



HAL
open science

Time-parallelization of sequential data assimilation problems

Sebastián Riffo, Félix Kwok, Julien Salomon

► **To cite this version:**

Sebastián Riffo, Félix Kwok, Julien Salomon. Time-parallelization of sequential data assimilation problems. 2022. hal-03884022

HAL Id: hal-03884022

<https://hal.science/hal-03884022>

Preprint submitted on 4 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TIME-PARALLELIZATION OF SEQUENTIAL DATA ASSIMILATION PROBLEMS

SEBASTIÁN RIFFO*, FELIX KWOK†, AND JULIEN SALOMON‡

Abstract. This paper is devoted to the problem of time parallelization of assimilation methods applying on unbounded time domain. In this way, we present a general procedure to couple the Luenberger observer with time parallelization algorithm. Our approach is based on *a posteriori* error estimates of the latter and preserves the rate of the non-parallelized observer. We then focus on the case where the Parareal algorithm is used as time parallelization algorithm, and derive a bound of the efficiency of our procedure. A variant devoted to the case a large number of processors is also proposed. We illustrate the performance of our approach with numerical experiments.

Key words. Parareal algorithm, Luenberger Observer, data assimilation, time parallelization.

AMS subject classifications. 49M27 , 68W10 , 65K10, 65F08 , 93B40

1. Introduction. The assumptions behind a mathematical model not only determine their range of applicability, but also induce an inevitable gap between predictions and reality. In order to narrow this difference, one can sacrifice the simplicity of the model or incorporate real data instead, by following a data assimilation (DA) procedure. Among these approaches, *sequential* methods construct a new system which uses the available observations (that arrive uninterrupted in time) to approximate the true state, whereas *variational* methods follow an optimal control approach using the information collected in a fixed amount of time.

In a deterministic context, sequential methods are often called *observers*. In the pioneering work [20], Luenberger introduced a dynamic which imitates the original model, by including an extra term that measures the misfit between the observations and its own predictions. As long as the original model is *observable*, this error can be driven to zero at exponential rate by properly choosing a certain matrix, meaning that the true state is recovered asymptotically. An alternative is the *Kalman filter* [17], which takes into account measurement errors and model inaccuracies represented by Gaussian white noises (both stationary and mutually uncorrelated), in order to compute a state estimate that minimizes the mean square error. Note that extensions to the nonlinear case have been developed, e.g., nonlinear versions of Luenberger observer [3] and *Extended Kalman filter* [16].

Significant difficulties appear when applying these techniques to, e.g., meteorology [15, 27] or oceanography problems [21, 28, 19, 4]. Here, the number of state variables and the vast amount of observations lead to very costly computations. To overcome this issue, one can consider space or time domain decomposition methods, which accelerate the numerical solution of PDEs using parallel computing. We now briefly recall the main approaches and refer to [9, 10] for their detailed description.

Since the seminal work of Schwarz [25], spatial domain decomposition and corresponding parallelization techniques have seen many qualitative and quantitative improvements, all of which are nowadays well documented [9, 8]. The time direction is significantly more complex to parallelize. The solution process of evolutionary

*CEREMADE, CNRS, UMR 7534, Université Paris-Dauphine, PSL Research University, 75016 Paris, France (reyesriffo@ceremade.dauphine.fr).

†Département de mathématiques et de statistique, Université Laval (felix.kwok@mat.ulaval.ca).

‡INRIA Paris, ANGE Project-Team, 75589 Paris Cedex 12, France and Sorbonne Université, CNRS, Laboratoire Jacques-Louis Lions, 75005 Paris, France (julien.salomon@inria.fr).

PDEs is indeed intrinsically sequential, so that time decomposition is, at first glance, not amenable to parallel computing. However, over the last 50 years, many parallel-in-time methods have been developed [10]. The origins of these approaches can be traced back to Nievergelt [23], who first introduced the concept that has later been dubbed *Multiple shooting*: decompose the time interval into disjoint subintervals and solve simultaneously a family of initial-value problems, breaking the intrinsic sequential nature of the time-dependent differential equation. Among these methods, one of the most recent ones is the Parareal algorithm [18].

Different procedures have been developed to couple space or time parallel methods with data assimilation problems. Trémolet and Le Dimet [26] were among the first to address the parallelization of *Variational* data assimilation problems in meteorology. In a continuous setting, they proposed a domain decomposition approach combined with an adjoint method, by assigning to each subdomain a local version of a continuous minimization problem, with an extra term on the local cost functional to enforce the continuity of the state between adjacent domains. Following this approach, Rao and Sandu [24] apply a quasi-Newton solver to the 4D-Var problem [6], and time-parallelize the computation of the gradient. A more sophisticated approach is proposed by D'Amore and Cacciapuoti [7], who combine the Parareal algorithm with the Multiplicative Parallel Schwarz method (MPS) to solve 4D-Var. Note finally that time parallelization has also been combined with optimization solvers in the neighbor field of control [22, 12].

Parallel-in-time algorithms could be quite useful when dealing with long time intervals, as is the case of *sequential* DA methods, where information can arrive uninterrupted. However, coupling these two approaches is not straightforward since the former generally applies on bounded time intervals. In this paper, we propose a first general method to time-parallelize an unbounded assimilation method, namely, the Luenberger observer. Our approach is based on a sequential treatment of time windows, each windows being processed in parallel.

Our paper is organized as follows: we start in Section 3 by proposing a procedure to couple sequential data assimilation methods with parallel-in-time algorithms, which involves splitting the unbounded time interval into subintervals of the same length (*windows*) and then applying, following a sequential order, the time-parallel solver on each window. By considering the Luenberger observer as an assimilation method, we provide an accuracy criterion that preserves its exponential rate of convergence, which yields an *a posteriori* estimate of the accuracy of the solver. In order to go further, in Section 4 we use the Parareal algorithm as a parallel-in-time solver. This allows us to design an alternative algorithm that provides an *a priori* estimate of the number of iterations required on each window, which also enables us to investigate the theoretical efficiency of the entire procedure. These results are based on a new convergence estimate that we derive for Parareal when the coarse solver is a contraction mapping. Finally, we present some numerical results in Section 5.

Throughout this paper, $\|\cdot\|$ represents the induced 2-norm of a matrix.

2. The Luenberger observer. Control theory usually requires a complete knowledge of the state vector. However, due to certain limitations related to a problem, for instance the number of available measurements, one can often have access only to partial information. An example which fits into this setting is given by

$$(2.1) \quad \begin{cases} \dot{x}(t) = Ax(t) + Bu(t), & x(0) = x_0, \\ y(t) = Cx(t), \end{cases}$$

where $A \in \mathcal{M}_{m \times m}(\mathbb{R})$, $B \in \mathcal{M}_{m \times p}(\mathbb{R})$ and $C \in \mathcal{M}_{q \times m}(\mathbb{R})$ are assumed to be known. Here $m, p, q \in \mathbb{N}^*$, with $p, q < m$ (and generally $q \ll m$ in the applications); $x \in \mathbb{R}^m$ is the state vector, $y \in \mathbb{R}^q$ is the measured output, $u \in \mathbb{R}^p$ and $t \in (0, +\infty)$. The initial condition $x(0) = x_0$ is unknown.

In such a situation, one can try to compute an estimate $\hat{x}(t)$ of $x(t)$, knowing only the input $u(t)$ and output $y(t)$. To tackle this problem, Luenberger [20] proposed to consider the dynamical system

$$(2.2) \quad \begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L[y(t) - \hat{y}(t)], & \hat{x}(0) = \hat{x}_0 \\ \hat{y}(t) = C\hat{x}(t) \end{cases}$$

with $L \in \mathcal{M}_{m \times q}(\mathbb{R})$ the *observer gain* and \hat{x}_0 an arbitrary initial condition. Equations (2.2) are known as the *Luenberger observer* or the *Identity observer*.

The matrix L needs to be specified, but let us already note that it plays an important role in the estimation error $\varepsilon(t) := x(t) - \hat{x}(t)$. Indeed, subtracting (2.1) and (2.2), and then solving the resultant ODE, one obtains

$$(2.3) \quad \varepsilon(t) = e^{(A-LC)t} (x(0) - \hat{x}(0)).$$

This last equality shows that the error will decay to zero if the eigenvalues of $A - LC$ lie in the open left half-plane $\{z \in \mathbb{C} : \operatorname{Re}\{z\} < 0\}$, where $\operatorname{Re}\{z\}$ denotes the real part of z . This property is related to the observability condition. More precisely, recall that (2.1) is *observable* if the rank of the matrix

$$\mathcal{C} := \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{m-1} \end{bmatrix}$$

is m . We then have the following result, often called the *Identity observer Theorem* [20, p.303]:

THEOREM 2.1. *Given a completely observable system (2.1), an identity observer of the form (2.2) exists. Moreover, the eigenvalues of $A - LC$ can be selected arbitrarily.*

This theorem shows that given a set $\{\mu_i\}_{i=1, \dots, m}$, there exists L satisfying

$$(2.4) \quad \det(sI - (A - LC)) = \phi(s),$$

where $\phi(s) = (s - \mu_1) \cdots (s - \mu_m)$, i.e., $\{\mu_i\}_{i=1, \dots, m}$ are the eigenvalues of $A - LC$.

Note that for a single-input single-output system, i.e. $p = q = 1$, one could determine a unique $L \in \mathbb{R}^m$ by equating the m coefficients of both polynomials in (2.4). However, this approach leads to highly nonlinear equations that are in practice not tractable. Another way to proceed is the Bass-Gura method [5], which requires the first companion form of A and the coefficients of $\phi(s)$. An even more direct method is the Ackermann's formula [1] for an observable system, given by

$$L = \phi(A)C^{-1}(0 \cdots 0 \ 1)^\top,$$

which follows from the Cayley-Hamilton Theorem. For its multi-input multi-output extension, see [2].

Due to Theorem 2.1, we obtain

PROPOSITION 2.2. *Suppose that (2.1) is observable and that the eigenvalues of $A - LC$ are negative and simple. Then, we have*

$$\left\| e^{(A-LC)t} \right\| \leq \gamma e^{-\mu t},$$

with $\mu := \min_{\nu \in \sigma(A-LC)} |\nu|$ and $\gamma := \text{cond}(V) = \|V^{-1}\| \|V\|$, where V is the matrix whose rows are the eigenvectors of $A - LC$.

Combining the latter with (2.3), we obtain in particular

$$(2.5) \quad \|\varepsilon(t)\| \leq \gamma \|x(0) - \hat{x}(0)\| e^{-\mu t}.$$

In practice, the term $\|x(0) - \hat{x}(0)\|$ is unknown, whereas μ is chosen by the procedure that designs L , hence known explicitly. Consequently, the previous formula provides in practice only a rate of convergence for the Luenberger observer.

3. Time-parallelization setting. In what follows, we propose to extend the combination between data assimilation algorithms and parallelization procedures to unbounded time intervals, by considering the Luenberger observer. In this case, we will manage to preserve the exponential rate of convergence of the problem, by an approach that we call the *Diamond strategy*.

Let us briefly describe our approach. We proceed by partitioning $[0, +\infty)$ into intervals of the same length that we call *windows*. Following a sequential order, we apply a parallel-in-time solver in each of them, up to some level of accuracy related to a specific accuracy criterion. We then develop an analysis which decomposes the estimation error into two terms, corresponding respectively to the Luenberger observer and the parallelization error. Based on that, we propose a suitable bound on the latter, so that our criterion preserves Luenberger's rate of convergence.

3.1. Framework. In order to accelerate the assimilation and take advantage of a time-parallelization procedure, we propose to divide the time interval $[0, +\infty)$ into windows of a given length $T > 0$ denoted by

$$W_\ell := (T_{\ell-1}, T_\ell), \quad \ell \geq 1,$$

where $T_\ell = \ell \cdot T$ with $\ell \in \mathbb{N}$. Then, we solve (2.2) on each window, in a sequential order, using a time-parallel algorithm. Let us describe how this last method applies.

Given $\ell \geq 1$ and a fixed window W_ℓ , we decompose the latter into N subintervals of length ΔT

$$W_\ell = \bigcup_{n=0}^{N-1} (t_n^\ell, t_{n+1}^\ell),$$

with $t_n^\ell = T_{\ell-1} + n\Delta T$ and $N\Delta T = T$, as shown in Figure 1.

Since time moves forward, parallelizing in this direction requires on each subinterval the introduction of initial conditions $\hat{X}_{\ell,n}^h$. These are assumed to be obtained from the time-parallelization procedure under consideration. In this setting, the parameter h is used in the notation to account for the accuracy of the procedure. In this way, we introduce a parallel version of (2.2) in each subinterval (t_n^ℓ, t_{n+1}^ℓ) , namely

$$(3.1) \quad \begin{cases} \dot{\hat{x}}_{\parallel}(t) = A\hat{x}_{\parallel}(t) + Bu(t) + L[y(t) - C\hat{x}_{\parallel}(t)] \\ \hat{x}_{\parallel}(t_n^{\ell+}) = \hat{X}_{\ell,n}^h, \end{cases}$$

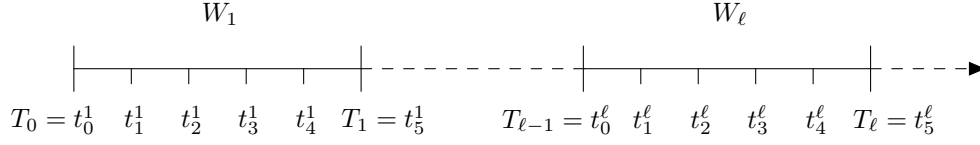


Figure 1: Notation associated with the parallelization setting in the case $N = 5$.

where $\hat{x}_\parallel(t)$ denotes the approximation of $\hat{x}(t)$ obtained by the parallel-in-time solver, see Figure 2. When $n = 0$, we consider as initial conditions $\hat{X}_{0,0}^h = \hat{x}_0$ and $\hat{X}_{\ell,0}^h = \hat{x}_\parallel(T_\ell^-)$ for $\ell > 0$, meaning that \hat{x}_\parallel is continuous at T_ℓ and that windows are treated sequentially, as announced above.

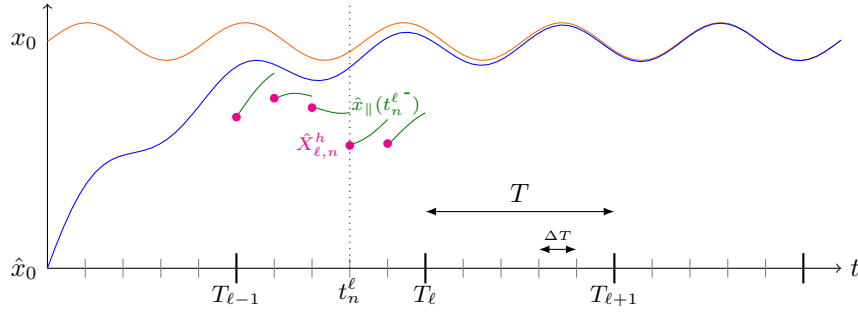


Figure 2: A time-parallelized observer

3.2. The Diamond strategy. Let $\ell \geq 1$. Imposing initial conditions induces discontinuities at the interfaces t_n^ℓ , $n = 1, \dots, N-1$ of the subintervals. These *jumps* are defined by $J_{\ell,n}^h := \hat{X}_{\ell,n}^h - \hat{x}_\parallel(t_n^\ell^-)$. The success of the parallel method relies on their decay to zero as ℓ increases. To analyze this decay we clarify the relation between the solution of (2.1) and the parallelized observer (3.1).

LEMMA 3.1. *Let $\ell > 0$. Under the assumptions of Proposition 2.2, we have*

$$(3.2) \quad \|\varepsilon_\parallel(T_\ell)\| \leq \gamma \left(\|x(0) - \hat{x}(0)\| + \sum_{j=1}^{\ell} e^{\mu_j T} \|J_j^h\| \right) e^{-\mu_\ell T}$$

where $\varepsilon_\parallel(t) := x(t) - \hat{x}_\parallel(t)$ is the error associated with the approximation (3.1) and

$$(3.3) \quad \mathcal{J}_\ell^h := - \sum_{n=1}^{N-1} e^{(A-LC)(N-n)\Delta T} J_{\ell,n}^h.$$

Proof. Let $\ell \geq 1$. We have

$$\begin{aligned} \varepsilon_{\parallel}(t_n^{\ell-}) &= x(t_n^{\ell}) - \hat{x}_{\parallel}(t_n^{\ell-}) = e^{(A-LC)\Delta T} e^{-(A-LC)\Delta T} (x(t_n^{\ell}) - \hat{x}_{\parallel}(t_n^{\ell-})) \\ &= e^{(A-LC)\Delta T} (x(t_{n-1}^{\ell}) - \hat{x}_{\parallel}(t_{n-1}^{\ell+})) \\ &= e^{(A-LC)\Delta T} (x(t_{n-1}^{\ell}) - \hat{X}_{\ell, n-1}^h) \\ &= e^{(A-LC)\Delta T} (x(t_{n-1}^{\ell}) - \hat{x}_{\parallel}(t_{n-1}^{\ell-}) - J_{\ell, n-1}^h) \\ &= e^{(A-LC)\Delta T} (\varepsilon_{\parallel}(t_{n-1}^{\ell-}) - J_{\ell, n-1}^h), \end{aligned}$$

so that

$$\varepsilon_{\parallel}(T_{\ell}) = \varepsilon_{\parallel}(t_N^{\ell}) = e^{(A-LC)T} \varepsilon_{\parallel}(t_0^{\ell}) + \mathcal{J}_{\ell}^h = e^{(A-LC)T} \varepsilon_{\parallel}(T_{\ell-1}) + \mathcal{J}_{\ell}^h,$$

where we have used the continuity of \hat{x}_{\parallel} in $T_{\ell} = t_N^{\ell}$ and $T_{\ell-1} = t_N^{\ell-1}$. In the same way, we obtain

$$\varepsilon_{\parallel}(T_{\ell}) = e^{(A-LC)\ell T} \varepsilon_{\parallel}(0) + \sum_{j=1}^{\ell} e^{(A-LC)(\ell-j)T} \mathcal{J}_j^h.$$

The result is obtained by taking the norm and using Proposition 2.2. \square

Recall that our approach aims at preserving Luenberger's rate of convergence. Thanks to Lemma 3.1, we can now define a criterion to reach this goal.

PROPOSITION 3.2. *Given an arbitrary parameter $\tilde{\gamma} > 0$, let us assume that h satisfies*

$$(3.4) \quad \gamma \sum_{n=1}^{N-1} e^{\mu n \Delta T} \|J_{\ell, n}^h\| \leq \tilde{\gamma} \frac{e^{-\mu(\ell-1)T}}{2^{\ell}},$$

for all $\ell \geq 0$. Then, the rate of convergence of $\hat{x}_{\parallel}(t)$ to $x(t)$ is bounded by μ , i.e.

$$(3.5) \quad \|\varepsilon_{\parallel}(T_{\ell})\| \leq \gamma (\|x(0) - \hat{x}(0)\| + \tilde{\gamma}) e^{-\mu \ell T}.$$

Proof. Using (3.4) and (3.3), we find that

$$\|\mathcal{J}_{\ell}^h\| \leq \sum_{n=1}^{N-1} \gamma e^{-\mu(N-n)\Delta T} \|J_{\ell, n}^h\| \leq \tilde{\gamma} \frac{e^{-\mu \ell T}}{2^{\ell}}.$$

Combining this inequality with (3.2), we deduce that

$$\begin{aligned} \|\varepsilon_{\parallel}(T_{\ell})\| &\leq \left(\|x(0) - \hat{x}(0)\| + \sum_{j=1}^{\ell} e^{\mu j T} \cdot \tilde{\gamma} \frac{e^{-\mu j T}}{2^j} \right) \gamma e^{-\mu \ell T} \\ &= \gamma \left(\|\varepsilon_{\parallel}(0)\| + \tilde{\gamma} \sum_{j=1}^{\ell} \frac{1}{2^j} \right) e^{-\mu \ell T} \\ &\leq \gamma (\|\varepsilon_{\parallel}(0)\| + \tilde{\gamma}) e^{-\mu \ell T}. \end{aligned}$$

The result then follows from $\varepsilon_{\parallel}(0) := x(0) - \hat{x}_{\parallel}(0) = x(0) - \hat{x}(0)$. \square

Note that (3.4) actually deals with an *a posteriori* quantity and, as such, can be used as a criterion to fix the level of accuracy of the time-parallelization procedure used in line on each window. We are now in a position to describe precisely our algorithm: the procedure for estimating $x(t)$ at $t = MT$ for some $M \in \mathbb{N}$ is detailed in Algorithm 3.1.

Algorithm 3.1 Diamond strategy

Input: $A, C, \hat{x}_0, T, N, (\mu_i)_{i=1, \dots, m}, \tilde{\gamma}, L$
Output: $(t_n^\ell)_{n=0, \dots, N}, (\hat{X}_{n, \ell}^h)_{n=0, \dots, N}$
 $L := \text{place}(A, C, (\mu_i)_{i=1, \dots, m})$ {place computes L , as in Theorem 2.1}
 $\mu := \min_{\nu \in \sigma(A-LC)} |\nu|$
 $\Delta t := \frac{T}{N}$
 $\ell := 0$
repeat
 $T_\ell := \ell T$
 $(\forall n \in \{1, \dots, N\}) \quad t_n^\ell := T_\ell + n \Delta T$
if $\ell = 0$ **then**
 $\hat{X}_{\ell, 0}^h := \hat{x}_0$
else
 $\hat{X}_{\ell, 0}^h := \hat{x}_\parallel(T_{\ell-1}^-)$
end if
Determine h such that (3.4) holds
 $\{\hat{X}_{\ell, n}^h\}_{n=1, \dots, N-1} := \text{GTP}(\hat{X}_{\ell, 0}^h)$ {Using a generic time-parallelization procedure (GTP)}
Compute \hat{x}_\parallel on W_ℓ , by (3.1)
Assign $\ell \leftarrow \ell + 1$
until $\ell = M$

4. Time Parallelization. Note that Algorithm 3.1 is defined independently of the chosen the parallel-in-time solver, since the jumps are computed *a posteriori*. However, by specifying it, we can study in more detail the conditions that makes the criterion (3.4) satisfied and the complexity of the overall procedure. Indeed, having in hand an *a priori* estimate of the jumps, one can determine the accuracy h required on each window and bound the efficiency of the *Diamond strategy*. In this way, we consider the Parareal algorithm as the time-parallel method (denoted by GTP in Algorithm 3.1).

4.1. The Parareal algorithm. Introduced by Lions, Maday and Turinici [18], the Parareal algorithm decomposes the solution of an evolution problem by means of a partition of the considered bounded time interval. Assigning each of the corresponding subintervals to a processor, it alternately updates the initial conditions and solves the (smaller) problems on the subintervals in parallel, reducing the computational cost of the overall solution. Let us describe the algorithm more precisely. Given the problem

$$(4.1) \quad \begin{cases} \dot{u}(t) = f(u(t)), & t \in [0, T] \\ u(0) = u_0, \end{cases}$$

decompose $[0, T]$ into a partition of M subintervals (t_{n-1}, t_n) . Consider then two solvers \mathcal{F} and \mathcal{G} , that compute a fine and a coarse numerical approximation of u , respectively. The former is considered to be computationally expensive and consequently restricted to the (parallel) solution of initial-value problems with high accuracy in each subinterval (t_{n-1}, t_n) , whereas the latter is faster and can be used for solving (sequentially) on large intervals as $[0, T]$. For an arbitrary initial condition \tilde{u} given in $t = t_{n-1}$, denote the corresponding local approximations of $u(t_n)$ by

$\mathcal{F}(t_n, t_{n-1}, \tilde{u})$ and $\mathcal{G}(t_n, t_{n-1}, \tilde{u})$, respectively. In this framework, $(u(t_n))_{n=1, \dots, M}$ is approximated by $(U_n^k)_{n=1, \dots, M}$, which is computed according to Algorithm 4.1.

Algorithm 4.1 Parareal algorithm

Input: u_0, T, M, Tol
Output: $(t_n)_{n=1, \dots, M}, (U_n^{k^*})_{n=1, \dots, M}$
 $\Delta T := \frac{T}{N}, t_0 := 0$
 $U_0^0 := u_0$ {Initialization of the initial conditions}
for $1 \leq n \leq M$ **do**
 $t_n := n\Delta T$
 $U_n^0 := \mathcal{G}(t_n, t_{n-1}, U_{n-1}^0)$
end for
 $k := 0$
repeat
 $U_0^k := u_0$
 for $1 \leq n \leq M$ **do**
 $U_n^{k+1} := \mathcal{F}(t_n, t_{n-1}, U_{n-1}^k) + \mathcal{G}(t_n, t_{n-1}, U_{n-1}^{k+1}) - \mathcal{G}(t_n, t_{n-1}, U_{n-1}^k)$
 $J_n^k := U_n^k - u(t_n^-)$
 $k \leftarrow k + 1$
 end for
until $\max_{1 \leq n \leq M} \|J_n^k\| \leq \text{Tol}$
 $k^* := k - 1$

Notice that the superscript k in Algorithm 4.1 plays the role of the parameter h , introduced in the previous section.

Gander and Vandewalle show in [14] that the parareal algorithm reads as a multi-shooting algorithm, in the sense that the method is an approximate Newton method. Indeed, solving the multiple shooting equations with the Newton's method yields

$$U_n^{k+1} = u_{n-1}(t_n, U_{n-1}^k) + \frac{\partial u_{n-1}}{\partial U_{n-1}}(t_n, U_{n-1}^k)(U_{n-1}^{k+1} - U_{n-1}^k),$$

where $u_{n-1}(t_n, U_{n-1}^k)$ denotes the exact solution of (4.1) at t_n , with initial condition U_{n-1}^k at t_{n-1} . Approximating the exact solution $u_{n-1}(t_n, U_{n-1}^k)$ using the fine solver and the Jacobian term $\frac{\partial u_{n-1}}{\partial U_{n-1}}(t_n, U_{n-1}^k)(U_{n-1}^{k+1} - U_{n-1}^k)$ by a difference on a coarse grid, gives

$$(4.2) \quad U_n^{k+1} = \mathcal{F}(t_n, t_{n-1}, U_{n-1}^k) + \mathcal{G}(t_n, t_{n-1}, U_{n-1}^{k+1}) - \mathcal{G}(t_n, t_{n-1}, U_{n-1}^k).$$

It follows that the convergence is super linear and that the number of iterations required to satisfy the criterion (3.4) will not necessarily increase linearly with respect to M . In addition, these authors obtain an estimate which shows that after k iterations, the algorithm gives rise to the fine solution on the first k subintervals. An improvement of their estimate, due to Gander and Hairer [11], assumes that the coarse solver must satisfy a Lipschitz condition

$$\|\mathcal{G}(t_n, t_{n-1}, y) - \mathcal{G}(t_n, t_{n-1}, z)\| \leq (1 + c\Delta T) \|y - z\|,$$

for a positive constant c . However, this result does not capture the enhanced convergence rate when the differential equation itself exhibits decaying behaviour, i.e.,

when $c < 0$. Since we are interested in coupling this algorithm with the Luenberger observer and taking advantage of its decaying behavior, we provide a result adapted to this case, which follows from [13].

THEOREM 4.1 (Convergence of the Parareal algorithm for decaying problems).

Given an initial condition z at time t_{n-1} , let $\mathcal{F}(t_n, t_{n-1}, z)$ and $\mathcal{G}(t_n, t_{n-1}, z)$ be the exact solution at time t_n and its approximation by a coarse integrator respectively. Assume that the local truncation error $\tau(t_n, z) := \mathcal{F}(t_n, t_{n-1}, z) - \mathcal{G}(t_n, t_{n-1}, z)$ satisfies for all y and z

$$(4.3) \quad \|\tau(t_n, z)\| \leq \alpha,$$

$$(4.4) \quad \|\tau(t_n, y) - \tau(t_n, z)\| \leq \beta \|y - z\|,$$

where $\alpha, \beta > 0$ are constants, and that \mathcal{F} and \mathcal{G} are Lipschitz with respect to the initial conditions:

$$(4.5) \quad \max \{ \|\mathcal{F}(t_n, t_{n-1}, y) - \mathcal{F}(t_n, t_{n-1}, z)\|, \|\mathcal{G}(t_n, t_{n-1}, y) - \mathcal{G}(t_n, t_{n-1}, z)\| \} \leq \eta \|y - z\|,$$

for a constant $\eta \in (0, 1)$. The error $\|U_n^k - u(t_n)\|$ at iteration k is bounded by B_n^k , defined by

$$(4.6) \quad B_n^k := \begin{cases} 0 & n \leq k \\ \alpha \beta^k \sum_{i=0}^{n-k-1} \binom{k+i}{k} \varepsilon^i & n > k. \end{cases}$$

REMARK 1. Suppose that (4.3) and (4.5) are satisfied for some $\alpha > 0$ and $\eta \in (0, 1)$, then (4.4) holds with $\beta := 2\eta$, so that (4.4) seems unnecessary. However, using $\beta := 2\eta$ does not necessarily give useful bounds when $0.5 < \eta < 1$.

Proof. Using Definition (4.2) and the fact that \mathcal{F} corresponds to the exact solution on (t_{n-1}, t_n) , we obtain

$$\begin{aligned} U_n^k - u(t_n) &= \mathcal{F}(t_n, t_{n-1}, U_{n-1}^{k-1}) + \mathcal{G}(t_n, t_{n-1}, U_{n-1}^k) - \mathcal{G}(t_n, t_{n-1}, U_{n-1}^{k-1}) \\ &\quad - \mathcal{F}(t_n, t_{n-1}, u(t_{n-1})) \\ &= \tau(t_n, U_{n-1}^{k-1}) - \tau(t_n, u(t_{n-1})) + \mathcal{G}(t_n, t_{n-1}, U_{n-1}^k) - \mathcal{G}(t_n, t_{n-1}, u(t_{n-1})). \end{aligned}$$

Taking norms and combining the resulting inequality with (4.4) and (4.5) gives

$$\|U_n^k - u(t_n)\| \leq \beta \|U_{n-1}^{k-1} - u(t_{n-1})\| + \varepsilon \|U_{n-1}^k - u(t_{n-1})\|.$$

The error in the initial condition can be estimated similarly. We have

$$\begin{aligned} U_n^0 - u(t_n) &= \mathcal{G}(t_n, t_{n-1}, U_{n-1}^0) - \mathcal{F}(t_n, t_{n-1}, u(t_{n-1})) \\ &\quad + \mathcal{G}(t_n, t_{n-1}, u(t_{n-1})) - \mathcal{G}(t_n, t_{n-1}, u(t_{n-1})) + \mathcal{F}(t_n, t_{n-1}, u(t_{n-1})) \\ &\leq \alpha + \eta \|U_{n-1}^0 - u(t_{n-1})\|, \end{aligned}$$

which gives, by means of (4.3) and (4.5)

$$\|U_n^0 - u(t_n)\| \leq \alpha + \eta \|U_{n-1}^0 - u(t_{n-1})\|.$$

Therefore, an upper bound B_n^k for $\|U_n^k - u(t_n)\|$ satisfies the recurrence relation

$$(4.7) \quad B_n^k = \beta B_{n-1}^{k-1} + \eta B_{n-1}^k,$$

$$(4.8) \quad B_n^0 = \alpha + \eta B_{n-1}^0,$$

with $B_0^k = 0$ for all k . This recurrence can be solved by means of generating functions, namely, by defining the formal power series

$$\rho_k(\zeta) = \sum_{n \geq 1} B_n^k \zeta^n.$$

Multiplying (4.7) and (4.8) by ζ^n and summing over $n \geq 1$ gives

$$\begin{aligned} \rho_k(\zeta) &= \beta \zeta \rho_{k-1}(\zeta) + \eta \zeta \rho_k(\zeta), \\ \rho_0(\zeta) &= \frac{\alpha \zeta}{1 - \zeta} + \eta \zeta \rho_0(\zeta), \end{aligned}$$

which can be solved by induction and yields the explicit formula

$$\rho_k(\zeta) = \frac{\alpha \beta^k \zeta^{k+1}}{(1 - \zeta)(1 - \eta \zeta)^{k+1}}.$$

Expanding $\rho_k(\zeta)$ in a power series leads to

$$\begin{aligned} \rho_k(\zeta) &= \alpha \beta^k \zeta^{k+1} \left(\sum_{i \geq 0} \zeta^i \right) \left(\sum_{j \geq 0} \binom{k+j}{k} (\eta \zeta)^j \right) = \alpha \beta^k \zeta^{k+1} \sum_{n \geq 0} \left(\sum_{i=0}^n \binom{k+i}{k} \eta^i \right) \zeta^n \\ &= \sum_{n \geq 0} \left(\alpha \beta^k \sum_{i=0}^n \binom{k+i}{k} \eta^i \right) \zeta^{n+k+1}. \end{aligned}$$

Then, for $n \leq k$ we have $B_0^k = \dots = B_k^k = 0$; whereas for $n > k$, we obtain

$$B_n^k = \alpha \beta^k \sum_{i=0}^{n-k-1} \binom{k+i}{k} \eta^i,$$

and the result follows. \square

We can derive from the previous result an estimate on the jumps.

COROLLARY 4.2. *After k iterations of Algorithm 4.1, the jump $\tilde{J}_n^k := U_n^k - u(t_n^-)$ satisfies $\|\tilde{J}_n^k\| \leq 2B_n^k$.*

Proof. Since \mathcal{F} is an exact solver, we have

$$\begin{aligned} \|\tilde{J}_n^k\| &= \|U_n^k - \mathcal{F}(t_n, t_{n-1}, U_{n-1}^k)\| \\ &\leq \|U_n^k - u(t_n)\| + \|u(t_n) - \mathcal{F}(t_n, t_{n-1}, U_{n-1}^k)\| \\ &\leq B_n^k + \|\mathcal{F}(t_n, t_{n-1}, u(t_{n-1})) - \mathcal{F}(t_n, t_{n-1}, U_{n-1}^k)\| \\ &\leq B_n^k + \eta \|u(t_{n-1}) - U_{n-1}^k\| = B_n^k + \eta B_{n-1}^k \leq 2B_n^k, \end{aligned}$$

since $\eta \in (0, 1)$ and $B_{n-1}^k \leq B_n^k$. The result follows from (4.7). \square

4.2. Combination with Luenberger observer. We now use the Parareal scheme to define the initial conditions $(\hat{X}_{n,\ell}^k)_{n=0,\dots,N}^{\ell \in \mathbb{N}}$ of the time-parallelized Luenberger observer (3.1), meaning that

$$(4.9) \quad \begin{cases} \hat{X}_{\ell,n}^k = \mathcal{F}(t_n^\ell, t_{n-1}^\ell, \hat{X}_{\ell,n-1}^{k-1}) + \mathcal{G}(t_n^\ell, t_{n-1}^\ell, \hat{X}_{\ell,n-1}^k) - \mathcal{G}(t_n^\ell, t_{n-1}^\ell, \hat{X}_{\ell,n-1}^{k-1}) \\ \hat{X}_{\ell,n}^0 = \mathcal{G}(t_n^\ell, t_{n-1}^\ell, \hat{X}_{\ell,n-1}^0), \quad \hat{X}_{\ell,0}^0 = \hat{x}_\parallel(T_\ell^-). \end{cases}$$

Recall that in this setting, jumps are obtained during the execution of Algorithm 4.1 and are consequently known *a posteriori*. In order to estimate the efficiency of Algorithm 3.1, we now propose on the contrary to derive *a priori* upper bounds of the number of iterations observed in practice and complexity. Given a tolerance parameter Tol and let define the corresponding efficiency of Algorithm 3.1 by

$$(4.10) \quad E := \frac{\tau_s}{N\tau_p}$$

where τ_s and τ_p are the CPU time required to achieve $\|\varepsilon(t)\| \leq \text{Tol}$ using a sequential solver and $\|\varepsilon_{\parallel}(t)\| \leq \text{Tol}$ using a parallel solver, respectively. Recall that N represents the number of available processors (and hence, subintervals).

THEOREM 4.3. *We neglect the time of interprocessor communication in the computational time. Let τ_G and τ_F be the computational times associated with one coarse and one fine solution of (2.2) on a interval of length T . The efficiency of the algorithm satisfies*

$$(4.11) \quad E \geq \frac{\ell_{\parallel}^{\text{Tol}} \tau_F}{\tau_F + N\tau_G} \left(\sum_{\ell=1}^{\ell_{\parallel}^{\text{Tol}}} k_{\ell} \right)^{-1},$$

where

$$k_{\ell} := \min_k \left\{ k : 2\gamma \sum_{n=k+1}^{N-1} e^{-\mu(N-n)\Delta T} B_n^k \leq \tilde{\gamma} \frac{e^{-\mu\ell T}}{2^{\ell}} \right\},$$

$$\ell^{\text{Tol}} := \left\lceil \frac{1}{\mu T} \log \left(\gamma \frac{\|x(0) - \hat{x}(0)\|}{\text{Tol}} \right) \right\rceil,$$

$$\ell_{\parallel}^{\text{Tol}} := \left\lceil \frac{1}{\mu T} \log \left(\gamma \frac{\|x(0) - \hat{x}(0)\| + \tilde{\gamma}}{\text{Tol}} \right) \right\rceil,$$

where $\lceil \cdot \rceil$ denotes the ceiling function.

The integers ℓ^{Tol} and $\ell_{\parallel}^{\text{Tol}}$ are upper bounds for the number of windows required to obtain $\|\varepsilon_{\parallel}(T_{\ell})\| \leq \text{Tol}$ and $\|\varepsilon_{\parallel}(T_{\ell})\| \leq \text{Tol}$, respectively. Let us denote by k_{ℓ}^{obs} the actual number of iterations performed in Algorithm 3.1 in the window W_{ℓ} . We will see that k_{ℓ}^{obs} is bounded by k_{ℓ} .

Proof. Using Corollary 4.2, we find that at if k_{ℓ} iterations of (4.9) are done by Algorithm 3.1 in the window W_{ℓ} , the left-hand side of (3.4) satisfies

$$\begin{aligned} \gamma \sum_{n=1}^{N-1} e^{-\mu(N-n)\Delta T} \|J_{\ell,n}^{k_{\ell}}\| &\leq 2\gamma \sum_{n=1}^{N-1} e^{-\mu(N-n)\Delta T} B_n^{k_{\ell}} \\ &= 2\gamma \sum_{n=k_{\ell}+1}^{N-1} e^{-\mu(N-n)\Delta T} B_n^{k_{\ell}} \leq \tilde{\gamma} \frac{e^{-\mu\ell T}}{2^{\ell}}, \end{aligned}$$

meaning that (3.4) is satisfied. As a consequence, we have

$$(4.12) \quad k_{\ell}^{\text{obs}} \leq k_{\ell}.$$

Because of (2.5) and (3.5), the number of windows required to get $\|\varepsilon(t)\| \leq \text{Tol}$ and $\|\varepsilon_{\parallel}(t)\| \leq \text{Tol}$ are bounded by ℓ^{Tol} and $\ell_{\parallel}^{\text{Tol}}$, respectively. In view of (4.9) and since we

neglect the time of interprocessor communication, the part of the total computational time spend required by Algorithm 3.1 to deal with the window W_ℓ is given by $\tau_{p,\ell} = k_\ell^{obs}(\frac{\tau_{\mathcal{F}}}{N} + \tau_{\mathcal{G}})$. On the other hand, the fine solver needs $\tau_{s,\ell} = \tau_{\mathcal{F}}$ to complete one solve of (2.2) on W_ℓ on the fine grid. Because of (4.12), the efficiency satisfies

$$E \geq \frac{\ell^{\text{Tot}} \tau_{\mathcal{F}}}{\tau_{\mathcal{F}} + N \tau_{\mathcal{G}}} \left(\sum_{\ell=1}^{\ell_{\parallel}^{\text{Tot}}} k_\ell^{obs} \right)^{-1} \geq \frac{\ell^{\text{Tot}} \tau_{\mathcal{F}}}{\tau_{\mathcal{F}} + N \tau_{\mathcal{G}}} \left(\sum_{\ell=1}^{\ell_{\parallel}^{\text{Tot}}} k_\ell \right)^{-1},$$

which is the desired estimate. \square

4.3. Variable window approach. Using the results of the previous section, we can propose a variant of Algorithm 3.1 devoted to the case of a large number of processors. Instead of always using the same window length T , we now choose the window length T_ℓ as a function of a prescribed number of iterations k_ℓ , in a way that ensures that the error estimate in Corollary 4.2 falls below the given tolerance after k_ℓ iterations. Since the parareal error must decrease at the same rate as the assimilation error as t increases, the number of iterations k_ℓ must increase with ℓ ; therefore, we will fix the number of parareal iterations applied on each window to $k'_\ell := \ell$, and determine the window length T_ℓ *a priori* using Corollary 4.2. For the sake of clarity, we denote by $(W'_\ell)_{\ell \in \mathbb{N}}$ the corresponding set of windows. Suppose that the window W'_ℓ is composed of N_ℓ subintervals of lengths ΔT , i.e., $W'_\ell = \bigcup_{n=0}^{N_\ell-1} (t_n^\ell, t_{n+1}^\ell)$ and define, for $\ell \in \mathbb{N}$, $T'_\ell := t_0^{\ell+1} = t_{N_\ell}^\ell$, meaning that $T'_\ell = \sum_{j=1}^\ell N_j \Delta T$ if $\ell > 0$ and $T'_0 = 0$. Since the number of iterations is now fixed for each window, we simply denote by $J_{\ell,n}$ (instead of $J_{\ell,n}^\ell$) the jumps observed at t_n^ℓ . Lemma 3.1 then translates as follows.

LEMMA 4.4. *Let $\ell > 0$. Under the assumptions of Proposition 2.2 and still denoting by $\varepsilon_{\parallel}(t) := x(t) - \hat{x}_{\parallel}(t)$ the error associated with the approximation (3.1), we have*

$$(4.13) \quad \|\varepsilon_{\parallel}(T'_\ell)\| \leq \gamma \left(\|x(0) - \hat{x}(0)\| + \sum_{j=1}^{\ell} e^{\mu T'_j} \|\mathcal{J}'_j\| \right) e^{-\mu T'_\ell}$$

where

$$(4.14) \quad \mathcal{J}'_\ell := - \sum_{n=1}^{N_\ell-1} e^{(A-LC)(N_\ell-n)\Delta T} J_{\ell,n}.$$

Proof. As in the proof of Lemma 3.1, we have

$$\varepsilon_{\parallel}(t_n^{\ell-}) = e^{(A-LC)\Delta T} (\varepsilon_{\parallel}(t_{n-1}^{\ell-}) - J_{\ell,n-1}),$$

hence

$$\varepsilon_{\parallel}(T'_\ell) = \varepsilon_{\parallel}(t_{N_\ell}^\ell) = e^{(A-LC)(T'_\ell - T'_{\ell-1})} \varepsilon_{\parallel}(t_0^\ell) + \mathcal{J}'_\ell = e^{(A-LC)(T'_\ell - T'_{\ell-1})} \varepsilon_{\parallel}(T'_{\ell-1}) + \mathcal{J}'_\ell,$$

where we have used the continuity of \hat{x}_{\parallel} in $T'_\ell = t_{N_\ell}^\ell$ and $T'_{\ell-1} = t_{N_\ell}^{\ell-1}$. In the same way, we obtain

$$\varepsilon_{\parallel}(T_\ell) = e^{(A-LC)T'_\ell} \varepsilon_{\parallel}(0) + \sum_{j=1}^{\ell} e^{(A-LC)(T'_\ell - T'_j)} \mathcal{J}'_j.$$

The result is obtained by taking the norm and using Proposition 2.2. \square

The rate of convergence can now be preserved *a priori*.

PROPOSITION 4.5. *Given $\tilde{\gamma} > 0$ an arbitrary parameter, define N_ℓ recursively by*

$$(4.15) \quad N_\ell := \max \left\{ N : 2\gamma \sum_{n=\ell}^{N-1} e^{\mu n \Delta T} B_n^\ell \leq \tilde{\gamma} \frac{e^{-\mu T'_{\ell-1}}}{2^\ell} \right\}.$$

Then, the rate of convergence of $\hat{x}_\parallel(t)$ to $x(t)$ is bounded by μ , i.e.

$$(4.16) \quad \|\varepsilon_\parallel(T'_\ell)\| \leq \gamma (\|x(0) - \hat{x}(0)\| + \tilde{\gamma}) e^{-\mu T'_\ell}.$$

Note that $B_n^\ell = 0$ for $n \leq \ell$, so that $\sum_{n=\ell}^{N-1} e^{\mu n \Delta T} B_n^\ell = 0$ if $N_\ell = \ell + 1$. Hence, $N_\ell \geq \ell + 1$.

Proof. Using successively Proposition 2.2, (4.14), Corollary 4.2 and (4.15), we find that:

$$\begin{aligned} \|\mathcal{J}'_\ell\| &\leq \gamma \sum_{n=1}^{N_\ell-1} e^{-\mu(N_\ell-n)\Delta T} \|\mathcal{J}_{\ell,n}\| \\ &\leq 2\gamma \sum_{n=1}^{N_\ell-1} e^{-\mu(N_\ell-n)\Delta T} B_n^\ell = 2\gamma \sum_{n=1}^{N_\ell-1} e^{\mu n \Delta T} B_n^\ell e^{-\mu(T'_\ell - T'_{\ell-1})} \\ &\leq \tilde{\gamma} \frac{e^{-\mu T'_\ell}}{2^\ell}. \end{aligned}$$

Combining this last equation with (4.13), we get:

$$\begin{aligned} \|\varepsilon_\parallel(T'_\ell)\| &\leq \gamma \left(\|x(0) - \hat{x}(0)\| + \sum_{j=1}^{\ell} e^{\mu T'_j} \tilde{\gamma} \frac{e^{-\mu T'_j}}{2^j} \right) e^{-\mu T'_\ell} \\ &\leq \gamma \left(\|x(0) - \hat{x}(0)\| + \tilde{\gamma} \sum_{j=1}^{\ell} \frac{1}{2^j} \right) e^{-\mu T'_\ell} \\ &\leq \gamma (\|x(0) - \hat{x}(0)\| + \tilde{\gamma}) e^{-\mu T'_\ell} \end{aligned}$$

The result then follows from $\varepsilon_\parallel(0) := x(0) - \hat{x}_\parallel(0) = x(0) - \hat{x}(0)$. \square

5. Numerical experiments. The present section is devoted to some numerical experiments for the Luenberger observer. For this purpose, we use

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix}, \quad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad C = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad v(t) = 3 + 0.5 \sin(0.75t).$$

We remark that the initial condition on System (2.1) is unknown, but we perform the experiments with $x(0) = (0, 0)^\top$. We then construct the observer $\hat{x}(t)$ by setting as initial condition $\hat{x}(0) = (2, 1)^\top$ and the eigenvalues of $A - LC$. For the latter, we consider $\{-0.25, -0.5\}$ and $\{-2, -4\}$ as possible choices.

To introduce the parareal procedure, given N available processors, we set

$$T = 1, \quad \delta T = \Delta T = \frac{T}{N}, \quad \text{Tol} = 10^{-8},$$

where δT denotes the time step associated with \mathcal{G} , chosen as a one step solver for the sake of simplicity. We use the Backward Euler method to define both propagators \mathcal{F} and \mathcal{G} .

5.1. Diagonalized system. We recall that the essential part of Theorem 4.1 is the contraction factor η . For the Luenberger observer (2.2), we have

$$\eta = \max \left\{ \left\| [I - \delta t(A - LC)]^{-\Delta T/\delta t} \right\|, \left\| [I - \Delta T(A - LC)]^{-1} \right\| \right\}.$$

where δt is the time step associated with \mathcal{F} , assumed to be constant. Even if we choose the eigenvalues of $A - LC$ to guarantee a decaying rate of convergence, η is not necessarily smaller than one. For this reason, we consider instead a diagonalized observer

$$(5.1) \quad \begin{cases} \dot{\hat{z}}(t) = D\hat{z}(t) + V^{-1}(Bu(t) + Ly(t)) \\ \hat{z}(0) = V^{-1}\hat{x}_0 \end{cases}$$

where $\hat{z} = V^{-1}\hat{x}$ and $D = V^{-1}(A - LC)V$.

Due to the change of variables, $\gamma = 1$. We determine the constants α , β and η by

PROPOSITION 5.1. *Let \mathcal{F} and \mathcal{G} be defined by the Backward Euler scheme, with time steps δt and δT , respectively. We assume that $\Delta TK \leq 1$ and (5.1) satisfies*

$$\begin{aligned} M &:= \sup_{(\hat{z}, t)} \|D\hat{z} + V^{-1}(Bu(t) + Ly(t))\| < \infty \\ K &:= \max \left\{ \|D\|, \sup_{t>0} \|V^{-1}(B\dot{u}(t) + L\dot{y}(t))\| \right\} < \infty. \end{aligned}$$

Then, the constants associated with both propagators in Theorem 4.1 are given by

$$(5.2) \quad \alpha = \Delta T^2 \left(\frac{K(M+1)}{2(1-\Delta TK)} \right),$$

$$(5.2) \quad \beta = \left\| [I - \delta tD]^{-\Delta T/\delta t} - [I - \Delta TD]^{-1} \right\|,$$

$$(5.3) \quad \eta = \max \left\{ \left\| [I - \delta tD]^{-\Delta T/\delta t} \right\|, \left\| [I - \Delta TD]^{-1} \right\| \right\}.$$

The proof is standard, but for the sake of completeness is presented in Appendix A.

5.2. Evolution of k_ℓ . As a first experiment, since the jumps involved in (3.4) allows us to compute the sequence $k^{obs} := \{k_\ell^{obs}\}_\ell$, we propose to compare its behavior with its *a priori* estimate

$$k^{th} := \{k_\ell\}_\ell,$$

where the latter sequence is provided by Theorem 4.3.

We observe in Figure 3 that increasing $\tilde{\gamma}$ leads to enlarge the number of windows in which the algorithm requires only 1 iteration. This is expected, due to the term $\tilde{\gamma}e^{-\mu\ell T}$ present in Proposition 3.2.

5.3. Observed efficiency. Our second experiment consists of comparing the observed efficiencies for both sequences k^{obs} and k^{th} , using different values of $\tilde{\gamma}$, N and δt . To evaluate E^{obs} , the execution time for the parallel and sequential solvers was computed with the functions `tic` and `toc` of MATLAB (version 9.4.0.813654 (R2018a)).

As we notice previously, increasing $\tilde{\gamma}$ improves the algorithm performance, but the difference between $E^{obs}(k^{obs})$ and $E^{obs}(k^{th})$ still remains, as observed in Figure 4a. In Figure 4c, the gap between these values varies slightly, showing that δt small

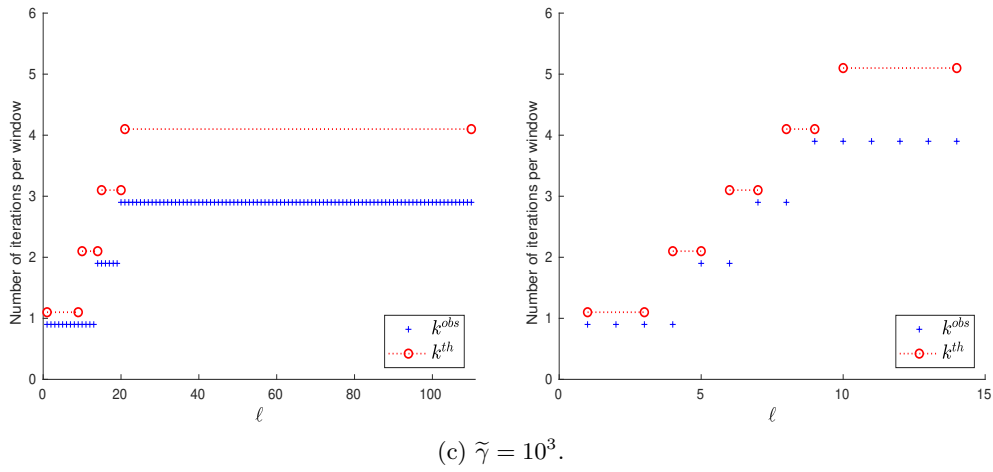
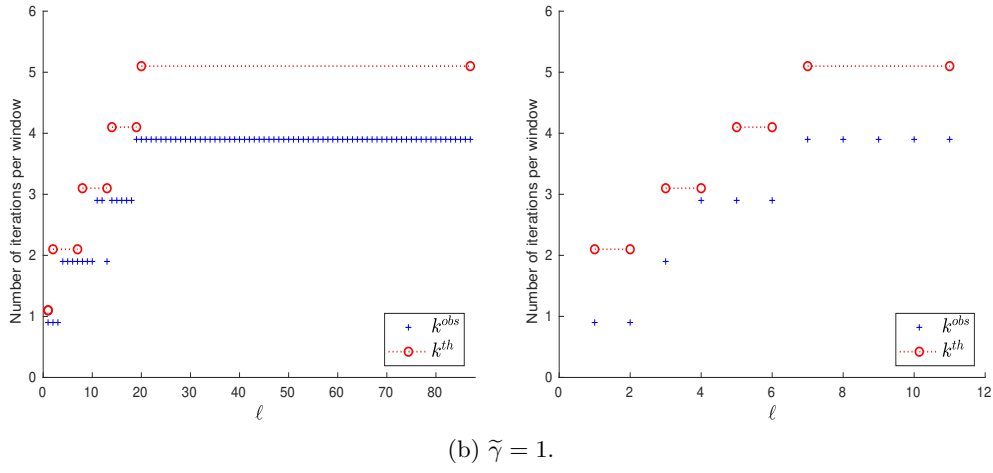
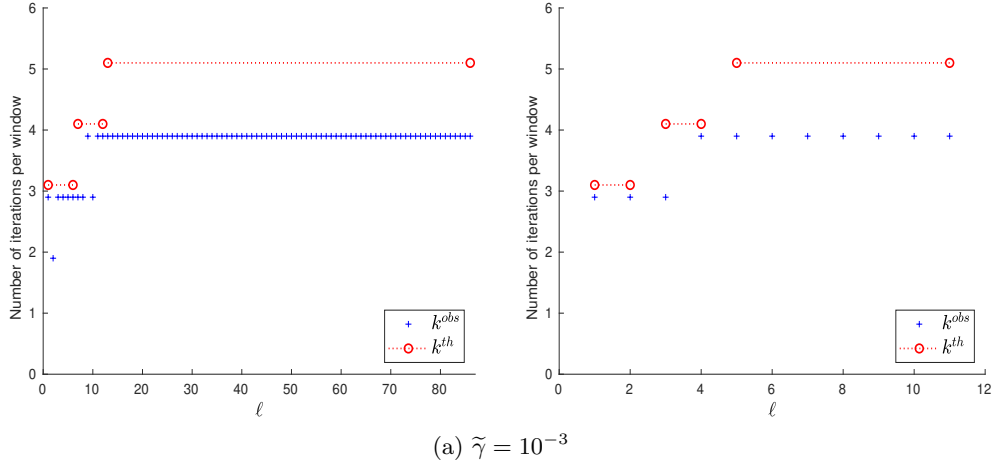


Figure 3: Comparison between k^{th} and k^{obs} , for $N = 16$ and $\delta t = \frac{\Delta T}{25}$. The eigenvalues of $A - LC$ are $\{-0.25, -0.5\}$ (left) and $\{-2, -4\}$ (right).

enough does not affect the efficiency significantly. Increasing the number of processors N makes this difference smaller and also improves the efficiency of the algorithm, as shown in Figure 4b. Another way to narrow this gap is choosing smaller eigenvalues for $A - LC$. As Figure 4 suggests, the comparison between $\{-0.25, -0.5\}$ and $\{-2, -4\}$ shows that $E^{obs}(k^{th})$ increases, whereas $E^{obs}(k^{obs})$ becomes smaller.

Figure 4 also shows that the observed efficiencies satisfy

$$E^{obs}(k^{th}) \leq E^{obs}(k^{obs}),$$

which is simply because the sequence k^{th} underperforms k^{obs} .

Finally, we recall that k^{th} is useful for estimating the efficiency. Assuming that $\tau_{\Delta T}^G$ is negligible, we denote this estimate by

$$E_0^{th} = \ell_{\parallel}^{\text{Tol}} \left(\sum_{\ell=1}^{\ell_{\parallel}^{\text{Tol}}} k_{\ell} \right)^{-1}.$$

with $\ell_{\parallel}^{\text{Tol}}$ given by Theorem 4.3. We note that this value predicts quite well $E^{obs}(k^{th})$ in all the tests.

5.4. Variable window approach. In the following, we set $M = 100$, $\Delta T = 1/16$ and $\{-0.8, -1\}$ as eigenvalues of $A - LC$.

The Diamond strategy and the Variable window approach are different in nature, but we can compare them by considering the number of parareal iterations as a function of time. Denoting by k_{vw} the sequence of parareal iterations associated with the latter, we observe in Figure 5 that starts performing better than k^{th} , but in the long term underperforms the *a priori* estimate.

Increasing $\tilde{\gamma}$ leads to a slightly better performance of the Variable window approach, but the behavior previously described still remains. As a consequence of this, the observed efficiency of this procedure is smaller than $E^{obs}(k^{th})$, as shown in Figure 6a.

In contrast to the previous subsection, when the observed efficiency depends on δt , we observe in Figure 6b a “jump” instead of a linear behaviour, due to a decrease in the total number of windows. This can be explained as a “blindness” to the tolerance: although the parallelized observer can be closer to the real solution at the end of a window, the Variable window approach does not take this into account and constructs the next one with more than enough subintervals. In principle, the Diamond Strategy faces the same problem, but it is solved using small windows.

Acknowledgments. This work was supported by ANR Ciné-Para (ANR-15-CE23-0019) and ANR/RGC ALLOWAP (ANR-19-CE46-0013/A-HKBU203/19), by Swiss National Science Foundation grant 200020 178752, and by Hong Kong Research Grants Council (ECS 22300115 and GRF 12301817).

REFERENCES

- [1] J. Ackermann. Der entwurf linearer regelungssysteme im zustandsraum. *Regelungstechnik*, 20:297–300, 1972.
- [2] J. Ackermann. On the synthesis of linear control systems with specified characteristics. *Automatica*, 13:89–94, 1977.
- [3] C. Afri, V. Andrieu, L. Bako, and P. Dufour. State and parameter estimation: A nonlinear Luenberger observer approach. *IEEE Transactions on Automatic Control*, 62(2):973–980, 2017.

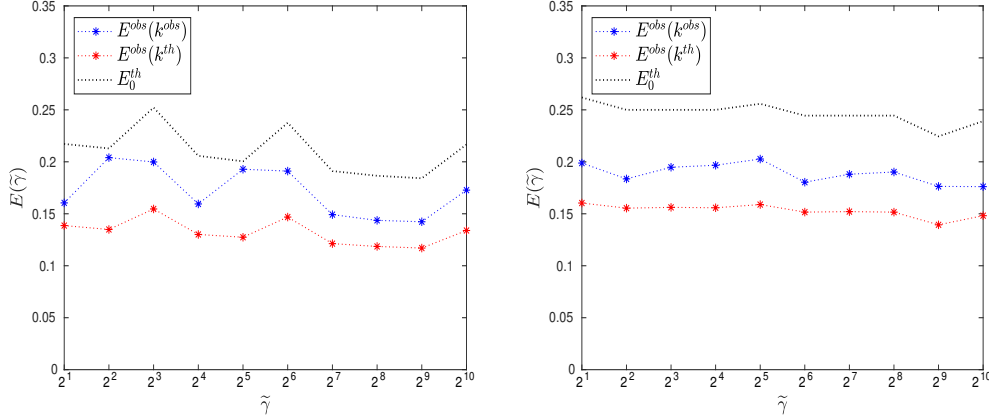
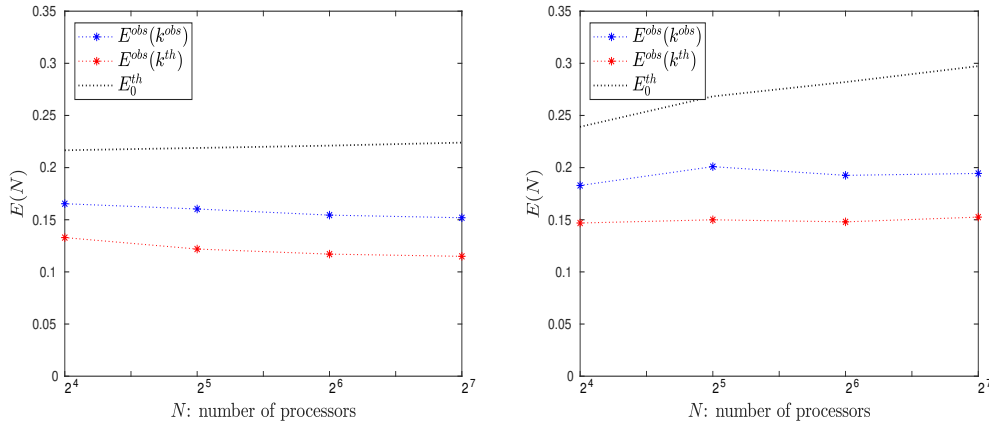
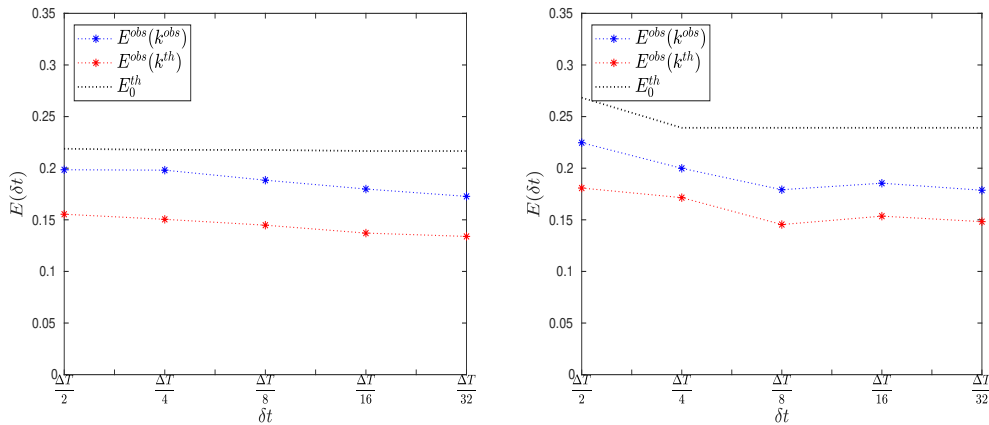

 (a) $E(\tilde{\gamma})$, for $N = 16$ and $\delta t = \frac{\Delta T}{2^5}$.

 (b) $E(N)$, for $\delta t = \frac{\Delta T}{2^5}$ and $\tilde{\gamma} = 2^{10}$.

 (c) $E(\delta t)$, for $N = 16$ and $\tilde{\gamma} = 2^{10}$.

 Figure 4: Comparison between $E^{obs}(k^{obs})$, $E^{obs}(k^{th})$ and E_0^{th} . The eigenvalues of $A - LC$ are $\{-0.25, -0.5\}$ (left) and $\{-2, -4\}$ (right).

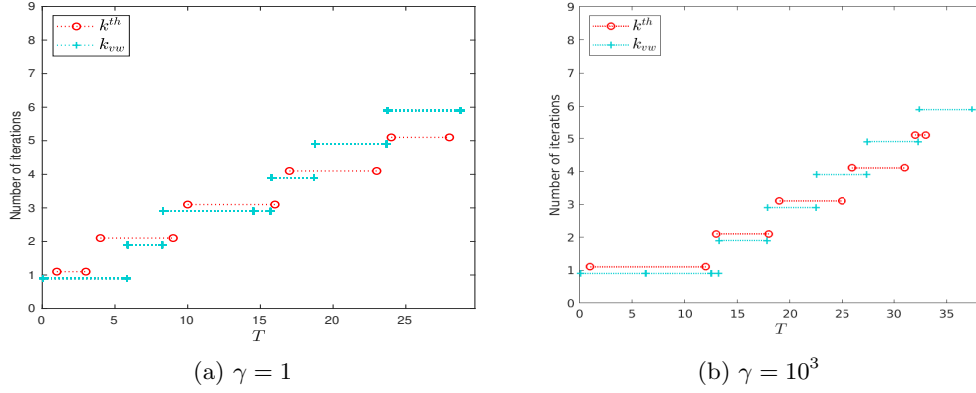
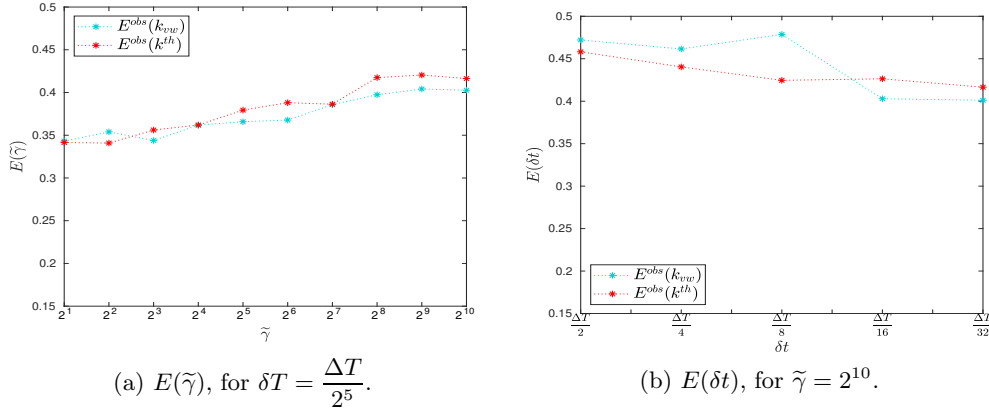


Figure 5: Assimilation for the Diamond strategy and Variable Window approach.

Figure 6: Comparison between the efficiencies $E^{obs}(k^{th})$ (Diamond strategy) and $E^{obs}(k_{vw})$ (Variable Window approach).

- [4] D. Auroux. The back and forth nudging algorithm applied to a shallow water model, comparison and hybridization with the 4D-VAR. *Int. J. Numer. Methods Fluids*, 61(8):911–929, 2009.
- [5] R. Bass and I. Gura. High-order system design via state-space considerations. In *Joint Automatic Control Conference*, volume 3, pages 311–319, New York, 1965.
- [6] P. Courtier, J.-N. Thépaut, and A. Hollingsworth. A strategy for operational implementation of 4d-var, using an incremental approach. *Quarterly Journal of the Royal Meteorological Society*, 120(519):1367–1387, 1994.
- [7] L. D’Amore and R. Cacciapuoti. DD-DA PinT-based model: A domain decomposition approach in space and time, based on parareal, for solving the 4D-Var data assimilation model. *ArXiv e-prints*, 2018.
- [8] V. Dolean, P. Jolivet, and F. Nataf. *An introduction to domain decomposition methods*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015. Algorithms, theory, and parallel implementation.
- [9] M. J. Gander. Schwarz methods over the course of time. *ETNA. Electronic Transactions on Numerical Analysis [electronic only]*, 31:228–255, 2008.
- [10] M. J. Gander. 50 years of time parallel time integration. In T. Carraro, M. Geiger, S. Körkel, and R. Rannacher, editors, *Multiple Shooting and Time Domain Decomposition Methods*, volume 9 of *Contributions in Mathematical and Computational Sciences*, pages 69–113.

- Springer, 2015.
- [11] M. J. Gander and E. Hairer. Nonlinear convergence analysis for the parareal algorithm. In O. B. Widlund and D. E. Keyes, editors, *Domain Decomposition Methods in Science and Engineering XVII*, volume 60 of *Lecture Notes in Computational Science and Engineering*, pages 45–56. Springer, 2008.
 - [12] M. J. Gander, F. Kwok, and J. Salomon. Paraopt: A parareal algorithm for optimality systems. *SIAM Journal on Scientific Computing*, 42(5):A2773–A2802, 2020.
 - [13] M. J. Gander, F. Kwok, and H. Zhang. Multigrid interpretations of the parareal algorithm leading to an overlapping variant and MGRIT. *Computing and Visualization in Science*, 19(3):59–74, 2018.
 - [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] J. HOKE and R. ANTHES. INITIALIZATION OF NUMERICAL-MODELS BY A DYNAMIC-INITIALIZATION TECHNIQUE. *MONTHLY WEATHER REVIEW*, 104(12):1551–1556, 1976.
 - [16] K. Ide, P. Courtier, M. Ghil, and A. C. Lorenc. Unified notation for data assimilation : Operational, sequential and variational. *Journal of the Meteorological Society of Japan. Ser. II*, 75(1B):181–189, 1997.
 - [17] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME—Journal of Basic Engineering*, 82(1):35–45, 1960.
 - [18] J.-L. Lions, Y. Maday, and G. Turinici. Résolution d’EDP par un schéma en temps «pararéel». *Comptes Rendus de l’Académie des Sciences - Série I - Mathématique*, 332(7):661–668, 2001.
 - [19] A. LORENC, R. BELL, and B. MACPHERSON. THE METEOROLOGICAL-OFFICE ANALYSIS CORRECTION DATA ASSIMILATION SCHEME. *QUARTERLY JOURNAL OF THE ROYAL METEOROLOGICAL SOCIETY*, 117(497, A):59–89, JAN 1991.
 - [20] D. Luenberger. *Introduction to Dynamic Systems: Theory, Models, and Applications*. John Wiley & Sons, New York, 1979.
 - [21] W. LYNE, R. SWINBANK, and N. BIRCH. A DATA ASSIMILATION EXPERIMENT AND THE GLOBAL CIRCULATION DURING THE FGGE SPECIAL OBSERVING PERIODS. *QUARTERLY JOURNAL OF THE ROYAL METEOROLOGICAL SOCIETY*, 108(457):575–594, 1982.
 - [22] Y. Maday, J. Salomon, and G. Turinici. Monotonic parareal control for quantum systems. *SIAM Journal on Numerical Analysis*, 45(6):2468–2482, 2007.
 - [23] J. Nievergelt. Parallel methods for integrating ordinary differential equations. *Commun. ACM*, 7(12):731–733, 1964.
 - [24] V. Rao and A. Sandu. A time-parallel approach to strong-constraint four-dimensional variational data assimilation. *Journal of Computational Physics*, 313:583–593, 2016.
 - [25] H. A. Schwarz. Über einen grenzübergang durch alternierendes verfahren. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zurich*, 15:272–286, 1870.
 - [26] Y. Trémolet and F.-X. Le Dimet. Parallel algorithms for variational data assimilation and coupling models. *Parallel Computing*, 22(5):657–674, 1996.
 - [27] F. Veersé, D. Auroux, and M. Fisher. Limited-memory BFGS diagonal preconditioners for a data assimilation problem in meteorology. *Optim. Engineer.*, 1.3:323–339, 2000.
 - [28] J. VERRON and W. HOLLAND. IMPACT OF SATELLITE ALTIMETER-DATA ON DIGITAL-SIMULATION OF MID-LATITUDE GENERAL OCEAN-CIRCULATION. *ANNALES GEOPHYSICAE-ATMOSPHERES HYDROSPHERES AND SPACE SCIENCES*, 7(1):31–46, FEB 1989.

Appendix A. Proof of Proposition 5.1. Let $\{t_n\}_{n=0}^N$ be a regular partition of the interval $[0, T]$, with $\Delta T = T/N$. Given \hat{z}_{n-1} an approximation of $\hat{z}(t_{n-1})$, we recall that the Backward Euler method applied to (5.1) is given by

$$\frac{\hat{z}_n - \hat{z}_{n-1}}{\Delta T} = f(\hat{z}_n, t_n)$$

where $f(s, t) = Ds + V^{-1}(Bv(t) + Ly(t))$.

Since δt is assumed to be constant, we then define \mathcal{F} by

$$\mathcal{F}(t_n, t_{n-1}, \hat{z}_{n-1}) = (I - \delta t D)^{-\Delta T/\delta t} [\delta t V^{-1}(Bv(t_{n-1}) + Lz(t_{n-1})) + \hat{z}_{n-1}]$$

and then, a direct calculation yields

$$(A.1) \quad \mathcal{F}(t_n, t_{n-1}, w) - \mathcal{F}(t_n, t_{n-1}, z) = (I - \delta t D)^{-\Delta T/\delta t} (w - z).$$

On the other hand, \mathcal{G} is defined as a one-step solver, which allows us to replace δt by ΔT in the previous expressions to obtain

$$(A.2) \quad \mathcal{G}(t_n, t_{n-1}, y) - \mathcal{G}(t_n, t_{n-1}, z) = (I - \Delta t D)^{-1} (y - z).$$

Hence, Definitions (5.2) and (5.3) of β and η follow from combining (A.1) and (A.2).

To bound the local truncation error, we proceed as follows. Starting at the exact solution $z_{n-1} = \hat{z}(t_{n-1})$, we define $z_n = \mathcal{G}(t_n, t_{n-1}, z_{n-1})$ and then

$$\begin{aligned} \tau(t_n, z_{n-1}) &= \mathcal{F}(t_n, t_{n-1}, z_{n-1}) - \mathcal{G}(t_n, t_{n-1}, z_{n-1}) \\ &= \hat{z}(t_n) - z_n \end{aligned}$$

since \mathcal{F} is an exact solver. We use that $z_n = \hat{z}(t_{n-1}) + \Delta T f(z_n, t_n)$ and then expand $\hat{z}(t_{n-1})$ around t_n to get

$$(A.3) \quad \tau(t_n, z_{n-1}) = \Delta T \left(\dot{\hat{z}}(\hat{z}(t_n), t_n) - f(z_n, t_n) \right) - \frac{(\Delta T)^2}{2} \ddot{\hat{z}}(\hat{z}(\xi), \xi)$$

where $\xi \in (t_{n-1}, t_n)$. Since $\dot{\hat{z}} = f(z, t)$, we can get rid of the derivatives of z . In particular, the definition of $f(s, t)$ shows that

$$\begin{aligned} \dot{\hat{z}}(\hat{z}(t_n), t_n) - f(z_n, t_n) &= f(\hat{z}(t_n), t_n) - f(z_n, t_n) = D\tau(t_n, z_{n-1}), \\ \ddot{\hat{z}}(\hat{z}(\xi), \xi) &= \frac{df}{dt}(\hat{z}(\xi), \xi) = \frac{\partial f}{\partial s}(\hat{z}(\xi), \xi) \cdot f(\hat{z}(\xi), \xi) + \frac{\partial f}{\partial t}(\hat{z}(\xi), \xi) \\ &= Df(\hat{z}(\xi), \xi) + V^{-1}(B\dot{v}(\xi) + L\dot{y}(\xi)). \end{aligned}$$

Replacing these expressions in (A.3) and rearranging terms yields

$$\tau(t_n, z_{n-1}) = -\frac{(\Delta T)^2}{2} (I - \Delta T D)^{-1} [Df(\hat{z}(\xi), \xi) + V^{-1}(B\dot{v}(\xi) + L\dot{y}(\xi))].$$

Finally, assuming that $\Delta T K < 1$, we take norms and use the definitions of K and M to obtain α .