\to\infty} y(t) = \left(A^{-1}q\right) x^\star.$$

The dynamics of the fast component y, for fixed slow component $x = x^\star$, is thus exponentially stable for all $x^\star$. It is then known (see Lemma 2 below and, for example, [34] and references therein) that, in the limit ϵ goes to zero, the solution $x(t)$ to (2.1) converges, on finite time intervals, to the solution $X(t)$ of

$$\dot{X} = \lambda X, \qquad X(0) = x_0, \qquad \lambda := \alpha + p^T A^{-1} q. \tag{2.4}$$

Comparing (2.1) with (2.4), one can see that the microscopic time-scale (namely the typical time-step required to integrate the full microscopic dynamics (2.1)) is of the order of ϵ, whereas the macroscopic time-scale (namely the typical time-step required to integrate the approximate macroscopic dynamics (2.4)) is independent of ϵ.

REMARK 1. *The asymptotic result that we mentioned above on the system* (2.1) *holds for more general cases. For instance, consider the dynamics*

$$\dot{x} = f(x,y), \quad \dot{y} = \frac{1}{\epsilon}\left(\eta(x) - Ay\right), \tag{2.5}$$

with again $x \in \mathbb{R}$, $y \in \mathbb{R}^{d-1}$ *and* $A \in \mathbb{R}^{(d-1)\times(d-1)}$, *and where* $f : \mathbb{R} \times \mathbb{R}^{d-1} \to \mathbb{R}$ *and* $\eta : \mathbb{R} \to \mathbb{R}^{d-1}$ *are two given, possibly nonlinear functions. Under Assumption* (2.3), *the solution* $x(t)$ *to* (2.5) *converges to* $X(t)$, *solution to*

$$\dot{X} = F(X), \qquad X(0) = x_0, \qquad F(X) = f(X, A^{-1}\eta(X)).$$

This result can also be extended to more general nonlinear cases [34].

For future reference, we introduce the exact time evolution operators,

$$u(t^* + \Delta t) = \Phi_{\Delta t}\left(u(t^*)\right), \quad X(t^* + \Delta t) = \rho_{\Delta t}\left(X(t^*)\right),$$

corresponding to (2.2) and (2.4), respectively. These equations are linear, hence the operators $\Phi_{\Delta t}$ and $\rho_{\Delta t}$ are linear:

$$\Phi_{\Delta t} = \exp(B^\epsilon \Delta t) \in \mathbb{R}^{d\times d}, \tag{2.6}$$

$$\rho_{\Delta t} = \exp(\lambda \Delta t) \in \mathbb{R}. \tag{2.7}$$

We now state some bounds on the solutions of (2.1), that will be useful in Section 4, when proving error bounds on the algorithms we propose.

LEMMA 2. *Consider the linear system* (2.1) *over the time range* $[0,T]$*, with the initial condition* $x(0)=x_0$*,* $y(0)=y_0$*. Introduce* $z(t)=y(t)-A^{-1}q\,x(t)\in\mathbb{R}^{d-1}$ *and* $z_0=z(0)$*. Under Assumption* (2.3)*, there exist* $\epsilon_0\in(0,1)$ *and* $C>0$*, that both only depend on* A*,* q*,* p*,* α *and* T*, such that, for all* $\epsilon<\epsilon_0$*,*

$$\sup_{t\in[0,T]}|x(t)-x_0\exp(\lambda t)|\leq C\epsilon(|x_0|+\|z_0\|), \tag{2.8}$$

$$\sup_{t\in[0,T]}\|z(t)-\exp\left(-At/\epsilon\right)z_0\|\leq C\epsilon\left(|x_0|+\|z_0\|\right). \tag{2.9}$$

Set

$$t_\epsilon^{\mathrm{BL}}=\frac{2\epsilon}{\lambda_-}\ln(1/\epsilon). \tag{2.10}$$

Then, for all $\epsilon<\epsilon_0$*, we have*

$$\sup_{t\in[t_\epsilon^{\mathrm{BL}},T]}\|z(t)\|\leq C\epsilon\left(|x_0|+\|z_0\|\right). \tag{2.11}$$

Hence, up to a boundary layer of size t_ϵ^{BL}*,* $z(t)$ *is of order* ϵ*, and the state* $u(t)$ *of the system is at a distance of the order of* ϵ *of the manifold*

$$\Sigma:=\left\{u=(x,y)\in\mathbb{R}^d;\ y=A^{-1}q\,x\right\}. \tag{2.12}$$

We call the manifold Σ the *slow manifold*. Note that the bound (2.11) is sharp in the sense that, after the initial time boundary layer, $z(t)$ is of order ϵ and not smaller. This can be checked for example on the analytically solvable system $\dot{x}=-x$, $\dot{y}=(x-y)/\epsilon$.

An important consequence of the above lemma is that the microscopic solution $u(t)=(x(t),y(t))$ remains bounded, independently of ϵ, on the time range $[0,T]$. The following result, which will be used repeatedly in the sequel, follows immediately from Lemma 2:

COROLLARY 3. *Consider the linear system* (2.1) *over the time range* $[0,T]$*, with initial condition* $x(0)=x_0$*,* $y(0)=y_0$*. Under Assumption* (2.3)*, there exist* $\epsilon_0\in(0,1)$ *and* $C>0$*, that both only depend on* A*,* q*,* p*,* α *and* T*, such that, for all* $\epsilon<\epsilon_0$*, we have*

$$\sup_{t\in[0,T]}|x(t)|\leq C\left(|x_0|+\epsilon\|y_0\|\right), \tag{2.13}$$

$$\sup_{t\in[t_\epsilon^{\mathrm{BL}},T]}\|y(t)\|\leq C\left(|x_0|+\epsilon\|y_0\|\right), \tag{2.14}$$

where the size t_ϵ^{BL} *of the boundary layer is defined by* (2.10)*.*

The proofs of these standard results are postponed until Appendix A. In view of (2.8), we see that, in the limit when ϵ goes to zero, the macroscopic dynamics (2.4) is exact. The aim of the algorithms we investigate below is to use these macroscopic dynamics to speed up the computation of the solution of the original model (2.1), for a fixed small but non-zero value of ϵ.

3. Micro-macro parareal algorithms. In this section, we describe two micro-macro parareal algorithms. As will become clear from the analysis in the forthcoming sections, the first one based on a lifting operator is inaccurate, whereas the second one based on a matching operator is extremely accurate. Both are generalizations of the parareal algorithm proposed in [26]. Our formulation follows most closely the description in [1]. We first introduce the necessary notation in Section 3.1, and we subsequently outline both algorithms in Section 3.2. For the sake of comparison, we also discuss in Section 3.3 the scheme proposed in [5, 28]. Let us emphasize that the two algorithms we introduce are not restricted to the linear system (2.1), and apply to any system of the form

$$\dot{x} = f(x, y), \quad \dot{y} = \frac{1}{\epsilon} g(x, y),$$

where $x \in \mathbb{R}^s$ is a slow component ($s \in \mathbb{N}^\star$), $y \in \mathbb{R}^m$ is a fast component ($m \in \mathbb{N}^\star$), and where the associated macroscopic dynamics (obtained in the limit of infinite time scale separation between the slow and the fast components, namely in the limit when ϵ goes to zero) reads $\dot{X} = F(X)$.

3.1. Notation. We introduce a time discretization $(t_n)_{n=0}^N$, with $t_n = n\Delta t$ and $N\Delta t = T$. Let $u^n = (x^n, y^n) \approx u(t_n) = (x(t_n), y(t_n))$ be the numerical approximation of the solution of the microscopic model (2.1), and let $X^n \approx X(t_n)$ be that of the solution to the macroscopic model (2.4).

Fine-scale and coarse propagators. The micro-macro parareal algorithm makes use of two propagators. First, we need a *fine-scale propagator*, that advances the microscopic model (2.1) over a time-range Δt:

$$u^{n+1} = \mathcal{F}_{\Delta t}(u^n). \tag{3.1}$$

To perform this, we may consider that we have at hand the *exact* propagator of the equation (2.1), in which case $\mathcal{F}_{\Delta t} \equiv \Phi_{\Delta t}$, where $\Phi_{\Delta t}$ is defined by (2.6). Alternatively, we may resort to a numerical integration of the dynamics (2.1) (using for example forward or backward Euler discretizations) over the time range Δt, using several steps of size δt. Typically, in the context of a system like (2.1), one would need δt to be of the order of ϵ to obtain accurate results.

Second, we need a *coarse propagator* for the macroscopic model (2.4),

$$X^{n+1} = \mathcal{C}_{\Delta t}(X^n), \tag{3.2}$$

where again we may assume that we can exactly integrate (2.4) and hence choose $\mathcal{C}_{\Delta t} \equiv \rho_{\Delta t}$, see (2.7). Alternatively, one may resort to a numerical integration of the dynamics (2.4), for which we can use a time-step independent of ϵ to obtain accurate results.

Restriction, lifting and matching operators. The parareal algorithm iteratively uses the fine-scale and the coarse propagators. In this work, these two propagators correspond to *different descriptions* of the system, either microscopic (using $u \in \mathbb{R}^d$) or macroscopic (using $X \in \mathbb{R}$). We thus need a way to go from one description to the other, as we discuss now.

We first introduce the *restriction* operator

$$\mathcal{R} : \left\{ \begin{array}{rcl} \mathbb{R}^d & \to & \mathbb{R} \\ u = (x, y) & \mapsto & x, \end{array} \right.$$

which maps a microscopic state to the corresponding macroscopic state. For notational convenience, we also introduce the complement of the restriction operator,

$$\mathcal{R}^\perp : \left\{ \begin{array}{rcl} \mathbb{R}^d & \to & \mathbb{R}^{d-1} \\ u = (x, y) & \mapsto & y, \end{array} \right.$$

such that we can write $u = (x, y) = (\mathcal{R}u, \mathcal{R}^\perp u)$.

Conversely, we will also need to reconstruct a microscopic state from a given macroscopic state. In contrast to the restriction operator, there is no unique obvious way to define this operator. We introduce two such operators, a *lifting* operator and a *matching* operator.

DEFINITION 4. *A* lifting operator $\mathcal{L}$ *is an operator*

$$\mathcal{L} : \left\{ \begin{array}{rcl} \mathbb{R} & \to & \mathbb{R}^d \\ X & \mapsto & u = \mathcal{L}(X) \end{array} \right.$$

that creates a microscopic state that is uniquely determined by a given macroscopic state and satisfies the consistency property

$$\mathcal{R} \circ \mathcal{L} = \mathrm{Id}\,. \tag{3.3}$$

A possible choice is to take $\mathcal{L}(X)$ such that

$$\mathcal{R}(\mathcal{L}(X)) = X \quad \text{and} \quad \mathcal{L}(X) \in \Sigma, \tag{3.4}$$

where Σ is the slow manifold associated to the multiscale problem.

In connection with the system (2.1), an example (and this is the choice we make in this work) is to choose

$$\mathcal{L}(X) = (X, (A^{-1}q)X). \tag{3.5}$$

This choice indeed satisfies (3.3) and (3.4), where the slow manifold Σ of the system (2.1) is defined by (2.12).

REMARK 5. *Other lifting operators can be introduced, using for example the constrained runs algorithm [18]. As soon as the lifting operator $\mathcal{L}$ is specified, u is uniquely determined by X: the lifting operator enforces a closure approximation on the microscopic state.*

Alternatively, one may reconstruct a microscopic state using a *matching operator.*

DEFINITION 6. *A* matching operator *is an operator*

$$\mathcal{P} : \left\{ \begin{array}{rcl} \mathbb{R} \times \mathbb{R}^d & \to & \mathbb{R}^d \\ (X, v) & \mapsto & \mathcal{P}_X(v), \end{array} \right.$$

that satisfies

$$X = (\mathcal{R} \circ \mathcal{P})\,(X, v) \text{ for any } v \in \mathbb{R}^d \text{ and } X \in \mathbb{R}, \tag{3.6}$$

and

$$\forall u \in \mathbb{R}^d \text{ such that } \mathcal{R}(u) = X, \quad \mathcal{P}_X(u) = u,$$

or, equivalently,

$$\forall u \in \mathbb{R}^d, \quad \mathcal{P}(\mathcal{R}(u), u) = u. \tag{3.7}$$

In contrast with a lifting operator, a matching operator requires a microscopic state as an input, and not only a macroscopic state.

The consistency property (3.6) may be seen as the equivalent for $\mathcal{P}$ of the property (3.3) for $\mathcal{L}$. We also note that, in view of (3.7), a microscopic state u which is already consistent with the macroscopic value X is unaltered by the operator $\mathcal{P}_X$. Combining (3.6) and (3.7), we observe that $\mathcal{P}_X \circ \mathcal{P}_X = \mathcal{P}_X$: the operator $\mathcal{P}_X : \mathbb{R}^d \to \mathbb{R}^d$ is thus a projection operator onto microscopic states $u \in \mathbb{R}^d$ that satisfy $\mathcal{R}(u) = X$. One may thus think of $\mathcal{P}_X$ as a projection operator that projects a microscopic state v to a microscopic state $u = \mathcal{P}_X(v)$, such that $\mathcal{R}(u) = X$ and u is as close to v as possible, in a sense to be made precise for the problem at hand.

In the following, we require in addition the following continuity property on $\mathcal{P}$: there exists $C > 0$ such that, for all $X \in \mathbb{R}$, $Y \in \mathbb{R}$, $u \in \mathbb{R}^d$ and $v \in \mathbb{R}^d$,

$$\|\mathcal{P}(X, u) - \mathcal{P}(Y, v)\| \leq C\Big[\, \|u - v\| + |X - Y| \,\Big]. \tag{3.8}$$

For the *analysis* of the algorithms described below, we only require $\mathcal{P}$ to satisfy (3.6), (3.7) and (3.8), and do not make any additional assumptions (see Section 4). For the *numerical experiments* reported on in Section 5, we choose, in the context of the system (2.1),

$$\mathcal{P}_X(v) := (X, \mathcal{R}^\perp v), \tag{3.9}$$

which consists in keeping the fast variables from v, while imposing the slow variable to be equal to X. This choice fulfills all the above conditions (3.6), (3.7) and (3.8).

REMARK 7. *The term* matching operator *has been chosen in reminiscence of the term "moment matching" that is commonly used in the Monte Carlo community, see e.g. [6].*

3.2. Algorithms 1 and 2. The parareal algorithm iteratively constructs approximations on the whole time domain $[0, T]$. We denote by u_k^n, resp. X_k^n, the approximate microscopic, resp. macroscopic, solution at time t_n, obtained at the k-th parareal iteration.

The first algorithm we consider is the following.

ALGORITHM 1. *Let $u(0) = u_0$ be the initial condition.*

1. *Initialization:*
   a) Compute $\{X_0^n\}_{0 \leq n \leq N}$ sequentially by using the coarse propagator:

$$X_0^0 = \mathcal{R}(u_0), \qquad X_0^{n+1} = \mathcal{C}_{\Delta t}(X_0^n).$$

   b) Lift the macroscopic approximation to the microscopic level:

$$u_0^0 = u_0 \quad \text{and, for all } 1 \leq n \leq N, \quad u_0^n = \mathcal{L}(X_0^n).$$

2. *Assume that, for some $k \geq 0$, the sequences $\{u_k^n\}_{0 \leq n \leq N}$ and $\{X_k^n\}_{0 \leq n \leq N}$ are known. Compute these sequences at the iteration $k+1$ by the following steps:*

a) For all $0 \leq n \leq N-1$, compute (in parallel) using the coarse and the fine-scale propagators

$$\overline{X}_k^{n+1} = \mathcal{C}_{\Delta t}(X_k^n), \quad \overline{u}_k^{n+1} = \mathcal{F}_{\Delta t}(u_k^n). \tag{3.10}$$

b) For all $0 \leq n \leq N-1$, evaluate the jumps (the difference between the two propagated values) at the macroscopic level*:*

$$J_k^{n+1} = \mathcal{R}(\overline{u}_k^{n+1}) - \overline{X}_k^{n+1}. \tag{3.11}$$

c) Compute $\left\{X_{k+1}^n\right\}_{0\leq n\leq N}$ sequentially by

$$X_{k+1}^0 = \mathcal{R}(u_0), \qquad X_{k+1}^{n+1} = \mathcal{C}_{\Delta t}(X_{k+1}^n) + J_k^{n+1}. \tag{3.12}$$

d) Compute $\left\{u_{k+1}^{n+1}\right\}_{0\leq n\leq N-1}$ by lifting the macroscopic solution:

$$u_{k+1}^0 = u_0 \quad \textit{and, for all } 0 \leq n \leq N-1, \quad u_{k+1}^{n+1} = \mathcal{L}(X_{k+1}^{n+1}). \tag{3.13}$$

We can recast the above algorithm as

$$u_{k+1}^{n+1} = \mathcal{L}\Big(\mathcal{C}_{\Delta t}\left(\mathcal{R}\left(u_{k+1}^n\right)\right) + \mathcal{R}\left(\mathcal{F}_{\Delta t}\left(u_k^n\right)\right) - \mathcal{C}_{\Delta t}\left(\mathcal{R}\left(u_k^n\right)\right)\Big). \tag{3.14}$$

Notice that this cannot be recast in the form of the original parareal algorithm (1.2). The above algorithm uses the following paradigm: each time we need to reconstruct a full microscopic solution u from a given macroscopic state X, we use the lifting operator $\mathcal{L}$. For example, for the system (2.1) and $\mathcal{L}$ given by (3.5), this amounts to creating a microscopic state exactly on the slow manifold (2.12).

We will see in the sequel that this algorithm leads to disappointing results. In particular, Algorithm 1 does not retain one of the properties of the parareal algorithm as originally proposed in [26], namely that the numerical trajectory is exact on the first k subintervals in time after k iterations of the parareal algorithm.

A much better algorithm is the following:

ALGORITHM 2. *Let $u(0) = u_0$ be the initial condition.*

1. *Initialization: proceed as in Step 1 of Algorithm 1.*
2. *Assume that, for some $k \geq 0$, the sequences $\{u_k^n\}_{0\leq n\leq N}$ and $\{X_k^n\}_{0\leq n\leq N}$ are known. To compute these sequences at the iteration $k+1$,*
   - *Proceed as in Steps 2a, 2b and 2c of Algorithm 1.*
   - *Compute $\left\{u_{k+1}^{n+1}\right\}_{0\leq n\leq N-1}$ by matching the result of the local microscopic computation, $\overline{u}_k^{n+1}$, on the corrected macroscopic state X_{k+1}^{n+1}:*

$$u_{k+1}^0 = u_0 \quad \textit{and, for all } 0 \leq n \leq N-1, \quad u_{k+1}^{n+1} = \mathcal{P}(X_{k+1}^{n+1}, \overline{u}_k^{n+1}). \tag{3.15}$$

The only difference between Algorithms 1 and 2 is how we reconstruct the microscopic solution u_{k+1}^{n+1}. In Algorithm 1, we simply choose u_{k+1}^{n+1} on the slow manifold defined by X_{k+1}^{n+1} (see (3.13)). In Algorithm 2, we use the quantity $\overline{u}_k^{n+1}$, which is the end point of a microscopic trajectory between times $n\Delta t$ and $(n+1)\Delta t$, and match this state onto the corrected macroscopic state X_{k+1}^{n+1}, obtained at the latest parareal iteration.

At the initial iteration $k = 0$, since no microscopic computation has been done, we cannot use the matching operator $\mathcal{P}$ to reconstruct a fine-scale solution. We thus resort to the lifting operator $\mathcal{L}$.

Algorithm 2 can be recast as

$$u_{k+1}^{n+1} = \mathcal{P}\Big(\mathcal{C}_{\Delta t}\left(\mathcal{R}\left(u_{k+1}^{n}\right)\right) + \mathcal{R}\left(\mathcal{F}_{\Delta t}\left(u_k^n\right)\right) - \mathcal{C}_{\Delta t}\left(\mathcal{R}\left(u_k^n\right)\right), \mathcal{F}_{\Delta t}(u_k^n)\Big), \tag{3.16}$$

which is to be compared with the original parareal algorithm (1.2) and (3.14) for Algorithm 1. For the linear system (2.1) and the choice (3.9) of matching operator, the equation (3.16) can be further simplified to

$$u_{k+1}^{n+1} = \mathcal{F}_{\Delta t}(u_k^n) + (1,0)^T\Big(\mathcal{C}_{\Delta t}\left(\mathcal{R}\left(u_{k+1}^{n}\right)\right) - \mathcal{C}_{\Delta t}\left(\mathcal{R}\left(u_k^n\right)\right)\Big). \tag{3.17}$$

This is exactly (1.2) with $\mathcal{F}_{\Delta t}$ as the fine propagator and $(1,0)^T\mathcal{C}_{\Delta t}\mathcal{R}$ as the coarse propagator.

Note that, in view of (3.3) and (3.13) (respectively (3.6) and (3.15)), the trajectories computed using Algorithm 1 (respectively Algorithm 2) satisfy

$$\forall k \geq 0, \quad \forall n \geq 0, \quad X_k^n = \mathcal{R}(u_k^n). \tag{3.18}$$

At any parareal iteration k, the macroscopic trajectory is *consistent* with the microscopic trajectory.

3.3. Comparison of Algorithms 1 and 2 with that of [5, 28]. As underlined in the introduction, a micro-macro version of the parareal algorithm has already been proposed in [28, 5]. In these works, the coarse propagator is an integrator of a reduced (DAE) model that contains *all* degrees of freedom in the system (both the fast and slow ones), in contrast to our algorithms, where the coarse propagator is an integrator for the effective dynamics of the slow degrees of freedom.

For the model problem (2.1), the reduced DAE considered in [28, 5] takes the form

$$\dot{x} = \alpha x + p^T y, \qquad Ay = qx. \tag{3.19}$$

The coarse propagator of [28, 5] is an integrator $\mathcal{G}_{\Delta t}$ of (3.19). This coarse integrator is combined with a fine-scale integrator $\mathcal{F}_{\Delta t}$ of (2.1) in the parareal fashion, following (1.2).

The obtained scheme, that we denote here Algorithm 3, differs from our Algorithm 2 in its treatment of the fast degrees of freedom. To show this, we note that, specifically for the model problem (2.1), an exact propagator for (3.19) can be obtained by first solving (2.4) exactly, and second solving the algebraic equation for y. Hence, we have

$$\mathcal{G}_{\Delta t}(u) = \mathcal{L} \circ \rho_{\Delta t} \circ \mathcal{R}\, u = \mathcal{L} \circ \mathcal{C}_{\Delta t} \circ \mathcal{R}\, u \tag{3.20}$$

where, we recall, $\mathcal{C}_{\Delta t}$ is the coarse propagator used in Algorithms 1 and 2. Using (1.2), we write Algorithm 3 as follows:

$$\begin{aligned} u_{k+1}^{n+1} &= \mathcal{F}_{\Delta t}(u_k^n) + \mathcal{G}_{\Delta t}\left(u_{k+1}^n\right) - \mathcal{G}_{\Delta t}\left(u_k^n\right) \\ &= \mathcal{F}_{\Delta t}(u_k^n) + \mathcal{L}\Big(\mathcal{C}_{\Delta t}\left(\mathcal{R}\left(u_{k+1}^{n}\right)\right) - \mathcal{C}_{\Delta t}\left(\mathcal{R}\left(u_k^n\right)\right)\Big), \end{aligned} \tag{3.21}$$

which can be compared with (3.17) and with (3.14). Notice in particular that Algorithm 2 differs from Algorithm 3 in the choice of the coarse propagator.

The three algorithms only differ in how the microscopic and macroscopic levels of description are coupled in the parareal iterations. These differences, however, have implications on (i) the computational complexity of the methods; (ii) the way they generalize to more complex multiscale systems; and (iii) the convergence behavior. The convergence properties of the three algorithms are analyzed in Section 4. We here briefly comment on the other two aspects. First, the computational complexity of the coarse propagator in Algorithm 3 is significantly higher than that of Algorithms 1 and 2, due to the presence of the fast degrees of freedom, which requires solving a large linear system in addition to the time-stepping of the slow degrees of freedom.

Second, in more complex situations, for instance when the microscopic and macroscopic systems are nonlinear, Algorithm 3 may require the use of a time integrator for DAEs. Although many such integrators exist, they are usually implicit, and less convenient than ODE solvers. In those cases, Algorithms 1 and 2 only require a reasonable model to propagate the macroscopic variables. In both cases, one may resort to computational multiscale methods that approximate the evolution of the approximate macroscopic model by using short microscopic simulations. The coarse propagator required in Algorithms 1 and 2 can be replaced by a coarse projective integration approach [22, 23]. The coarse propagator for the DAE system required in Algorithm 3 can be replaced by a projective integration method [17, 19]. Remark that the computational cost of both methods is not identical: projective integration requires a computational cost of $O(\log(1/\epsilon))$, whereas the computational cost of coarse projective integration is independent of ϵ. This shows again that Algorithms 1 and 2 are cheaper to implement than Algorithm 3. We will see in the next Section to what extent the higher computational cost of Algorithm 3 allows for a better accuracy.

4. Analysis. In this section, we analyze the convergence of the two micro-macro parareal algorithms introduced above on the linear model problem (2.1). We also give a detailed analysis for Algorithm 3, introduced in [5, 28]. To keep the analysis simple, we focus on the error due to the fact that the models are different at the macroscopic and microscopic levels. We thus track the dependency of the error bounds on the parameter ϵ, and consider, at both levels, the exact propagators (2.6) and (2.7). Thus, the fine-scale and coarse propagators in (3.1) and (3.2) are given by

$$\mathcal{F}_{\Delta t}(u) = \Phi_{\Delta t} u, \qquad \mathcal{C}_{\Delta t}(X) = \rho_{\Delta t} X,$$

for a fixed Δt, which is chosen typically much larger than ϵ (so that Δt is a macroscopic time-scale). We recall that the lifting operator $\mathcal{L}$ is defined by (3.5), and that we work with a matching operator $\mathcal{P}$ satisfying (3.6), (3.7) and (3.8).

We first derive an error recursion formula in Section 4.1. Using this formula, we derive a sharp error bound on the trajectories computed by Algorithm 1, where the microscopic state is reconstructed using the lifting operator $\mathcal{L}$ (see Section 4.2). We next turn to Algorithm 2, where the microscopic state is reconstructed using a matching operator $\mathcal{P}$. We first show that, at a given parareal iteration k, the computed trajectories (both at the macro and the micro scales) are exact up to the time $k\Delta t$ (see Section 4.3.1), reproducing thereby a property of the standard parareal algorithm. We subsequently derive a sharp error bound in terms of ϵ, showing that, at iteration k, Algorithm 2 converges to the exact solution of the full microscopic system with an error of the order of $\epsilon^{k/2}$ (see Section 4.3.2 for precise statements).

These two properties (exactness of the trajectories up to time $k\Delta t$ after k iterations, and improvement of the convergence rate to the exact solution as k increases) are not satisfied for Algorithm 1. We eventually consider Algorithm 3. Being based on (1.2), this algorithm automatically satisfies the local exactness property. We then prove a sharp error bound in terms of ϵ, showing, in agreement with [28], that, at iteration k, Algorithm 3 converges to the exact solution of the full microscopic system with an error of the order of ϵ^k (see Section 4.4 for precise statements).

The analysis below closely follows that of [26], but is significantly extended. We explicitly relate to the case considered in [26] when appropriate.

Before proceeding, we introduce two notions of error:

DEFINITION 8 (Microscopic error). *Let $u(t_n)$ be the exact microscopic solution of (2.2) at time $t_n = n\Delta t$, and let u_k^n be the parareal microscopic solution after k parareal iterations, using Algorithm 1 or 2. The microscopic error*

$$e_k^n = u_k^n - u(t_n) \tag{4.1}$$

is defined as the difference of the solutions at the microscopic level.

DEFINITION 9 (Macroscopic error). *Let $u(t_n)$ be the exact microscopic solution of (2.2) at time $t_n = n\Delta t$, and let X_k^n be the parareal macroscopic solution after k parareal iterations, using Algorithm 1 or 2. The macroscopic error*

$$E_k^n = X_k^n - \mathcal{R}u(t_n) \tag{4.2}$$

is defined as the difference of the solutions at the macroscopic level.

Note that, in view of (3.18) and using the linearity of $\mathcal{R}$, we have

$$E_k^n = \mathcal{R}e_k^n. \tag{4.3}$$

4.1. Error recursion formula. A first step in the analysis of the algorithms described above is the derivation of a recursion formula for the error, which is valid for both algorithms and for any choice of the operators $\mathcal{R}$, $\mathcal{L}$ and $\mathcal{P}$. Starting from (3.12) and (3.11), we write X_{k+1}^n as a function of the microscopic and macroscopic solutions at the parareal iteration k: for $n \geq 2$,

$$\begin{aligned}
X_{k+1}^n &= \mathcal{C}_{\Delta t}(X_{k+1}^{n-1}) + J_k^n \\
&= \rho_{\Delta t} X_{k+1}^{n-1} + \left(\mathcal{R}\Phi_{\Delta t} u_k^{n-1} - \rho_{\Delta t} X_k^{n-1}\right) \\
&= \mathcal{R}\Phi_{\Delta t} u_k^{n-1} + \rho_{\Delta t}\left(X_{k+1}^{n-1} - X_k^{n-1}\right) \\
&= \mathcal{R}\Phi_{\Delta t} u_k^{n-1} + \rho_{\Delta t}\left(\mathcal{R}\Phi_{\Delta t} u_k^{n-2} + \rho_{\Delta t}\left(X_{k+1}^{n-2} - X_k^{n-2}\right) - X_k^{n-1}\right) \\
&= \mathcal{R}\Phi_{\Delta t} u_k^{n-1} + \rho_{\Delta t}\left(\mathcal{R}\Phi_{\Delta t} u_k^{n-2} - X_k^{n-1}\right) + \rho_{\Delta t}^2\left(X_{k+1}^{n-2} - X_k^{n-2}\right) \\
&= \mathcal{R}\Phi_{\Delta t} u_k^{n-1} + \sum_{p=1}^{n-1} \rho_{\Delta t}^p \left(\mathcal{R}\Phi_{\Delta t} u_k^{n-p-1} - X_k^{n-p}\right) \\
&= \mathcal{R}\Phi_{\Delta t} u_k^{n-1} + \sum_{p=1}^{n-1} \rho_{\Delta t}^{n-p} \left(\mathcal{R}\Phi_{\Delta t} u_k^{p-1} - X_k^{p}\right).
\end{aligned} \tag{4.4}$$

This formula is also valid for $n = 1$ using the convention $\sum_{p=1}^{0} \cdot = 0$. Note that we have used the linearity of the coarse propagator. We then obtain a recursion for the

macroscopic error, using the linearity of the fine-scale propagator:

$$\begin{aligned}
E_{k+1}^n &= X_{k+1}^n - \mathcal{R}\Phi_{\Delta t}^n u_0 \\
&= \mathcal{R}\Phi_{\Delta t} u_k^{n-1} - \mathcal{R}\Phi_{\Delta t}^n u_0 + \sum_{p=1}^{n-1} \rho_{\Delta t}^{n-p} \left(\mathcal{R}\Phi_{\Delta t} u_k^{p-1} - X_k^p \right) \\
&= \mathcal{R}\Phi_{\Delta t} e_k^{n-1} + \sum_{p=1}^{n-1} \rho_{\Delta t}^{n-p} \left(\mathcal{R}\Phi_{\Delta t} u_k^{p-1} - \mathcal{R}\Phi_{\Delta t}^p u_0 + \mathcal{R}\Phi_{\Delta t}^p u_0 - X_k^p \right) \\
&= \mathcal{R}\Phi_{\Delta t} e_k^{n-1} + \sum_{p=1}^{n-1} \rho_{\Delta t}^{n-p} \left(\mathcal{R}\Phi_{\Delta t} e_k^{p-1} - E_k^p \right) \\
&= \mathcal{R}\Phi_{\Delta t} e_k^{n-1} + \sum_{p=0}^{n-2} \rho_{\Delta t}^{n-p-1} \mathcal{R}\Phi_{\Delta t} e_k^p - \sum_{p=1}^{n-1} \rho_{\Delta t}^{n-p} E_k^p \\
&= \sum_{p=1}^{n-1} \rho_{\Delta t}^{n-p-1} \left(\mathcal{R}\Phi_{\Delta t} e_k^p - \rho_{\Delta t} E_k^p \right), \qquad (4.5)
\end{aligned}$$

where, in the last line, we have used the fact that the microscopic error at $t = 0$ vanishes: $e_k^0 = 0$ for any k.

We remark that the formula (4.5) is not closed, in the sense that it couples the macroscopic error at parareal iteration $k+1$ *to both the macroscopic and the microscopic errors* at parareal iteration k. To close the formula, and to transform it into specific bounds on the errors, we will need to make use of specific properties of $\Phi_{\Delta t}$ and of the lifting, matching and restriction operators $\mathcal{L}$, $\mathcal{P}$ and $\mathcal{R}$. This is where the analysis of Algorithms 1 and 2 differ.

REMARK 10. *Using* (4.5)*, it is possible to recover standard error bounds on the parareal algorithm, when the microscopic and the macroscopic models are linear and written at the same level of description, using a common state variable* $u \in \mathbb{R}$*, as in [26] for example. In this case, we have* $\mathcal{R} = \mathcal{L} = \mathcal{P} = \mathrm{Id}$ *(there is only one model, and one level of description), and* $e_k^n = E_k^n$*. The coarse and fine-scale propagators are linear operators, denoted respectively by* $\mathcal{C}_{\Delta t}(u) = \mathcal{G}_{\Delta t}(u) = \rho_{\Delta t}^G u$ *and* $\mathcal{F}_{\Delta t}(u) = \rho_{\Delta t}^F u$*. Since* u *is scalar, the propagators are simply multiplications by two scalars* $\rho_{\Delta t}^G$ *and* $\rho_{\Delta t}^F$*. The equation* (4.5) *then reads*

$$E_{k+1}^n = \sum_{p=1}^{n-1} (\rho_{\Delta t}^G)^{n-p-1} \left(\rho_{\Delta t}^F - \rho_{\Delta t}^G \right) E_k^p. \qquad (4.6)$$

Assume, as in the classical analysis of the parareal algorithm presented in [26], that the fine-scale propagator is exact, whereas the coarse propagator is a scheme of order s*:* $\left| \rho_{\Delta t}^F - \rho_{\Delta t}^G \right| = O(\Delta t^{s+1})$*. We consider a range of* Δt *such that* $\rho_{\Delta t}^G > 0$ *(which is possible since* $\rho_{\Delta t}^G = 1 + O(\Delta t)$*). Fix a time range* $[0, T]$*. We show that, using* (4.6)*, one can recover the classical result of [26]: at any parareal iteration* k*, there exists* c_k *such that, for any* Δt*,*

$$\sup_{0 \le n\Delta t \le T} |E_k^n| \le c_k \Delta t^{s(k+1)}. \qquad (4.7)$$

This bound is satisfied at $k = 0$ *since the coarse propagator is of order* s*. Assume now that* (4.7) *holds at some parareal iteration* k*. We then deduce from* (4.6) *that, for all*

$n \geq 0$ *such that* $n\Delta t \leq T$,

$$\left|E_{k+1}^n\right| \leq c_k \Delta t^{s(k+1)} \left|\rho_{\Delta t}^F - \rho_{\Delta t}^G\right| \sum_{p=1}^{n-1} (\rho_{\Delta t}^G)^{n-p-1} \leq C c_k \Delta t^{s(k+1)} \Delta t^{s+1} \sum_{p=0}^{N-1} (\rho_{\Delta t}^G)^p. \tag{4.8}$$

Remark now that

$$\sum_{p=0}^{N-1} (\rho_{\Delta t}^G)^p = \frac{(\rho_{\Delta t}^G)^N - 1}{\rho_{\Delta t}^G - 1} = \frac{(\rho_{\Delta t}^G)^N - (\rho_{\Delta t}^F)^N}{\rho_{\Delta t}^G - 1} + \frac{(\rho_{\Delta t}^F)^N - 1}{\rho_{\Delta t}^G - 1}.$$

Since the fine-scale propagator is exact, we have $(\rho_{\Delta t}^F)^N = \rho_{N\Delta t}^F = \rho_T^F$, *which is independent of* Δt. *Thus*

$$\left|\sum_{p=0}^{N-1} (\rho_{\Delta t}^G)^p\right| \leq \frac{C\Delta t^s + C}{\left|\rho_{\Delta t}^G - 1\right|} \leq \frac{C}{\Delta t}. \tag{4.9}$$

Collecting (4.8) *and* (4.9), *we deduce* (4.7) *at the parareal iteration* $k+1$. *This concludes the proof.*

4.2. Error bounds for Algorithm 1. We consider Algorithm 1, where the reconstruction at each parareal iteration is done using the lifting operator $\mathcal{L}$ defined by (3.5). We show in this section that the accuracy of the numerical trajectory does not improve (neither at the microscale nor at the macroscale) as the number of parareal iterations k goes to infinity.

THEOREM 11. *Consider Algorithm 1, where* $\mathcal{F}_{\Delta t}$ *is the exact propagator of the microscopic problem* (2.1), $\mathcal{C}_{\Delta t}$ *is the exact propagator of the associated macroscopic problem* (2.4), *and* $\mathcal{L}$ *is the lifting operator defined by* (3.5). *We fix the time range* $[0, T]$, *and recall that the size of the boundary layer* $t_\epsilon^{\rm BL}$ *in* (2.1) *is defined by* (2.10).

Then, there exists $\epsilon_0 \in (0,1)$, *that only depends on* A, q, p, α *and* T, *such that, for all* $\epsilon < \epsilon_0$ *and all* $\Delta t > t_\epsilon^{\rm BL}$, *there exists* C, *that depends on* A, q, p, α, Δt *and* T *such that*

$$\sup_{0\leq n\leq N} |E_0^n| \leq C\epsilon \quad \textit{and, for all } k \geq 1, \quad \sup_{0\leq n\leq N} |E_k^n| \leq C\epsilon^2, \tag{4.10}$$

$$\textit{for all } k \geq 0, \quad \sup_{0\leq n\leq N} \|e_k^n\| \leq C\epsilon, \tag{4.11}$$

where $N = T/\Delta t$ *and where the macroscopic (resp. microscopic) error* E_k^n *(resp.* e_k^n*) is defined by* (4.2) *(resp.* (4.1)*). Note that* C *is independent from* ϵ *and* k.

The numerical experiments described in Section 5 show that these error estimates are sharp. Recall that $t_\epsilon^{\rm BL} = C\epsilon\ln(1/\epsilon)$ for some constant C only depending on the matrix A of (2.1) (see (2.10)). The assumption $\Delta t > t_\epsilon^{\rm BL} = C\epsilon\ln(1/\epsilon)$ is therefore automatically satisfied for sufficiently small ϵ, and in particular when the time-step Δt is of the order of the macroscopic time scale.

Proof. Using the definitions (4.1) and (4.2) of the microscopic and the macroscopic errors, and the fact that the microscopic state is reconstructed via the lifting operator $\mathcal{L}$,

$$u_k^n = \mathcal{L}X_k^n,$$

we have

$$\mathcal{L}\, E_k^n = \mathcal{L}\, X_k^n - \mathcal{L}\mathcal{R}\, \Phi_{\Delta t}^n u_0 = u_k^n - \mathcal{L}\mathcal{R}\, \Phi_{\Delta t}^n u_0 = e_k^n + \Phi_{\Delta t}^n u_0 - \mathcal{L}\mathcal{R}\, \Phi_{\Delta t}^n u_0,$$

or, equivalently,

$$e_k^n = \mathcal{L}\, E_k^n - (\mathrm{Id} - \mathcal{L}\mathcal{R})\ \Phi_{\Delta t}^n u_0. \tag{4.12}$$

As a consequence, we can write the recursion (4.5) for E_{k+1}^n in terms of E_k^p only, by eliminating the microscopic errors e_k^p. We have

$$\begin{aligned} \mathcal{R}\Phi_{\Delta t} e_k^p - \rho_{\Delta t} E_k^p &= \mathcal{R}\Phi_{\Delta t}\left[\mathcal{L}\, E_k^p - (\mathrm{Id} - \mathcal{L}\mathcal{R})\ \Phi_{\Delta t}^p u_0\right] - \rho_{\Delta t} E_k^p \\ &= (\mathcal{R}\Phi_{\Delta t}\mathcal{L} - \rho_{\Delta t})\ E_k^p - \mathcal{R}\Phi_{\Delta t}\,(\mathrm{Id} - \mathcal{L}\mathcal{R})\ \Phi_{\Delta t}^p u_0. \end{aligned} \tag{4.13}$$

The first term in (4.13) stems from the difference between the macroscopic evolution of the microscopic system and the evolution of the approximate macroscopic equation. The second term stems from the difference in evolution between a microscopic state and the (unique) microscopic state that is obtained by lifting its restriction. To bound the first term in (4.13), we observe that

$$|\mathcal{R}\Phi_{\Delta t}\mathcal{L} - \rho_{\Delta t}| \le C\epsilon. \tag{4.14}$$

Consider indeed the system (2.1) with the initial condition $(x_0, y_0) = \mathcal{L}(x_0)$. Then $z_0 = 0$ (because $\mathcal{L}(x_0) \in \Sigma$, see (3.5)), and we deduce from (2.8) that

$$|x(\Delta t) - x_0 \exp(\lambda \Delta t)| \le C\epsilon |x_0|,$$

that reads

$$|\mathcal{R}\Phi_{\Delta t}\mathcal{L}(x_0) - \rho_{\Delta t} x_0| \le C\epsilon |x_0|,$$

from which we infer (4.14).

We now turn to the second term of equation (4.13). We introduce the shorthand notation for the exact solution

$$\Phi_{\Delta t}^p u_0 = \tilde{u}^p = \left(\mathcal{R}\tilde{u}^p, \mathcal{R}^\perp \tilde{u}^p\right) = (\tilde{x}^p, \tilde{y}^p)\,.$$

First, using the definition (3.5) for $\mathcal{L}$, we write

$$(\mathrm{Id} - \mathcal{L}\mathcal{R})\, \Phi_{\Delta t}^p u_0 = \begin{bmatrix} 0 \\ \tilde{y}^p - (A^{-1}q)\, \tilde{x}^p \end{bmatrix}.$$

Second, using (2.13) with the initial condition $\overline{x}_0 = 0$, $\overline{y}_0 = \tilde{y}^p - (A^{-1}q)\, \tilde{x}^p$, we get

$$|\mathcal{R}\Phi_{\Delta t}\,(\mathrm{Id} - \mathcal{L}\mathcal{R})\, \Phi_{\Delta t}^p u_0| \le C\epsilon\, \left\|\tilde{y}^p - (A^{-1}q)\, \tilde{x}^p\right\|. \tag{4.15}$$

We are now left with bounding $\left\|\tilde{y}^p - (A^{-1}q)\, \tilde{x}^p\right\|$. We note that $\tilde{y}^p - (A^{-1}q)\, \tilde{x}^p = z(p\Delta t)$, thus, using (2.11) for the solution $u(p\Delta t) = \Phi_{\Delta t}^p u_0$, we deduce that

$$\left\|\tilde{y}^p - (A^{-1}q)\, \tilde{x}^p\right\| = \|z(p\Delta t)\| \le C\epsilon\, (|x_0| + \|z_0\|) \le C\epsilon.$$

Note that we have used the fact that $p \ge 1$ and $\Delta t \ge t_\epsilon^{\mathrm{BL}}$, hence $p\Delta t \ge t_\epsilon^{\mathrm{BL}}$. We then deduce from (4.15) that

$$|\mathcal{R}\Phi_{\Delta t}\,(\mathrm{Id} - \mathcal{L}\mathcal{R})\, \Phi_{\Delta t}^p u_0| \le C\epsilon^2. \tag{4.16}$$

Collecting (4.13), (4.14) and (4.16), we obtain

$$|\mathcal{R}\Phi_{\Delta t} e_k^p - \rho_{\Delta t} E_k^p| \leq C\epsilon \left(|E_k^p| + \epsilon\right),$$

where C only depends on A, q, p, α, T and Δt.

Inserting this bound into the error recursion (4.5), and using that $\rho_{\Delta t} > 0$ (see (2.7)), we get

$$\left|E_{k+1}^n\right| \leq C\epsilon \sum_{p=1}^{n-1} \rho_{\Delta t}^{n-p-1} \left(|E_k^p| + \epsilon\right).$$

We now introduce $\widetilde{E}_k := \max_{0\leq n\leq T/\Delta t} |E_k^n|$, and write

$$\left|E_{k+1}^n\right| \leq C\epsilon \left(\widetilde{E}_k + \epsilon\right) \sum_{p=1}^{n-1} \rho_{\Delta t}^{n-p-1} = C\epsilon \left(\widetilde{E}_k + \epsilon\right) \frac{1-\rho_{\Delta t}^{n-1}}{1-\rho_{\Delta t}}.$$

Let $m := \max\limits_{0\leq n\leq N} \dfrac{1-\rho_{\Delta t}^n}{1-\rho_{\Delta t}}$, which only depends on Δt, T and λ. We obtain

$$\widetilde{E}_{k+1} \leq Cm\epsilon \left(\widetilde{E}_k + \epsilon\right),$$

where Cm only depends on A, q, p, α, T and Δt (and is in particular independent of k and ϵ). We thus have

$$0 \leq \widetilde{E}_k \leq v_k$$

where the sequence $\{v_k\}_{k\in\mathbb{N}}$ is recursively defined by $v_{k+1} = Cm\epsilon\,(v_k + \epsilon)$ and $v_0 = \widetilde{E}_0$, so that

$$v_k = \widetilde{E}_0 (Cm\epsilon)^k + Cm\,\epsilon^2 \frac{1-(Cm\epsilon)^k}{1-Cm\epsilon}.$$

Note that the bound (2.8) reads $|x(t) - X(t)| \leq C\epsilon$, hence $v_0 = \widetilde{E}_0 \leq C\epsilon$.

Let us choose $\epsilon_0 = 1/(Cm)$. Notice that ϵ_0 only depends on A, q, p, α, T and Δt. For any $\epsilon \in (0, \epsilon_0)$, the sequence v_k has a limit as k goes to infinity and there exists C, independent of k and ϵ, such that

$$0 \leq \widetilde{E}_0 \leq C\epsilon \quad \text{and} \quad \forall k \geq 1, \quad 0 \leq \widetilde{E}_k \leq v_k \leq C\epsilon^2.$$

This proves the bound (4.10) on the macroscopic error.

To prove the error bound on the microscopic error, we notice, using the definition (3.5) of $\mathcal{L}$, that

$$(\mathrm{Id} - \mathcal{L}\mathcal{R})\ \Phi_{\Delta t}^n u_0 = (\mathrm{Id} - \mathcal{L}\mathcal{R})\ u(n\Delta t) = \Big(0, y(n\Delta t) - (A^{-1}q)x(n\Delta t)\Big).$$

Since $\Delta t \geq t_\epsilon^{\mathrm{BL}}$, we deduce from (2.11) that

$$\forall n \geq 1, \quad \|(\mathrm{Id} - \mathcal{L}\mathcal{R})\ \Phi_{\Delta t}^n u_0\| \leq C\epsilon. \tag{4.17}$$

Collecting (4.12), (4.10) and (4.17), we obtain, for any $k \geq 0$,

$$\forall n \geq 1, \quad \|e_k^n\| \leq C|E_k^n| + \|(\mathrm{Id} - \mathcal{L}\mathcal{R})\ \Phi_{\Delta t}^n u_0\| \leq C\epsilon.$$

Note that the microscopic error is always dominated by the lifting error (the second term of (4.12)).

Since, at any parareal iteration k, we start with the correct initial condition, we have $e_k^0 = 0$ and we thus have proved (4.11). □

4.3. Error bounds for Algorithm 2. We now consider Algorithm 2, where the reconstruction at each parareal iteration is done using *any* matching operator $\mathcal{P}$ satisfying (3.6) and (3.7). The continuity assumption (3.8) will be added when needed. As pointed out above, we do not assume any specific expression for $\mathcal{P}$ here. We show in Section 4.3.2 that, in contrast to Algorithm 1, the convergence rate obtained with Algorithm 2 increases as the number of parareal iterations k increases. Before that, we show in Section 4.3.1 that, at a given parareal iteration k, the computed trajectories (again both at the macro and the micro scales) are exact up to the time $k\Delta t$.

4.3.1. Local exactness of the algorithm. One of the important properties of the parareal algorithm (1.2) is that it results, after k parareal iterations, in a solution that is exact at all times up to $k\Delta t$. The word "exact" here means that the parareal solution is equal to the solution that would have been obtained using only, in a sequential fashion, the fine-scale propagator up to time $k\Delta t$. We now show that this exactness property holds for the micro-macro parareal algorithm we propose.

THEOREM 12. *Consider Algorithm 2, where $\mathcal{F}_{\Delta t}$ is the exact propagator of the microscopic problem* (2.1)*, $\mathcal{C}_{\Delta t}$ is the exact propagator of the associated macroscopic problem* (2.4)*, $\mathcal{L}$ is the lifting operator defined by* (3.5) *and $\mathcal{P}$ is a matching operator satisfying* (3.6) *and* (3.7)*.*

Denote by u_k^n the microscopic solution obtained at the n-th time-step and k-th parareal iteration, using Algorithm 2. Then, at any parareal iteration $k \geq 1$, we have

$$\forall p \leq k, \quad u_k^p = \Phi_{\Delta t}^p u_0. \tag{4.18}$$

Proof. The proof goes by induction. Consider the parareal iteration $k = 1$. We obviously have $u_1^0 = u_0 = \Phi_{\Delta t}^0 u_0$. At time iteration $n = 1$, in view of (3.15), we have

$$u_1^1 = \mathcal{P}(X_1^1, \overline{u}_0^1),$$

with (see (3.12))

$$X_1^1 = \mathcal{C}_{\Delta t}(X_1^0) + \mathcal{R}(\overline{u}_0^1) - \overline{X}_0^1 = \mathcal{C}_{\Delta t}(X_1^0) + \mathcal{R}(\overline{u}_0^1) - \mathcal{C}_{\Delta t}(X_0^0) = \mathcal{R}(\overline{u}_0^1).$$

Hence, using the fundamental property (3.7),

$$u_1^1 = \mathcal{P}(\mathcal{R}(\overline{u}_0^1), \overline{u}_0^1) = \overline{u}_0^1 = \mathcal{F}_{\Delta t}(u_0^0) = \Phi_{\Delta t} u_0.$$

This proves (4.18) for $k = 1$.

Assume now that, at some parareal iteration $k \geq 1$, we have (4.18). In view of (3.18), this implies that $X_k^p = \mathcal{R}\left[\Phi_{\Delta t}^p u_0\right]$ for any $p \leq k$. Using (4.4) and the fact that $\Phi_{\Delta t} u_k^{p-1} = \Phi_{\Delta t}^p u_0$ for all $p \leq k$, we deduce that

$$\forall n \leq k+1, \quad X_{k+1}^n = \mathcal{R}\Phi_{\Delta t} u_k^{n-1} = \mathcal{R}\Phi_{\Delta t}^n u_0. \tag{4.19}$$

Hence, at the parareal iteration $k+1$, the macroscopic solution is exact up to time $(k+1)\Delta t$. Using (3.15), we now write, for any $n \leq k$,

$$u_{k+1}^{n+1} = \mathcal{P}(X_{k+1}^{n+1}, \overline{u}_k^{n+1}) = \mathcal{P}(\mathcal{R}\Phi_{\Delta t}^{n+1} u_0, \Phi_{\Delta t} u_k^n) = \mathcal{P}(\mathcal{R}\Phi_{\Delta t}^{n+1} u_0, \Phi_{\Delta t}^{n+1} u_0) = \Phi_{\Delta t}^{n+1} u_0,$$

where we have used (4.19) and (3.10) in the first equality, the exactness assumption of the microscopic solution at iteration k in the second equality, and the fundamental property (3.7) of the matching operator $\mathcal{P}$ in the last equality. This proves the relation (4.18) at the iteration $k+1$ and concludes the proof. ∎

This result also directly follows from our above remark that, in its form (3.17), Algorithm 2 is of the form (1.2).

4.3.2. Error bounds. We now establish error bounds on Algorithm 2 that show that the microscopic solution converges towards the exact microscopic dynamics when the modeling error ϵ decreases, and that the convergence rate improves as the number of parareal iterations k increases. This is in contrast with Algorithm 1, where the error does not improve even if k goes to infinity (see Section 4.2). With Algorithm 2, we recover the behavior of the standard parareal algorithm, as recalled in Remark 10 (see e.g. (4.7)).

THEOREM 13. *Consider Algorithm 2, where $\mathcal{F}_{\Delta t}$ is the exact propagator of the microscopic problem* (2.1)*, $\mathcal{C}_{\Delta t}$ is the exact propagator of the associated macroscopic problem* (2.4)*, $\mathcal{L}$ is the lifting operator defined by* (3.5)*, and $\mathcal{P}$ is a matching operator satisfying* (3.6)*,* (3.7) *and* (3.8)*. We fix the time range $[0,T]$, and recall that the size of the boundary layer $t_\epsilon^{\rm BL}$ in* (2.1) *is defined by* (2.10)*.*

Then, there exists $\epsilon_0 \in (0,1)$, that only depends on A, q, p, α and T, such that, for all $\epsilon < \epsilon_0$ and all $\Delta t > t_\epsilon^{\rm BL}$, there exists a constant C_k, independent of ϵ, such that

$$\text{for all } k \geq 0, \quad \sup_{0\leq n\leq N} |E_k^n| \leq C_k \epsilon^{1+\lceil k/2\rceil}, \tag{4.20}$$

$$\text{for all } k \geq 0, \quad \sup_{0\leq n\leq N} \|e_k^n\| \leq C_k \epsilon^{1+\lfloor k/2\rfloor}, \tag{4.21}$$

where $N = T/\Delta t$ and where the macroscopic (resp. microscopic) error E_k^n (resp. e_k^n) is defined by (4.2) *(resp.* (4.1)*). The constant C_k is independent from ϵ, but a priori depends on k, A, q, p, α, Δt and T.*

In (4.20) and (4.21), we used the notation: for any $x \in \mathbb{R}$, $\lceil x\rceil \in \mathbb{Z}$ and $\lfloor x\rfloor \in \mathbb{Z}$ are respectively defined by: $\lfloor x\rfloor \leq x < \lfloor x\rfloor + 1$ and $\lceil x\rceil - 1 < x \leq \lceil x\rceil$.

The above result shows that the parareal iterations alternatingly improve the macroscopic and the microscopic errors by an order of magnitude in ϵ. The numerical results of Section 5 show that (4.20) and (4.21) are sharp error estimates. As already mentioned above, the assumption $\Delta t > t_\epsilon^{\rm BL}$ is automatically satisfied for sufficiently small ϵ, in particular when the time-step Δt is of the order of the macroscopic time-scale.

The bounds (4.20) and (4.21) show that, as k increases, the rate of convergence (with respect to ϵ) of the error increases. The dependence of the constant C_k in these two bounds on Δt and k will be analyzed in details on the numerical test case considered in Section 5.1 (see (5.4) and (5.5)).

Proof. Using (3.15), (3.7) and the definition (4.1) of the microscopic error, we have

$$e_{k+1}^n = u_{k+1}^n - \Phi_{\Delta t}^n u_0 = \mathcal{P}(X_{k+1}^n, \overline{u}_k^n) - \mathcal{P}(\mathcal{R}(\Phi_{\Delta t}^n u_0), \Phi_{\Delta t}^n u_0).$$

Hence, using (3.8), we deduce that

$$\begin{aligned} \left\|e_{k+1}^n\right\| &\leq C\left(\|\overline{u}_k^n - \Phi_{\Delta t}^n u_0\| + \left|X_{k+1}^n - \mathcal{R}(\Phi_{\Delta t}^n u_0)\right|\right) \\ &\leq C\left(\left\|\Phi_{\Delta t} u_k^{n-1} - \Phi_{\Delta t}^n u_0\right\| + \left|E_{k+1}^n\right|\right) \\ &\leq C\left(\left\|\Phi_{\Delta t} e_k^{n-1}\right\| + \left|E_{k+1}^n\right|\right). \end{aligned} \tag{4.22}$$

Since $\Delta t \geq t_\epsilon^{\rm BL}$, we infer from (2.13) and (2.14) that

$$\left\|\Phi_{\Delta t} e_k^{n-1}\right\| \leq C\left(\left|\mathcal{R} e_k^{n-1}\right| + \epsilon \left\|\mathcal{R}^\perp e_k^{n-1}\right\|\right) = C\left(\left|E_k^{n-1}\right| + \epsilon \left\|\mathcal{R}^\perp e_k^{n-1}\right\|\right),$$

where we have used (4.3). We then deduce from (4.22) that

$$\left\|e_{k+1}^n\right\| \leq C\left(\left|E_k^{n-1}\right| + \epsilon\left\|\mathcal{R}^\perp e_k^{n-1}\right\| + \left|E_{k+1}^n\right|\right) \leq C\left(\left|E_k^{n-1}\right| + \epsilon\left\|e_k^{n-1}\right\| + \left|E_{k+1}^n\right|\right). \tag{4.23}$$

We now bound the macroscopic error E_{k+1}^n, using the recursion formula (4.5), that reads

$$E_{k+1}^n = \sum_{p=1}^{n-1} \rho_{\Delta t}^{n-p-1}\, T_k^p, \tag{4.24}$$

with $T_k^p := \mathcal{R}\Phi_{\Delta t} e_k^p - \rho_{\Delta t} E_k^p$. Consider the solution $(\widetilde{x}(t), \widetilde{y}(t))$ to the system (2.1) with initial condition $\widetilde{u}(0) = e_k^p$, that is $\widetilde{x}(0) = \mathcal{R}(e_k^p) = E_k^p$ and $\widetilde{y}(0) = \mathcal{R}^\perp(e_k^p)$. We then have

$$T_k^p = \widetilde{x}(\Delta t) - \widetilde{X}(\Delta t),$$

where $\widetilde{X}(\Delta t)$ is the solution to (2.4) with initial condition $\widetilde{X}(0) = E_k^p$. In view of (2.8), we have

$$|T_k^p| = \left|\widetilde{x}(\Delta t) - \widetilde{X}(\Delta t)\right| \leq C\epsilon\left(|\widetilde{x}(0)| + \|\widetilde{y}(0) - A^{-1}q\widetilde{x}(0)\|\right) \leq C\epsilon\left(|E_k^p| + \|e_k^p\|\right),$$

where C is independent from p, k and ϵ. We are now in position to use the recursion (4.24), from which we infer

$$\left|E_{k+1}^n\right| \leq \sum_{p=1}^{n-1} \rho_{\Delta t}^{n-p-1}\left|T_k^p\right| \leq C\epsilon \sum_{p=1}^{n-1} \rho_{\Delta t}^{n-p-1}\left(|E_k^p| + \|e_k^p\|\right). \tag{4.25}$$

We now prove the theorem by induction, using the two fundamental estimates (4.23) and (4.25). At the parareal iteration $k = 0$, Algorithm 2 is identical to Algorithm 1. In view of (4.10) and (4.11), we thus have

$$\sup_{0\leq n\leq N} |E_0^n| \leq C_0\epsilon \quad \text{and} \quad \sup_{0\leq n\leq N} \|e_0^n\| \leq C_0\epsilon,$$

that is (4.20) and (4.21) for $k = 0$.

Let us now assume that (4.20) and (4.21) hold at any iteration $k' \leq k$, with k an even integer. We prove the bounds at iteration $k+1$. Setting $m = k/2 + 1$ (so that $\lfloor k/2\rfloor = \lceil k/2\rceil = m-1$, $\lfloor (k-1)/2\rfloor = m-2$ and $\lceil (k-1)/2\rceil = m-1$), we thus assume that

$$\sup_{0\leq p\leq N} \left|E_{k-1}^p\right| \leq C_{k-1}\epsilon^m, \quad \sup_{0\leq p\leq N} \left\|e_{k-1}^p\right\| \leq C_{k-1}\epsilon^{m-1},$$
$$\sup_{0\leq p\leq N} |E_k^p| \leq C_k\epsilon^m, \quad \sup_{0\leq p\leq N} \|e_k^p\| \leq C_k\epsilon^m.$$

Then, we infer from (4.25) that, for any $0 \leq n \leq N$,

$$\left|E_{k+1}^n\right| \leq CC_k\epsilon^{m+1} \sum_{p=1}^{n-1} \rho_{\Delta t}^{n-p-1} \leq CC_k\epsilon^{m+1}\frac{1-\rho_{\Delta t}^{n-1}}{1-\rho_{\Delta t}} \leq \frac{CC_k\epsilon^{m+1}}{1-\rho_{\Delta t}} \leq C_{k+1}\epsilon^{m+1},$$

where C_{k+1} is independent from ϵ, but depends on k and Δt. We next deduce from (4.23) that, for any $0 \leq n \leq N$,

$$\left\|e_{k+1}^n\right\| \leq C\left(C_k\epsilon^m + C_k\epsilon^{m+1} + C_{k+1}\epsilon^{m+1}\right) \leq \tilde{C}_{k+1}\epsilon^m,$$

where $\tilde{C}_{k+1}$ is again independent from ϵ, but depends on k. We thus have proved (4.20) and (4.21) at iteration $k+1$.

We now assume that (4.20) and (4.21) hold at any iteration $k' \leq k$, with k an odd integer. We prove the bounds at iteration $k+1$. Setting $m = (k-1)/2+1$ (so that $\lfloor k/2 \rfloor = m-1$, $\lceil k/2 \rceil = m$ and $\lfloor (k-1)/2 \rfloor = \lceil (k-1)/2 \rceil = m-1$), we thus assume that

$$\sup_{0\leq p\leq N} \left|E_{k-1}^p\right| \leq C_{k-1}\epsilon^m, \quad \sup_{0\leq p\leq N} \left\|e_{k-1}^p\right\| \leq C_{k-1}\epsilon^m,$$
$$\sup_{0\leq p\leq N} |E_k^p| \leq C_k\epsilon^{m+1}, \quad \sup_{0\leq p\leq N} |e_k^p| \leq C_k\epsilon^m.$$

Using again equations (4.25) and (4.23), we find that, for any $0 \leq n \leq N$,

$$\left|E_{k+1}^n\right| \leq C_{k+1}\epsilon^{m+1} \quad \text{and} \quad \left\|e_{k+1}^n\right\| \leq C_{k+1}\epsilon^{m+1},$$

where C_{k+1} is again independent from ϵ. We thus have proved (4.20) and (4.21) at iteration $k+1$. This concludes the proof. □

4.4. Error bounds for Algorithm 3. Since Algorithm 3 uses the standard parareal iteration (1.2), local exactness is automatically satisfied. We proceed to proving error bounds on Algorithm 3, which can be compared to those of Algorithm 2.

THEOREM 14. *Consider Algorithm 3 given by* (3.21)*, where $\mathcal{F}_{\Delta t}$ is the exact propagator of the microscopic problem* (2.1)*, $\mathcal{G}_{\Delta t}$ is the exact propagator of the associated macroscopic DAE problem* (3.20)*, and $\mathcal{L}$ is the lifting operator defined by* (3.5)*. We fix the time range $[0,T]$, and recall that the size of the boundary layer t_ϵ^{BL} in* (2.1) *is defined by* (2.10)*.*

Then, there exists $\epsilon_0 \in (0,1)$, that only depends on A, q, p, α and T, such that, for all $\epsilon < \epsilon_0$ and all $\Delta t > t_\epsilon^{\mathrm{BL}}$, there exists a constant C_k, independent of ϵ, such that

$$\text{for all } k \geq 0, \quad \sup_{0\leq n\leq N} |E_k^n| \leq C_k\epsilon^{k+1}, \tag{4.26}$$

$$\text{for all } k \geq 0, \quad \sup_{0\leq n\leq N} \|e_k^n\| \leq C_k\epsilon^{k+1}, \tag{4.27}$$

where $N = T/\Delta t$ and where the macroscopic (resp. microscopic) error E_k^n (resp. e_k^n) is defined by (4.2) *(resp.* (4.1)*). The constant C_k is independent from ϵ, but a priori depends on k, A, q, p, α, Δt and T.*

These results are in agreement with [28, Theorem 2.1]. The numerical results of Section 5 show that (4.26) and (4.27) are sharp error estimates.

The above result shows that, in contrast to Algorithm 2, Algorithm 3 improves the order of convergence (in terms of ϵ) of both the macroscopic and the microscopic errors by an order of magnitude in ϵ at each iteration. As noted in Section 3.3, this improved convergence rate comes at the price of a larger computational cost per iteration.

Proof. Using (3.21) and the definition (4.1) of the microscopic error, we have

$$e_{k+1}^n = u_{k+1}^n - \Phi_{\Delta t}^n u_0 = \Phi_{\Delta t} e_k^{n-1} + \mathcal{L}\rho_{\Delta t}\mathcal{R}\left(e_{k+1}^{n-1} - e_k^{n-1}\right).$$

Using (4.3), we deduce from the above equation that

$$\left\|e_{k+1}^n\right\| \leq \left\|\Phi_{\Delta t} e_k^{n-1} - \mathcal{L}\rho_{\Delta t}\mathcal{R}e_k^{n-1}\right\| + \left\|\mathcal{L}\rho_{\Delta t}E_{k+1}^{n-1}\right\|. \tag{4.28}$$

The first term is decomposed as

$$\begin{aligned}\left\|\Phi_{\Delta t} e_k^{n-1} - \mathcal{L}\rho_{\Delta t}\mathcal{R} e_k^{n-1}\right\| \\ \leq C\left(\left|\mathcal{R}\Phi_{\Delta t} e_k^{n-1} - \rho_{\Delta t}\mathcal{R} e_k^{n-1}\right| + \left\|\mathcal{R}^\perp \Phi_{\Delta t} e_k^{n-1} - R^\perp \mathcal{L}\rho_{\Delta t}\mathcal{R} e_k^{n-1}\right\|\right).\end{aligned}$$

Since $\Delta t \geq t_\epsilon^{\mathrm{BL}}$, we infer an estimate on the first (resp. second) term of the above right-hand side using (2.8) (resp. (2.11)), resulting in

$$\left\|\Phi_{\Delta t} e_k^{n-1} - \mathcal{L}\rho_{\Delta t}\mathcal{R} e_k^{n-1}\right\| \leq C\epsilon \left\|e_k^{n-1}\right\|. \tag{4.29}$$

Collecting (4.28) and (4.29), we deduce that there exists a constant C, independent of ϵ, such that

$$\left\|e_{k+1}^n\right\| \leq C\left(\epsilon\left\|e_k^{n-1}\right\| + \left|E_{k+1}^{n-1}\right|\right). \tag{4.30}$$

This estimate is to be compared with (4.23) in the proof of Theorem 13. Using (4.30), the proof of Theorem 14 is completed via induction, in a way that is similar to (but simpler than) the proof of Theorem 13. □

5. Numerical experiments (linear test-case). In this section, we numerically illustrate the above convergence results on a linear problem. We first consider the case when both the microscopic and the macroscopic models are integrated exactly (Section 5.1). We next consider the case when the macroscopic propagator is a forward Euler discretization, thus introducing some finite step-size error (Section 5.2).

We consider the example system

$$\left\{\begin{array}{rcl} \dot{x} & = & -\dfrac{x}{2} - \dfrac{y_1+y_2}{4} \\ \begin{bmatrix}\dot{y}_1 \\ \dot{y}_2\end{bmatrix} & = & \dfrac{1}{\epsilon}\left(\begin{bmatrix}1\\1\end{bmatrix} x - \begin{bmatrix}1/2 & 1/2 \\ 0 & 1/3\end{bmatrix}\begin{bmatrix}y_1\\y_2\end{bmatrix}\right)\end{array}\right., \tag{5.1}$$

which is of the form (2.1). The associated macroscopic, slow dynamics is given by (2.4) with $\lambda = -1$. The initial condition is $x(0) = 1$, $y_1(0) = y_2(0) = 0$, and we consider the solution on the interval $[0, T]$ with $T = N\Delta t = 10$.

The fine-scale propagator $\mathcal{F}_{\Delta t}$ is the exact integrator of (5.1). The coarse propagator $\mathcal{C}_{\Delta t}$ is the exact integrator of (2.4) in Section 5.1, and a forward Euler discretization of (2.4) in Section 5.2. We choose the parareal time-step $\Delta t = 10^{-1}$, and consider $\epsilon \in [10^{-5}, 10^{-1}]$. The lifting operator $\mathcal{L}$ is defined by (3.5), and we use the matching operator $\mathcal{P}$ defined by (3.9).

We look at the relative macroscopic error $\left|E_k^N\right| / \left|x(T)\right|$ and the relative microscopic error $\left\|e_k^N\right\| / \left\|u(T)\right\|$ at the final time $T = t_N = N\Delta t$ for different iteration numbers k, satisfying $0 \leq k \leq K$.

5.1. Results with exact microscopic and macroscopic integrations. In this section, we take both the fine-scale and the coarse propagators to be the exact integrators.

5.1.1. Algorithm 1. We first consider Algorithm 1 (analyzed in Section 4.2), where the reconstruction at the end of each parareal iteration is done using the lifting operator $\mathcal{L}$.

We set the maximal number of parareal iterations at $K = 2$. Figure 5.1 shows the macroscopic and microscopic errors as a function of ϵ for the chosen values of k.

We see that the macroscopic error is of the order of $O(\epsilon^2)$ as soon as $k \geq 1$ (and is of the order of $O(\epsilon)$ at $k = 0$). The macroscopic error at $k = 2$ is equal to that at $k = 1$. We also observe that the microscopic error is always of the order of ϵ (for any k), although the value of the error is smaller at $k = 1$ than at $k = 0$. These results are in agreement with Theorem 11, and confirm the fact that the accuracy of Algorithm 1 does not improve when k goes to infinity.

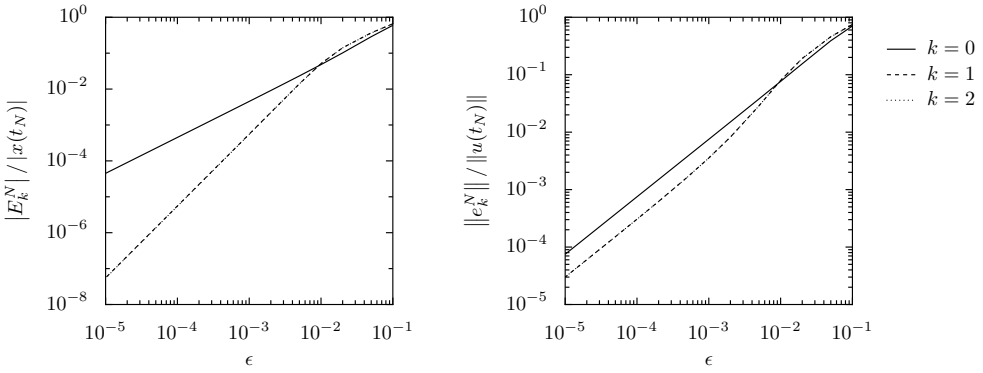

FIG. 5.1. *Algorithm 1 for the system* (5.1)*, with exact fine-scale and coarse propagators: errors as a function of* ϵ *for different values of* k *(left: macroscopic error; right: microscopic error). Note that the lines for* $k \geq 1$ *visually overlap.*

When ϵ is too large, the macroscopic error is not anymore of the order of $O(\epsilon^2)$ at the iteration $k \geq 1$. This is due to the fact that the assumption $\Delta t \geq t_\epsilon^{\mathrm{BL}}$ is no longer satisfied. Recall indeed that we keep Δt fixed, and $t_\epsilon^{\mathrm{BL}} = C\epsilon \ln(1/\epsilon)$ increases if ϵ increases. Hence, for too large values of ϵ, the time step Δt is too small to correct for the initial boundary layer.

5.1.2. Algorithm 2. We now consider Algorithm 2 (analyzed in Section 4.3), where the reconstruction at the end of each parareal iteration is performed using the matching operator $\mathcal{P}$.

The maximal number of parareal iterations is set at $K = 6$. Figure 5.2 shows the macroscopic and microscopic errors as a function of ϵ, for the chosen values of k. The numerical results are in agreement with Theorem 13. At each odd parareal iteration, the order of convergence (in terms of ϵ) of the macroscopic error increases by 1, whereas the microscopic error decreases, but remains of the same order in ϵ. At each even iteration, the converse holds: the order of convergence (in terms of ϵ) increases by 1 for the microscopic error, whereas the macroscopic error decreases but its order remains alike. Note also that, for the smallest considered values of ϵ, the algorithm reaches machine precision in 5 to 6 iterations.

As with Algorithm 1, when ϵ is too large, the numerical results do not agree with the theoretical results, because the chosen time-step Δt does not satisfy the assumption $\Delta t \geq t_\epsilon^{\mathrm{BL}}$.

At this point, we have numerically verified our theoretical results, and know that the macroscopic error is bounded from above by, and actually roughly of the order of,

$$\sup_n |E_k^n| \approx C_{k,\Delta t} \left(\frac{\epsilon}{\Delta t}\right)^{1+\lceil k/2 \rceil}, \tag{5.2}$$

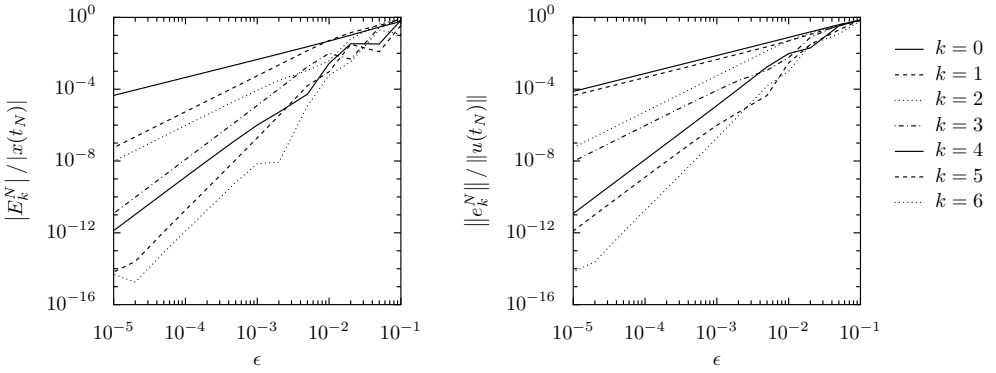

FIG. 5.2. *Algorithm 2 for the system* (5.1)*, with exact fine-scale and coarse propagators: errors as a function of ϵ for different values of k (left: macroscopic error; right: microscopic error).*

where $C_{k,\Delta t}$ a priori depends on k and Δt, but is independent of ϵ (and likewise for the microscopic error).

On Figure 5.3, we plot the macroscopic and microscopic errors as a function of the iteration number k, $0 \leq k \leq K = 30$, for various values of ϵ. We observe an exponential convergence to the exact solution as a function of k, with a convergence rate that increases when ϵ decreases. We deduce from (5.2) how the constant $C_{k,\Delta t}$ depends on k: there exists $C^0_{\Delta t}$ and $C^1_{\Delta t}$, independent of k and ϵ, such that

$$\sup_n |E_k^n| \approx C^0_{\Delta t} \left(C^1_{\Delta t}\right)^{1+\lceil k/2 \rceil} \left(\frac{\epsilon}{\Delta t}\right)^{1+\lceil k/2 \rceil}. \tag{5.3}$$

Note that, for $\epsilon = 10^{-1}$ (which is quite a large value compared to $\Delta t = 10^{-1}$), the convergence is very slow, which is in agreement with the previous observations.

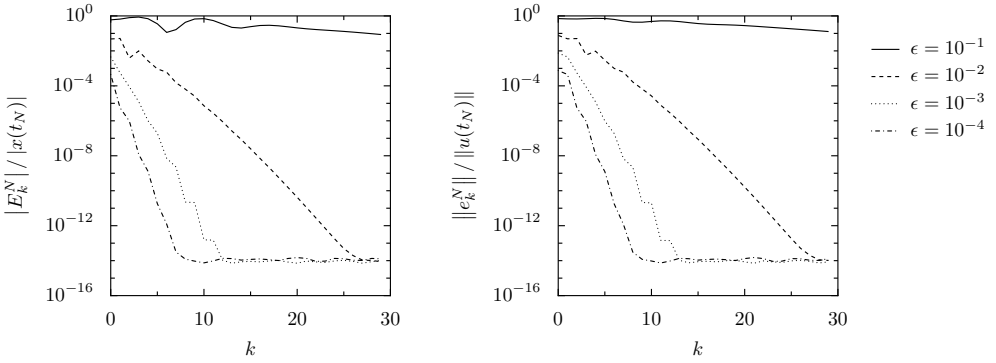

FIG. 5.3. *Algorithm 2 for the system* (5.1)*, with exact fine-scale and coarse propagators and parareal time step $\Delta t = 10^{-1}$: errors as a function of k for different values of ϵ (left: macroscopic error; right: microscopic error).*

To understand how $C^0_{\Delta t}$ and $C^1_{\Delta t}$ depend on Δt, we perform another experiment, in which we fix $\epsilon = 10^{-5}$ and vary Δt. We then plot the error as a function of Δt^{-1} for different values of k (see Figure 5.4). These results show that the macroscopic error varies proportionally to $\left(\Delta t^{-1}\right)^{1+\lceil k/2 \rceil}$. Combined with (5.3), we therefore deduce

that, on this numerical test-case, the macroscopic error satisfies

$$\sup_n |E_k^n| \approx C^0_{\text{macro}} \left(C^1_{\text{macro}} \frac{\epsilon}{\Delta t} \right)^{1+\lceil k/2 \rceil}, \tag{5.4}$$

and likewise for the microscopic error:

$$\sup_n |e_k^n| \approx C^0_{\text{micro}} \left(C^1_{\text{micro}} \frac{\epsilon}{\Delta t} \right)^{1+\lfloor k/2 \rfloor}. \tag{5.5}$$

We in particular see that, if $\epsilon/\Delta t$ is sufficiently small, then the parareal trajectory converges to the exact trajectory when k goes to ∞.

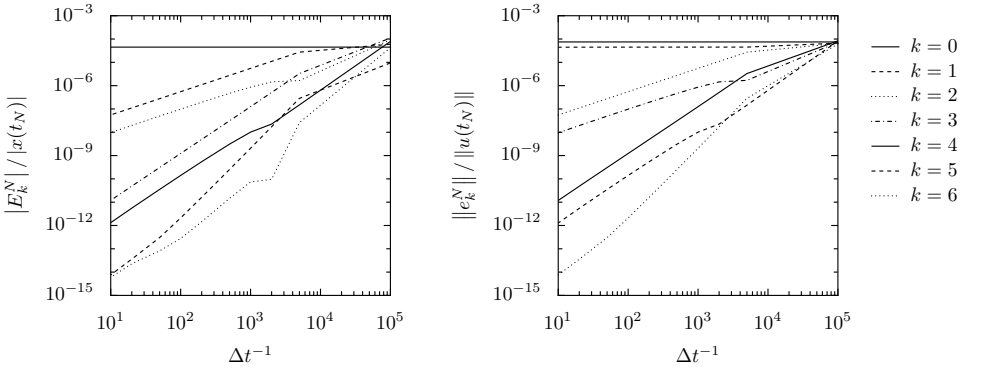

FIG. 5.4. *Algorithm 2 for the system* (5.1)*, with exact fine-scale and coarse propagators and* $\epsilon = 10^{-5}$*: errors as a function of* Δt^{-1} *for different values of* k *(left: macroscopic error; right: microscopic error).*

5.1.3. Algorithm 3. To complete these numerical tests, we consider Algorithm 3 originally proposed in [5, 28] (which we analyzed in Section 4.4), and repeat the previous experiment. The results, shown in Figure 5.5, are in agreement with the theoretical results.

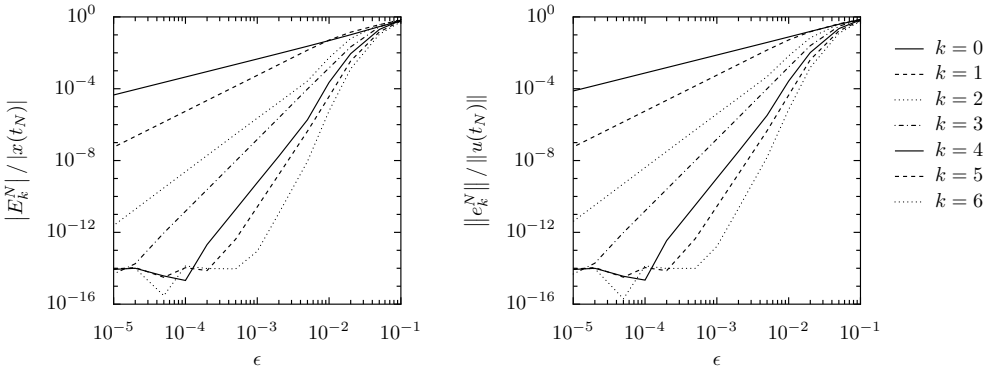

FIG. 5.5. *Algorithm 3 for the system* (5.1)*, with exact fine-scale and coarse propagators: errors as a function of* ϵ *for different values of* k *(left: macroscopic error; right: microscopic error).*

5.2. Results with exact microscopic and approximate macroscopic integration. In this section, we repeat the above experiments, but now using a forward Euler time discretization for the coarse propagator, using the time step Δt (hence, to propagate the system over the time range Δt, the scheme $\mathcal{C}_{\Delta t}$ consists in doing a *single* step of the forward Euler algorithm). The fine-scale propagator is again the exact one.

5.2.1. Algorithm 1. We first consider Algorithm 1 (for which the reconstruction is done using the lifting operator $\mathcal{L}$), and set the maximal number of parareal iterations to $K = 3$. Figure 5.6 shows the errors as a function of ϵ for the chosen values of k.

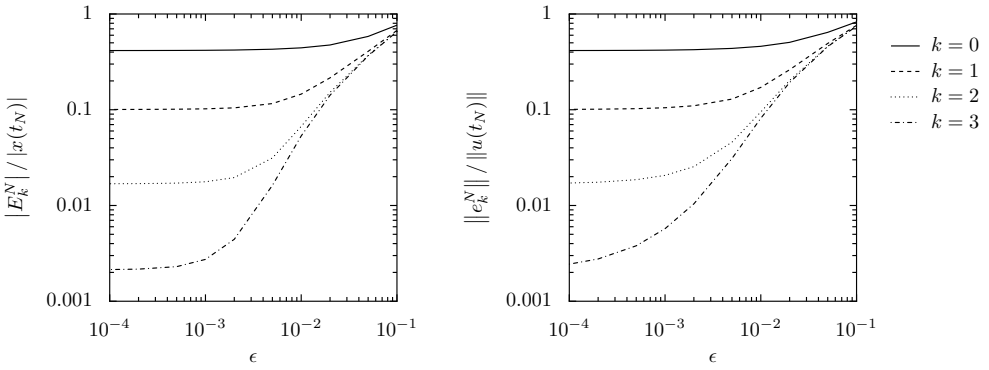

FIG. 5.6. *Algorithm 1 for the system* (5.1)*, with exact fine-scale propagator and approximate coarse propagator: errors as a function of ϵ for different values of k (left: macroscopic error; right: microscopic error).*

When comparing Figure 5.6 with Figure 5.1 (in which the macroscopic dynamics is exactly integrated), we notice that the behavior of the algorithm is similar for large values of ϵ. For small values of ϵ, the errors approach an asymptotic value as ϵ goes to zero (rather than converging to 0 as in Figure 5.1), with an asymptotic value that depends on the number of iterations k. The larger k is, the smaller this asymptotic value is, and the wider the range of ϵ for which results of Figures 5.6 and 5.1 (with approximate or exact integration at the macroscopic scale) agree.

This observation is confirmed in Figure 5.7, where we show the errors as a function of the iteration number k, $1 \leq k \leq K = 8$, for various values of ϵ. We see that the errors first converge exponentially to 0 as k increases, and then reach a plateau. The residual macroscopic (resp. microscopic) error is of the order of $O(\epsilon^2)$ (resp. $O(\epsilon)$).

We explain this behavior as follows. The parareal iterations iteratively correct the approximation made using the coarse propagator. When the coarse propagator is a forward Euler discretization of the approximate macroscopic equation, there are two sources of error: a modeling error (due to the fact that the macroscopic equation (2.4) is only an *approximation* of the slow part of the reference model (2.1)), and a time discretization error. For large values of ϵ, the modeling error is dominant, and the error behaves as if the coarse propagator were exact. For small ϵ, the time discretization is dominant, and the error becomes therefore independent of ϵ. Due to the parareal iterations, the time discretization error is iteratively removed. However, due to the fact that the reconstruction is performed using the lifting operator $\mathcal{L}$, the modeling error never vanishes when $\epsilon > 0$. Hence, when k goes to infinity, Algorithm 1 using an approximate coarse propagator converges to the solution given by a parareal algorithm

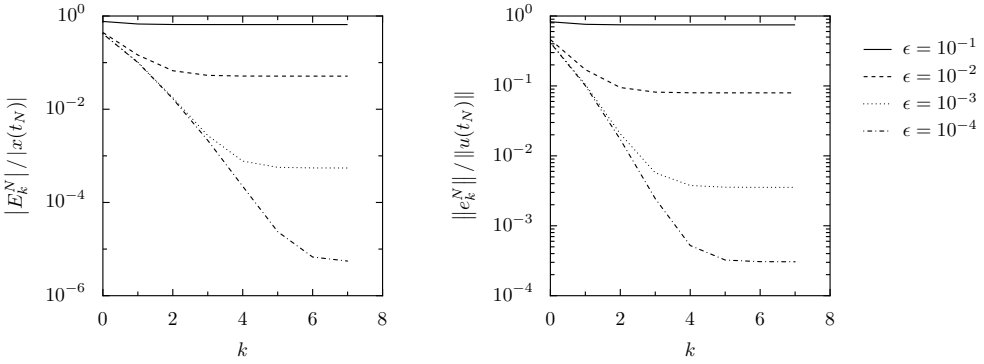

FIG. 5.7. *Algorithm 1 for the system* (5.1)*, with exact fine-scale propagator and approximate coarse propagator: errors as a function of k for different values of ϵ (left: macroscopic error; right: microscopic error).*

with no time-step discretisation error (this latter has been removed by the iterations in k), but with some modeling error. The solution hence converges to that given by Algorithm 1 with exact coarse propagation.

5.2.2. Algorithm 2. We now consider Algorithm 2 (for which the reconstruction is performed using a matching operator $\mathcal{P}$), and set the maximal number of parareal iterations at $K = 6$. Figure 5.8 shows the errors as a function of ϵ for the chosen values of k.

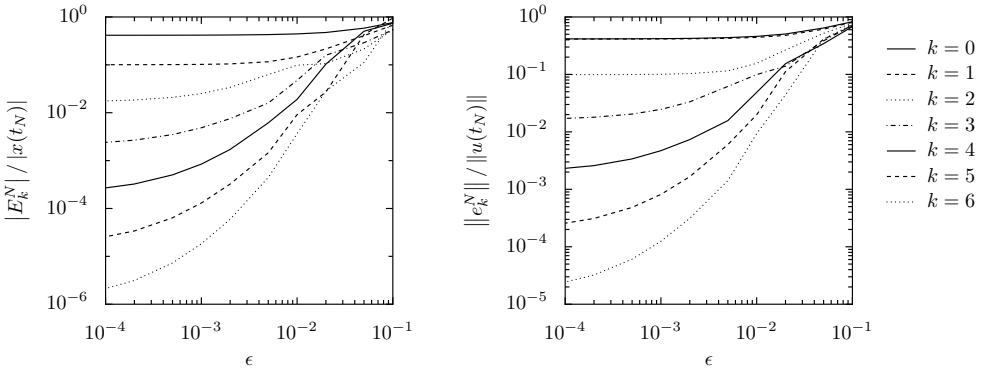

FIG. 5.8. *Algorithm 2 for the system* (5.1)*, with exact fine-scale propagator and approximate coarse propagator: errors as a function of ϵ for different values of k (left: macroscopic error; right: microscopic error).*

We compare Figure 5.8 to the corresponding Figure 5.2 (where the macroscopic equation is exactly integrated). We again notice that, for small ϵ, the algorithm behaves differently: in particular, the convergence when ϵ goes to zero slows down when the macroscopic equation is only approximately integrated. However, the behavior with respect to k is left unchanged. We show on Figure 5.9 the evolution of the errors as a function of the parareal iteration number k, $0 \leq k \leq K = 30$, for a number of fixed values of ϵ. As in Figure 5.3, the computed trajectory again converges to the exact microscopic solution up to machine precision, exponentially with respect to k, despite the presence of time discretization errors at the macroscopic level. Moreover,

comparing these results with those obtained when using an exact coarse propagator (see Figure 5.3), we see that only a few extra parareal iterations are needed.

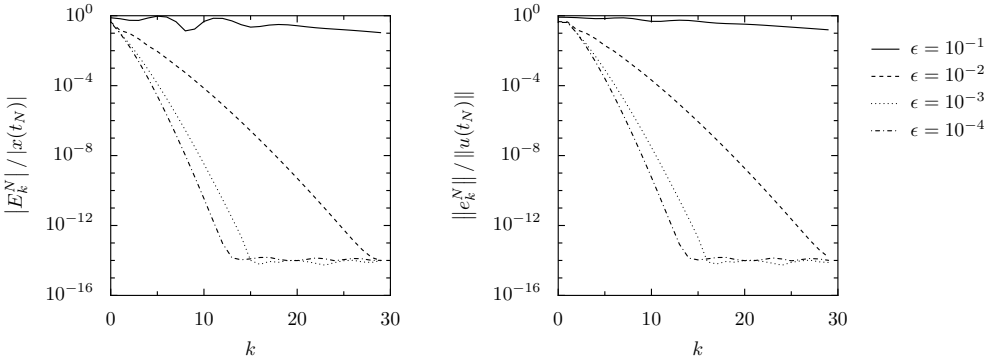

FIG. 5.9. *Algorithm 2 for the system* (5.1)*, with exact fine-scale propagator and approximate coarse propagator: errors as a function of k for different values of ϵ (left: macroscopic error; right: microscopic error).*

6. Nonlinear examples. We finally illustrate the performance of our Algorithm 2 on two nonlinear examples. Such cases are not covered by the theoretical analysis of Section 4, where we considered a linear problem.

The first nonlinear example we consider is a straightforward extension of problem (2.1), and reads

$$\dot{x} = -\lambda x - y, \qquad \dot{y} = \frac{1}{\epsilon}(x^2 - y), \tag{6.1}$$

which is of the form (2.5). The corresponding macroscopic model is

$$\dot{X} = -\lambda X - X^2. \tag{6.2}$$

We use Algorithm 2 to integrate this system. The fine propagator $\mathcal{F}_{\Delta t}$ is a forward Euler scheme for (6.1) with the time step $\delta t = 10^{-5}$. The coarse propagator $\mathcal{C}_{\Delta t}$ is a forward Euler scheme for (6.2) with the time step $\Delta t = 10^{-1}$ (which is equal to the parareal time step). The lifting operator reads $\mathcal{L}(X) = (X, X^2)$, and the matching operator is again given by (3.9). The remaining parameters are chosen identical to those in the previous experiments. On Figure 6.1, we show the error as a function of ϵ for different values of k. Comparing that figure with Figure 5.2, we see that Algorithm 2 performs equally well on this nonlinear case as on the linear problem considered in Section 5.

The second nonlinear example we consider is the so-called Brusselator problem, which was already considered in e.g. [19]. It reads

$$\begin{aligned} \dot{x}_1 &= A - (y+1)x_1 + x_1^2 x_2, \\ \dot{x}_2 &= y x_1 - x_1^2 x_2, \\ \dot{y} &= \frac{1}{\epsilon}(B_0 - y) - y x_1. \end{aligned} \tag{6.3}$$

It models the evolution of the concentration of three chemical species. The concentration of y is reduced via reaction with x, but restored to its base level B_0 with a

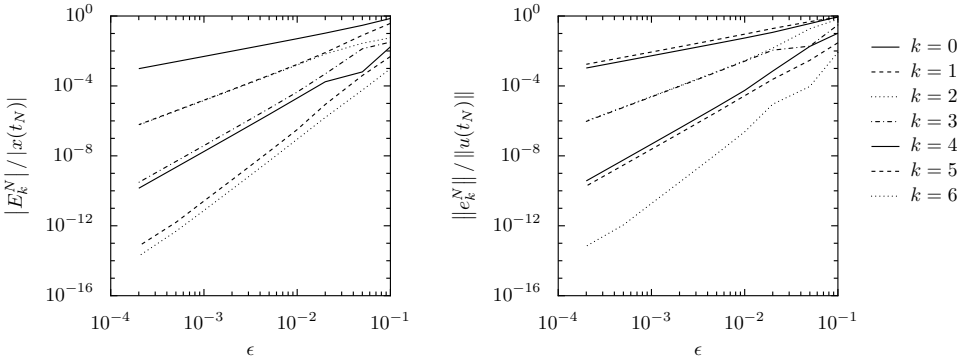

FIG. 6.1. *Algorithm 2 for the system* (6.1)*, with parareal time step* $\Delta t = 10^{-1}$*: errors as a function of* ϵ *for different values of* k *(left: macroscopic error; right: microscopic error).*

characteristic time of the order of ϵ. We choose $A = 1$ and $B_0 = 3$. The fine propagator $\mathcal{F}_{\Delta t}$ is a forward Euler discretization of (6.3) with the time step $\delta t = 10^{-5}$. The coarse propagator $\mathcal{C}_{\Delta t}$ is a forward Euler discretization of the macroscopic model

$$\dot{x}_1 = A - (B_0 + 1)x_1 + x_1^2 x_2, \qquad \dot{x}_2 = B_0\, x_1 - x_1^2 x_2,$$

with the time step $\Delta t = 10^{-1}$ (equal to the parareal time step). This system thus has two slow variables and one fast one: $u = (x, y)$, with $x = (x_1, x_2)$. Note that this case does not enter our theoretical framework for two reasons: (i) the example is nonlinear; and (ii) the equation for y in the microscopic model is not purely a fast equation (the second term in the right-hand side of the equation for $\dot{y}$ in (6.3) is *not* scaled by ϵ^{-1}).

We show on Figure 6.2 the results obtained. Algorithm 2 again performs very well. Actually, on this problem, the convergence behavior of Algorithm 2 closely resembles that of Algorithm 3: at parareal iteration k, the order of convergence (in terms of ϵ) seems to be equal to k, both for the macroscopic and the microscopic errors.

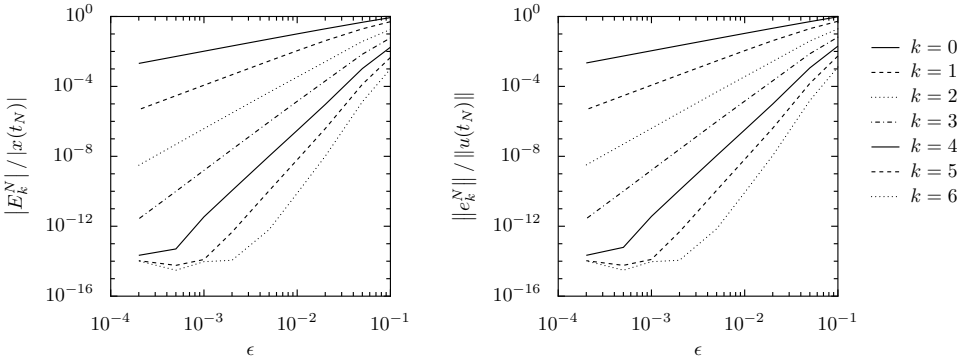

FIG. 6.2. *Algorithm 2 for the system* (6.3)*, with parareal time step* $\Delta t = 10^{-1}$*: errors as a function of* ϵ *for different values of* k *(left: macroscopic error; right: microscopic error).*

7. Discussion and conclusions. We have introduced and analyzed two micro-macro parareal algorithms for the time-parallel integration of singularly perturbed ordinary differential equations, and provided a numerical analysis for the case where

the problem is linear and the coarse and fine-scale propagators integrate the macroscopic, resp. microscopic models exactly. The analysis shows that, when an *appropriate matching operator* is used to update the microscopic state after correction of the macroscopic state (which corresponds to using Algorithm 2), the rate of convergence (in terms of the modelling error ϵ, which quantifies the time scale separation between the microscopic and the macroscopic evolutions) improves at each parareal iteration, and is roughly equal to $\epsilon^{k/2}$. We have illustrated this theoretical result with numerical experiments, and also numerically investigated the effect of using a numerical scheme to integrate the macroscopic model (thereby introducing some discretization error). The results show that the proposed micro-macro parareal algorithm, Algorithm 2, is robust with respect to time discretization errors at the macroscopic level. It can hence be viewed as an interesting way of using an approximate macroscopic model to speed up simulations of high-dimensional multiscale systems of singularly perturbed ordinary differential equations, provided that special care is taken when transferring information between the microscopic and macroscopic levels of description.

Several questions remain open. First, while the analysis reveals that it is important to choose the parareal time step Δt sufficiently large (to average out the initial time boundary layers in the full microscopic dynamics), the dependence on Δt of the numerical error and the convergence rate have not been analyzed. In particular, one may expect an optimal time step Δt to exist that leads to a required accuracy with a minimal cost. Assume again (as for the original parareal algorithm, see the introduction) that the cost of a single evaluation of the fine-scale propagator $\mathcal{F}_{\Delta t}$ is much larger than the cost of propagating the macroscopic system using $\mathcal{C}_{\Delta t}$ over the complete time range $[0, T]$ (This assumption is all the more justified as the macroscopic system is a low-dimensional problem compared to the microscopic problem). Then the cost of the parareal algorithm, after K iterations, is proportional to $K\Delta t/\epsilon$ (We have assumed that the cost of $\mathcal{F}_{\Delta t}$ is proportional to $\Delta t/\epsilon$, since we need to use a time step of the order of ϵ over a time range of length Δt). This cost is to be compared with the cost of the full microscopic sequential integration, which is proportional to $N\Delta t/\epsilon$. The computational speed-up is thus N/K. We saw on Figure 5.2 that, for reasonably small values of ϵ, results obtained at the iteration $K = 6$ were satisfactory. For the test-case considered in Section 5.1.2, the computational speed-up is thus

$$\frac{N}{K} = \frac{T/\Delta t}{K} = 16.6.$$

Second, we expect Algorithm 2 to extend to more general dissipative systems. As pointed out above, we considered here the simple linear problem (2.1) to focus on the issues stemming from using two different levels of description of the same system. We have already checked in Section 6 that Algorithm 2 behaves equally well on nonlinear systems of singularly perturbed ODEs. We currently investigate the extension of the algorithm to a setting, motivated by molecular simulations, where the reference model is a high-dimensional stochastic differential equation (modeling the evolution of all the degrees of freedom of the atomistic system) and the macroscopic model is the effective dynamics of the slow component of the microscopic model, derived under time scale separation assumptions following [24, 25].

Acknowledgements. FL and TL thank Sorin Mitran for enlightening discussions that eventually led to this work. All authors thank the anonymous referees for their comments that lead to a substancial improvement of the manuscript. Part of this work was performed during a research stay of GS at CERMICS (ENPC – Paris),

when he was a Postdoctoral Fellow of the Research Foundation – Flanders (FWO). GS warmly thanks the whole CERMICS team for its hospitality, and both CERMICS and FWO for funding this stay. This work was (partially) completed while FL and TL were visiting the Institute for Mathematical Sciences, National University of Singapore in 2012. This work was partially supported by the Research Council of the K.U. Leuven through grant OT/09/27, by the Interuniversity Attraction Poles Programme of the Belgian Science Policy Office under grant IUAP/V/22, and by the Agence Nationale de la Recherche under grant ANR-09-BLAN-0216-01 (MEGAS). The scientific responsibility rests with its authors.

REFERENCES

[1] L. Baffico, S. Bernard, Y. Maday, G. Turinici, and G. Zérah. Parallel-in-time molecular-dynamics simulations. *Physical Review E*, 66(5):057701, 2002.

[2] G. Bal. Parallelization in time of (stochastic) ordinary differential equations. preprint available at `http://www.columbia.edu/~gb2030/PAPERS/paralleltime.pdf`, 2003.

[3] G. Bal. On the convergence and the stability of the parareal algorithm to solve partial differential equations. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain decomposition methods in science and engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 425–432. Springer Berlin Heidelberg, 2005.

[4] G. Bal and Y. Maday. A "parareal" time discretization for non-linear PDE's with application to the pricing of an american put. In L.F. Pavarino and A. Toselli, editors, *Recent developments in domain decomposition methods*, volume 23 of *Lecture Notes in Computational Science and Engineering*, pages 189–202. Springer Berlin Heidelberg, 2002.

[5] A. Blouza, L. Boudin, and S.-M. Kaber. Parallel in time algorithms with reduction methods for solving chemical kinetics. *Communications in Applied Mathematics and Computational Science*, 5(2):241–263, 2010.

[6] R.E. Caflisch. Monte Carlo and Quasi-Monte Carlo methods. *Acta Numerica*, 7:1–49, 1998.

[7] X. Dai, C. Le Bris, F. Legoll, and Y. Maday. Symmetric parareal algorithms for Hamiltonian systems. *Mathematical Modelling and Numerical Analysis*, 2013. in press (preprint arXiv:1011.6222).

[8] W. E and B. Engquist. The heterogeneous multi-scale methods. *Communications in Mathematical Sciences*, 1(1):87–132, 2003.

[9] W. E, B. Engquist, X. Li, W. Ren, and E. Vanden-Eijnden. Heterogeneous multiscale methods: A review. *Communications in Computational Physics*, 2(3):367–450, 2007.

[10] S. Engblom. Parallel in time simulation of multiscale stochastic chemical kinetics. *Multiscale Modeling and Simulation*, 8:46–68, 2009.

[11] C. Farhat and M. Chandesris. Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid–structure applications. *International Journal for Numerical Methods in Engineering*, 58(9):1397–1434, 2003.

[12] P. Fischer, F. Hecht, and Y. Maday. A parareal in time semi-implicit approximation of the Navier-Stokes equations. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain decomposition methods in science and engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 433–440. Springer Berlin Heidelberg, 2005.

[13] M.J. Gander and E. Hairer. Nonlinear convergence analysis for the parareal algorithm. In U. Langer, M. Discacciati, D.E. Keyes, O. Widlund, and W. Zulehner, editors, *Domain decomposition methods in science and engineering XVII*, volume 60 of *Lect. Notes Comput. Sci. Eng.*, pages 45–56. Springer, 2008.

[14] M.J. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM Journal on Scientific Computing*, 29:556–578, 2007.

[15] I. Garrido, M. Espedal, and G. Fladmark. A convergent algorithm for time parallelization applied to reservoir simulation. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 469–476. Springer Berlin Heidelberg, 2005.

[16] I. Garrido, B. Lee, G.E. Fladmark, and M.S. Espedal. Convergent iterative schemes for time parallelization. *Mathematics of Computation*, 75(255):1403–1428, 2006.

[17] C.W. Gear. Towards explicit methods for differential algebraic equations. *BIT Numerical Mathematics*, 46(3):505–514, 2006.
[18] C.W. Gear, T.J. Kaper, I.G. Kevrekidis, and A. Zagaris. Projecting to a slow manifold: Singularly perturbed systems and legacy codes. *SIAM Journal on Applied Dynamical Systems*, 4(3):711–732, 2005.
[19] C.W. Gear and I.G. Kevrekidis. Projective methods for stiff differential equations: problems with gaps in their eigenvalue spectrum. *SIAM Journal on Scientific Computing*, 24(4):1091–1106, 2003.
[20] D. Givon, R. Kupferman, and A. Stuart. Extracting macroscopic dynamics: model problems and algorithms. *Nonlinearity*, 17(6):R55–R127, 2004.
[21] H.B. Keller. *Numerical methods for two-point boundary-value problems*. Blaisdell (Waltham, MA), 1968.
[22] I.G. Kevrekidis, C.W. Gear, J.M. Hyman, P.G. Kevrekidis, O. Runborg, and C. Theodoropoulos. Equation-free, coarse-grained multiscale computation: enabling microscopic simulators to perform system-level tasks. *Communications in Mathematical Sciences*, 1(4):715–762, 2003.
[23] I.G. Kevrekidis and G. Samaey. Equation-free multiscale computation: Algorithms and applications. *Annual Review on Physical Chemistry*, 60:321–344, 2009.
[24] F. Legoll and T. Lelièvre. Effective dynamics using conditional expectations. *Nonlinearity*, 23(9):2131–2163, 2010.
[25] F. Legoll and T. Lelièvre. Some remarks on free energy and coarse-graining. In B. Engquist, O. Runborg, and R. Tsai, editors, *Numerical Analysis and Multiscale Computations*, volume 82 of *Lect. Notes Comput. Sci. Eng.*, pages 279–329. Springer, 2012.
[26] J.-L. Lions, Y. Maday, and G. Turinici. Résolution d'EDP par un schéma en temps pararéel [A "parareal" in time discretization of PDE's]. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 332(7):661–668, 2001.
[27] C. Lubich and A. Ostermann. Multi-grid dynamic iteration for parabolic equations. *BIT Numerical Mathematics*, 27(2):216–234, 1987.
[28] Y. Maday. Parareal in time algorithm for kinetic systems based on model reduction. In A. Bandrauk, M.C. Delfour, and C. Le Bris, editors, *High-dimensional partial differential equations in science and engineering*, volume 41 of *CRM Proceedings and Lecture Notes*, pages 183–194. American Mathematical Society, 2007.
[29] Y. Maday. The parareal in time algorithm. In F. Magoulès, editor, *Substructuring Techniques and Domain Decomposition Methods*, pages 19–44 (Chapter 2). Saxe-Coburg Publications, Stirlingshire, UK, 2010.
[30] Y. Maday and G. Turinici. A parareal in time procedure for the control of partial differential equations. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 335(4):387–392, 2002.
[31] Y. Maday and G. Turinici. The parareal in time iterative solver: a further direction to parallel implementation. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain decomposition methods in science and engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 441–448. Springer Berlin Heidelberg, 2005.
[32] S. Mitran. Time parallel kinetic-molecular interaction algorithm for CPU/GPU computers. *Procedia Computer Science*, 1:745–752, 2010.
[33] J. Nievergelt. Parallel methods for integrating ordinary differential equations. *Communications of the ACM*, 7(12):731–733, 1964.
[34] G.A. Pavliotis and A.M. Stuart. *Multiscale methods: averaging and homogenization*. Springer, 2008.
[35] G. Staff and E. Rønquist. Stability of the parareal algorithm. In R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain decomposition methods in science and engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 449–456. Springer, 2005.
[36] S. Vandewalle and R. Piessens. Efficient parallel algorithms for solving initial-boundary value and time-periodic parabolic partial differential equations. *SIAM Journal on Scientific and Statistical Computing*, 13:1330–1346, 1992.

Appendix A. Proofs of Lemma 2 and Corollary 3.

Before proving Lemma 2 and Corollary 3, we start with a preliminary result. Here and in all what follows, $\|\cdot\|$ denotes the Euclidean norm when applied to vectors, and the associated operator norm when applied to matrices.

LEMMA 15. *Let M be a matrix in $\mathbb{R}^{d\times d}$ such that the real part of the spectrum of M is positive. Then, there exist $C>0$ and $\mu>0$ such that, for all time $t\geq 0$,*

$$\|\exp(-Mt)\| \leq C\exp(-\mu t). \tag{A.1}$$

One can choose $\mu = \inf_{\nu\in\sigma(M)} \mathrm{Re}(\nu)/2$, where $\sigma(M)$ denotes the spectrum of M. We also have

$$\left\|M^{-1}\right\| \leq \frac{C}{\mu}. \tag{A.2}$$

Proof. We introduce the Jordan form of the matrix M. Let us assume for simplicity of notation that M has only two Jordan blocks associated to two complex eigenvalues $\lambda_1 \neq \lambda_2$ with $0<\mathrm{Re}(\lambda_1)\leq \mathrm{Re}(\lambda_2)$. The generalization to any number of Jordan blocks is straightforward. Let us denote $\mu = \inf_{\nu\in\sigma(M)} \mathrm{Re}(\nu)/2 = \mathrm{Re}(\lambda_1)/2 > 0$.

Since M has only two Jordan blocks, there exists an invertible matrix $Q\in\mathbb{R}^{d\times d}$ such that

$$M = Q^{-1}\begin{bmatrix} N_{d_1}(\lambda_1) & 0 \\ 0 & N_{d_2}(\lambda_2)\end{bmatrix} Q,$$

where $d_1+d_2=d$, and, for any $m\in\mathbb{N}^\star$ and any $\lambda\in\mathbb{C}$,

$$N_m(\lambda) = \begin{bmatrix} \lambda & 1 & 0 & 0 & & \\ 0 & \lambda & 1 & 0 & & \\ & & \cdots & & & \\ & & & \cdots & & \\ & & & 0 & \lambda & 1 \\ & & & & 0 & \lambda \end{bmatrix} \in \mathbb{R}^{m\times m}.$$

We compute, for any $t\in\mathbb{R}$,

$$\exp(Mt) = Q^{-1}\begin{bmatrix} \exp(\lambda_1 t)P_{d_1}(t) & 0 \\ 0 & \exp(\lambda_2 t)P_{d_2}(t)\end{bmatrix} Q, \tag{A.3}$$

where, for any $m\in\mathbb{N}^\star$ and any $t\in\mathbb{R}$,

$$P_m(t) = \begin{bmatrix} 1 & t & t^2/2 & t^3/6 & \dots & \dots & t^{m-1}/((m-1)!) \\ 0 & 1 & t & t^2/2 & \dots & \dots & t^{m-2}/((m-2)!) \\ & & \cdots & & & & \\ & & & \cdots & & & \\ & & & & 0 & 1 & t \\ & & & & & 0 & 1 \end{bmatrix} \in \mathbb{R}^{m\times m}.$$

Since $P_m(t)$ is a matrix with entries which are polynomial functions of t, there exists a constant C that only depends on the matrix M such that

$$\forall t\geq 0, \quad \|P_{d_1}(-t)\| + \|P_{d_2}(-t)\| \leq C\exp(\mu t).$$

We then infer from (A.3) that there exists a constant that only depends on M such that

$$\forall t\geq 0, \quad \|\exp(-Mt)\| \leq C\exp(-\mu t).$$

This yields (A.1). Then, (A.2) is obtained using the fact that

$$\left\|M^{-1}\right\| = \left\|\int_0^\infty \exp(-Mt)\,dt\right\| \leq \int_0^\infty \|\exp(-Mt)\|\,dt \leq C\int_0^\infty \exp(-\mu t)\,dt = \frac{C}{\mu}.$$

This concludes the proof of Lemma 15. □

We are now in position to prove Lemma 2 and Corollary 3.

Proof of Lemma 2. We start by writing

$$\begin{aligned}\dot z &= \dot y - A^{-1}q\,\dot x\\ &= \frac{1}{\epsilon}\left(qx - Ay\right) - A^{-1}q\left(\alpha x + p^T y\right)\\ &= -\left[\frac{A}{\epsilon} + \left(A^{-1}q\right)p^T\right] z - \lambda\left(A^{-1}q\right)x, \end{aligned} \tag{A.4}$$

where λ is defined by (2.4). Introducing

$$M^\epsilon := A + \epsilon\left(A^{-1}q\right)p^T \in \mathbb{R}^{(d-1)\times(d-1)}, \quad V := \lambda\left(A^{-1}q\right) \in \mathbb{R}^{(d-1)},$$

we recast (A.4) as

$$\dot z = -\frac{M^\epsilon}{\epsilon} z - Vx. \tag{A.5}$$

From the definition of M^ϵ, and in view of Assumption (2.3), it is clear that there exists a critical value $\epsilon_0(A,q,p)$ such that for all $\epsilon < \epsilon_0(A,q,p)$, the matrix M^ϵ has a spectrum with a real part bounded from below by $\lambda_-/2 > 0$, where λ_- is independent of ϵ. Up to a modification of $\epsilon_0(A,q,p,\alpha)$, the same property holds true for the matrix $M^\epsilon + \epsilon\lambda\,\mathrm{Id}$ that will appear below (where λ is defined by (2.4)). In the sequel of the proof, we will systematically work with $\epsilon < \epsilon_0(A,q,p,\alpha)$.

By explicit integration of (A.5), we have

$$z(t) - \exp(-M^\epsilon t/\epsilon)z_0 = -\int_0^t \exp\left[-M^\epsilon(t-s)/\epsilon\right] V\,x(s)\,ds. \tag{A.6}$$

From (2.1), we have $\dot x = \lambda x + p^T z$. Using equation (A.6), we thus obtain

$$\begin{aligned} x(t) - x_0\exp(\lambda t) &= p^T\int_0^t \exp\left(\lambda(t-s)\right) z(s)ds\\ &= p^T\int_0^t \exp\left(\lambda(t-s)\right)\exp\left(-M^\epsilon s/\epsilon\right) z_0 ds\\ &\quad - p^T\int_0^t \exp\left(\lambda(t-s)\right)\int_0^s \exp\left(-M^\epsilon(s-r)/\epsilon\right) Vx(r)drds. \end{aligned} \tag{A.7}$$

To bound the first term of (A.7), we write, using Lemma 15,

$$\begin{aligned}
&\left\| \int_0^t \exp\left(\lambda(t-s)\right) \exp\left(-M^\epsilon s/\epsilon\right) ds \right\| \\
&= \exp(\lambda t) \left\| \int_0^t \exp\left(-\left(M^\epsilon/\epsilon + \lambda \,\mathrm{Id}\right) s\right) ds \right\| \\
&\leq \left\| \left(M^\epsilon/\epsilon + \lambda \,\mathrm{Id}\right)^{-1} \right\| \left\| \exp(\lambda t)\,\mathrm{Id} - \exp\left(-\left(M^\epsilon/\epsilon\right) t\right) \right\| \\
&\leq \epsilon \left\| \left(M^\epsilon + \epsilon\lambda \,\mathrm{Id}\right)^{-1} \right\| \Big(\left\| \exp(\lambda t)\,\mathrm{Id} \right\| + \left\| \exp\left(-\left(M^\epsilon/\epsilon\right) t\right) \right\| \Big) \\
&\leq \epsilon \frac{C(A,q,p,\alpha)}{\lambda_-/4} \Big(\exp(\lambda T) + C(A,q,p,\alpha) \exp\left(-\lambda_- t/(4\epsilon)\right) \Big) \\
&\leq C(A,q,p,\alpha,T)\,\epsilon, \qquad \text{(A.8)}
\end{aligned}$$

when $\epsilon \leq \epsilon_0(A,q,p,\alpha)$. Turning to the second term of (A.7), we use Fubini's theorem, and write

$$\begin{aligned}
&\int_0^t \exp\left(\lambda(t-s)\right) \int_0^s \exp\left(-M^\epsilon(s-r)/\epsilon\right) V\, x(r)\, dr ds \\
&= \exp(\lambda t) \int_0^t \exp(M^\epsilon r/\epsilon) \left[\int_r^t \exp\left(-(M^\epsilon/\epsilon + \lambda \,\mathrm{Id}) s\right) ds \right] V\, x(r)\, dr \\
&= \exp(\lambda t) \int_0^t \exp(M^\epsilon r/\epsilon) \Big[\exp\left(-(M^\epsilon/\epsilon + \lambda \,\mathrm{Id}) r\right) - \exp\left(-(M^\epsilon/\epsilon + \lambda \,\mathrm{Id}) t\right) \Big] \\
&\qquad\qquad \times \left(M^\epsilon/\epsilon + \lambda \,\mathrm{Id}\right)^{-1} V\, x(r)\, dr \\
&= \int_0^t \left[\exp\left(\lambda(t-r)\right) \mathrm{Id} - \exp\left(-M^\epsilon(t-r)/\epsilon\right) \right] \left(M^\epsilon/\epsilon + \lambda \,\mathrm{Id}\right)^{-1} V\, x(r)\, dr.
\end{aligned}$$

Therefore, for $\epsilon \leq \epsilon_0(A,q,p,\alpha)$, using Lemma 15, we obtain

$$\begin{aligned}
&\left\| \int_0^t \exp\left(\lambda(t-s)\right) \int_0^s \exp\left(-M^\epsilon(s-r)/\epsilon\right) V x(r) dr ds \right\| \\
&\leq \left\| \left(M^\epsilon/\epsilon + \lambda \,\mathrm{Id}\right)^{-1} \right\| \ \|V\| \sup_{0\leq r\leq t} |x(r)| \int_0^t \left\| \exp\left(\lambda(t-r)\right) \mathrm{Id} - \exp\left(-M^\epsilon(t-r)/\epsilon\right) \right\| dr \\
&\leq \epsilon \left\| \left(M^\epsilon + \epsilon\lambda \,\mathrm{Id}\right)^{-1} \right\| C(A,q,p,\alpha,T) \sup_{0\leq r\leq t} |x(r)| \\
&\leq C(A,q,p,\alpha,T)\,\epsilon \sup_{0\leq r\leq t} |x(r)|. \qquad \text{(A.9)}
\end{aligned}$$

Combining equations (A.7), (A.8) and (A.9), we get

$$\begin{aligned}
|x(t) - x_0 \exp(\lambda t)| &\leq C(A,q,p,\alpha,T) \|p\| \, \|z_0\| \, \epsilon + C(A,q,p,\alpha,T) \, \|p\| \, \epsilon \sup_{0\leq r\leq t} |x(r)| \\
&\leq C(A,q,p,\alpha,T) \epsilon \left(\|z_0\| + \sup_{0\leq r\leq t} |x(r)| \right),
\end{aligned}$$

and hence,

$$\begin{aligned}
&\sup_{0\le t\le T} |x(t) - x_0 \exp(\lambda t)| \\
&\le C(A,q,p,\alpha,T)\ \epsilon \left(\|z_0\| + \sup_{0\le r\le T} |x(r)| \right) \\
&\le C(A,q,p,\alpha,T)\ \epsilon \left(\|z_0\| + |x_0| + \sup_{0\le r\le T} |x(r) - x_0 \exp(\lambda r)| \right).
\end{aligned}$$

We deduce that, for $\epsilon \le \epsilon_0(A,q,p,\alpha,T)$,

$$\begin{aligned}
\sup_{0\le t\le T} |x(t) - x_0 \exp(\lambda t)| &\le \frac{C(A,q,p,\alpha,T)\,\epsilon}{1 - C(A,q,p,\alpha,T)\,\epsilon}\ (\|z_0\| + |x_0|) \\
&\le \overline{C}(A,q,p,\alpha,T)\ \epsilon\ (\|z_0\| + |x_0|). \qquad \text{(A.10)}
\end{aligned}$$

This proves (2.8).

We now turn to proving the bound (2.9) on $z(t)$. Introducing

$$B := \left(A^{-1}q\right) p^T \in \mathbb{R}^{(d-1)\times(d-1)},$$

we now recast (A.4) as

$$\dot{z} = -\left[\frac{A}{\epsilon} + B\right] z - Vx = -\frac{A}{\epsilon} z - [Bz + Vx].$$

By explicit integration, we have

$$z(t) - \exp(-At/\epsilon) z_0 = -\int_0^t \exp\left[-A(t-s)/\epsilon\right] (Bz(s) + Vx(s))\, ds. \qquad \text{(A.11)}$$

Using (A.1) and Assumption (2.3), we obtain

$$\begin{aligned}
\|z(t) - \exp(-At/\epsilon) z_0\| &\le C(A) \int_0^t \exp\left[\lambda_- \frac{s-t}{2\epsilon}\right] \|Bz(s) + Vx(s)\|\, ds \\
&\le C(A,q,p,\alpha) \int_0^t \exp\left[\lambda_- \frac{s-t}{2\epsilon}\right] (\|z(s)\| + |x(s)|)\, ds \\
&\le C(A,q,p,\alpha) \sup_{s\in[0,t]} (\|z(s)\| + |x(s)|)\, \frac{2\epsilon}{\lambda_-} \\
&\le \epsilon C(A,q,p,\alpha) \left(\sup_{s\in[0,t]} \|z(s) - \exp(-As/\epsilon) z_0\| \right. \\
&\qquad \left. + \sup_{s\in[0,t]} \|\exp(-As/\epsilon) z_0\| + \sup_{s\in[0,t]} |x(s)| \right).
\end{aligned}$$

Taking the supremum over $t \in [0,T]$, we obtain, for $\epsilon \le \epsilon_0(A,q,p,\alpha)$,

$$\begin{aligned}
\sup_{t\in[0,T]} \|z(t) - \exp(-At/\epsilon) z_0\| &\le \frac{\epsilon C(A,q,p,\alpha)}{1 - \epsilon C(A,q,p,\alpha)} \left(\sup_{s\in[0,T]} \|\exp(-As/\epsilon) z_0\| + \sup_{s\in[0,T]} |x(s)| \right) \\
&\le \frac{\epsilon C(A,q,p,\alpha)}{1 - \epsilon C(A,q,p,\alpha)} \left(C(A)\|z_0\| + \sup_{s\in[0,T]} |x(s)| \right).
\end{aligned}$$

We then deduce from (A.10) that, for $\epsilon \leq \epsilon_0(A, q, p, \alpha, T)$,

$$\sup_{t\in[0,T]} \|z(t) - \exp(-At/\epsilon)z_0\| \leq \frac{\epsilon C(A, q, p, \alpha, T)}{1 - \epsilon C(A, q, p, \alpha)} \left(\|z_0\| + |x_0|\right)$$
$$\leq \epsilon \overline{C}(A, q, p, \alpha, T) \left(\|z_0\| + |x_0|\right). \qquad \text{(A.12)}$$

This proves (2.9).

We finally turn to proving (2.11). Using (A.1), we see that

$$\|\exp(-At/\epsilon)\| \leq C(A) \exp(-\lambda_- t/(2\epsilon)),$$

thus, for times $t \geq t_\epsilon^{\mathrm{BL}} = \dfrac{2\epsilon}{\lambda_-} \ln(1/\epsilon)$, we have $\|\exp(-At/\epsilon)\| \leq C(A)\epsilon$. We then deduce from (A.12) the bound (2.11). This concludes the proof of Lemma 2.

Proof of Corollary 3. The first assertion follows directly from (2.8) and the fact that $\|z_0\| \leq \|y_0\| + C|x_0|$. The second assertion follows from (2.11) and (2.13).