



HAL
open science

A Parallel in Time Algorithm Based on ParaExp for Optimal Control Problems

Felix Kwok, Djahou N Tognon

► **To cite this version:**

Felix Kwok, Djahou N Tognon. A Parallel in Time Algorithm Based on ParaExp for Optimal Control Problems. 63 rd IEEE Conference on Decision and Control, Dec 2024, Milan (Italie), Italy. hal-04612018v3

HAL Id: hal-04612018

<https://inria.hal.science/hal-04612018v3>

Submitted on 5 Sep 2024

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.



Distributed under a Creative Commons Attribution 4.0 International License

A Parallel in Time Algorithm Based on ParaExp for Optimal Control Problems

Felix Kwok and Djahou N. Tognon

Abstract—We propose a new parallel-in-time algorithm for solving optimal control problems constrained by discretized partial differential equations. Our approach, which is based on a deeper understanding of ParaExp, considers an overlapping time-domain decomposition in which we combine the solution of homogeneous problems using exponential propagation with the local solutions of inhomogeneous problems. The algorithm yields a linear system whose matrix-vector product can be fully performed in parallel. We then propose a preconditioner to speed up the convergence of GMRES in the special cases of the heat and wave equations. Numerical experiments are provided to illustrate the efficiency of our preconditioners.

I. INTRODUCTION

The efficient solution of optimal control problems constrained by time-dependent partial differential equations (PDEs) has received much interest in recent decades, given its importance in applications and the increasing availability of powerful modern computing clusters. The main computational challenges lie in the large amounts of data that must be processed, the intensive computation required to simulate the underlying PDEs, and the optimization procedure needed to determine the best control function. Parallel numerical algorithms, specifically those of the domain decomposition type, are well suited for tackling these huge problems by subdividing them into manageable chunks that can be handled simultaneously by many processors. Parallelism also has the obvious advantage of reducing the wall-clock time required to find the optimal solution, which is especially important for real-time applications.

There exists a rich literature on spatial domain decomposition methods for elliptic and initial value problems, see [18], [7], [12], [13] and the references therein. There is also a growing literature on parallel-in-time methods, in which multiple processors solve the initial value problem on different parts of the time interval simultaneously; see [5], [16] for a survey of these methods. Time parallelization can also be applied to optimal control problems, either by decomposing the optimization problem directly, see [10], [14], [1], [8], or by using a parallel-in-time method to solve the forward and adjoint problems required by gradient computations [20].

In this paper, we focus on the efficient solution of linear-quadratic optimal control problems when a cheap exponential integrator is available, i.e., when one is able to evaluate

$\exp(t\mathcal{L})$ quickly for a matrix \mathcal{L} arising from spatial discretization. For initial value problems, the authors of [6] have used such integrators to construct a parallel-in-time method known as ParaExp. In practice, many different methods can be used to evaluate $\exp(t\mathcal{L})$, and the choice depends on the particular properties of the operator \mathcal{L} . In this paper, we show how similar ideas can be used to solve optimal control problems; we will focus on the particular cases of the heat and wave equations, where specific approximations are used to construct the preconditioner for solving for the optimal adjoint state. Numerical experiments are provided to illustrate the behavior of these preconditioners.

II. LINEAR OPTIMAL CONTROL PROBLEM

We consider the optimal control problem constrained by a system of ordinary differential equations (ODEs)

$$\min_{\mathbf{v}} \frac{1}{2} \|y(T) - y_{tg}\|^2 + \frac{\alpha}{2} \int_0^T \|\mathbf{v}(t)\|^2 dt,$$

$$\text{subject to } \dot{y}(t) = \mathcal{L}y(t) + \mathbf{v}(t), y(0) = y_{in}, t \in (0, T),$$

where the constraint arises from a semi-discretization in space of a time-dependent PDE. Here, α is a regularization parameter and y_{in}, y_{tg} are the initial and target states respectively. The state function y and the control \mathbf{v} are mappings from $(0, T)$ to \mathbb{R}^r , r a positive integer. The operator $\mathcal{L} \in \mathbb{R}^{r \times r}$ is independent of the time variable. Using the Lagrange multiplier approach, one can characterize the optimal trajectory $y(t)$ via the forward and adjoint problems

$$\begin{cases} \dot{y} = \mathcal{L}y + \mathbf{v} & \text{on } (0, T), \\ y(0) = y_{in}, \end{cases} \quad \begin{cases} \dot{\lambda} = -\mathcal{L}^T \lambda & \text{on } (0, T) \\ \lambda(T) = y(T) - y_{tg}, \end{cases}$$

where the control \mathbf{v} and adjoint state λ are related by the algebraic equation $\lambda(t) = \alpha \mathbf{v}(t)$ for all $t \in (0, T)$. Eliminating \mathbf{v} , the above system can also be written as

$$\dot{y} = \mathcal{L}y - \frac{1}{\alpha} \lambda, \quad \dot{\lambda} = -\mathcal{L}^T \lambda \quad \text{on } (0, T) \quad (1)$$

with the initial and final condition $y(0) = y_{in}$ and $\lambda(T) = y(T) - y_{tg}$ respectively.

III. PARALLEL-IN-TIME ALGORITHM

For a positive integer L , we consider the non-overlapping sub-intervals $(T_{\ell-1}, T_{\ell}), \ell = 1, \dots, L$ of $(0, T)$ with $T_{\ell} = \ell \Delta T$ and $\Delta T = T/L$. We define two sets of intermediate states $(Y_{\ell})_{\ell=1, \dots, L}$ and $(\Lambda_{\ell})_{\ell=1, \dots, L}$, corresponding to the state y and the adjoint λ at times T_1, \dots, T_L . We now introduce a new parallelization idea for solving (1) inspired by the ParaExp algorithm [6], namely to decompose the ODEs into

F. Kwok, Université Laval, Département de mathématiques et de statistique, Canada felix.kwok@mat.ulaval.ca

D. N. Tognon, INRIA Paris : ANGE Project-Team, and Sorbonne Université: Laboratoire Jacques-Louis Lions (LJLL), France djahou-norbert.tognon@inria.fr

homogeneous and inhomogeneous parts. First, we consider the homogeneous sub-problems on λ , which require the backward propagation of λ over each $(T_{\ell-1}, T_L)$:

$$\dot{\lambda}_\ell(t) = -\mathcal{L}^T \lambda_\ell(t), \quad \lambda_\ell(T_L) = \Lambda_L, \quad \text{on } (T_{\ell-1}, T_L), \quad (2)$$

Next, we define the inhomogeneous part of the problem on y , given by

$$\dot{w}_\ell(t) = \mathcal{L} w_\ell(t) - \frac{1}{\alpha} \lambda_\ell(t), \quad w_\ell(T_{\ell-1}) = 0, \quad \text{on } (T_{\ell-1}, T_L), \quad (3)$$

for $\ell = 1, \dots, L$. Note that the w_ℓ are solved with no initial conditions; they are instead carried by the homogeneous sub-problems u_ℓ defined below, which perform the forward propagation of y over $(T_{\ell-1}, T_L)$:

$$\begin{aligned} \dot{u}_1(t) &= \mathcal{L} u_1(t), \quad u_1(T_0) = y_{in}, \quad \text{on } (T_0, T_L) \\ \dot{u}_\ell(t) &= \mathcal{L} u_\ell(t), \quad u_\ell(T_{\ell-1}) = w_{\ell-1}(T_{\ell-1}), \quad \text{on } (T_{\ell-1}, T_L), \end{aligned} \quad (4)$$

for $\ell = 2, \dots, L$. Finally, $y(T_\ell)$ is obtained by superposition:

$$y(T_\ell) = w_\ell(T_\ell) + \sum_{j=1}^{\ell} u_j(T_\ell), \quad \ell = 1, \dots, L. \quad (5)$$

Figure 1 summarizes the dependency between λ_ℓ , w_ℓ and u_ℓ . Thus, for L available processors, it is natural to assign λ_ℓ , w_ℓ and $u_{\ell+1}$ to the same processor for $\ell = 1, \dots, L-1$ and λ_L , w_L and u_1 to the L^{th} processor. Note that to calculate $y(T_\ell)$, the processors only need to exchange the values of u and w at the interfaces T_ℓ , rather than whole trajectories.

Once the $Y_\ell = y(T_\ell)$ and $\Lambda_{\ell+1} = \lambda(T_{\ell+1})$ are found, the trajectories $y(t)$ and $\lambda(t)$ on $t \in [T_\ell, T_{\ell+1}]$ can easily be constructed in parallel using (1).

Let \mathcal{P}_ℓ and \mathcal{Q}_ℓ be the solution operators for u_ℓ and λ_ℓ defined by the homogeneous sub-problems (4) and (2). (Even though \mathcal{L} is identical on each interval, we write the index ℓ explicitly to indicate the sub-interval on which the propagation is performed.) In other words, the solution of (2) is given by

$$\lambda_\ell(t) = \mathcal{Q}_\ell(t) \cdot \Lambda_L, \quad \text{on } (T_{\ell-1}, T_L), \quad (6)$$

and the one of (4) is given by

$$\begin{aligned} u_1(t) &= \mathcal{P}_1(t) \cdot y_{in}, \quad \text{on } (T_0, T_L) \\ u_\ell(t) &= \mathcal{P}_\ell(t) \cdot w_{\ell-1}(T_{\ell-1}), \quad \text{on } (T_{\ell-1}, T_L). \end{aligned} \quad (7)$$

Thus, substituting (6) into (3) on $(T_{\ell-1}, T_\ell)$, we get

$$\dot{w}_\ell(t) = \mathcal{L} w_\ell(t) - \frac{1}{\alpha} \mathcal{Q}_\ell(t) \cdot \Lambda_L, \quad w_\ell(T_{\ell-1}) = 0. \quad (8)$$

Since $w_\ell(T_{\ell-1}) = 0$, the above implies

$$w_\ell(t) = -\frac{1}{\alpha} \left[\int_{T_{\ell-1}}^t e^{(t-s)\mathcal{L}} \mathcal{Q}_\ell(s) ds \right] \cdot \Lambda_L.$$

Let \mathcal{R}_ℓ denote the solution operator of w_ℓ on $(T_{\ell-1}, T_\ell)$, such that

$$w_\ell(t) = -\frac{1}{\alpha} \mathcal{R}_\ell(t) \cdot \Lambda_L. \quad (9)$$

Therefore, substituting (9) into (7), we obtain

$$\begin{aligned} u_1(t) &= \mathcal{P}_1(t) \cdot y_{in} \quad \text{on } (T_0, T_L), \\ u_\ell(t) &= -\frac{1}{\alpha} \mathcal{P}_\ell(t) \cdot \mathcal{R}_{\ell-1}(T_{\ell-1}) \cdot \Lambda_L, \quad \text{on } (T_{\ell-1}, T_L), \end{aligned} \quad (10)$$

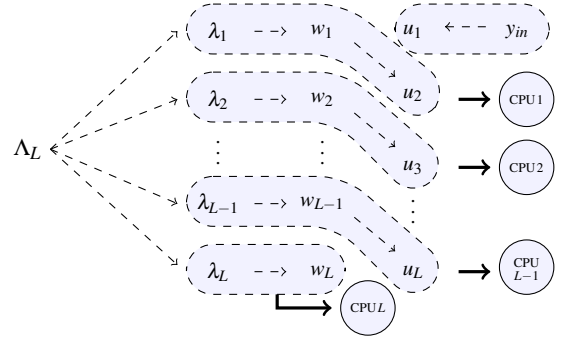


Fig. 1. Data dependency and processor assignment for the parallel computation of $\mathcal{M} \cdot \Lambda_L$. Note that u_1 is independent of Λ_L and is calculated only once when forming the right-hand side \mathbf{b} in (13).

and inserting (9) and (10) into (5) yields

$$\begin{aligned} y(T_\ell) &= \mathcal{P}_1(T_\ell) \cdot y_{in} - \frac{1}{\alpha} \mathcal{R}_\ell(T_\ell) \cdot \Lambda_L \\ &\quad - \frac{1}{\alpha} \sum_{j=2}^{\ell} \mathcal{P}_j(T_\ell) \cdot \mathcal{R}_{j-1}(T_{j-1}) \cdot \Lambda_L, \quad \text{on } (T_{\ell-1}, T_\ell). \end{aligned}$$

Hence, the optimality system (1) in terms of $Y_\ell = y(T_\ell)$ and $\Lambda_\ell = \lambda(T_\ell)$ becomes

$$\begin{aligned} Y_\ell &= \mathcal{P}_1(T_\ell) \cdot y_{in} - \frac{1}{\alpha} \mathcal{R}_\ell(T_\ell) \cdot \Lambda_L \\ &\quad - \frac{1}{\alpha} \sum_{j=2}^{\ell} \mathcal{P}_j(T_\ell) \cdot \mathcal{R}_{j-1}(T_{j-1}) \cdot \Lambda_L, \quad \ell = 1, \dots, L, \end{aligned} \quad (11)$$

and $\Lambda_\ell = \mathcal{Q}_\ell(T_\ell) \cdot \Lambda_L$, $\ell = 1, \dots, L-1$. Now that we expressed the Y_ℓ and Λ_ℓ in terms of a single unknown Λ_L , we only need to impose one last equation to close the system: we use the final condition in (1), given by

$$\Lambda_L - Y_L + y_{ig} = 0. \quad (12)$$

Substituting (11) for $\ell = L$ into the above, we obtain

$$\mathcal{M} \cdot \Lambda_L + \mathbf{b} = 0, \quad (13)$$

where $\mathbf{b} := y_{ig} - \mathcal{P}_1(T_L) \cdot y_{in}$ and

$$\mathcal{M} := I + \frac{1}{\alpha} \mathcal{R}_L(T_L) + \frac{1}{\alpha} \sum_{j=2}^L \mathcal{P}_j(T_L) \cdot \mathcal{R}_{j-1}(T_{j-1}).$$

with I being the identity matrix. Just like for (5), the matrix-vector product $\mathcal{M} \cdot \Lambda_L$ can be computed fully in parallel: each term of the form $\mathcal{P}_j(T_L) \cdot \mathcal{R}_{j-1}(T_{j-1}) \cdot \Lambda_L$ in the sum above can be performed on the $(j-1)^{\text{th}}$ processor for $j = 2, \dots, L$, while the term $\mathcal{R}_L(T_L) \cdot \Lambda_L$ can be handled by the L^{th} processor.

In practice, the solution operators \mathcal{P}_ℓ and \mathcal{R}_ℓ are matrix exponentials that need to be approximated numerically. Computing this approximation efficiently is a non-trivial task: for small and medium-sized dense matrices ($r \leq 1000$), various methods can be found in [15], [11] and [9]. For large sparse matrices, however, the task frequently requires an understanding of the spectral properties of the underlying operator \mathcal{L} . One possibility proposed by [6] is projection-based methods. There, the authors claim that “we clearly

find that it is beneficial to use the Arnoldi method rather than a time-stepping method for the propagation of the linear homogeneous problems. Note that the difference between the projection and time-stepping methods gets large for high accuracy," meaning that the Arnoldi method can produce accurate approximations efficiently.

IV. PRECONDITIONER DESIGN

It remains to solve the linear system (13) using an iterative method, e.g. GMRES. Since the matrix \mathcal{M} is generally ill-conditioned, a preconditioner is needed, which we now construct for the semi-discretized heat and wave equations.

A. Heat equation

In this section, we propose a preconditioner $\widehat{\mathcal{M}}^{-1}$ for the heat equation to speed up the convergence of (13) in GMRES. Our approach is inspired by the work in the Master's thesis [19], where the author derived a preconditioner based on the behavior of \mathcal{L} at low and high frequencies. Let us consider the heat equation given by

$$\dot{y} = \Delta y + v, \text{ on } \Omega \times (0, T), \quad (14)$$

with $y(x, 0) = y_{in}(x)$ on Ω . We consider the 1D case where $\Omega = (0, 1)$ with Dirichlet boundary conditions $y(0, t) = y(1, t) = 0, t \in (0, T)$ with $T \geq 1$. Then, a semi-discretization in space of (14) using second-order centered finite differences leads to the following ODE system

$$\dot{y}(t) = \mathcal{L}y(t) + v(t), y(0) = y_{in}, t \in (0, T), \quad (15)$$

where $y(t), y_0, v(t) \in \mathbb{R}^r$, with r being the number of unknowns in space and \mathcal{L} being the discretization of Δ in space. Since $\mathcal{L} = \mathcal{L}^T$, the optimality system (1) becomes

$$\dot{y} = \mathcal{L}y - \frac{1}{\alpha}\lambda, \quad \dot{\lambda} = -\mathcal{L}\lambda, \quad (16)$$

with $y(0) = y_{in}$ and $\lambda(T) = y(T) - y_{tg}$.

To obtain an effective preconditioner, we need to study the eigenvalue properties of the matrix \mathcal{M} in (13). To do so, we consider the eigenvalue decomposition $\mathcal{L} = \mathcal{U}\mathcal{D}\mathcal{U}^T$, where \mathcal{D} is a diagonal matrix with the eigenvalues of \mathcal{L} and \mathcal{U} a unitary matrix. The transformation $y(t) \mapsto \mathcal{U}^T \cdot y(t)$ and $\lambda(t) \mapsto \mathcal{U}^T \cdot \lambda(t)$ allows us to diagonalize (16) and obtain scalar equations of the form

$$\dot{y}(t) = \sigma y(t) - \frac{1}{\alpha}\lambda(t), \quad \dot{\lambda}(t) = -\sigma\lambda(t), \quad t \in (0, T), \quad (17)$$

with $y(t), \lambda(t) \in \mathbb{R}$ and $\sigma \leq 0$ an eigenvalue of \mathcal{L} (see [3]). From (17), it is easy to see that $\lambda(t) = e^{\sigma(T-t)}\lambda(T)$ and

$$y(t) = e^{\sigma t}y_{in} - \frac{1}{\alpha} \int_0^t e^{\sigma(t-s)}\lambda(s)ds.$$

Substituting $\lambda(t)$ into $y(t)$, we obtain

$$y(t) = e^{\sigma t}y_{in} - \frac{1}{\alpha}e^{\sigma(T+t)} \left(\int_0^T e^{-2\sigma s} ds \right) \lambda(T).$$

Evaluating y at $t = T$ and inserting the result into the final condition $\lambda(T) = y(T) - y_{tg}$ of (17) (where $\Lambda_L = \lambda(T)$), we get

$$f(\sigma)\Lambda_L = e^{\sigma T}y_{in} - y_{tg}, \quad (18)$$

where $f(\sigma) := 1 + (e^{2\sigma T} - 1)/2\alpha\sigma$.

Note that $f(\sigma)$ is an eigenvalue of \mathcal{M} whenever σ is an eigenvalue of \mathcal{L} . Thus, if $\text{Sp}(\mathcal{L}) = \{\sigma_j, j = 1, \dots, r\}$ is the discrete approximation of the continuous eigenvalues of Δ given by $-j^2\pi^2$, then $f(\sigma_j) \approx 1 - 1/2\alpha\sigma_j, j = 1, \dots, r$, so that $\mathcal{U}f(\mathcal{D})\mathcal{U}^T \approx (I - \frac{1}{2\alpha}\mathcal{L}^{-1})$. Hence, we propose the following preconditioner

$$\widehat{\mathcal{M}}^{-1} = \mathcal{L}(\mathcal{L} - \frac{1}{2\alpha}I)^{-1}. \quad (19)$$

Note that there is no longer any mention of the eigenvalue decomposition in the definition of $\widehat{\mathcal{M}}$, meaning that no such computation is needed when applying the preconditioner. Each matrix-vector multiplication of the form $\widehat{\mathcal{M}}^{-1} \cdot \Lambda_L$ only requires a multiplication by \mathcal{L} and the solution of an elliptic problem of the form $(\mathcal{L} - \frac{1}{2\alpha}I)v = \mathbf{f}$, which can be done cheaply using e.g. algebraic multigrid [17].

The following result shows that for small δt , the eigenvalues of the preconditioned matrix is clustered around 1. Thus, preconditioned GMRES is expected to converge in a small number of iterations [2].

Theorem 4.1: Let a positive integer N be given and \mathcal{R}_ℓ be approximated using the Implicit Euler method with N fine sub-intervals over each $(T_{\ell-1}, T_\ell)$. Then any eigenvalue μ of $\mathcal{M}\widehat{\mathcal{M}}^{-1}$ satisfies

$$1 < \mu < 1 + \frac{\delta t}{\alpha}, \quad (20)$$

where $\delta t = T/LN$.

Proof: By definition, $\mathcal{R}_\ell(T_\ell)$ for (17) is given by

$$\mathcal{R}_\ell(T_\ell) = e^{(T+T_\ell)\sigma} \int_{T_{\ell-1}}^{T_\ell} e^{-2\sigma s} ds.$$

Then, the discrete form of $\mathcal{R}_\ell(T_\ell)$ using Implicit Euler reads $\mathcal{R}_\ell(T_\ell) = \delta t e^{(T+T_\ell-2T_{\ell-1})\sigma} \sum_{n=1}^N e^{-2n\sigma\delta t}$, so that,

$$\mathcal{R}_\ell(T_\ell) = \delta t e^{2(\Delta T - \delta t)\sigma} e^{(L-\ell)\Delta T\sigma} \left(\frac{e^{-2\Delta T\sigma} - 1}{e^{-2\delta t\sigma} - 1} \right). \quad (21)$$

Next, from (11), we have

$$Y_L = \mathcal{P}_1(T) \cdot y_{in} - \frac{1}{\alpha} \mathcal{R}_L(T) \Lambda_L - \frac{1}{\alpha} \sum_{j=2}^L \mathcal{P}_j(T) \mathcal{R}_{j-1}(T_{j-1}) \Lambda_L. \quad (22)$$

Since $\mathcal{P}_\ell(T) = e^{(T-T_{\ell-1})\sigma} = e^{(L-\ell+1)\Delta T\sigma}$, replacing (21) into (22), we obtain

$$\begin{aligned} Y_L &= e^{L\Delta T\sigma} \cdot y_{in} \\ &\quad - \frac{\delta t}{\alpha} e^{2(\Delta T - \delta t)\sigma} \left(\frac{e^{-2\Delta T\sigma} - 1}{e^{-2\delta t\sigma} - 1} \right) \left(\frac{e^{2T\sigma} - 1}{e^{2\Delta T\sigma} - 1} \right) \Lambda_L \\ &= e^{L\Delta T\sigma} \cdot y_{in} - \frac{\delta t}{\alpha} \left(\frac{1 - e^{2T\sigma}}{1 - e^{2\delta t\sigma}} \right) \Lambda_L. \end{aligned}$$

Therefore, inserting Y_L into $\Lambda_L = Y_L - y_{tg}$, we get $f_{\delta t}(\sigma)\Lambda_L = e^{T\sigma}y_{in} - y_{tg}$, where $f_{\delta t}(\sigma) := 1 + \frac{\delta t}{\alpha} \left(\frac{1 - e^{2T\sigma}}{1 - e^{2\delta t\sigma}} \right)$. Note that for $\sigma \in \text{Sp}(\mathcal{L})$, $f_{\delta t}(\sigma)$ is eigenvalue of the discrete \mathcal{M} .

The largest eigenvalue of \mathcal{L} as function of r is given by $\sigma_1(r) = -4(r+1)^2 \sin^2(\frac{\pi}{2(r+1)}), r \geq 1$ (see [3]). Then,

$$\begin{aligned} \sigma_1'(r) &= -8(r+1) \sin\left(\frac{\pi}{2(r+1)}\right) \\ &\quad \times \left[\sin\left(\frac{\pi}{2(r+1)}\right) - \frac{\pi}{2(r+1)} \cos\left(\frac{\pi}{2(r+1)}\right) \right]. \end{aligned}$$

Since $r \geq 1$, $\frac{\pi}{2(r+1)} \in (0, \frac{\pi}{4}]$. Using the fact that $\tan \theta - \theta > 0$, i.e., $\sin \theta - \theta \cos \theta > 0$ for $0 \leq \theta \leq \frac{\pi}{4}$, we conclude that $\sigma_1'(r) < 0$. Thus, $\sigma_1(r)$ decreases on $(1, +\infty)$ and its maximum is $\sigma_1(1) = -8$.

Now, let $\psi_0(\sigma) := f_{\delta t}(\sigma) f^{-1}(\sigma)$ for $\sigma \leq -8$. Then, $\psi_0(x) - 1 = \varphi(\sigma) / (2\alpha\sigma - 1)(1 - e^{2\delta t\sigma})$, where $\varphi(\sigma) := 2\sigma\delta t(1 - e^{2T\sigma}) + (1 - e^{2\delta t\sigma})$. We have $\varphi'(\sigma) = 2\delta t(1 - e^{2\delta\sigma}) - 2\delta t(2\sigma T + 1)e^{2T\sigma} > 0$, since $(2\sigma T + 1) < 0$. But $\varphi(-8) = -16\delta t(1 - e^{-16T}) + (1 - e^{-16\delta t}) < 0$ for $\delta t > 0$, which means $\varphi(\sigma) < 0$ for $\sigma \leq -8$. Thus, $\psi_0(\sigma) - 1 > 0$, i.e., $\psi_0(\sigma) > 1$.

Next, we consider $f_{\delta t}^0(\sigma) := 1 + \frac{\delta t}{\alpha(1 - e^{2\delta t\sigma})} \geq f_{\delta t}(\sigma)$ and $\psi(\sigma) := f_{\delta t}^0(\sigma) f^{-1}(\sigma)$ for $\sigma \in (-\infty, 0)$, whose derivative is $\psi'(\sigma) = \psi_1(\sigma) / \alpha(2\delta t\alpha - 1)^2(1 - e^{2\delta t\sigma})^2$, where $\psi_1(\sigma) := -2\alpha(1 - e^{2\delta t\sigma})^2 - \delta t(1 - e^{2\delta t\sigma}) + 4\delta t^2\sigma(2\alpha\sigma - 1)e^{2\delta t\sigma}$. We have $\psi_1'(\sigma) = 8\delta t e^{2\delta t\sigma} \psi_2(\sigma)$, where $\psi_2(\sigma) := \alpha(1 - e^{2\delta t\sigma}) + \sigma\delta t^2(2\alpha\sigma - 1) + 2\alpha\delta t\sigma$. Since $\psi_2''(\sigma) = 4\alpha\delta t^2(1 - e^{2\delta t\sigma}) > 0$ and $\psi_2'(0) = -\delta t^2 < 0$, we get $\psi_2'(\sigma) < 0$. Also, since $\psi_2(0) = 0$, $\psi_2(\sigma) \geq 0$, which implies $\psi_1'(\sigma) > 0$. Finally, since $\psi_1(0) = 0$, $\psi_1(\sigma) \leq 0$ and $\psi'(\sigma) \leq 0$. Hence, ψ is decreases on $(-\infty, 0)$ and tends to $1 + \delta t/\alpha$ as σ tends to $-\infty$, such that $\psi(\sigma) < 1 + \frac{\delta t}{\alpha}$.

Now, the quantity $\psi_0(\sigma)$ is the eigenvalue of $\mathcal{M}\widehat{\mathcal{M}}^{-1}$ associated with the fixed $\sigma \in \text{Sp}(\mathcal{L})$. Then we have $1 < \psi_0(\sigma) \leq \psi(\sigma) \leq 1 + \frac{\delta t}{\alpha}$, which concludes the proof. ■

Remark 1: Note that $\widehat{\mathcal{M}}^{-1}$ is derived from the continuous form of \mathcal{M} . Therefore, we expect the preconditioner to be more efficient for a high-order approximation of \mathcal{R}_ℓ than for a lower-order one.

Analogously, we can prove that (19) yields an effective preconditioner for (13) when the boundary condition in (14) is Dirichlet-Neumann in 1D or Dirichlet in 2D.

In our numerical test, we use the example studied in [4]: we set $T = 1, L = 10, r = 100, y_{in}(x) = \exp(-100(x - 1/2)^2)$ and $y_{ig}(x) = \frac{1}{2} \exp(-100(x - 1/4)^2) + \frac{1}{2}(-100(x - 3/4)^2)$ with Dirichlet boundary conditions. We use Matlab R2021b as our numerical environment. Moreover, to get \mathcal{R}_ℓ , we use two quadrature formulas, namely Implicit Euler and the Singly Diagonal Implicit Runge-Kutta (SDIRK) method of order 3, whose stages and weights are $c = [1/2 + \sqrt{3}/6, 1/2 - \sqrt{3}/6]$ and $d = [1/2, 1/2]$ respectively.

First, we investigate the numerical behavior of the upper bound in (20) and the maximal eigenvalue σ_{max} of $\mathcal{M}\widehat{\mathcal{M}}^{-1}$ for various $N = 100k, k = 1, \dots, 10$, i.e., $\delta t = 1/N$ and fixed $\alpha = 10^{-4}$ with only Implicit Euler for getting \mathcal{R}_ℓ . The result is shown in Fig. 2, where we observe that the upper bound in (20) follows closely the evolution of σ_{max} for various δt . Thus, (20) provides an efficient estimation of the upper bound of the eigenvalues of $\mathcal{M}\widehat{\mathcal{M}}^{-1}$.

Next, we investigate the convergence speed of the linear system (13) in GMRES for $N = 1000$ and $\alpha = \{10^{-4}, 10^{-6}\}$. We use for this instance the function `gmres` in Matlab with `restart=1, tol=1e-8` and `maxit=500`.

For $\alpha = 10^{-4}$, the result is shown in Table I. We observe that when solving (13), unpreconditioned GMRES reaches 500 iterations without satisfying the tolerance for either quadrature formula. Moreover, the eigenvalues of \mathcal{M} for both quadrature formulas lie in $(1, 500)$. In contrast, for the preconditioned case, when SDIRK is used, GMRES reaches the tolerance with only 2 iterations, and 9 iterations for the Implicit Euler case. We observe a similar behavior for $\alpha = 10^{-6}$, with poor convergence without preconditioning, fairly good convergence for Implicit Euler (44 iterations) and excellent convergence for SDIRK (4 iterations). The results are shown in Table II.

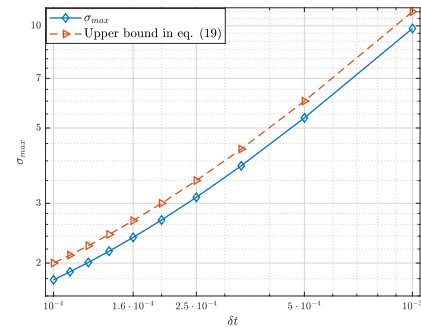


Fig. 2. The maximal eigenvalue of $\mathcal{M}\widehat{\mathcal{M}}^{-1}$ and the upper bound in (20) for various δt with $\alpha = 10^{-4}$.

TABLE I

GMRES FOR SOLVING (13) FOR THE HEAT EQUATION, WITH $\alpha = 10^{-4}$.

Matrix \mathcal{M}				
	σ_{min}	σ_{max}	# Iters	Res
Euler	2.0	4.9e2	500	1.3e-8
SDIRK	1.08	4.9e2	500	4.08e-7
Preconditioned matrix $\mathcal{M}\widehat{\mathcal{M}}^{-1}$				
	σ_{min}	σ_{max}	# Iters	Res
Euler	1.0	1.78	9	7.7e-9
SDIRK	0.97	1.0	2	9.64e-9

TABLE II

GMRES FOR SOLVING (13) FOR THE HEAT EQUATION, WITH $\alpha = 10^{-6}$.

Matrix \mathcal{M}				
	σ_{min}	σ_{max}	# Iters	Res
Euler	1e2	4.97e4	500	1.08e-6
SDIRK	9.68	4.96e4	500	4.2e-5
Preconditioned matrix $\mathcal{M}\widehat{\mathcal{M}}^{-1}$				
	σ_{min}	σ_{max}	# Iters	Res
Euler	1.0	7.76	44	8.37e-9
SDIRK	0.74	1.0	4	7.31e-9

In both Tables I and II, we observe that preconditioned GMRES converges more quickly for SDIRK than for Implicit

Euler. This is consistent with Remark 1, since SDIRK is of order 3, which is higher than the order 1 of Implicit Euler.

Theorem 4.1 asserts that the condition number $\text{cond}(\mathcal{M}\widehat{\mathcal{M}}^{-1})$ is bounded by $1 + \frac{\delta t}{\alpha}$ when Implicit Euler is used to compute \mathcal{R}_ℓ . Since $\mathcal{M}\widehat{\mathcal{M}}^{-1}$ is symmetric (as \mathcal{M} and $\widehat{\mathcal{M}}$ commute and are both symmetric), we deduce that the number of preconditioned GMRES iterations remains bounded as $r \rightarrow \infty$, which is confirmed by the results in Table III. We also observe that GMRES converges faster for a higher order method like SDIRK than for Implicit Euler.

TABLE III

GMRES FOR SOLVING (13) WITH PRECONDITIONING FOR THE HEAT EQUATION, WITH $\alpha = 10^{-4}$ AND FOR VARIOUS r .

r	# Iters(Euler)		# Iters(SDIRK)	
	$L = 10^3$	$L = 3 \cdot 10^3$	$L = 10^3$	$L = 3 \cdot 10^3$
100	9	6	3	2
200	10	7	3	3
250	11	7	3	3
600	11	7	3	3

Finally, one can use an explicit method as a numerical integrator to approximate \mathcal{R}_ℓ , as long as a CFL condition is satisfied.

B. Wave equation

We now consider the optimal control problem constrained by the 1D wave equation given by

$$\partial_{tt}u = \Delta u + v, \text{ on } (0, 1) \times (0, T), \quad (23)$$

with $u(x, 0) = u_0(x)$, $\partial_t u(x, 0) = 0$, $x \in (0, 1)$. For simplicity, we will consider homogeneous Dirichlet boundary conditions in space. Discretizing in space using second-order centered finite differences leads to the ODE system

$$\dot{y} = \mathcal{L}y + \mathcal{B}v, \text{ with } \mathcal{L} = \begin{bmatrix} 0 & I \\ \Delta_h & 0 \end{bmatrix}, \mathcal{B} = \begin{bmatrix} 0 \\ I \end{bmatrix}, \quad (24)$$

where $y = \begin{bmatrix} u_h \\ \partial_t u_h \end{bmatrix}$ and Δ_h is the discrete Laplacian.

To derive a preconditioner for the system (13), we again look to the continuous problem. Since \mathcal{L} is no longer symmetric, the continuous analogue of \mathcal{M} takes the form

$$\mathcal{M}_c := I + \frac{1}{\alpha} \int_0^T e^{s\mathcal{L}} \mathcal{B} \mathcal{B}^T e^{s\mathcal{L}^T} ds.$$

Let us evaluate this integral. First note that

$$e^{s\mathcal{L}} = I + s\mathcal{L} + \frac{s^2}{2!}\mathcal{L}^2 + \frac{s^3}{3!}\mathcal{L}^3 + \dots$$

Introducing the notation $A := -\Delta_h$, we see that

$$\mathcal{L}^2 = \begin{bmatrix} \Delta_h & 0 \\ 0 & \Delta_h \end{bmatrix} = - \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} =: -\mathcal{A}$$

is a symmetric negative definite matrix, so we can write

$$\begin{aligned} e^{s\mathcal{L}} &= \left(I - \frac{s^2}{2!}\mathcal{A} + \frac{s^4}{4!}\mathcal{A}^2 + \dots \right) \\ &\quad + s\mathcal{L} \left(I - \frac{s^2}{3!}\mathcal{A} + \frac{s^4}{5!}\mathcal{A}^2 + \dots \right) \\ &= \cos(s\mathcal{A}^{1/2}) + s\mathcal{L}\text{sinc}(s\mathcal{A}^{1/2}), \end{aligned}$$

where $\text{sinc}(x) = \sin(x)/x$ is the sinc function and $\mathcal{A}^{1/2}$ is the symmetric square root of \mathcal{A} (i.e., a symmetric matrix with positive eigenvalues whose square is \mathcal{A}). This leads to

$$\begin{aligned} e^{s\mathcal{L}} \mathcal{B} \mathcal{B}^T e^{s\mathcal{L}^T} &= [\cos(s\mathcal{A}^{1/2}) + s\mathcal{L}\text{sinc}(s\mathcal{A}^{1/2})] \cdot \mathcal{B} \\ &\quad \cdot \mathcal{B}^T \cdot [\cos(s\mathcal{A}^{1/2}) + s\text{sinc}(s\mathcal{A}^{1/2})\mathcal{L}^T]. \end{aligned}$$

By direct calculation and using $x \cdot \text{sinc}(xA^{1/2}) = A^{-1/2} \sin(xA^{1/2})$, we obtain

$$e^{s\mathcal{L}} \mathcal{B} \mathcal{B}^T e^{s\mathcal{L}^T} = \begin{bmatrix} \mathcal{C}_{11}(s) & \mathcal{C}_{12}(s) \\ \mathcal{C}_{21}(s) & \mathcal{C}_{22}(s) \end{bmatrix},$$

where $\mathcal{C}_{11}(s) := A^{-1} \sin^2(sA^{1/2})$, $\mathcal{C}_{22}(s) := \cos^2(sA^{1/2})$, $\mathcal{C}_{12}(s) := A^{-1/2} \cos(sA^{1/2}) \sin(sA^{1/2})$ and $\mathcal{C}_{21}(s) := \mathcal{C}_{12}(s)$.

We can now integrate these coefficients to obtain

$$\mathcal{M}_c = \begin{bmatrix} \mathcal{M}_{11} & \mathcal{M}_{12} \\ \mathcal{M}_{21} & \mathcal{M}_{22} \end{bmatrix},$$

where $\mathcal{M}_{12} := \frac{1}{2\alpha} A^{-1} (I - \cos(2TA^{1/2}))$, $\mathcal{M}_{21} := \mathcal{M}_{12}$,

$$\mathcal{M}_{11} := I + \frac{T}{2\alpha} A^{-1} - \frac{1}{4\alpha} A^{-3/2} \sin(2TA^{1/2}),$$

$$\mathcal{M}_{22} := \left(1 + \frac{T}{2\alpha}\right) I + \frac{1}{4\alpha} A^{-1/2} \sin(2TA^{1/2}).$$

To find a good preconditioner for \mathcal{M}_c , let us analyze \mathcal{M}_c^{-1} . Since all \mathcal{M}_{ij} are square and they all commute, we have

$$\mathcal{M}_c^{-1} = \begin{bmatrix} \mathcal{H}^{-1} \mathcal{M}_{22} & -\mathcal{H}^{-1} \mathcal{M}_{12} \\ -\mathcal{H}^{-1} \mathcal{M}_{21} & \mathcal{H}^{-1} \mathcal{M}_{11} \end{bmatrix},$$

with $\mathcal{H} := \mathcal{M}_{11}\mathcal{M}_{22} - \mathcal{M}_{12}\mathcal{M}_{21}$. Just like for the heat equation, we replace A by a scalar $\sigma > 0$ and study the eigenvalues of \mathcal{M}_c^{-1} when $\sigma \approx k^2 \pi^2$ approximates the eigenvalues of the negative Laplacian. We observe for large k that $\mathcal{H} \approx \frac{1}{4\alpha^2} (2\alpha + T) A^{-1} (TI + 2\alpha A)$, $\mathcal{H}^{-1} \mathcal{M}_{12} = \mathcal{H}^{-1} \mathcal{M}_{21} \approx 0$ and

- $\mathcal{H}^{-1} \mathcal{M}_{11} \approx I - (TI + 2\alpha A)^{-1}$,
- $\mathcal{H}^{-1} \mathcal{M}_{22} \approx \left(1 - \frac{1}{(2\alpha + T)}\right) I$.

These observations lead us to approximate \mathcal{M}_c^{-1} by a block diagonal matrix $\widehat{\mathcal{M}}^{-1}$ of the form

$$\widehat{\mathcal{M}}^{-1} := I - \begin{bmatrix} (aI + bA)^{-1} & 0 \\ 0 & cI \end{bmatrix}, \quad (25)$$

and use it as a preconditioner for (13), where $a = T, b = 2\alpha$ and finally $c = 1/(2\alpha + T)$. As before, one need not compute the eigenvalue decomposition in order to apply the preconditioner $\widehat{\mathcal{M}}$: each application only requires solving an elliptic problem of the form $(aI + bA)\mathbf{v} = \mathbf{f}$, which again can be done cheaply using e.g. multigrid.

In our numerical experiment, we take $T = 2.3$, $u_0(x) = \exp(-100(x - 1/2)^2)$, so that $y_{in} = (u_0(x), 0)$ and $y_{ig}(x) = [(\frac{1}{2} \exp(-100(x - 1/4)^2) + \frac{1}{2}(-100(x - 3/4)^2), 0)]$. We set $L = 10, r = 100$, and we use the same Matlab functions for computing \mathcal{R}_ℓ and \mathcal{P}_ℓ as in Section IV-A. We first investigate the convergence speed of the linear system (13) in GMRES for $N = 1000$ and $\alpha = 10^{-6}$. We use for this instance the function `gmres` in Matlab with `restart=[]`, `tol=1e-8` and `maxit=size(M, 1)`. The results are

shown in Table IV, where we observe that unpreconditioned GMRES applied to (13) converges in 84 iterations for both quadrature formulas, whereas the preconditioned iteration converges in only 4 iterations.

TABLE IV

GMRES FOR SOLVING (13) FOR THE WAVE EQUATION, WITH $\alpha = 10^{-6}$.

Matrix \mathcal{M}			
	cond(\mathcal{M})	# Iters	Res
Euler	3.8e4	84	8.74e-9
SDIRK	3.8e4	84	8.74e-9
Preconditioned matrix $\mathcal{M}\widehat{\mathcal{M}}^{-1}$			
	cond($\mathcal{M}\widehat{\mathcal{M}}^{-1}$)	# Iters	Res
Euler	2.59	4	1.58e-9
SDIRK	2.59	4	1.58e-9

Unlike for the heat equation, the higher order of SDIRK does not lead to better convergence compared to Implicit Euler. Nonetheless, for both methods, we have observed that the number of GMRES iterations does not change as we increase the number of time steps N from 100 to 1000, as long as preconditioning is used.

These results show that for fixed r , the discretization parameters in time do not change the behavior of the convergence of (13) in GMRES. We also observe that when r increases, the number of iterations of the unpreconditioned system increases, whereas the number of iterations of the preconditioned system decreases. The results are shown in Tables V and VI.

TABLE V

GMRES FOR SOLVING (13) FOR THE WAVE EQUATION WITH IMPLICIT EULER WITH $\alpha = 10^{-6}$ FOR VARIOUS r .

r	cond(\mathcal{M})	# Iters	cond($\mathcal{M}\widehat{\mathcal{M}}^{-1}$)	# Iters
10	4.83e2	10	2.47	5
150	7.75e4	76	2.81	3
350	2.48e5	104	2.49	3

TABLE VI

GMRES FOR SOLVING (13) FOR THE WAVE EQUATION WITH IMPLICIT EULER FOR VARIOUS α .

α	cond(\mathcal{M})	# Iters	cond($\mathcal{M}\widehat{\mathcal{M}}^{-1}$)	# Iters
1e-5	2.26e4	52	2.41	3
1e-3	4.98e2	20	1.09	3
1e-1	6.03	9	1.01	3
1e1	1.05	4	1.0	2

V. ONGOING WORK

We are currently studying the behavior of our preconditioners when \mathcal{M} is obtained from an explicit method while respecting the CFL condition, in order to better understand the impact of using explicit versus implicit methods. We are also working on more general convergence properties of the algorithm, its error analysis, and on understanding its performance compared to existing parallel-in-time algorithms.

VI. ACKNOWLEDGMENTS

The work described in this paper is partially supported by a grant from the ANR/RGC joint research scheme sponsored by the Research Grants Council of the Hong Kong Special Administrative Region, China (Project no. A-CityU203/19), by the French National Research Agency (Project AL-LOWAPP, grant ANR-19-CE46-0013-01 and Projet DEEP-NUM, grant ANR-21-CE23-0017) and by Mitacs-Globalink Canada and INRIA France (grant IT34057, 2023).

REFERENCES

- [1] Andrew T Barker and Martin Stoll. Domain decomposition in time for PDE-constrained optimization. *Computer Physics Communications*, 197:136–143, 2015.
- [2] Michele Benzi. Preconditioning techniques for large linear systems: a survey. *Journal of Computational Physics*, 182(2):418–477, 2002.
- [3] Albrecht Böttcher and Sergei M. Grudsky. *Spectral Properties of Banded Toeplitz Matrices*. Society for Industrial and Applied Mathematics, 2005.
- [4] 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.
- [5] Martin J Gander. 50 years of time parallel time integration. In *Multiple Shooting and Time Domain Decomposition Methods: MuS-TDD, Heidelberg, May 6-8, 2013*, pages 69–113. Springer, 2015.
- [6] Martin J. Gander and Stefan Güttel. Paraexp: A parallel integrator for linear initial-value problems. *SIAM Journal on Scientific Computing*, 35(2):C123–C142, 2013.
- [7] Martin J Gander and Laurence Halpern. Optimized schwarz waveform relaxation methods for advection reaction diffusion problems. *SIAM Journal on Numerical Analysis*, 45(2):666–697, 2007.
- [8] Martin J Gander and Felix Kwok. Schwarz methods for the time-parallel solution of parabolic control problems. In *Domain decomposition methods in science and engineering XXII*, pages 207–216. Springer, 2016.
- [9] G.H. Golub and C.F. Van Loan. *Matrix computations*. The John Hopkins University Press, Baltimore, 2013.
- [10] Matthias Heinkenschloss. A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems. *Journal of Computational and Applied Mathematics*, 173(1):169–198, 2005.
- [11] Nicholas J. Higham. *Functions of Matrices*. Society for Industrial and Applied Mathematics, 2008.
- [12] Thi-Thao-Phuong Hoang, Jérôme Jaffré, Caroline Japhet, Michel Kern, and Jean E Roberts. Space-time domain decomposition methods for diffusion problems in mixed formulations. *SIAM Journal on Numerical Analysis*, 51(6):3532–3559, 2013.
- [13] Shishun Li and Xiao-Chuan Cai. Convergence analysis of two-level space-time additive Schwarz method for parabolic equations. *SIAM Journal on Numerical Analysis*, 53(6):2727–2751, 2015.
- [14] Yvon Maday, Julien Salomon, and Gabriel Turinici. Monotonic parareal control for quantum systems. *SIAM Journal on Numerical Analysis*, 45(6):2468–2482, 2007.
- [15] Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49, 2003.
- [16] Benjamin W Ong and Jacob B Schroder. Applications of time parallelization. *Computing and Visualization in Science*, 23:1–15, 2020.
- [17] John W Ruge and Klaus Stüben. Algebraic multigrid. In *Multigrid methods*, pages 73–130. SIAM, 1987.
- [18] Andrea Toselli and Olof Widlund. *Domain Decomposition Methods – Algorithms and Theory*. Springer-Verlag, Berlin, 2005.
- [19] Colin S.C. Tsang. Preconditioners for linear parabolic optimal control problems. Master’s thesis, Hong Kong Baptist University, Hong Kong, China, August 2017.
- [20] Stefan Ulbrich. Generalized SQP methods with “parareal” time-domain decomposition for time-dependent pde-constrained optimization. In *Real-time PDE-constrained optimization*, pages 145–168. SIAM, 2007.