PARAOPT: A PARAREAL ALGORITHM FOR OPTIMALITY SYSTEMS[*]

MARTIN J. GANDER[†], FELIX KWOK[‡], AND JULIEN SALOMON[§]

Abstract. The time parallel solution of optimality systems arising in PDE constrained optimization could be achieved by simply applying any time parallel algorithm, such as Parareal, to solve the forward and backward evolution problems arising in the optimization loop. We propose here a different strategy by devising directly a new time parallel algorithm, which we call ParaOpt, for the coupled forward and backward non-linear partial differential equations. ParaOpt is inspired by the Parareal algorithm for evolution equations, and thus is automatically a two-level method. We provide a detailed convergence analysis for the case of linear parabolic PDE constraints. We illustrate the performance of ParaOpt with numerical experiments both for linear and nonlinear optimality systems.

Key words. Parareal algorithm, Optimal control, Preconditionning

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

1. Introduction.

Time parallel time integration has become an active research area over the last decade; there is even an annual workshop now dedicated to this topic called the PinT (Parallel in Time) workshop, which started with the first such dedicated workshop at the USI in Lugano in June 2011. The main reason for this interest is the advent of massively parallel computers [5] with so many computing cores that spatial parallelization of an evolution problem saturates long before all cores have been effectively used. There are four classes of such algorithms: methods based on multiple shooting leading to the parareal algorithm [44, 2, 30, 33, 20, 11, 18, 40], methods based on waveform relaxation [32, 8, 19, 21, 13, 14, 31, 39, 1, 15], methods based on multigrid [27, 34, 53, 28, 6, 17, 7, 41, 4], and direct time parallel methods [42, 49, 50, 35, 10]; for a review of the development of PinT methods, see [9],[45] and the references therein.

A natural area where this type of parallelization could be used effectively is in PDE constrained optimization on bounded time intervals, when the constraint is a time dependent PDE. In these problems, calculating the descent direction within the optimization loop requires solving both a forward and a backward evolution problem, so one could directly apply time parallelization techniques to each of these solves [23, 24, 25, 26]. Parareal can also be useful in one-shot methods where the preconditioning operator requires the solution of initial value problems, see e.g. [52]. Another method, which has been proposed in [36, 47] in the context of quantum control, consists of decomposing the time interval into sub-intervals and defining intermediate states at sub-interval boundaries; this allows one to construct a set of independent optimization problems associated with each sub-interval in time. Each iteration of the method then requires the solution of these independent sub-problems in parallel, followed by a cheap update of the intermediate states. In this paper, we propose yet another approach: based on a fundamental understanding of the parareal algorithm invented in [33] as a

[*]Submitted to the editors 07.10.2019.

[†]Section of Mathematics, University of Geneva, 1211 Geneva 4, Switzerland (martin.gander@unige.ch).

[‡]Department of Mathematics, Hong Kong Baptist University, Hong-Kong (felix_kwok@hkbu.edu.hk).

[§]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).

specific approximation of a multiple shooting method [20], we construct a new time-parallel method called ParaOpt for solving directly the coupled forward and backward evolution problems arising in the optimal control context. Our approach is related to the multiple shooting paradigm [43], where the time horizon is decomposed into non-overlapping sub-intervals, and we solve for the unknown interface state and adjoint variables using an inexact Newton method so that the trajectories are continuous across sub-intervals. Additionally, a parareal-like approximation is used to obtain a cheap approximate Jacobian for the Newton solve. There are two potential benefits to our approach: firstly, it is known that for some control problems, long time horizons lead to difficulties in convergence for the optimization loop. Therefore, a multiple shooting approach allows us to deal with local subproblems on shorter time horizons, where we obtain faster convergence. Such convergence enhancement has also been observed in [3, 37, 38], and also more recently in [47]. Secondly, if we use parareal to parallelize the forward and backward sweeps, then the speedup ratio will be bounded above by L/K, where L is the number of sub-intervals and K is the number of parareal iterations required for convergence. For many problems, especially the non-diffusive ones like the Lotka-Volterra problem we consider in Section 4.2, this ratio does not go above 4–5; this limits the potential speedup that can be obtained from this classical approach. By decomposing the control problem directly and conserving the globally coupled structure of the problem, we obtain higher speedup ratios, closer to ones that are achievable for two-level methods for elliptic problems.

Our paper is organized as follows: in Section 2, we present our PDE constrained optimization model problem, and ParaOpt for its solution. In Section 3 we give a complete convergence analysis of ParaOpt for the case when the PDE constraint is linear and of parabolic type. We then illustrate the performance of ParaOpt by numerical experiments in Section 4, both for linear and non-linear problems. We present our conclusions and an outlook on future work in Section 5.

2. ParaOpt: a two-grid method for optimal control.

Consider the optimal control problem associated with the cost functional

$$J(c) = \frac{1}{2}\|y(T) - y_{target}\|^2 + \frac{\alpha}{2}\int_0^T \|c(t)\|^2 dt,$$

where $\alpha > 0$ is a fixed regularization parameter, y_{target} is a target state, and the evolution of the state function y: $[0,T] \to \mathbb{R}^n$ is described by the non-linear equation

$$\dot{y}(t) = f(y(t)) + c(t), \tag{2.1}$$

with initial condition $y(0) = y_{init}$, where $c(t)$ is the control, which is assumed to enter linearly in the forcing term. The first-order optimality condition then reads

$$\dot{y} = f(y) - \frac{\lambda}{\alpha}, \quad \dot{\lambda} = -(f'(y))^T\lambda, \tag{2.2}$$

with the final condition $\lambda(T) = y(T) - y_{target}$, see [16] for a detailed derivation.

We now introduce a parallelization algorithm for solving the coupled problem (2.1–2.2). The approach we propose follows the ideas of the parareal algorithm, combining a sequential coarse integration on $[0,T]$ and parallel fine integration on subintervals.

Consider a subdivision of $[0,T] = \cup_{\ell=0}^{L-1}[T_\ell, T_{\ell+1}]$ and two sets of intermediate states $(Y_\ell)_{\ell=0,\cdots,L}$ and $(\Lambda_\ell)_{\ell=1,\cdots,L}$ corresponding to approximations of the state y

and the adjoint state λ at times $T_0, \cdots, T_L$ and $T_1, \cdots, T_L$ respectively. We denote by P and Q the nonlinear solution operators for the boundary value problem (2.2) on the subinterval $[T_\ell, T_{\ell+1}]$ with initial condition $y(T_l) = Y_l$ and final condition $\lambda(T_{\ell+1}) = \Lambda_{\ell+1}$, defined so that P propagates the state y forward to $T_{\ell+1}$ and Q propagates the adjoint backward to T_ℓ:

$$\begin{pmatrix} y(T_{\ell+1}) \\ \lambda(T_\ell) \end{pmatrix} = \begin{pmatrix} P(Y_\ell, \Lambda_{\ell+1}) \\ Q(Y_\ell, \Lambda_{\ell+1}) \end{pmatrix}. \tag{2.3}$$

Using these solution operators, we can write the boundary value problem as a system of subproblems, which have to satisfy the matching conditions

$$\begin{array}{rcl@{\qquad}rcl} Y_0 - y_{init} & = & 0, & & & \\ Y_1 - P(Y_0, \Lambda_1) & = & 0, & \Lambda_1 - Q(Y_1, \Lambda_2) & = & 0, \\ Y_2 - P(Y_1, \Lambda_2) & = & 0, & \Lambda_2 - Q(Y_2, \Lambda_3) & = & 0, \\ & \vdots & & & \vdots & \\ Y_L - P(Y_{L-1}, \Lambda_L) & = & 0, & \Lambda_L - Y_L + y_{target} & = & 0. \end{array} \tag{2.4}$$

This nonlinear system of equations can be solved using Newton's method. Collecting the unknowns in the vector $(Y^T, \Lambda^T) := (Y_0^T, Y_1^T, \ldots, Y_L^T, \Lambda_1^T, \Lambda_2^T, \ldots, \Lambda_L^T)$, we obtain the nonlinear system

$$\mathcal{F}\begin{pmatrix} Y \\ \Lambda \end{pmatrix} := \begin{pmatrix} Y_0 - y_{init} \\ Y_1 - P(Y_0, \Lambda_1) \\ Y_2 - P(Y_1, \Lambda_2) \\ \vdots \\ Y_L - P(Y_{L-1}, \Lambda_L) \\ \Lambda_1 - Q(Y_1, \Lambda_2) \\ \Lambda_2 - Q(Y_2, \Lambda_3) \\ \vdots \\ \Lambda_L - Y_L + y_{target} \end{pmatrix} = 0.$$

Using Newton's method to solve this system gives the iteration

$$\mathcal{F}'\begin{pmatrix} Y^n \\ \Lambda^n \end{pmatrix} \begin{pmatrix} Y^{n+1} - Y^n \\ \Lambda^{n+1} - \Lambda^n \end{pmatrix} = -\mathcal{F}\begin{pmatrix} Y^n \\ \Lambda^n \end{pmatrix}, \tag{2.5}$$

where the Jacobian matrix of $\mathcal{F}$ is given by

$$\mathcal{F}'\begin{pmatrix} Y \\ \Lambda \end{pmatrix} = \tag{2.6}$$

$$\left(\begin{array}{ccccc|cccc} I & & & & & & & & \\ -P_y(Y_0,\Lambda_1) & I & & & & -P_\lambda(Y_0,\Lambda_1) & & & \\ & \ddots & \ddots & & & & \ddots & & \\ & & & -P_y(Y_{L-1},\Lambda_L) & I & & & & -P_\lambda(Y_{L-1},\Lambda_L) \\ \hline & -Q_y(Y_1,\Lambda_2) & & & & I & -Q_\lambda(Y_1,\Lambda_2) & & \\ & & \ddots & & & & \ddots & \ddots & \\ & & & -Q_y(Y_{L-1},\Lambda_L) & & & & I & -Q_\lambda(Y_{L-1},\Lambda_L) \\ & & & & -I & & & & I \end{array}\right).$$

Using the explicit expression for the Jacobian gives us the componentwise linear system we have to solve at each Newton iteration:

$$
\begin{aligned}
Y_0^{n+1} &= y_{init},\\
Y_1^{n+1} &= -P(Y_0^n,\Lambda_1^n) + P_y(Y_0^n,\Lambda_1^n)(Y_0^{n+1}-Y_0^n) + P_\lambda(Y_0^n,\Lambda_1^n)(\Lambda_1^{n+1}-\Lambda_1^n),\\
Y_2^{n+1} &= -P(Y_1^n,\Lambda_2^n) + P_y(Y_1^n,\Lambda_2^n)(Y_1^{n+1}-Y_1^n) + P_\lambda(Y_1^n,\Lambda_2^n)(\Lambda_2^{n+1}-\Lambda_2^n),\\
&\ \vdots\\
Y_L^{n+1} &= -P(Y_{L-1}^n,\Lambda_L^n) + P_y(Y_{L-1}^n,\Lambda_L^n)(Y_{L-1}^{n+1}-Y_{L-1}^n) + P_\lambda(Y_{L-1}^n,\Lambda_L^n)(\Lambda_L^{n+1}-\Lambda_L^n),\\
\Lambda_1^{n+1} &= Q(Y_1^n,\Lambda_2^n) + Q_\lambda(Y_1^n,\Lambda_2^n)(\Lambda_2^{n+1}-\Lambda_2^n) + Q_y(Y_1^n,\Lambda_2^n)(Y_1^{n+1}-Y_1^n),\\
\Lambda_2^{n+1} &= Q(Y_2^n,\Lambda_3^n) + Q_\lambda(Y_2^n,\Lambda_3^n)(\Lambda_3^{n+1}-\Lambda_3^n) + Q_y(Y_2^n,\Lambda_3^n)(Y_2^{n+1}-Y_2^n),\\
&\ \vdots\\
\Lambda_{L-1}^{n+1} &= Q(Y_{L-1}^n,\Lambda_L^n) + Q_\lambda(Y_{L-1}^n,\Lambda_L^n)(\Lambda_L^{n+1}-\Lambda_L^n) + Q_y(Y_{L-1}^n,\Lambda_L^n)(Y_{L-1}^{n+1}-Y_{L-1}^n),\\
\Lambda_L^{n+1} &= Y_L^{n+1} - y_{target}.
\end{aligned}
\tag{2.7}
$$

Note that this system is not triangular: the Y_ℓ^{n+1} are coupled to the Λ_ℓ^{n+1} and vice versa, which is clearly visible in the Jacobian in (2.6). This is in contrast to the initial value problem case, where the application of multiple shooting leads to a block lower triangular system.

The parareal approximation idea is to replace the derivative term by a difference computed on a coarse grid in (2.7), i.e., to use the approximations

$$
\begin{aligned}
P_y(Y_{\ell-1}^n,\Lambda_\ell^n)(Y_{\ell-1}^{n+1}-Y_{\ell-1}^n) &\approx P^G(Y_{\ell-1}^{n+1},\Lambda_\ell^n) - P^G(Y_{\ell-1}^n,\Lambda_\ell^n),\\
P_\lambda(Y_{\ell-1}^n,\Lambda_\ell^n)(\Lambda_\ell^{n+1}-\Lambda_\ell^n) &\approx P^G(Y_{\ell-1}^n,\Lambda_\ell^{n+1}) - P^G(Y_{\ell-1}^n,\Lambda_\ell^n),\\
Q_\lambda(Y_{\ell-1}^n,\Lambda_\ell^n)(\Lambda_\ell^{n+1}-\Lambda_\ell^n) &\approx Q^G(Y_{\ell-1}^n,\Lambda_\ell^{n+1}) - Q^G(Y_{\ell-1}^n,\Lambda_\ell^n),\\
Q_y(Y_{\ell-1}^n,\Lambda_\ell^n)(Y_{\ell-1}^{n+1}-Y_{\ell-1}^n) &\approx Q^G(Y_{\ell-1}^{n+1},\Lambda_\ell^n) - Q^G(Y_{\ell-1}^n,\Lambda_\ell^n),
\end{aligned}
\tag{2.8}
$$

where P^G and Q^G are propagators obtained from a coarse discretization of the subinterval problem (2.3), e.g., by using only one time step for the whole subinterval. This is certainly cheaper than evaluating the derivative on the fine grid; the remaining expensive fine grid operations $P(Y_{\ell-1}^n,\Lambda_\ell^n)$ and $Q(Y_{\ell-1}^n,\Lambda_\ell^n)$ in (2.7) can now all be performed in parallel. However, since (2.7) does not have a block triangular structure, the resulting non-linear system would need to be solved iteratively; each of these outer iterations is now very expensive, since one must evaluate the propagators $P^G(Y_{\ell-1}^{n+1},\Lambda_\ell^n)$, etc., by solving a *coupled non-linear* local control problem. This is in contrast to initial value problems, where the additional cost of solving non-linear local problems is justified, because the block lower triangular structure allows one to solve the outer problem by forward substitution, without the need to iterate. In order to reduce the cost of computing outer residuals, our idea is not to use the parareal approximation (2.8), but to use the so-called "derivative parareal" variant, where we approximate the derivative by effectively computing it for a coarse problem, see [12],

$$
\begin{aligned}
P_y(Y_{\ell-1}^n,\Lambda_\ell^n)(Y_{\ell-1}^{n+1}-Y_{\ell-1}^n) &\approx P_y^G(Y_{\ell-1}^n,\Lambda_\ell^n)(Y_{\ell-1}^{n+1}-Y_{\ell-1}^n),\\
P_\lambda(Y_{\ell-1}^n,\Lambda_\ell^n)(\Lambda_\ell^{n+1}-\Lambda_\ell^n) &\approx P_\lambda^G(Y_{\ell-1}^n,\Lambda_\ell^n)(\Lambda_\ell^{n+1}-\Lambda_\ell^n),\\
Q_\lambda(Y_{\ell-1}^n,\Lambda_\ell^n)(\Lambda_\ell^{n+1}-\Lambda_\ell^n) &\approx Q_\lambda^G(Y_{\ell-1}^n,\Lambda_\ell^n)(\Lambda_\ell^{n+1}-\Lambda_\ell^n),\\
Q_y(Y_{\ell-1}^n,\Lambda_\ell^n)(Y_{\ell-1}^{n+1}-Y_{\ell-1}^n) &\approx Q_y^G(Y_{\ell-1}^n,\Lambda_\ell^n)(Y_{\ell-1}^{n+1}-Y_{\ell-1}^n).
\end{aligned}
\tag{2.9}
$$

The advantage of this approximation is that the computation of P_y^G, P_λ^G, etc. only involves linear problems. Indeed, for a small perturbation δy in $Y_{\ell-1}$, the quantities $P_y^G(Y_{\ell-1},\Lambda_\ell)\delta y$ and $Q_y^G(Y_{\ell-1},\Lambda_\ell)\delta y$ can be computed by discretizing and solving the coupled differential equations obtained by differentiating (2.2). If (y, λ) is the solution of (2.2) with $y(T_{\ell-1}) = Y_{\ell-1}$ and $\lambda(T_\ell) = \Lambda_\ell$, then solving the *linear* derivative system

$$\begin{aligned} \dot{z} &= f'(y)z + \mu/\alpha, & \dot{\mu} &= -f'(y)^T\mu - H(y,z)^T\lambda, \\ z(T_{\ell-1}) &= \delta y, & \mu(T_\ell) &= 0 \end{aligned} \tag{2.10}$$

on a coarse time grid leads to

$$z(T_\ell) = P_y^G(Y_{\ell-1}, \Lambda_\ell)\delta y, \qquad \mu(T_{\ell-1}) = Q_y^G(Y_{\ell-1}, \Lambda_\ell)\delta y,$$

where $H(y, z) = \lim_{r\to 0} \frac{1}{r}(f'(y + rz) - f'(y))$ is the Hessian of f multiplied by z, and is thus linear in z. Similarly, to compute $P_\lambda^G(Y_{\ell-1}, \Lambda_\ell)\delta\lambda$ and $Q_\lambda^G(Y_{\ell-1}, \Lambda_\ell)\delta\lambda$ for a perturbation $\delta\lambda$ in Λ_ℓ, it suffices to solve the same ODE system as (2.10), except the end-point conditions must be replaced by $z(T_{\ell-1}) = 0$, $\mu(T_\ell) = \delta\lambda$. Therefore, if GMRES is used to solve the Jacobian system (2.5), then each matrix-vector multiplication requires only the solution of coarse, *linear* subproblems in parallel, which is much cheaper than solving coupled non-linear subproblems in the standard parareal approximation (2.8).

To summarize, our new ParaOpt method consists of solving for $n = 0, 1, 2, \ldots$ the system

$$\mathcal{J}^G \begin{pmatrix} Y^n \\ \Lambda^n \end{pmatrix} \begin{pmatrix} Y^{n+1} - Y^n \\ \Lambda^{n+1} - \Lambda^n \end{pmatrix} = -\mathcal{F} \begin{pmatrix} Y^n \\ \Lambda^n \end{pmatrix}, \tag{2.11}$$

for Y^{n+1} and Λ^{n+1}, where

$$\mathcal{J}^G \begin{pmatrix} Y \\ \Lambda \end{pmatrix} = \tag{2.12}$$

$$\left(\begin{array}{cccc|cccc}
I & & & & & & & \\
-P_y^G(Y_0,\Lambda_1) & I & & & -P_\lambda^G(Y_0,\Lambda_1) & & & \\
 & \ddots & \ddots & & & \ddots & & \\
 & & -P_y^G(Y_{L-1},\Lambda_L) & I & & & & -P_\lambda^G(Y_{L-1},\Lambda_L) \\
\hline
 & -Q_y^G(Y_1,\Lambda_2) & & & I & -Q_\lambda^G(Y_1,\Lambda_2) & & \\
 & & \ddots & & & \ddots & \ddots & \\
 & & & -Q_y^G(Y_{L-1},\Lambda_L) & & & I & -Q_\lambda^G(Y_{L-1},\Lambda_L) \\
 & & & -I & & & & I
\end{array}\right)$$

is an approximation of the true Jacobian in (2.6). If the system (2.11) is solved using a matrix-free method, the action of the sub-blocks P_y^G, P_λ^G, etc. can be obtained by solving coarse linear subproblems of the type (2.10). Note that the calculation of $\mathcal{J}^G$ times a vector (without preconditioning) is embarrassingly parallel, since it only requires the solution of local subproblems of the type (2.10), with no additional coupling to other sub-intervals. Global communication is only required in two places: within the Krylov method itself (e.g. when calculating inner products), and possibly within the preconditioner. The design of an effective preconditioner is an important and technical topic that will be the subject of a future paper. Of course, for problems with small state spaces (e.g. for ODE control problems), direct methods may also be used, once the coefficients of $\mathcal{J}^G$ are calculated by solving (2.10) for suitable choices of δy and $\delta\lambda$.

Regardless of how (2.11) is solved, since we use an approximation of the Jacobian, the resulting inexact Newton method will no longer converge quadratically, but only linearly; this is true even in the case where the differential equation is linear. In the next section, we will analyze in detail the convergence of the method for the case of a diffusive linear problem.

3. Implicit Euler for the diffusive linear case.

We now consider the method in a linear and discrete setting. More precisely, we focus on a control problem

$$\dot{y}(t) = Ay(t) + c(t), \tag{3.1}$$

where A is a real, symmetric matrix with **negative** eigenvalues. The matrix A can for example be a finite difference discretization of a diffusion operator in space. We will consider a *discretize-then-optimize* strategy, so the analysis that follows is done in a discrete setting.

3.1. Discrete formulation.

To fix ideas, we choose the implicit Euler[1] method for the time discretization; other discretizations will be studied in a future paper. Let $M \in \mathbb{N}$, and $\delta t = T/M$. Then the implicit Euler method gives[2]

$$y_{n+1} = y_n + \delta t(Ay_{n+1} + c_{n+1}), \tag{3.2}$$

or, equivalently,

$$y_{n+1} = (I - \delta tA)^{-1}(y_n + \delta tc_{n+1}).$$

We minimize the cost functional

$$J_{\delta t}(c) = \frac{1}{2}\|y_M - y_{target}\|^2 + \frac{\alpha}{2}\delta t\sum_{n=0}^{M-1}\|c_{n+1}\|^2.$$

For the sake of simplicity, we keep the notations y, λ and c for the discrete variables, that is $y = (y_n)_{n=0,\cdots,M}$, $\lambda = (\lambda_n)_{n=0,\cdots,M}$ and $c = (c_n)_{n=0,\cdots,M}$. Introducing the Lagrangian (see [29, 16] and also [22, 51, 46] for details)

$$\mathcal{L}_{\delta t}(y, \lambda, c) = J_{\delta t}(c) - \sum_{n=0}^{M-1}\left\langle \lambda_{n+1}, y_{n+1} - (I - \delta tA)^{-1}(y_n + \delta tc_{n+1})\right\rangle,$$

the optimality systems reads:

$$\begin{aligned}
&(3.3) && y_0 = y_{init}, && \\
&(3.4) && y_{n+1} = (I - \delta tA)^{-1}(y_n + \delta tc_{n+1}), && n = 0, 1, \ldots, M-1, \\
&(3.5) && \lambda_M = y_M - y_{target}, && \\
&(3.6) && \lambda_n = (I - \delta tA)^{-1}\lambda_{n+1}, && n = 0, 1, \ldots, M-1, \\
&(3.7) && \alpha c_{n+1} = -(I - \delta tA)^{-1}\lambda_{n+1}, && n = 0, 1, \ldots, M-1,
\end{aligned}$$

[1] We use the term 'implicit Euler' instead of 'Backward Euler' because the method is applied forward and backward in time.

[2] If the ODE system contains mass matrices arising from a finite element discretization, e.g.,

$$\mathcal{M}y_{n+1} = \mathcal{M}y_n + \delta t(Ay_{n+1} + \mathcal{M}c_{n+1}),$$

then one can analyze ParaOpt by introducing the change of variables $\bar{y}_n := \mathcal{M}^{1/2}y_n$, $\bar{c}_n := \mathcal{M}^{1/2}c_n$, so as to obtain

$$\bar{y}_{n+1} = \bar{y}_n + \delta t(\bar{A}\bar{y}_{n+1} + \bar{c}_{n+1}),$$

with $\bar{A} := \mathcal{M}^{-1/2}A\mathcal{M}^{-1/2}$. Since $\bar{A}$ is symmetric positive definite whenever A is, the analysis is identical to that for (3.2), even though one would never calculate $\mathcal{M}^{1/2}$ and $\bar{A}$ in actual computations.

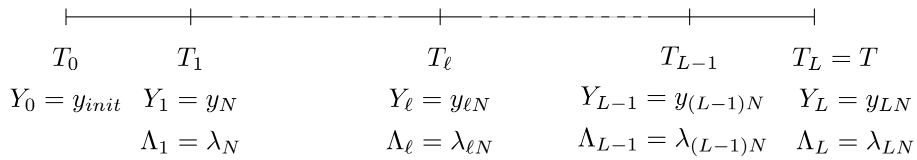

FIG. 1. *Notations associated with the parallelization setting.*

where we used the fact that A is symmetric. If $A = VDV^T$ is the eigenvalue decomposition of A, then the transformation $y_n \mapsto V^T y_n$, $\lambda_n \mapsto V^T \lambda_n$, $c_n \mapsto V^T c_n$ allows us to diagonalize the equations (3.3)–(3.7) and obtain a family of **decoupled** optimality systems of the form

$$
\begin{aligned}
&(3.8) && y_0 = y_{init}, && \\
&(3.9) && y_{n+1} = (I - \sigma\delta t)^{-1}(y_n + \delta t c_{n+1}), && n = 0, 1, \ldots, M-1, \\
&(3.10) && \lambda_M = y_M - y_{target}, && \\
&(3.11) && \lambda_n = (I - \sigma\delta t)^{-1}\lambda_{n+1}, && n = 0, 1, \ldots, M-1, \\
&(3.12) && \alpha c_{n+1} = -(I - \sigma\delta t)^{-1}\lambda_{n+1}, && n = 0, 1, \ldots, M-1,
\end{aligned}
$$

where the y_n, λ_n and c_n are now scalars, and $\sigma < 0$ is an eigenvalue of A. This motivates us to study the scalar **Dahlquist** problem

$$\dot{y}(t) = \sigma y(t) + c(t),$$

where σ is a real, negative number. For the remainder of this section, we will study the ParaOpt algorithm applied to the scalar variant (3.8)–(3.12), particularly its convergence properties as a function of σ.

Let us now write the linear ParaOpt algorithm for (3.8)–(3.12) in matrix form. For the sake of simplicity, we assume that the subdivision is uniform, that is $T_\ell = \ell\Delta T$, where N satisfies $\Delta T = N\delta t$ and $M = NL$, see Figure 1. We start by eliminating interior unknowns, i.e., ones that are not located at the time points $T_0, T_1, \ldots T_L$. For $0 \le n_1 \le n_2 \le M$, (3.9) and (3.12) together imply

$$
\begin{aligned}
y_{n_2} &= (1 - \sigma\delta t)^{n_1 - n_2} y_{n_1} - \delta t \sum_{j=0}^{n_2 - n_1 - 1} (1 - \sigma\delta t)^{n_1 - n_2 + j} c_{n_1 + j + 1} \\
&= (1 - \sigma\delta t)^{n_1 - n_2} y_{n_1} - \frac{\delta t}{\alpha} \sum_{j=0}^{n_2 - n_1 - 1} (1 - \sigma\delta t)^{n_1 - n_2 + j - 1} \lambda_{n_1 + j + 1}. \qquad (3.13)
\end{aligned}
$$

On the other hand, (3.11) implies

$$\lambda_{n_1 + j} = (1 - \sigma\delta t)^{n_1 - n_2 + j} \lambda_{n_2}. \qquad (3.14)$$

Combining (3.13) and (3.14) then leads to

$$y_{n_2} = (1 - \sigma\delta t)^{n_1 - n_2} y_{n_1} - \frac{\delta t}{\alpha} \Big[\sum_{j=0}^{n_2 - n_1 - 1} (1 - \sigma\delta t)^{2(n_1 - n_2 + j)} \Big] \lambda_{n_2}. \qquad (3.15)$$

Setting $n_1 = (\ell-1)N$ and $n_2 = \ell N$, and using the notation $Y_\ell = y_{\ell N}$, $\Lambda_\ell = \lambda_{\ell N}$ (see Figure 1), we obtain from (3.14) and (3.15) the equations

$$\begin{aligned} Y_0 &= y_{init} \\ -\beta_{\delta t} Y_{\ell-1} + Y_\ell + \frac{\gamma_{\delta t}}{\alpha}\Lambda_\ell &= 0, & 1 \le \ell \le M, \\ \Lambda_{\ell-1} - \beta_{\delta t}\Lambda_\ell &= 0, & 0 \le \ell \le M-1, \\ Y_\ell + \Lambda_\ell &= y_{target}, \end{aligned}$$

where

$$\beta_{\delta t} := (1-\sigma\delta t)^{-\Delta T/\delta t}, \tag{3.16}$$

$$\gamma_{\delta t} := \delta t \sum_{j=0}^{N-1} (1-\sigma\delta t)^{2(j-N)} = \frac{\beta_{\delta t}^2 - 1}{\sigma(2-\sigma\delta t)}. \tag{3.17}$$

In matrix form, this can be written as

$$\left(\begin{array}{cccc|cccc} 1 & & & & 0 & & & \\ -\beta_{\delta t} & \ddots & & & \gamma_{\delta t}/\alpha & & \ddots & \\ & \ddots & \ddots & & & & \ddots & 0 \\ & & -\beta_{\delta t} & 1 & & & & \gamma_{\delta t}/\alpha \\ \hline & & & & 1 & -\beta_{\delta t} & & \\ & & & & & \ddots & \ddots & \\ & & & & & & \ddots & -\beta_{\delta t} \\ & & & -1 & & & & 1 \end{array}\right) \left(\begin{array}{c} Y_0 \\ \vdots \\ \vdots \\ Y_L \\ \hline \Lambda_1 \\ \vdots \\ \vdots \\ \Lambda_L \end{array}\right) = \left(\begin{array}{c} y_{init} \\ 0 \\ \\ \vdots \\ \\ 0 \\ -y_{target} \end{array}\right),$$

or, in a more compact form,

$$A_{\delta t} X = b. \tag{3.18}$$

Note that this matrix has the same structure as the Jacobian matrix $\mathcal{F}$ in (2.6), except that $Q_\lambda = 0$ for the linear case. In order to solve (3.18) numerically, we consider a second time step Δt such that $\delta t \le \Delta t \le \Delta T$. In other words, for each sub-interval of length ΔT, the derivatives of the propagators P_y, Q_y, P_λ, Q_λ are approximated using a coarser time discretization with time step $\Delta t \le \Delta T$. The optimality system for this coarser time discretization has the form

$$A_{\Delta t}\hat{X} = b,$$

where $A_{\Delta t}$ has the same form as above, except that $\beta_{\delta t}$ and $\gamma_{\delta t}$ are replaced by $\beta_{\Delta t}$ and $\gamma_{\Delta t}$, i.e., the values obtained from the formulas (3.16) and (3.17) when one replaces δt by Δt. Then the ParaOpt algorithm (2.11–2.12) for the linear Dahlquist problem can be written as

$$A_{\Delta t}(X^{k+1} - X^k) = -\mathcal{F}(X^k) = -(A_{\delta t}X^k - b),$$

or, equivalently

$$X^{k+1} = \left(I - A_{\Delta t}^{-1}A_{\delta t}\right)X^k + A_{\Delta t}^{-1}b. \tag{3.19}$$

Note that using this iteration, only a coarse matrix needs to be inverted.

3.2. Eigenvalue problem. In order to study the convergence of the iteration (3.19), we study the eigenvalues of the matrix $I - A_{\Delta t}^{-1} A_{\delta t}$, which are given by the generalized eigenvalue problem

$$(A_{\Delta t} - A_{\delta t})x = \mu A_{\Delta t} x, \tag{3.20}$$

with $x = (v_0, v_1, \cdots, v_L, w_1, \cdots, w_L)^T$ being the eigenvector associated with the eigenvalue μ. Since $A_{\Delta t} - A_{\delta t}$ has two zero rows, the eigenvalue $\mu = 0$ must have multiplicity at least two. Now let $\mu \neq 0$ be a non-zero eigenvalue. (If no such eigenvalue exists, then the preconditioning matrix is nilpotent and the iteration converges in a finite number of steps.) Writing (3.20) componentwise yields

$$v_0 = 0 \tag{3.21}$$

$$\mu(v_\ell - \beta v_{\ell-1} + \gamma w_\ell/\alpha) = -\delta\beta v_{\ell-1} + \delta\gamma w_\ell/\alpha \tag{3.22}$$

$$\mu(w_\ell - \beta w_{\ell+1}) = -\delta\beta w_{\ell+1} \tag{3.23}$$

$$\mu\,(w_L - v_L) = 0, \tag{3.24}$$

where we have introduced the simplified notation

$$\beta = \beta_{\Delta t},\ \gamma = \gamma_{\Delta t},\ \delta\beta = \beta_{\Delta t} - \beta_{\delta t},\ \delta\gamma = \gamma_{\Delta t} - \gamma_{\delta t}. \tag{3.25}$$

The recurrences (3.22) and (3.23) are of the form

$$v_\ell = a v_{\ell-1} + b w_\ell, \qquad w_\ell = a w_{\ell+1}, \tag{3.26}$$

where

$$a = \beta - \mu^{-1}\delta\beta, \qquad b = \frac{-\gamma + \mu^{-1}\delta\gamma}{\alpha}.$$

Solving the recurrence (3.26) in v together with the initial condition (3.21) leads to

$$v_L = \sum_{\ell=1}^{L} a^{L-\ell} b w_\ell, \tag{3.27}$$

whereas the recurrence (3.26) in w simply gives

$$w_\ell = a^{L-\ell} w_L. \tag{3.28}$$

Combining (3.27) and (3.28), we obtain

$$v_L = \left(\sum_{\ell=1}^{L} a^{2(L-\ell)} b\right) w_L,$$

so that (3.24) gives rise to $P(\mu) w_L = 0$, with

$$P(\mu) = \alpha\mu^{2L-1} + (\mu\gamma - \delta\gamma)\sum_{\ell=0}^{L-1} \mu^{2(L-\ell-1)}(\mu\beta - \delta\beta)^{2\ell}. \tag{3.29}$$

Since we seek a non-trivial solution, we can assume $w_L \neq 0$. Therefore, the eigenvalues of $I - A_{\Delta t}^{-1} A_{\delta t}$ consist of the number zero (with multiplicity two), together with the $2L - 1$ roots of $P(\mu)$, which are all non-zero. In the next subsection, we will give a precise characterization of the roots of $P(\mu)$, which depend on α, as well as on σ via the parameters β, $\delta\beta$, γ and $\delta\gamma$.

3.3. Characterization of eigenvalues. In the next two results, we describe the location of the roots of $P(\mu)$ from the last section, or equivalently, the non-zero eigenvalues of the iteration matrix $I - A_{\Delta t}^{-1} A_{\delta t}$. We first establish the sign of a few parameters in the case $\sigma < 0$, which is true for diffusive problems.

LEMMA 3.1. *Let $\sigma < 0$. Then we have $0 < \beta < 1$, $0 < \delta\beta < \beta$, $\gamma > 0$ and $\delta\gamma < 0$.*

Proof. By the definitions (3.16) and (3.25), we see that

$$\beta = \beta_{\Delta t} = (1 - \sigma\Delta t)^{-\Delta T/\Delta t},$$

which is between 0 and 1, since $1 - \sigma\Delta t > 1$ for $\sigma < 0$. Moreover, $\beta_{\Delta t}$ is an increasing function of Δt by direct calculation, so that

$$\delta\beta = \beta_{\Delta t} - \beta_{\delta t} > 0,$$

which shows that $0 < \delta\beta < \beta$. Next, we have by definition

$$\gamma = \frac{\beta^2 - 1}{\sigma(2 - \sigma\Delta t)}.$$

Since $\beta < 1$ and $\sigma < 0$, both the numerator and the denominator are negative, so $\gamma > 0$. Finally, we have

$$\delta\gamma = \frac{1}{|\sigma|}\left(\frac{1 - \beta_{\Delta t}^2}{2 + |\sigma|\Delta t} - \frac{1 - \beta_{\delta t}^2}{2 + |\sigma|\delta t}\right) < 0,$$

since $1 - \beta_{\Delta t}^2 < 1 - \beta_{\delta t}^2$ and $2 + |\sigma|\Delta t > 2 + |\sigma|\delta t$, so the first quotient inside the parentheses is necessarily smaller than the second quotient. $\square$

We are now ready to prove a first estimate for the eigenvalues of the matrix $I - A_{\Delta t}^{-1} A_{\delta t}$.

THEOREM 3.2. *Let P be the polynomial defined in (3.29). For $\sigma < 0$, the roots of P are contained in the set $D_\sigma \cup \{\mu^*\}$, where*

$$D_\sigma = \{\mu \in \mathbb{C} : |\mu - \mu_0| < \delta\beta/(1 - \beta^2)\}, \tag{3.30}$$

where $\mu_0 = -\beta\delta\beta/(1 - \beta^2)$, and $\mu^* < 0$ is a real negative number.

Proof. Since zero is not a root of $P(\mu)$, we can divide $P(\mu)$ by μ^{2L-1} and see that $P(\mu)$ has the same roots as the function

$$\hat{P}(\mu) = \alpha + (\gamma - \mu^{-1}\delta\gamma) \sum_{\ell=0}^{L-1} (\beta - \mu^{-1}\delta\beta)^{2\ell}.$$

Recall the change of variables

$$a = \beta - \mu^{-1}\delta\beta \quad \Longleftrightarrow \quad \mu = \frac{\delta\beta}{\beta - a}\ ;$$

substituting a into $\hat{P}(\mu)$ and multiplying the result by $\delta\beta/|\delta\gamma|$ shows that $P(\mu) = 0$ is equivalent to

$$Q(a) := \frac{\alpha\delta\beta}{|\delta\gamma|} + (C - a) \sum_{\ell=0}^{L-1} a^{2\ell} = 0,$$

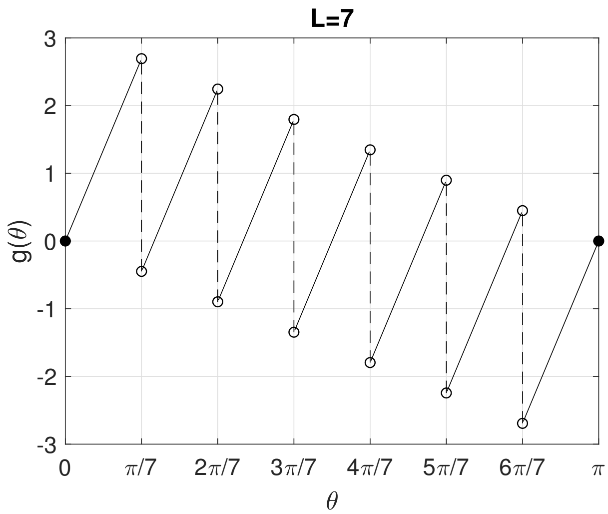

FIG. 2. *Plot of* $g(\theta) = \arg\left(\sum_{\ell=0}^{L-1} e^{2i\ell\theta}\right)$ *for* $L = 7$.

with

$$C := \beta + \gamma\delta\beta/|\delta\gamma| > 0. \tag{3.31}$$

We will now show that $Q(a)$ has at most one root inside the unit disc $|a| \leq 1$; since the transformation from μ to a maps circles to circles, this would be equivalent to proving that $P(\mu)$ has at most one root outside the disc D_σ. We now use the argument principle from complex analysis, which states that the difference between the number of zeros and poles of Q inside a closed contour $\mathcal{C}$ is equal to the winding number of the contour $Q(\mathcal{C})$ around the origin. Since Q is a polynomial and has no poles, this would allow us to count the number of zeros of Q inside the unit disc. Therefore, we consider the winding number of the contour $\Gamma = \{f(e^{i\theta}) : 0 \leq \theta \leq 2\pi\}$ with

$$f(a) = (C - a)\sum_{\ell=0}^{L-1} a^{2\ell}$$

around the point $-\alpha\delta\beta/|\delta\gamma|$, which is a real negative number. If we can show that Γ intersects the negative real axis at at most one point, then it follows that the winding number around any negative real number cannot be greater than 1.

We now concentrate on finding values of θ such that $\arg(f(e^{i\theta})) = \pi \pmod{2\pi}$. Since $f(\overline{a}) = \overline{f(a)}$, it suffices to consider the range $0 \leq \theta \leq \pi$, and the other half of the range will follow by conjugation. Since f is a product, we deduce that

$$\arg(f(e^{i\theta})) = \arg(C - e^{i\theta}) + \arg\left(1 + e^{2i\theta} + \cdots + e^{2(L-1)i\theta}\right).$$

We consider the two terms on the right separately.

- For the first term, we have for all $0 < \theta < \pi$

$$\theta - \pi < \arg(-e^{i\theta}) < \arg(C - e^{i\theta}) < 0,$$

since C is real and positive. For $\theta = \pi$, we obviously have $\arg(C - e^{i\theta}) = 0$, whereas for $\theta = 0$, we have $\arg(C - e^{i\theta}) = -\pi$ if $C < 1$, and $\arg(C - e^{i\theta}) = 0$ otherwise.

- For the second term, observe that for $0 < \theta < \pi$, we have

$$1 + e^{2i\theta} + \cdots + e^{2(L-1)i\theta} = \frac{1 - e^{2iL\theta}}{1 - e^{2i\theta}} = e^{(L-1)i\theta} \cdot \frac{\sin(L\theta)}{\sin(\theta)}.$$

  Therefore, the second term is piecewise linear with slope $L - 1$, with a jump of size π whenever $\sin(L\theta)$ changes sign, i.e., at $\theta = k\pi/L$, $k = 1, \ldots, L-1$. Put within the range $(-\pi, \pi)$, we can write

$$\arg\left(\frac{1 - e^{2iL\theta}}{1 - e^{2i\theta}}\right) = (L-1)\theta - \left\lfloor \frac{L\theta}{\pi} \right\rfloor \pi =: g(\theta), \qquad 0 < \theta < \pi.$$

  We also have $g(0) = g(\pi) = 0$ by direct calculation. The function g satisfies the property $-\theta \le g(\theta) \le \pi - \theta$, see Figure 2.

From the above, we deduce that $\arg(f(e^{i\theta})) < \pi$ for all $0 \le \theta \le \pi$. Moreover,

$$\arg(f(e^{i\theta})) = \begin{cases} 0, & \text{if } \theta = 0 \text{ and } C > 1, \\ -\pi, & \text{if } \theta = 0 \text{ and } C < 1, \\ \arg(C - e^{i\theta}) + g(\theta) > -\pi, & \text{if } 0 < \theta < \pi, \\ 0, & \text{if } \theta = \pi. \end{cases}$$

Thus, the winding number around the point $-\alpha\delta\beta/|\delta\gamma|$ cannot exceed one, so at most one of the roots of Q can lie inside the unit disc. If there is indeed such a root a^*, it must be real, since the conjugate of any root of Q is also a root. Moreover, it must satisfy $a^* > C$, since $Q(a) > 0$ for any $a \le C$. This implies

$$\beta - a^* < \beta - C = -\frac{\gamma\delta\beta}{|\delta\gamma|} < 0, \tag{3.32}$$

so the corresponding $\mu^* = \delta\beta/(\beta - a^*)$ must also be negative. □

We have seen that the existence of μ^* depends on whether the constant C is larger than 1. The following lemma shows that we indeed have $C < 1$.

LEMMA 3.3. *Let $\sigma < 0$. Then the constant $C = \beta + \gamma\delta\beta/|\delta\gamma|$, defined in (3.31), satisfies $C < 1$.*

Proof. We first transform the relation $C < 1$ into a sequence of equivalent inequalities. Starting with the definition of C, we have

$$\begin{aligned} C = \beta_{\Delta t} + \frac{\gamma_{\Delta t}(\beta_{\Delta t} - \beta_{\delta t})}{\gamma_{\delta t} - \gamma_{\Delta t}} < 1 &\iff \beta_{\Delta t}(\gamma_{\delta t} - \gamma_{\Delta t}) + \gamma_{\Delta t}(\beta_{\Delta t} - \beta_{\delta t}) < \gamma_{\delta t} - \gamma_{\Delta t} \\ &\iff \gamma_{\Delta t}(1 - \beta_{\delta t}) < \gamma_{\delta t}(1 - \beta_{\Delta t}) \\ &\iff \frac{(1 - \beta_{\Delta t}^2)(1 - \beta_{\delta t})}{|\sigma|(2 + |\sigma|\Delta t)} < \frac{(1 - \beta_{\delta t}^2)(1 - \beta_{\Delta t})}{|\sigma|(2 + |\sigma|\delta t)} \\ &\iff \frac{1 + \beta_{\Delta t}}{2 + |\sigma|\Delta t} < \frac{1 + \beta_{\delta t}}{2 + |\sigma|\delta t}, \end{aligned}$$

where the last equivalence is obtained by multiplying both sides of the penultimate inequality by $|\sigma|$ and then dividing it by $(1 - \beta_{\Delta t})(1 - \beta_{\delta t})$. By the definition of $\beta_{\Delta t}$ and $\beta_{\delta t}$, the last inequality can be written as $f(|\sigma|\Delta t) < f(|\sigma|\delta t)$, where

$$f(x) := \frac{1 + (1 + x)^{-k/x}}{2 + x}$$

with $k = |\sigma|\Delta T > 0$. Therefore, it suffices to show that $f(x)$ is decreasing for $0 < x \le k$. In other words, we need to show that

$$f'(x) = \frac{(1+x)^{-k/x}}{2+x}\left[\frac{k\ln(1+x)}{x^2} - \frac{k}{x(1+x)}\right] - \frac{1+(1+x)^{-k/x}}{(2+x)^2} < 0.$$

This is equivalent to showing

$$(2+x)\left[\frac{k\ln(1+x)}{x^2} - \frac{k}{x(1+x)}\right] - 1 < (1+x)^{k/x}. \tag{3.33}$$

Using the fact that $\ln(1+x) \le x$, we see that the left hand side is bounded above by

$$\begin{aligned}(2+x)\left[\frac{k\ln(1+x)}{x^2} - \frac{k}{x(1+x)}\right] - 1 &\le (2+x)\left[\frac{kx}{x^2} - \frac{k}{x(1+x)}\right] - 1\\ &= k\left(\frac{2+x}{1+x}\right) - 1.\end{aligned}$$

But for every $k > 0$ and $0 < x < k$ we have

$$(1+x)^{k/x} > k\left(\frac{2+x}{1+x}\right) - 1, \tag{3.34}$$

see proof in the appendix. Therefore, (3.33) is satisfied by all $k > 0$ and $0 < x < k$, so f is in fact decreasing. It follows that $C < 1$, as required. ∎

THEOREM 3.4. *Let $\sigma < 0$ be fixed, and let*

$$L_0 := \frac{C-\beta}{\gamma(1-C)}. \tag{3.35}$$

Then the spectrum of $I - A_{\Delta t}^{-1}A_{\delta t}$ has an eigenvalue μ^* outside the disc D_σ defined in (3.30) if and only if the number of subintervals L satisfies $L > \alpha L_0$, where α is the regularization parameter.

Proof. The isolated eigenvalue exists if and only if the winding number of $Q(e^{i\theta})$ about the origin is non-zero. Since $Q(e^{i\theta})$ only intersects the negative real axis at most once, we see that the winding number is non-zero when $Q(-1) < 0$, i.e., when

$$\frac{\alpha\delta\beta}{|\delta\gamma|} + (C-1)L < 0.$$

Using the definition of C, this leads to

$$\frac{\alpha(C-\beta)}{\gamma} + (C-1)L < 0 \iff L > \frac{\alpha(C-\beta)}{\gamma(1-C)},$$

hence the result. ∎

3.4. Spectral radius estimates. The next theorem now gives a more precise estimate on the isolated eigenvalue μ^*.

THEOREM 3.5. *Suppose that the number of intervals L satisfies $L > \alpha L_0$, with L_0 defined in (3.35). Then the real negative eigenvalue μ^* outside the disc D_σ is bounded below by*

$$\mu^* > -\frac{|\delta\gamma| + \alpha\delta\beta(1+\beta)}{\gamma + \alpha(1-\beta^2)}.$$

Proof. Suppose $a^* = \beta - \delta\beta/\mu^*$ is a real root of $Q(a)$ inside the unit disc. We have seen at the end of the proof of Theorem 3.2 (page 12, immediately before Equation (3.32)) that $a^* > C$; moreover, since a^* is assumed to be inside the unit circle, we must have $a^* < 1$. Therefore, a^* satisfies $C < a^* < 1$. This implies

$$\frac{\alpha\delta\beta}{|\delta\gamma|} + \frac{C - a^*}{1-(a^*)^2} = \frac{(C-a^*)(a^*)^{2L}}{1-(a^*)^2} < 0.$$

Therefore, a^* satisfies

$$(1-(a^*)^2)\alpha\delta\beta + |\delta\gamma|(C - a^*) < 0,$$

which means

$$\begin{aligned} a^* >& \frac{-|\delta\gamma| + \sqrt{|\delta\gamma|^2 + 4\alpha\delta\beta(\alpha\delta\beta + C|\delta\gamma|)}}{2\alpha\delta\beta} \\ =& \frac{-|\delta\gamma| + \sqrt{(|\delta\gamma|^2 + 2\alpha\delta\beta)^2 - 4(1-C)\alpha\delta\beta|\delta\gamma|}}{2\alpha\delta\beta}. \end{aligned}$$

Therefore,

$$\begin{aligned} \mu^* &= \frac{\delta\beta}{\beta - a^*} \\ &> \frac{2\alpha\delta\beta^2}{(2\alpha\beta\delta\beta + |\delta\gamma|) - \sqrt{(|\delta\gamma| + 2\alpha\delta\beta)^2 - 4(1-C)\alpha\delta\beta|\delta\gamma|}} \\ &= \frac{2\alpha\delta\beta^2\left[(2\alpha\beta\delta\beta + |\delta\gamma|) + \sqrt{(|\delta\gamma| + 2\alpha\delta\beta)^2 - 4(1-C)\alpha\delta\beta|\delta\gamma|}\right]}{(2\alpha\beta\delta\beta + |\delta\gamma|)^2 - (|\delta\gamma| + 2\alpha\delta\beta)^2 + 4(1-C)\alpha\delta\beta|\delta\gamma|} \\ &= \frac{\delta\beta\left[(2\alpha\beta\delta\beta + |\delta\gamma|) + \sqrt{(|\delta\gamma| + 2\alpha\delta\beta)^2 - 4(1-C)\alpha\delta\beta|\delta\gamma|}\right]}{2(\beta - C)|\delta\gamma| + 2\alpha\delta\beta(\beta^2-1)} \\ &= -\frac{(2\alpha\beta\delta\beta + |\delta\gamma|) + \sqrt{(|\delta\gamma| + 2\alpha\delta\beta)^2 - 4(1-C)\alpha\delta\beta|\delta\gamma|}}{2\gamma + 2\alpha(1-\beta^2)} \\ &> -\frac{|\delta\gamma| + \alpha\delta\beta(1+\beta)}{\gamma + \alpha(1-\beta^2)}, \end{aligned}$$

where the last inequality is obtained by dropping the term containing $(1 - C)$ inside the square root, which makes the square root larger since $C < 1$. ∎

To illustrate the above theorems, we show in Figures 3 and 4 the spectrum of the iteration matrix $I - A_{\Delta t}^{-1}A_{\delta t}$ for different values of σ and for $\alpha = 1$ and 1000. Here, the time interval $[0, T]$ is subdivided into $L = 30$ subintervals, and each subinterval contains 50 coarse time steps and 5000 fine time steps. Table 1 shows the values of the relevant parameters. For $\alpha = 1$, we see that there is always one isolated eigenvalue on the negative real axis, since $L > L_0$ in all cases, and its location is predicted rather accurately by the formula (3.36). The rest of the eigenvalues all lie within the disc D_σ defined in (3.30). For $\alpha = 1000$, the bounding disc is identical to the previous case; however, since we have $L < \alpha L_0$ for all cases except for $\sigma = -16$, we observe no eigenvalue outside the disc, except for the very last case. In that very last case, we have $|\delta\gamma| = 0.0107$, so (3.36) gives the lower bound $\mu^* > -1.07 \times 10^{-5}$, which again is quite accurate when compared with the bottom right panel of Figure 4.

TABLE 1
Parameter values for $T = 100$, $L = 30$, $\Delta T/\Delta t = 50$, $\Delta t/\delta t = 100$.

σ	β	γ	C	L_0	Radius of D_σ	μ^* bound ($\alpha = 1$)
$-1/8$	0.6604	2.2462	0.8268	0.4280	2.00×10^{-3}	-6.08×10^{-3}
$-1/4$	0.4376	1.6037	0.6960	0.5300	3.67×10^{-3}	-9.34×10^{-3}
$-1/2$	0.1941	0.9466	0.4713	0.5539	5.35×10^{-3}	-1.24×10^{-2}
-1	0.0397	0.4831	0.1588	0.2930	3.97×10^{-3}	-1.36×10^{-2}
-2	0.0019	0.2344	0.0116	0.0417	6.36×10^{-4}	-1.30×10^{-2}
-16	1.72×10^{-16}	0.0204	5×10^{-16}	1.61×10^{-14}	1.72×10^{-16}	-1.05×10^{-2}

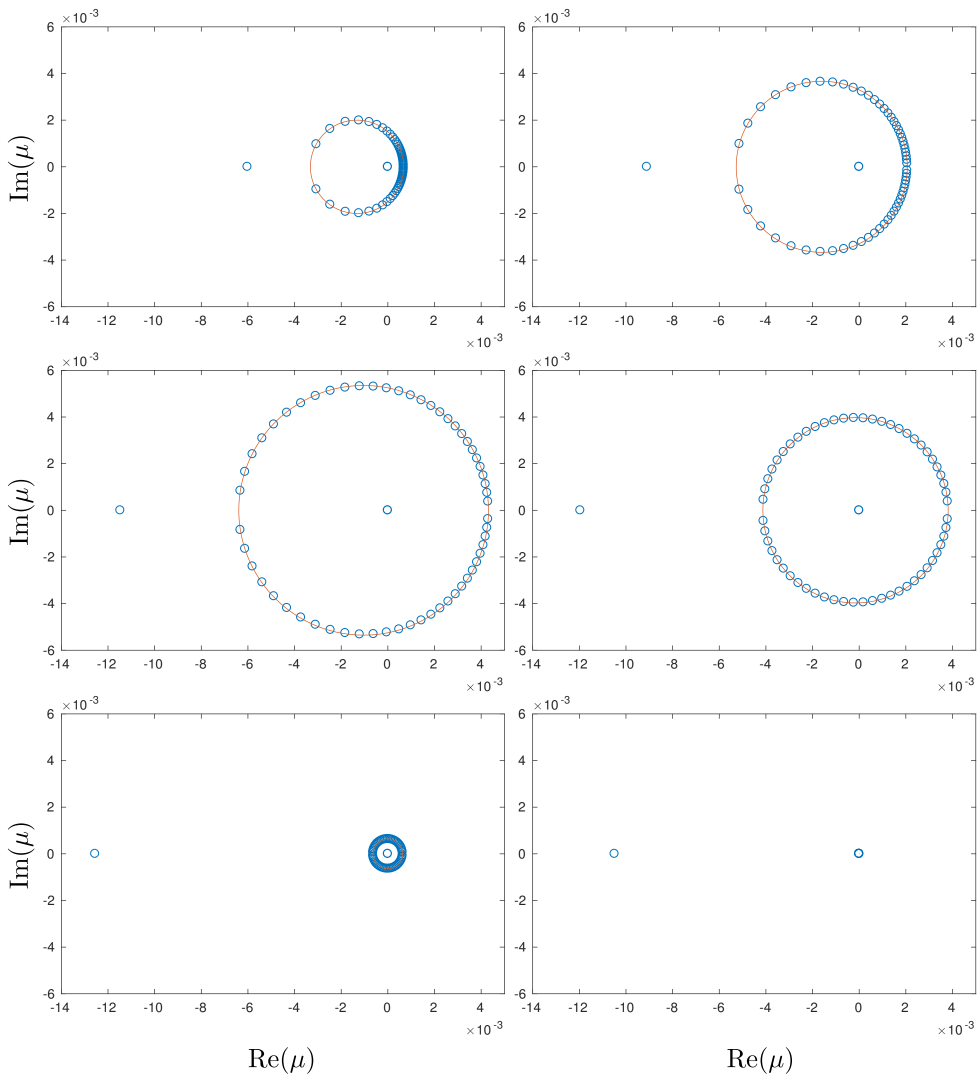

FIG. 3. *Spectrum of the iteration matrix for $T = 100$, $L = 30$, $\Delta T/\Delta t = 50$, $\Delta t/\delta t = 100$, $\alpha = 1$, and for $\sigma = -1/8, -1/4, -1/2, -1, -2, -16$, from top left to bottom right.*

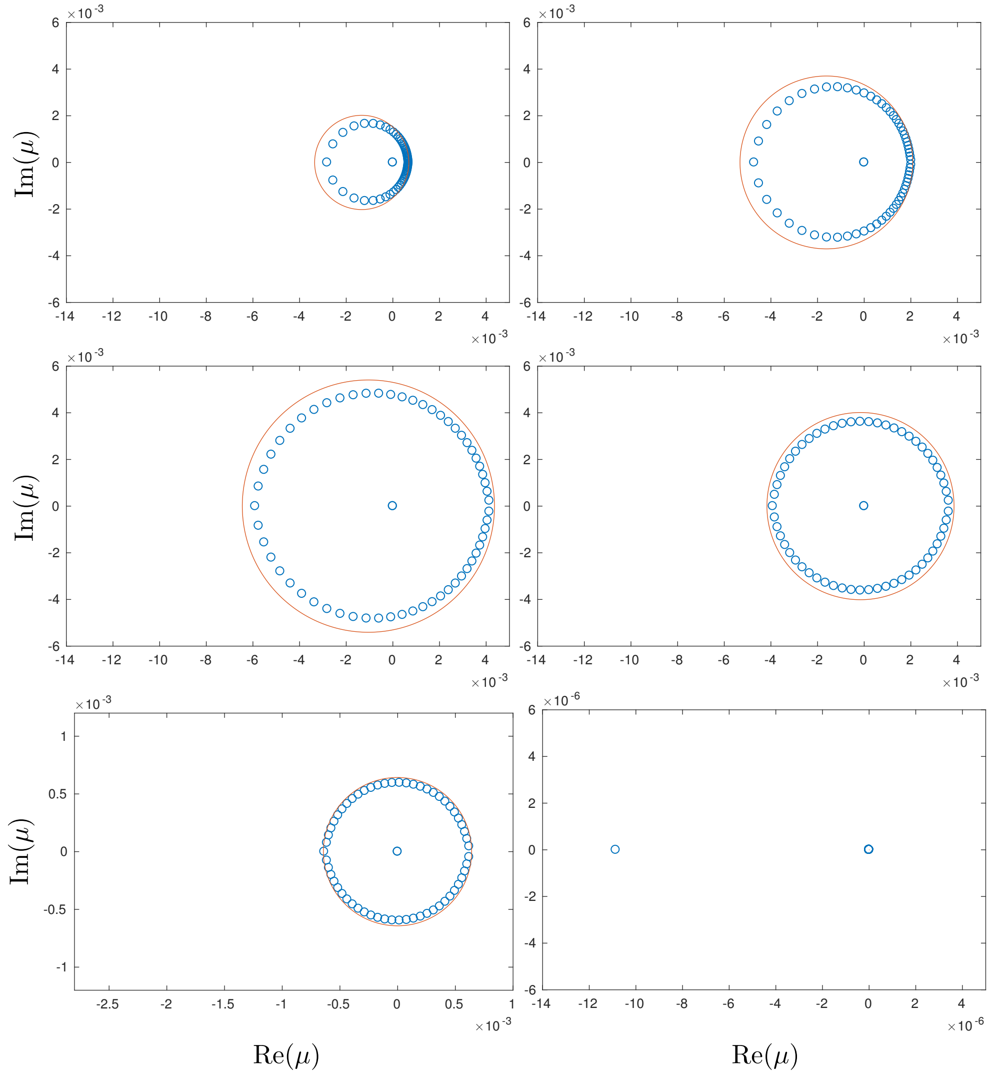

FIG. 4. *Spectrum of the iteration matrix for* $T = 100$*,* $L = 30$*,* $\Delta T/\Delta t = 50$*,* $\Delta t/\delta t = 100$*,* $\alpha = 1000$*, and for* $\sigma = -1/8, -1/4, -1/2, -1, -2, -16$*, from top left to bottom right.*

COROLLARY 3.6. *Let* T*,* ΔT*,* Δt*,* δt*,* α *and* σ *be fixed. Then the spectral radius* ρ *of the matrix* $I - A_{\Delta t}^{-1} A_{\delta t}$ *satisfies*

$$\rho \leq \frac{|\delta\gamma| + \alpha\delta\beta(1+\beta)}{\gamma + \alpha(1-\beta^2)}. \tag{3.36}$$

Note that the inequality (3.36) is valid for all $L > 0$, i.e., regardless of whether the isolated eigenvalue μ^* exists.

Proof. When the number of sub-intervals L satisfies $L > \alpha L_0$, the spectral radius is determined by the isolated eigenvalue, which according to Theorem 3.5 is estimated by

$$|\mu^*| < \frac{|\delta\gamma| + \alpha\delta\beta(1+\beta)}{\gamma + \alpha(1-\beta^2)}.$$

Otherwise, when $L \le \alpha L_0$, all the eigenvalues lie within the bounding disc D_σ, so no eigenvalue can be farther away from the origin than

$$\mathrm{Radius}(D_\sigma) + |\mathrm{Center}(D_\sigma)| = \frac{\delta\beta}{1-\beta^2} + \frac{\beta\delta\beta}{1-\beta^2} = \frac{\delta\beta}{1-\beta}.$$

A straightforward calculation shows that

$$\frac{|\delta\gamma| + \alpha\delta\beta(1+\beta)}{\gamma + \alpha(1-\beta^2)} > \frac{\delta\beta}{1-\beta} \quad \text{if and only if} \quad \beta + \frac{\gamma\delta\beta}{|\delta\gamma|} < 1,$$

which is true by Lemma 3.3. Thus, the inequality (3.36) holds in both cases. ∎

The above corollary is of interest when we apply our ParaOpt method to a large system of ODEs (arising from the spatial discretization of a PDE, for example), where the eigenvalues lie in the range $\sigma \in [-\sigma_{\max}, -\sigma_{\min}]$, with $\sigma_{\max} \to \infty$ when the spatial grid is refined. As we can see from Figure 5, the upper bound follows the actual spectral radius rather closely for most values of σ, and its maximum occurs roughly at the same value of σ as the one that maximizes the spectral radius. In the next two results, we will use the estimate (3.36) of the spectral radius of $I - A_{\Delta t}^{-1} A_{\delta t}$ to derive a criterion for the convergence of the method.

LEMMA 3.7. *Let T, ΔT, Δt, δt be fixed. Then for all $\sigma < 0$, we have*

$$\frac{|\delta\gamma|}{\gamma} \le 1.58|\sigma|(\Delta t - \delta t), \qquad \frac{\delta\beta}{1-\beta} \le 0.3. \tag{3.37}$$

Proof. To bound $|\delta\gamma|/\gamma$, we start by bounding a scaled version of the quantity. We first use the definition of γ and $\gamma_{\delta t}$ (cf. (3.17)) to obtain

$$\begin{aligned}
\frac{|\delta\gamma|}{\gamma} \cdot \frac{1}{|\sigma|(\Delta t - \delta t)} &= \frac{\gamma_{\delta t} - \gamma}{\gamma|\sigma|(\Delta t - \delta t)} \\
&= \frac{2 + |\sigma|\Delta t}{(1-\beta^2)|\sigma|(\Delta t - \delta t)} \left(\frac{1-\beta_{\delta t}^2}{2 + |\sigma|\delta t} - \frac{1-\beta^2}{2+|\sigma|\Delta t} \right) \\
&= \frac{1-\beta_{\delta t}^2}{(2+\sigma\delta t)(1-\beta^2)} + \frac{\beta^2 - \beta_{\delta t}^2}{|\sigma|(\Delta t - \delta t)(1-\beta^2)} =: A + B.
\end{aligned}$$

To estimate the terms A and B above, we define the mapping

$$h_{\Delta T}(\tau) := (1 + |\sigma|\tau)^{-\Delta T/\tau},$$

so that $\beta = h_{\Delta T}(\Delta t)$, $\beta_{\delta t} = h_{\Delta T}(\delta t)$. Using the fact that $\ln(1+x) > \frac{x}{1+x}$ for $x > 0$ (see Lemma A.2 in Appendix A), we see that

$$h'_{\Delta T}(\tau) = h_{\Delta T}(\tau) \left[\frac{\Delta T}{\tau^2} \ln(1 + |\sigma|\tau) - \frac{|\sigma|\Delta T}{\tau(1+|\sigma|\tau)} \right] > 0,$$

so $h_{\Delta T}$ is increasing. Therefore, we have

$$\lim_{\tau \to 0} h_{\Delta T}(\tau) = e^{-|\sigma|\Delta T} \le \beta_{\delta t} \le \beta \le \frac{1}{1 + |\sigma|\Delta T} = h_{\Delta T}(\Delta T). \tag{3.38}$$

It then follows that

$$A := \frac{1-\beta_{\delta t}^2}{(2+\sigma\delta t)(1-\beta^2)} \le \frac{1 - e^{-2|\sigma|\Delta T}}{(2+\sigma\delta t)(1 - (1+|\sigma|\Delta T)^{-2})} \le \frac{(1-e^{-2|\sigma|\Delta T})(1+|\sigma|\Delta T)^2}{2|\sigma|\Delta T(2+|\sigma|\Delta T)}.$$

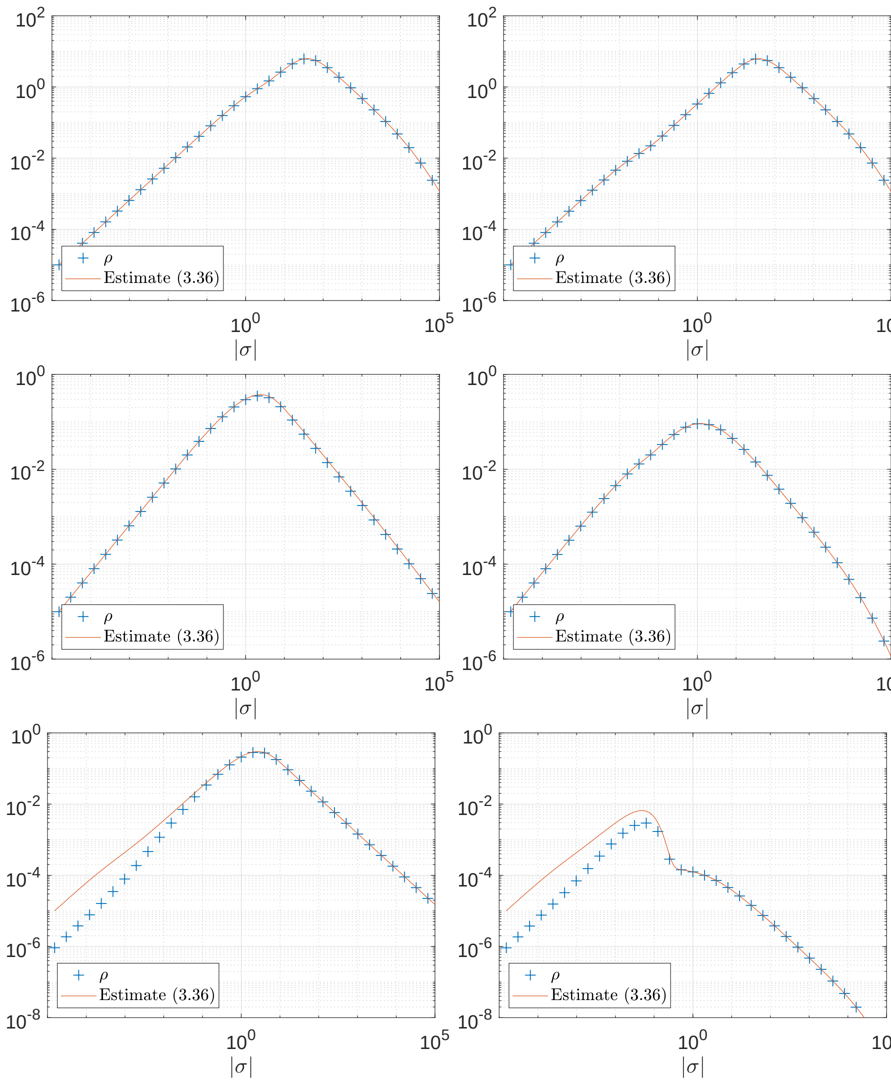

FIG. 5. *Behaviour of* $\mu_{\max}$ *as a function of* σ *for* $\alpha = 0.001, 1, 1000$ *(top to bottom). Left: 150 subintervals, 1 coarse step per subinterval. Right: 3 subintervals, 50 coarse steps per subinterval. All examples use* $T = 100$, $\Delta t = 2/3$ *and* $\Delta t/\delta t = 10^4$.

The last quotient is a function in $|\sigma|\Delta T$ only, whose maximum over all $|\sigma|\Delta T > 0$ is approximately $0.5773 < 0.58$; therefore, we have

$$A \leq 0.58.$$

For the second term, we use the mean value theorem and the fact that $\beta^2 = h_{2\Delta T}(\Delta t)$, $\beta_{\delta t}^2 = h_{2\Delta T}(\delta t)$ to obtain

$$\beta^2 - \beta_{\delta t}^2 = (\Delta t - \delta t) h'_{2\Delta T}(\tau^*)$$

for some $\delta t < \tau^* < \Delta t$, with

$$h'_{2\Delta T}(\tau) = h_{2\Delta T}(\tau) \left[\frac{2\Delta T}{\tau^2} \ln(1 + |\sigma|\tau) - \frac{2|\sigma|\Delta T}{\tau(1 + |\sigma|\tau)} \right].$$

Using the fact that $\ln(1+x) \leq x$ for all $x \geq 0$, we deduce that

$$h'_{2\Delta T}(\tau^*) \leq h_{2\Delta T}(\tau^*)\frac{2|\sigma|^2\Delta T}{1+|\sigma|\tau^*} \leq \frac{2\beta^2|\sigma|^2\Delta T}{1+|\sigma|\delta t},$$

so that

$$B := \frac{\beta^2-\beta_{\delta t}^2}{|\sigma|(\Delta t-\delta t)(1-\beta^2)} \leq \frac{2|\sigma|\Delta T}{(1+|\sigma|\Delta T)^2}\cdot\frac{(1+|\sigma|\Delta T)^2}{|\sigma|\Delta T(2+|\sigma|\Delta T)} \leq 1.$$

Combining the estimates for A and B and multiplying by $|\sigma|(\Delta t-\delta t)$ gives the first inequality in (3.37). For the second inequality, we use (3.38) to obtain

$$\frac{\beta-\beta_{\delta t}}{1-\beta} \leq \frac{(1+|\sigma|\Delta T)^{-1}-e^{-|\sigma|\Delta T}}{1-(1+|\sigma|\Delta T)^{-1}} = \frac{1-(1+|\sigma|\Delta T)e^{-|\sigma|\Delta T}}{|\sigma|\Delta T}.$$

This is again a function in a single variable $|\sigma|\Delta T$, whose maximum over all $|\sigma|\Delta T > 0$ is approximately $0.2984 < 0.3$. ∎

THEOREM 3.8. *Let ΔT, Δt, δt and α be fixed. Then for all $\sigma < 0$, the spectral radius of $I - A_{\Delta t}^{-1}A_{\delta t}$ satisfies*

$$\max_{\sigma<0}\rho(\sigma) \leq \frac{0.79\Delta t}{\alpha+\sqrt{\alpha\Delta t}} + 0.3. \tag{3.39}$$

Thus, if $\alpha > 0.4544\Delta t$, then the linear ParaOpt algorithm (3.19) converges.

Proof. Starting with the spectral radius estimate (3.36), we divide the numerator and denominator by γ, then substitute its definition in (3.17) to obtain

$$\begin{aligned}\rho(\sigma) &< \frac{|\delta\gamma|+\alpha\delta\beta(1+\beta)}{\gamma+\alpha(1-\beta^2)} = \frac{\frac{|\delta\gamma|}{\gamma}+\frac{\delta\beta}{1-\beta}\alpha|\sigma|(2+|\sigma|\Delta t)}{1+\alpha|\sigma|(2+|\sigma|\Delta t)}\\ &\leq \frac{|\delta\gamma|}{\gamma(1+\alpha|\sigma|(2+|\sigma|\Delta t))}+\frac{\delta\beta}{1-\beta}.\end{aligned}$$

Now, by Lemma 3.7, the first term is bounded above by

$$f(\sigma) := \frac{1.58|\sigma|\Delta t}{1+\alpha|\sigma|(2+|\sigma|\Delta t)},$$

whose maximum occurs at $\sigma^* = -1/\sqrt{\alpha\Delta t}$ with

$$f(\sigma^*) = \frac{0.79\Delta t}{\sqrt{\alpha\Delta t}+\alpha}.$$

Together with the estimate on $\delta\beta/(1-\beta)$ in Lemma 3.7, this proves (3.39). Thus, a sufficient condition for the method (3.19) to converge can be obtained by solving the inequality

$$\frac{0.79\Delta t}{\alpha+\sqrt{\alpha\Delta t}} + 0.3 < 1.$$

This is a quadratic equation in $\sqrt{\alpha}$; solving it leads to $\alpha > 0.4544\Delta t$, as required. ∎

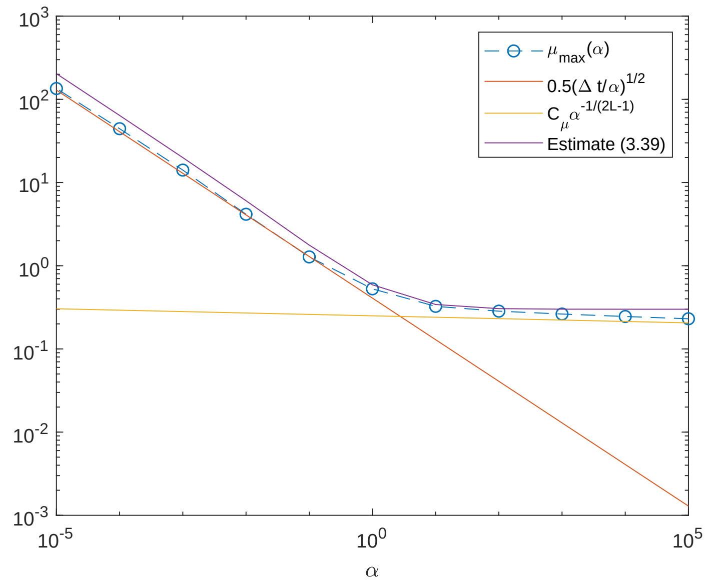

FIG. 6. *Behaviour of* $\max_{\sigma<0}\rho(\sigma)$ *as a function of* α, $T = 100$, $L = 30$, $\Delta T = \Delta t$, $\Delta t/\delta t = 10^{-4}$. *The data for* $\mu_{\max}(\alpha)$ *has been generated by solving the generalized eigenvalue problem* (3.20) *using* `eig` *in* MATLAB.

In Figure 6, we show the maximum spectral radius of $I - A_{\Delta t}^{-1}A_{\delta t}$ over all negative σ for different values of α for a model decomposition with $T = 100$, 30 subintervals, one coarse time step per sub-interval, and a refinement ratio of 10^4 between the coarse and fine grid. We see in this case that the estimate (3.39) is indeed quite accurate.

Remarks.
1. (Dependence on α) Theorem 3.8 states that in order to guarantee convergence, one should make sure that the coarse time step Δt is sufficiently small relative to α. In that case, the method converges.
2. (Weak scalability) Note that the estimate (3.39) depends on the coarse time step Δt, but not explicitly on the number of sub-intervals L. One may then consider weak scalability, i.e. cases where the problem size per processor is fixed[3], under two different regimes: (i) keeping the sub-interval length ΔT and refinement ratios $\Delta T/\Delta t$, $\Delta t/\delta t$ fixed, such that adding subintervals increases the overall time horizon $T = L\Delta T$; and (ii) keeping the time horizon T fixed and refinement ratios $\Delta T/\Delta t$, $\Delta t/\delta t$ fixed, such that adding sub-intervals decreases their length $\Delta T = T/L$. In the first case, Δt remains fixed, so the bound (3.39) remains bounded as $L \to \infty$. In the second case, $\Delta t \to 0$ as $L \to \infty$, so in fact (3.39) decreases to 0.3 as $L \to \infty$. Therefore, the method is weakly scalable *under both regimes*.
3. (Contraction rate for high and low frequencies) Let $\alpha > 0$ be fixed, and let $\rho(\sigma)$ be the spectral radius of $I - A_{\Delta t}^{-1}A_{\delta t}$ as a function of σ given by (3.36). Then for

[3]On the contrary, strong scalability deals with cases where the total problem size is fixed.

$\Delta t/\delta t \geq 2$, an asymptotic expansion shows that we have

$$\rho(\sigma) = \begin{cases} |\sigma|(\Delta t - \delta t) + O(|\sigma|^2) & \text{as } |\sigma| \to 0, \\ \frac{1}{|\sigma|\Delta t} + O(|\sigma|^{-2}) & \text{as } |\sigma| \to \infty \text{ if } \Delta T = \Delta t, \\ \frac{1}{\alpha \delta t}|\sigma|^{-2} + O(|\sigma|^{-3}) & \text{as } |\sigma| \to \infty \text{ if } \Delta T/\Delta t \geq 2. \end{cases}$$

In other words, the method reduces high and low frequency error modes very quickly, and the overall contraction rate is dominated by mid frequencies (where "mid" depends on α, Δt, etc). This is also visible in Figure 5, where ρ attains its maximum at $|\sigma| = O(1/\sqrt{\alpha})$ and decays quickly for both large and small $|\sigma|$.

Finally, we note that for the linear problem, it is possible to use Krylov acceleration to solve for the fixed point of (3.19), even when the spectral radius is greater than 1. However, the goal of this linear analysis is to use it as a tool for studying the asymptotic behaviour of the *non-linear* method (2.11); since a contractive fixed point map must have a Jacobian with spectral radius less than 1 at the fixed point, Theorem 3.8 shows which conditions are sufficient to ensure asymptotic convergence of the non-linear ParaOpt method.

4. Numerical results.

In the previous section, we have presented numerical examples related to the efficiency of our bounds with respect to σ and α. We now study in more detail the quality of our bounds with respect to the discretization parameters. We complete these experiments with a non-linear example and a PDE example.

4.1. Linear scalar ODE: sensitivity with respect to the discretization parameters.

In this part, we consider the case where $\alpha = 1$, $\sigma = -16$ and $T = 1$ and investigate the dependence of the spectral radius of $I - A_{\Delta t}^{-1} A_{\delta t}$ when L, Δt, δt vary.

We start with variations in Δt and δt, and a fixed number of sub-intervals $L = 10$. In this way, we compute the spectral radius of $I - A_{\Delta t}^{-1} A_{\delta t}$ for three cases: first with a fixed $\Delta t = 10^{-4}$ and $\delta t = \frac{\Delta t}{2^k}$, $k = 1, \ldots, 15$; then with a fixed $\delta t = 10^{-2} \cdot 2^{-20}$ and $\Delta t = 2^{-k}$, $k = 0, \ldots, 20$; and finally with a fixed ratio $\frac{\Delta t}{\delta t} = 10^2$ with $\Delta t = 2^k$, $k = 1, \ldots, 15$. The results are shown in Figure 7. In all cases, we observe a very good agreement between the estimate obtained in (3.36) and the true spectral radius. Note that the largest possible Δt for this problem is when Δt equals the length of the sub-interval, i.e., when $\Delta t = \Delta T = 0.1$; for this Δt, the estimates (3.36) and (3.39) are very close to each other, because (3.39) is obtained from (3.36) by making Δt as large as possible, i.e., by letting $\Delta t = \Delta T$.

We next study the scalability properties of ParaOpt. More precisely, we examine the behaviour of the spectral radius of the preconditioned matrix when the number of subintervals L varies. In order to fit with the paradigm of numerical efficiency, we set $\Delta T = \Delta t$ which corresponds somehow to a coarsening limit. We consider two cases: the first case uses a fixed value of T, namely $T = 1$, and the second case uses $T = L\Delta T$ for the fixed value of $\Delta T = 1$. The results are shown in Figure 8. In both cases, we observe perfect scalability of ParaOpt, in the sense that the spectral radius is uniformly bounded with respect to the number of subintervals considered in the time parallelization.

4.2. A nonlinear example.

We now consider a control problem associated with a non-linear vectorial dynamics, namely the *Lotka-Volterra* system. The problem

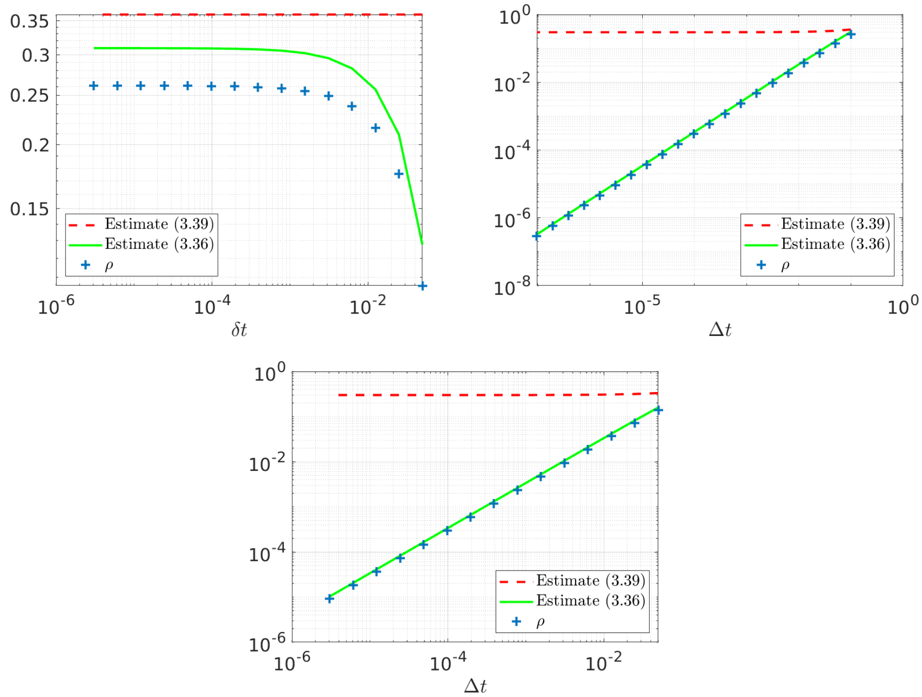

FIG. 7. *Spectral radius of the preconditioned matrix. Top left: varying δt (with fixed Δt), top right: varying Δt (with fixed δt), bottom: varying Δt (with fixed $\frac{\Delta t}{\delta t}$).*

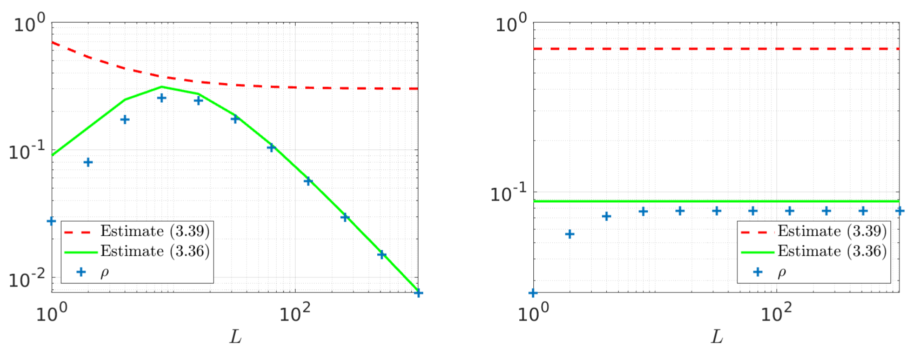

FIG. 8. *Spectral radius of the preconditioned matrix as a function of L. Left: fixed value of T (with $T = 1$), Right: $T = L\Delta T$.*

consists of minimizing the cost functional

$$J(c) = \frac{1}{2}|y(T) - y_{target}|^2 + \frac{\alpha}{2}\int_0^T |c(t)|^2\, dt$$

with $y_{\mathsf{target}} = (100, 20)^T$, subject to the Lotka-Volterra equations

$$\begin{aligned} \dot{y}_1 &= g(y) := a_1 y_1 - b_1 y_1 y_2 + c_1, \\ \dot{y}_2 &= \widetilde{g}(y) := a_2 y_1 y_2 - b_2 y_2 + c_2 \end{aligned} \tag{4.1}$$

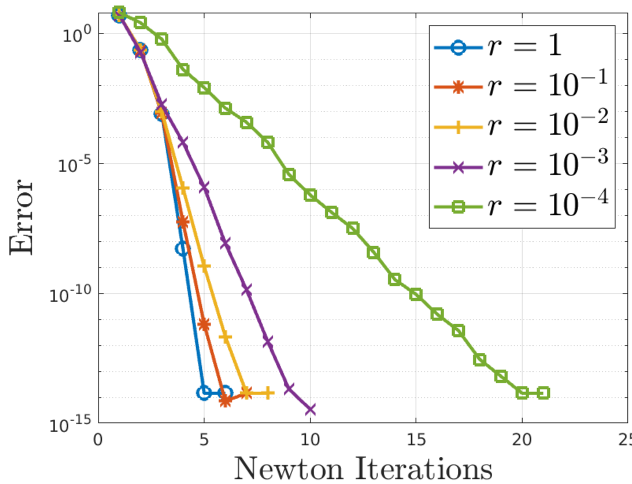

FIG. 9. *L^∞ error as a function of the number of exact ($r = 1$) or inexact ($r < 1$) Newton iterations, for various values of the ratio $r = \frac{\delta t}{\Delta t}$.*

with $a_1 = b_2 = 10$, $b_1 = a_2 = 0.2$ and initial conditions $y(0) = (20, 10)^T$. In this non-linear setting, the computation of each component of $\mathcal{F}(Y, \Lambda)$ for given Y and Λ requires a series of independent iterative inner loops. In our test, these computations are carried out using a Newton method. As in Section 3, the time discretization of (2.2) is performed with an implicit Euler scheme.

In a first test, we set $T = 1/3$ and $\alpha = 5\times10^{-2}$ and fix the fine time discretization step to $\delta t = 10^{-5} \cdot T$. In Figure 9, we show the rate of convergence of ParaOpt for $L = 10$ and various values of the ratio $r = \frac{\delta t}{\Delta t}$. Here, the error is defined as the maximum difference between the interface state and adjoint values obtained from a converged fine-grid solution, and the interface values obtained at each inexact Newton iteration by ParaOpt. As can be expected when using a Newton method, we observe that quadratic convergence is obtained in the case $r = 1$. When r becomes smaller, the preconditioning becomes a coarser approximation of the exact Jacobian, and thus convergence becomes a bit slower.

In our experiments, we observed that the initial guess plays a significant role in the convergence of the method. This follows from the fact that ParaOpt is an exact (if $\Delta t = \delta t$) or approximate (otherwise) Newton method. The initial guess we consider is $c(t) = 1$, $y(T_\ell) = (1 - T_\ell/T)y_0 + T_\ell/T y_{target}$, and $\lambda(T_\ell) = (1, 1)^T$. While for $T = 1/3$ we observe convergence for all L, if we increase T to $T = 1$, we do not observe convergence any more for $L < 10$; in fact, without decomposing the time domain, the sequential version of our solver with $L = 1$ does not converge, even if we use the exact Jacobian without the coarse approximation. This shows that using a time-domain decomposition actually helps in solving the non-linear problem, a phenomenon already observed for a different time-parallelization method in [48]. These convergence problems we observed are also related to the existence of multiple solutions. Indeed, if we coarsen the outer iteration by replacing the Newton iteration with a Gauss-Newton iteration, i.e., by removing the second order derivatives of g and $\widetilde{g}$ in Newton's iterative formula, we obtain another solution, as illustrated in Figure 10 on the left for $T = 1$ and $\Delta t = \delta t$. For both solutions, we observe that the eigenvalues

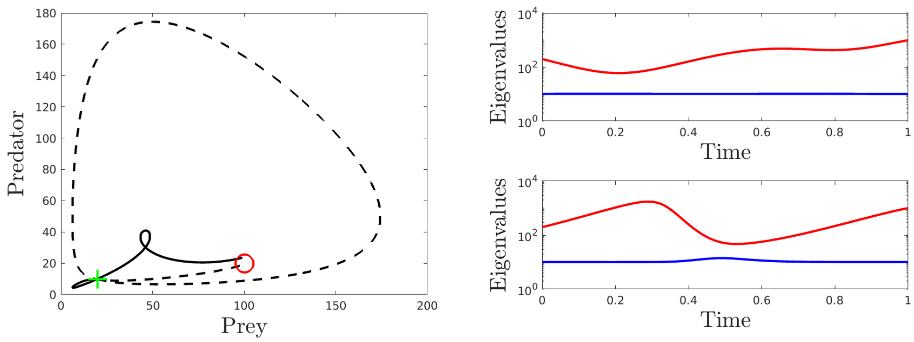

Fig. 10. *Left: two local minima of the cost functional J, obtained with Newton (plain line) and Gauss-Newton (dashed line) in the outer loop, for $T = 1$. The cost functional values are $J \approx 1064.9$ and $J \approx 15.75$. The green cross and the red circle indicate y_0 and y_{target}. Right: eigenvalues associated with the linearized dynamics in a neighborhood of the local minima obtained with Newton (top) and Gauss-Newton (bottom).*

Table 2

Performance of the algorithm (total computing time T_{cpu} and also parallel computing time only, in seconds).

L	Newton Its.	T_{cpu}	Parallel computing time
3	10	183.51	180.71
6	9	82.50	80.38
12	9	44.42	44.19
24	9	24.29	23.83

associated with the linearized dynamics

$$\delta\dot{y}_1 = a_1\delta y_1 - b_1\delta y_1 y_2 - b_1 y_1 \delta y_2 + \delta c_1, \quad \delta\dot{y}_2 = a_2\delta y_1 y_2 + a_2 y_1 \delta y_2 - b_2\delta y_2 + \delta c_2$$

in a neighborhood of the local minima remain strictly positive along the trajectories, in contrast to the situation analyzed in Section 3. Their values are presented in Figure 10 on the right.

We next test the numerical efficiency of our algorithm. corresponds to the last curve of Figure 9, i.e. $T = 1/3$ and $r = 10^{-4}$, except that we use various values of $L \in \{3, 6, 12, 24\}$ using the corresponding number of processors. We execute our code in parallel on workers of a parallel pool, using Matlab's Parallel Processing Toolbox on a 24-core machine that is part of the SciBlade cluster at Hong Kong Baptist University. The results are presented in Table 2, where we also indicate the total parallel computing time without communication, as well as the number of outer Newton iterations required for convergence to a tolerance of 10^{-13}. We observe that our cluster enables us to get very good scalability, the total computing time is roughly divided by two when the number of processors is doubled. To see how this compares with speedup ratios that can be expected from more classical approaches, we run parareal on the initial value problem (4.1) with the same initial conditions and no control, i.e., $c_1 = c_2 = 0$. For $L = 3, 6, 12$ and 24 sub-intervals and a tolerance of 10^{-13}, parareal requires $K = 3, 6, 8$ and 13 iterations to converge. (For a more generous tolerance of 10^{-8}, parareal requires $K = 3, 6, 6$ and 7 iterations.) Since the speedup obtained by parareal cannot exceed L/K, the maximum speedup that can be obtained if parareal is used as a subroutine for forward and backward sweeps does

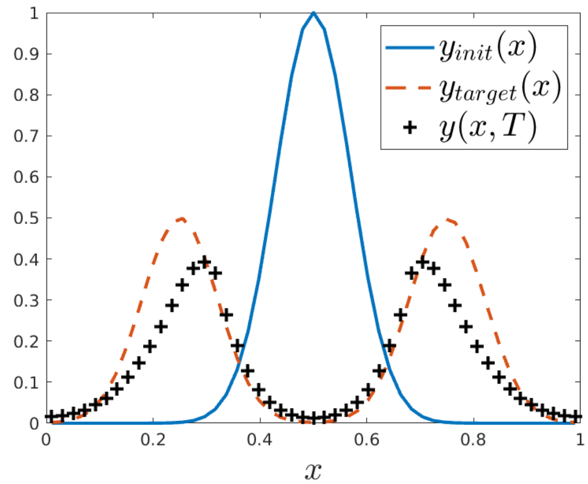

FIG. 11. *Example of initial condition, target state and final state of the solution of the control problem.*

not exceed 4 for our problem.

4.3. A PDE example. We finally consider a control problem involving the heat equation. More precisely, Eq. (2.1) is replaced by

$$\partial_t y - \Delta y = Bc,$$

where the unknown $y = y(x,t)$ is defined on $\Omega = [0,1]$ with periodic boundary conditions, and on $[0,T]$ with $T = 10^{-2}$. Initial and target states are

$$y_{init} = \exp(-100(x-1/2)^2),$$
$$y_{target} = \frac{1}{2}\exp(-100(x-1/4)^2) + \frac{1}{2}\exp(-100(x-3/4)^2).$$

The operator B is the indicator function of a sub-interval Ω_c of Ω; in our case, $\Omega_c = [1/3, 2/3]$. We also set $\alpha = 10^{-4}$. The corresponding solution is shown in Figure 11. We use a finite difference scheme with 50 grid points for the spatial discretization. As in the previous subsection, an implicit Euler scheme is used for the time discretization, and we consider a parallelization involving $L = 10$ subintervals, with $\delta t = 10^{-7}$. We then study the rate of convergence for various values of $r = \frac{\delta t}{\Delta t}$. For $\alpha = 10^{-4}$, the evolution of the error along the iterations is shown in Figure 12. Here, the error is defined as the maximum difference between the iterates and the reference discrete solution, evaluated at sub-interval interfaces. Cases of divergence can also be observed, in particular for $T = 1$ and small values of α and r, as shown in Figure 13.

Of course, one can envisage using different spatial discretizations for the coarse and fine propagators; this may provide additional speedup, provided suitable restriction and prologation operators are used to communicate between the two discretizations. This will be the subject of investigation in a future paper.

5. Conclusions. We introduced a new time-parallel algorithm we call ParaOpt for time-dependent optimal control problems. Instead of applying Parareal to solve

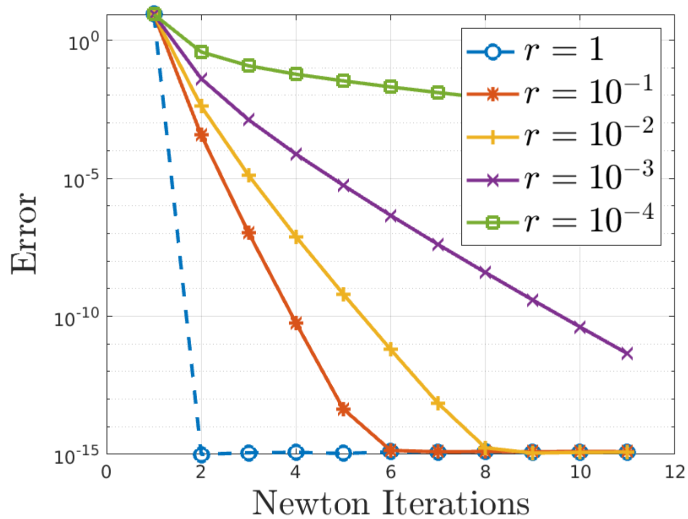

FIG. 12. *Convergence of the method for various values of the ratio* $r = \frac{\delta t}{\Delta t}$.

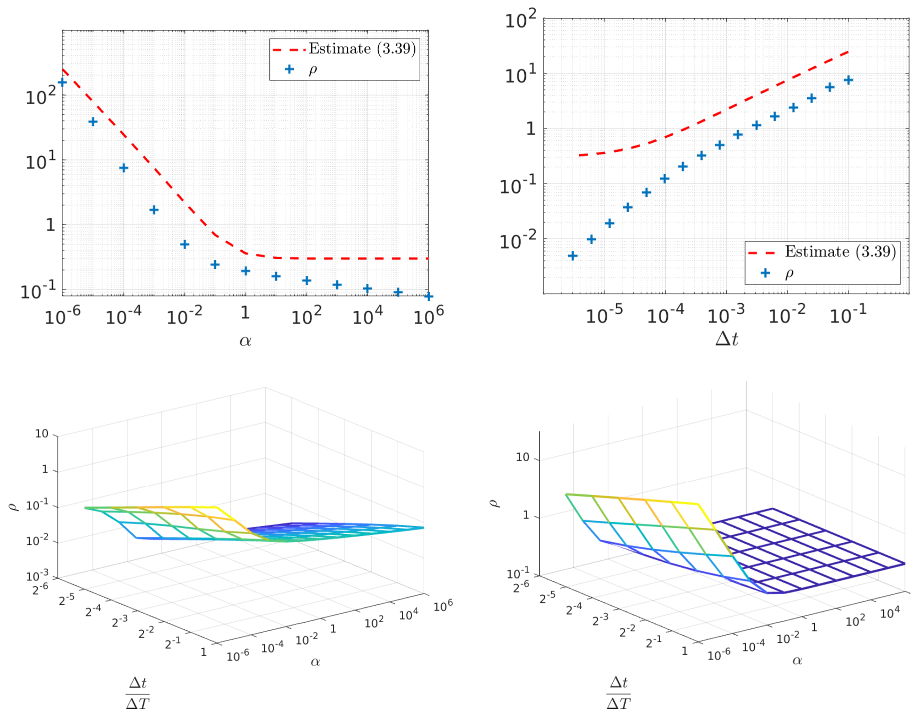

FIG. 13. *Top left: Spectral radius of the preconditioned matrix as a function of* α*, with* $\delta t = 10^{-5}$ *and* $\Delta t = \Delta T = 10^{-1}$*. Top right: Spectral radius of the preconditioned matrix as a function of* $\Delta t/\Delta T$*, with* $\delta t = 10^{-8}$ *and* $\alpha = 10^{-4}$*. Bottom left: Spectral radius of the preconditioned matrix as a function of* α *and* $\Delta t/\Delta T$*, with* $\delta t = 10^{-7}$*. Bottom right: Estimate* (3.39) *as a function of* α *and* Δt*.*

separately the forward and backward equations as they appear in an optimization loop, we propose in ParaOpt to partition the coupled forward-backward problem directly in time, and to use a Parareal-like iteration to incorporate a coarse correction when solving this coupled problem. We analyzed the convergence properties of ParaOpt, and proved in the linear diffusive case that its convergence is independent of the number of sub-intervals in time, and thus scalable. We also tested ParaOpt on scalar linear optimal control problems, a non-linear optimal control problem involving the Lotka-Volterra system, and also on a control problem governed by the heat equation. A small scale parallel implementation of the Lotka-Volterra case also showed scalability of ParaOpt for this non-linear problem.

Our ongoing work consists of analyzing the algorithm for non-dissipative problems. Also, for problems with large state spaces, e.g., for discretized PDEs in three spatial dimensions, the approximate Jacobian $\mathcal{J}^G$ in (2.12) may become too large to solve by direct methods; we are currently working on designing efficient preconditioners for solving such systems iteratively. Finally, we are currently studying ParaOpt by applying it to realistic problems from applications, in order to better understand its behaviour in such complex cases.

Acknowledgments. The authors acknowledge support from ANR Ciné-Para (ANR-15-CE23-0019), ANR ALLOWAP (ANR-19-CE46-0013/A-HKBU203/19), the Swiss National Science Foundation grant no. 200020_178752, and the Hong Kong Research Grants Council (ECS 22300115 and GRF 12301817). We also thank the anonymous referees for their valuable suggestions, which greatly improved our paper.

Appendix A. Proof of Inequality (3.34). Our goal is to prove the following lemma, which is needed for the proof of Lemma 3.3.

LEMMA A.1. *For every $k > 0$ and $0 < x \leq k$, we have*

$$(1+x)^{k/x} > k\left(\frac{2+x}{1+x}\right) - 1. \tag{A.1}$$

First, we need the following property of logarithmic functions.

LEMMA A.2. *For any $x > 0$, we have*

$$\ln(1+x) \geq \frac{x}{x+1} + \frac{1}{2}\left(\frac{x}{x+1}\right)^2.$$

Proof. Let $u = \frac{x}{x+1} < 1$. Then

$$\begin{aligned}\ln(1+x) &= -\ln\left(\frac{1}{1+x}\right) = -\ln(1-u)\\ &= u + \frac{u^2}{2} + \frac{u^3}{3} + \cdots \geq u + \frac{u^2}{2}.\end{aligned}$$

The conclusion now follows. ∎

Proof. (Lemma A.1) Let g and h denote the left and right hand sides of (A.1) respectively. We consider two cases, namely when $0 < k \leq 1$ and when $k > 1$. When $k \leq 1$, we have

$$h(x) \leq \frac{2+x}{1+x} - 1 = \frac{1}{1+x} < 1 < (1+x)^{k/x} = g(x).$$

For the case $k > 1$, we will show that $g(k) > h(k)$ and $g'(x) - h'(x) < 0$ for $0 < x < k$, which together imply that $g(x) > h(x)$ for all $0 < x \leq k$. The first assertion follows from the fact that

$$g(k) - h(k) = 1 + k - k \cdot \frac{k+2}{k+1} + 1 = 2 - \frac{k}{k+1} > 0.$$

To prove the second part, we note that

$$\begin{aligned} g'(x) &= (1+x)^{k/x} \left[-\frac{k}{x^2} \ln(1+x) + \frac{k}{x(1+x)} \right] \\ &= \frac{-k}{x^2}(1+x)^{k/x-1} \left[(1+x)\ln(1+x) - x \right] \\ &< \frac{-k}{x^2}(1+x)^{k/x-1} \cdot \frac{x^2}{2(x+1)} = \frac{-k}{2}(1+x)^{k/x-2} < 0, \\ h'(x) &= -\frac{k}{(1+x)^2} < 0. \end{aligned}$$

Therefore, we have

$$g'(x) - h'(x) < -\frac{k}{(1+x)^2} \left[\frac{1}{2}(1+x)^{k/x} - 1 \right] \leq -\frac{k}{(1+x)^2} \underbrace{\left[\frac{1+k}{2} - 1 \right]}_{> 0 \text{ since } k > 1} < 0.$$

Thus, $g(x) > h(x)$ for all $0 < x < k$, as required. ∎

REFERENCES

[1] M. D. Al-Khaleel, M. J. Gander, and A. E. Ruehli, *Optimization of transmission conditions in waveform relaxation techniques for RC circuits*, SIAM Journal on Numerical Analysis, 52 (2014), pp. 1076–1101.

[2] A. Bellen and M. Zennaro, *Parallel algorithms for initial-value problems for difference and differential equations*, J. Comput. Appl. Math., 25 (1989), pp. 341–350.

[3] H. Bock and K. Plitt, *A multiple shooting algorithm for direct solution of optimal control problems**, IFAC Proceedings Volumes, 17 (1984), pp. 1603 – 1608. 9th IFAC World Congress: A Bridge Between Control Science and Technology, Budapest, Hungary, 2-6 July 1984.

[4] V. Dobrev, T. Kolev, N. A. Petersson, and J. B. Schroder, *Two-level convergence theory for multigrid reduction in time (MGRIT)*, SIAM Journal on Scientific Computing, 39 (2017), pp. S501–S527.

[5] J. Dongarra, P. Beckman, T. Moore, P. Aerts, G. Aloisio, J.-C. Andre, D. Barkai, J.-Y. Berthou, T. Boku, B. Braunschweig, F. Cappello, B. Chapman, X. Chi, A. Choudhary, S. Dosanjh, T. Dunning, S. Fiore, A. Geist, B. Gropp, R. Harrison, M. Hereld, M. Heroux, A. Hoisie, K. Hotta, Z. Jin, Y. Ishikawa, F. Johnson, S. Kale, R. Kenway, D. Keyes, B. Kramer, J. Labarta, A. Lichnewsky, T. Lippert, B. Lucas, B. Maccabe, S. Matsuoka, P. Messina, P. Michielse, B. Mohr, M. S. Mueller, W. E. Nagel, H. Nakashima, M. E. Papka, D. Reed, M. Sato, E. Seidel, J. Shalf, D. Skinner, M. Snir, T. Sterling, R. Stevens, F. Streitz, B. Sugar, S. Sumimoto, W. Tang, J. Taylor, R. Thakur, A. Trefethen, M. Valero, A. van der Steen, J. Vetter, P. Williams, R. Wisniewski, and K. Yelick, *The international exascale software project roadmap*, International Journal of High Performance Computing Applications, 25 (2011), pp. 3–60.

[6] M. Emmett and M. L. Minion, *Toward an efficient parallel in time method for partial differential equations*, Comm. App. Math. and Comp. Sci, 7 (2012), pp. 105–132.

[7] R. Falgout, S. Friedhoff, T. V. Kolev, S. MacLachlan, and J. B. Schroder, *Parallel time integration with multigrid*, SIAM Journal on Scientific Computing, 36 (2014), pp. C635–C661.

[8] M. J. Gander, *Overlapping Schwarz for linear and nonlinear parabolic problems*, in Proceedings of the 9th International Conference on Domain Decomposition, ddm.org, 1996, pp. 97–104.

[9] M. J. Gander, *50 years of time parallel time integration*, in Multiple Shooting and Time Domain Decomposition Methods, Springer, 2015, pp. 69–113.

[10] M. J. Gander and S. Güttel, *Paraexp: A parallel integrator for linear initial-value problems*, SIAM Journal on Scientific Computing, 35 (2013), pp. C123–C142.

[11] M. J. Gander and E. Hairer, *Nonlinear convergence analysis for the parareal algorithm*, in Domain Decomposition Methods in Science and Engineering XVII, O. B. Widlund and D. E. Keyes, eds., vol. 60 of Lecture Notes in Computational Science and Engineering, Springer, 2008, pp. 45–56.

[12] M. J. Gander and E. Hairer, *Analysis for parareal algorithms applied to Hamiltonian differential equations*, Journal of Computational and Applied Mathematics, 259 (2014), pp. 2–13.

[13] M. J. Gander and L. Halpern, *Absorbing boundary conditions for the wave equation and parallel computing*, Math. of Comp., 74 (2004), pp. 153–176.

[14] ———, *Optimized Schwarz waveform relaxation methods for advection reaction diffusion problems*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 666–697.

[15] M. J. Gander, F. Kwok, and B. Mandal, *Dirichlet-Neumann and Neumann-Neumann waveform relaxation algorithms for parabolic problems*, ETNA, 45 (2016), pp. 424–456.

[16] M. J. Gander, F. Kwok, and G. Wanner, *Constrained optimization: From Lagrangian mechanics to optimal control and PDE constraints*, in Optimization with PDE Constraints, Springer, 2014, pp. 151–202.

[17] M. J. Gander and M. Neumüller, *Analysis of a new space-time parallel multigrid algorithm for parabolic problems*, SIAM Journal on Scientific Computing, 38 (2016), pp. A2173–A2208.

[18] M. J. Gander and M. Petcu, *Analysis of a Krylov subspace enhanced parareal algorithm for linear problems*, in ESAIM: Proceedings, vol. 25, EDP Sciences, 2008, pp. 114–129.

[19] M. J. Gander and A. M. Stuart, *Space-time continuous analysis of waveform relaxation for the heat equation*, SIAM Journal on Scientific Computing, 19 (1998), pp. 2014–2031.

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

[21] E. Giladi and H. B. Keller, *Space time domain decomposition for parabolic problems*, Numerische Mathematik, 93 (2002), pp. 279–313.

[22] R. Glowinski and J. Lions, *Exact and approximate controllability for distributed parameter systems*, Acta numerica, 3 (1994), pp. 269–378.

[23] S. Götschel and M. L. Minion, *Parallel-in-time for parabolic optimal control problems using PFASST*, in Domain Decomposition Methods in Science and Engineering XXIV, Springer, 2018, pp. 363–371.

[24] S. Götschel and M. L. Minion, *An efficient parallel-in-time method for optimization with parabolic PDEs*, SIAM Journal on Scientific Computing, 41 (2019), pp. C603–C626.

[25] S. Günther, N. R. Gauger, and J. B. Schroder, *A non-intrusive parallel-in-time adjoint solver with the XBraid library*, Computing and Visualization in Science, 19 (2018), pp. 85–95.

[26] ———, *A non-intrusive parallel-in-time approach for simultaneous optimization with unsteady PDEs*, Optimization Methods and Software, 34 (2019), pp. 1306–1321.

[27] W. Hackbusch, *Parabolic multi-grid methods*, in Computing Methods in Applied Sciences and Engineering, VI, R. Glowinski and J.-L. Lions, eds., North-Holland, 1984, pp. 189–197.

[28] G. Horton and S. Vandewalle, *A space-time multigrid method for parabolic partial differential equations*, SIAM Journal on Scientific Computing, 16 (1995), pp. 848–864.

[29] K. Ito and K. Kunisch, *Lagrange multiplier approach to variational problems and applications*, vol. 15, Siam, 2008.

[30] M. Kiehl, *Parallel multiple shooting for the solution of initial value problems*, Parallel computing, 20 (1994), pp. 275–295.

[31] F. Kwok, *Neumann-Neumann waveform relaxation for the time-dependent heat equation*, in Domain decomposition methods in science and engineering, DD21, Springer, 2014.

[32] E. Lelarasmee, A. E. Ruehli, and A. L. Sangiovanni-Vincentelli, *The waveform relaxation method for time-domain analysis of large scale integrated circuits*, IEEE Trans. on CAD of IC and Syst., 1 (1982), pp. 131–145.

[33] J.-L. Lions, Y. Maday, and G. Turinici, *Résolution d'edp par un schéma en temps 'pararéel'*, Comptes Rendus de l'Académie des Sciences-Series I-Mathematics, 332 (2001), pp. 661–668.

[34] C. Lubich and A. Ostermann, *Multi-grid dynamic iteration for parabolic equations*, BIT, 27

(1987), pp. 216–234.

[35] Y. Maday and E. M. Rønquist, *Parallelization in time through tensor-product space–time solvers*, Comptes Rendus Mathematique, 346 (2008), pp. 113–118.

[36] Y. Maday, J. Salomon, and G. Turinici, *Monotonic parareal control for quantum systems*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 2468–2482.

[37] Y. Maday and G. Turinici, *A parareal in time procedure for the control of partial differential equations*, Comptes Rendus Mathematique, 335 (2002), pp. 387 – 392.

[38] Y. Maday and G. Turinici, *Parallel in time algorithms for quantum control: Parareal time discretization scheme*, International Journal of Quantum Chemistry, 93 (2003), pp. 223–228.

[39] B. Mandal, *A time-dependent Dirichlet-Neumann method for the heat equation*, in Domain decomposition methods in science and engineering, DD21, Springer, 2014.

[40] M. L. Minion, *A hybrid parareal spectral deferred corrections method*, Communications in Applied Mathematics and Computational Science, 5 (2010), pp. 265–301.

[41] M. L. Minion, R. Speck, M. Bolten, M. Emmett, and D. Ruprecht, *Interweaving PFASST and parallel multigrid*, SIAM journal on scientific computing, 37 (2015), pp. S244–S263.

[42] W. L. Miranker and W. Liniger, *Parallel methods for the numerical integration of ordinary differential equations*, Math. Comp., 91 (1967), pp. 303–320.

[43] D. D. Morrison, J. D. Riley, and J. F. Zancanaro, *Multiple shooting method for two-point boundary value problems*, Commun. ACM, 5 (1962), p. 613614.

[44] J. Nievergelt, *Parallel methods for integrating ordinary differential equations*, Comm. ACM, 7 (1964), pp. 731–733.

[45] B. W. Ong and J. B. Schroder, *Applications of time parallelization*, Computing and Visualization in Science, submitted, (2019).

[46] J. W. Pearson, M. Stoll, and A. J. Wathen, *Regularization-robust preconditioners for time-dependent pde-constrained optimization problems*, SIAM Journal on Matrix Analysis and Applications, 33 (2012), pp. 1126–1152.

[47] M. K. Riahi, J. Salomon, S. J. Glaser, and D. Sugny, *Fully efficient time-parallelized quantum optimal control algorithm*, PHYSICAL REVIEW A, 93 (2016).

[48] ———, *Fully efficient time-parallelized quantum optimal control algorithm*, PHYSICAL REVIEW A, 93 (2016).

[49] D. Sheen, I. H. Sloan, and V. Thomée, *A parallel method for time discretization of parabolic equations based on Laplace transformation and quadrature*, IMA Journal of Numerical Analysis, 23 (2003), pp. 269–299.

[50] V. Thomée, *A high order parallel method for time discretization of parabolic type equations based on Laplace transformation and quadrature*, Int. J. Numer. Anal. Model, 2 (2005), pp. 121–139.

[51] F. Tröltzsch, *Optimal control of partial differential equations: theory, methods, and applications*, vol. 112, American Mathematical Soc., 2010.

[52] S. Ulbrich, *Preconditioners based on "parareal" time-domain decomposition for time-dependent PDE-constrained optimization*, in Multiple Shooting and Time Domain Decomposition Methods, Springer, 2015, pp. 203–232.

[53] S. Vandewalle and E. Van de Velde, *Space-time concurrent multigrid waveform relaxation*, Annals of Numer. Math, 1 (1994), pp. 347–363.