



HAL
open science

A parallel in time algorithm based 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 ParaExp for optimal control problems. 2024. hal-04612018

HAL Id: hal-04612018

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

Preprint submitted on 14 Jun 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 PARAREXP FOR OPTIMAL CONTROL PROBLEMS

Felix Kwok

Département de mathématiques et
de statistique, Université Laval
Canada

felix.kwok@mat.ulaval.ca

Djahou N. Tognon

INRIA Paris, ANGE Project-Team,
Sorbonne Université, (LJLL),
France

djahou-norbert.tognon@inria.fr

ABSTRACT

We propose a new parallel-in-time algorithm for solving optimal control problems constrained by 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.

Keywords ParaExp Algorithm · Optimality systems · Preconditioning

1 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 for finding 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 [17, 7, 12, 13] and the references therein. There is also a growing literature in parallel-in-time methods, in multiple processors solve the initial value problem on different parts of the time interval simultaneously; see [5, 16] for a survey of these methods for initial value problems. Time parallelization can also be applied to optimal control problems, either by decomposing the optimization problem directly [10, 14, 1, 8], or by using a parallel-in-time method to solve the forward and adjoint problems required by gradient computations [19].

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 linear operator \mathcal{L} . 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.

The rest of the paper is organized in follows. We present our model optimal control problem in Section 2. The parallel-in-time algorithm is defined in Section 3, where we show how computations from different parts of the time interval can be performed in parallel. The algorithm yields a linear system whose matrix-vector product can be performed in

parallel, but the system is generally ill-conditioned; a preconditioner is therefore needed to solve it efficiently using iterative methods. Section 4 is devoted to the construction of the preconditioner for two model problems, namely the heat and wave equations. Numerical experiments are provided to illustrate the behaviour of these preconditioners. Finally, we summarize and outline some future work in Section 5.

2 Linear optimal control problem

We consider the optimal control problem given by

$$\begin{aligned} \min_{\nu} \quad & \frac{1}{2} \|y(T) - y_{tg}\|^2 + \frac{\alpha}{2} \int_0^T \|\nu(t)\|^2 dt, \\ \text{subject to} \quad & \dot{y}(t) = \mathcal{L}y(t) + \nu(t), \quad y(0) = y_{in}, \quad t \in (0, T), \end{aligned}$$

where α is a regularization parameter and y_{in}, y_{tg} are the initial and target states respectively. The state function y and the control ν are defined from $(0, T)$ to $\mathbb{R}^r, r \in \mathbb{N}^*$. The operator $\mathcal{L} \in \mathbb{R}^{r \times r}$ is independent of the time variable. Using the Lagrange multiplier approach (see e.g. the historical review [3]), one can derive the forward and adjoint problems to be

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

where the control ν and adjoint state λ are related by the algebraic equation $\lambda(t) = \alpha\nu(t)$ for all $t \in (0, T)$. Eliminating ν , 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.

3 Parallel-in-time algorithm

For a given $L \in \mathbb{N}^*$, 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 approximations of the state y and the adjoint state 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 homogeneous and inhomogeneous parts. First, we consider the homogeneous sub-problems on λ , which require the backward propagation of λ over each $(T_{\ell-1}, T_{\ell})$ according to

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

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

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

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 $(T_{\ell-1}, T_{\ell})$:

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

for $\ell = 2, \dots, L$. Finally, $Y_{\ell} \approx y_{\ell}(T_{\ell})$ is directly obtained by superposition principle

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

Thus, for a given Λ_L , we have the following data dependency: the λ_{ℓ} is independent of everything, w_{ℓ} depends only on λ_{ℓ} as a source term, and $u_{\ell+1}$ depends on w_{ℓ} for initial conditions. Moreover, the computation for different indices ℓ are independent. 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, as shown in Figure 1. As we only need y_{ℓ} at the interfaces to recreate the optimal trajectory (5), the only communication needed between processors are the values of u and w at the interfaces T_{ℓ} .

Let \mathcal{P}_ℓ and \mathcal{Q}_ℓ be the exponential propagators for u_ℓ and λ_ℓ , as 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.) Then the solution of (2) becomes

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

Substituting (6) into (3) on $(T_{\ell-1}, T_\ell)$ gives

$$\dot{w}_\ell(t) = \mathcal{L}w_\ell(t) - \frac{1}{\alpha} \mathcal{Q}_\ell(t) \cdot \Lambda_L, \quad w_\ell(T_{\ell-1}) = 0. \quad (7)$$

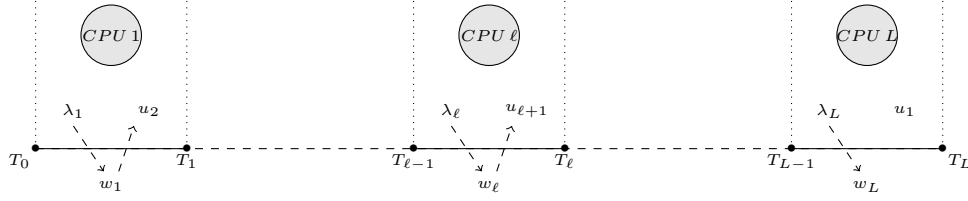


Figure 1: Parallel distribution on the processors

Moreover, because $w_\ell(T_{\ell-1}) = 0$ in (7), a continuous solution of (7) is given by

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

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

$$w_\ell(t) = -\frac{1}{\alpha} \mathcal{R}_\ell(t) \cdot \Lambda_L. \quad (8)$$

Therefore, using (8), (4) become

$$\begin{aligned} u_1(t) &= \mathcal{P}_1(t) \cdot y_{in} \text{ on } (T_0, T_L), \\ u_\ell(t) &= -\frac{1}{\alpha} \mathcal{P}_\ell(t) \cdot \mathcal{R}_{\ell-1}(T_{\ell-1}) \cdot \Lambda_L, \text{ on } (T_{\ell-1}, T_L). \end{aligned} \quad (9)$$

Now, replacing (8) and (9) into (5) leads to

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

Hence, the optimality system (1) in terms of Y_ℓ and Λ_ℓ becomes

$$Y_\ell = \mathcal{P}_1(T_\ell) \cdot y_{in} - \frac{1}{\alpha} \mathcal{R}_\ell(T_\ell) \cdot \Lambda_L - \frac{1}{\alpha} \sum_{j=2}^{\ell} \mathcal{P}_j(T_\ell) \cdot \mathcal{R}_{j-1}(T_{j-1}) \cdot \Lambda_L, \quad \ell = 1, \dots, L, \quad (10)$$

with

$$\Lambda_\ell = \mathcal{Q}_\ell(T_\ell) \cdot \Lambda_L, \quad \ell = 1, \dots, L-1.$$

We remark that each Y_ℓ depends on the same unknown Λ_L . Moreover, Y_L and Λ_L satisfy the discrete final condition of (1) given by

$$\Lambda_L - Y_L + y_{tg} = 0. \quad (11)$$

It follows that, for $\ell = L$, substituting (10) into (11) yields

$$M \cdot \Lambda_L + \mathbf{b} = 0, \quad (12)$$

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

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

with I being the identity matrix. Note that $M\Lambda_L$ contains a sum of terms, each one of which can be computed in parallel over the sub-intervals $[T_{\ell-1}, T_\ell]$. We remark that the problem of approximating \mathcal{P} and \mathcal{R} is non-trivial: 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 is more challenging and 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.

It remains to solve the linear system (12) using an iterative method, e.g. GMRES. Since the matrix M is generally ill-conditioned, an appropriate preconditioner is needed, which we will construct in the next section for the heat and wave equations.

4 Preconditioner design

4.1 Heat equation case

In this section, we propose a preconditioner P^{-1} for the heat equation to speed up the convergence of (12) in GMRES. Our approach is inspired by the work in the Master’s thesis [18], where the author derived a preconditioner based the behaviour of \mathcal{L} at low and high frequencies. Let us consider the heat equation given by

$$\dot{y} = \Delta y + \nu, \text{ on } \Omega \times (0, T), \quad (13)$$

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 (13) using finite differences leads the following ODE system

$$\dot{y}(t) = \mathcal{L}y(t) + \nu(t), \quad y(0) = y_{in}, \quad t \in (0, T), \quad (14)$$

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

$$\dot{y} = \mathcal{L}y - \frac{1}{\alpha}\lambda, \quad \dot{\lambda} = -\mathcal{L}\lambda, \quad (15)$$

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

We now consider the eigenvalue decomposition $\mathcal{L} = QDQ^T$, where D is a diagonal matrix with the eigenvalues of \mathcal{L} and Q a unitary matrix. The transformation $y(t) \mapsto Q^T \cdot y(t)$ and $\lambda(t) \mapsto Q^T \cdot \lambda(t)$ allows us to diagonalize (15) to obtain scalar equations of the form

$$\dot{y}(t) = \sigma y(t) - \frac{1}{\alpha}\lambda(t), \quad \dot{\lambda}(t) = -\sigma\lambda(t), \quad t \in (0, T), \quad (16)$$

with $y(t), \lambda(t) \in \mathbb{R}$ and σ an eigenvalue of \mathcal{L} , which is real and negative. From (16), 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 $y(t)$ gives

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

Evaluating y at $t = T$ and inserting the result into the final condition $\lambda(T) = y(T) - y_{tg}$ of (16) (where $\Lambda_L = \lambda(T)$), we get

$$f(\sigma)\Lambda_L = e^{\sigma T}y_{in} - y_{tg}, \quad (17)$$

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

Note that $f(\sigma)$ is an eigenvalue of 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 $Qf(D)Q^T \approx (I - \frac{1}{2\alpha}\mathcal{L}^{-1})$. Hence, we propose the following preconditioner

$$\widehat{M}^{-1} = \mathcal{L}(\mathcal{L} - \frac{1}{2\alpha}I)^{-1}, \quad (18)$$

where I is the identity matrix. Each application of $\widehat{\mathcal{M}}^{-1}$ thus requires the solution of the elliptic problem $(\mathcal{L} - \frac{1}{\alpha}I)v = \mathbf{f}$, which is cheaper in terms of numerical computing cost than the inverse of the full matrix M .

To understand the behaviour of GMRES applied to the preconditioned system, we prove the following result.

Theorem 4.1. *Let N be given and \mathcal{R}_ℓ be approximated using implicit Euler with N fine sub-intervals over each $[T_{\ell-1}, T_\ell]$. Then any eigenvalue μ of $M\widehat{\mathcal{M}}^{-1}$ satisfies*

$$1 < \mu < 1 + \frac{\delta t}{\alpha}, \quad (19)$$

where $\delta t = T/LN$.

Proof. By definition, $\mathcal{R}_\ell(T_\ell)$ for (16) is given by

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

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,

$$\mathcal{R}_\ell(T_\ell) = \delta t e^{2(\Delta T - \delta t)\sigma} \cdot e^{(L-\ell)\Delta T\sigma} \left(\frac{e^{-2\Delta T\sigma} - 1}{e^{-2\delta t\sigma} - 1} \right). \quad (20)$$

Next, from (5), we have

$$Y_L = \mathcal{P}_1(T) \cdot y_{in} - \frac{1}{\alpha} \mathcal{R}_L(T) \Lambda_L - \frac{1}{\alpha} \sum_{j=2}^L \mathcal{P}_j(T) \mathcal{R}_{j-1}(T_{j-1}) \Lambda_L. \quad (21)$$

Since $\mathcal{P}_\ell(T) = e^{(T-T_{\ell-1})\sigma} = e^{(L-\ell+1)\Delta T\sigma}$, we obtain using (20) that

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

Inserting Y_L into $\Lambda_L = Y_L - y_{tg}$ yields $f_{\delta t}(\sigma)\Lambda_L = e^{T\sigma} y_{in} - y_{tg}$, 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 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 [2]). Then, we have

$$\sigma'_1(r) = -8(r+1) \sin\left(\frac{\pi}{2(r+1)}\right) \left[\sin\left(\frac{\pi}{2(r+1)}\right) - \frac{\pi}{2(r+1)} \cos\left(\frac{\pi}{2(r+1)}\right) \right].$$

Note that $\frac{\pi}{2(r+1)} \in (0, \frac{\pi}{4}]$ since $r \geq 1$. Then, 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 $\sigma'_1(r) < 0$. Hence, $\sigma_1(r)$ is decreasing as r increases and the maximum of $\sigma_1(r)$ for $r \geq 1$ is -8 .

Now, let consider $\psi_0(\sigma) = f_{\delta t}(\sigma)f^{-1}(\sigma)$ for $\sigma \leq -8$. Then, we have $\psi_0(x) - 1 = \varphi(\sigma)/(2\alpha\sigma - 1)(1 - e^{2\delta t\sigma})$, where $\varphi(\sigma) := 2\sigma\delta t(1 - e^{2T\sigma}) + (1 - e^{2\delta t\sigma})$. Next, $\varphi'(\sigma) = 2\delta t(1 - e^{2\delta\sigma}) - 2\delta t(2\sigma T + 1)e^{2T\sigma} > 0$, since $(2\sigma T + 1) < 0$. Moreover, $\varphi(-8) = -16\delta t(1 - e^{-16T}) + (1 - e^{-16\delta t}) < 0$ for $\delta t > 0$ then, $\varphi(\sigma) < 0$. It follows that, $\psi_0(\sigma) - 1 > 0$, i.e., $\psi_0(\sigma) > 1$.

Next, since $f_{\delta t}(\sigma) > 0$, let $f_{\delta t}(\sigma) \leq f_{\delta t}^0(\sigma) := 1 + \frac{\delta t}{\alpha(1 - e^{2\delta t\sigma})}$ and $\psi(\sigma) := f_{\delta t}^0(\sigma)f^{-1}(\sigma)$ for $\sigma \in (-\infty, 0)$. The derivative of ψ is given by $\psi'(\sigma) = \psi_1(\sigma)/\alpha(2\delta t\alpha - 1)^2(1 - e^{2\delta t\sigma})^2$, where

$$\psi_1(\sigma) := -2\alpha(1 - e^{2\delta t\sigma})^2 - \delta t(1 - e^{2\delta t\sigma}) + 4\delta t^2\sigma(2\alpha\sigma - 1)e^{2\delta t\sigma}.$$

Moreover, $\psi'_1(\sigma) = 8\delta te^{2\delta t\sigma}\psi_2(\sigma)$, where $\psi_2(\sigma) := \alpha(1 - e^{2\delta t\sigma}) + \sigma\delta t^2(2\alpha\sigma - 1) + 2\alpha\delta t\sigma$. Also, $\psi''_2(\sigma) = 4\alpha\delta t^2(1 - e^{2\delta t\sigma}) > 0$ and $\psi'_2(0) = -\delta t^2 < 0$ then $\psi'_2(\sigma) < 0$. Since $\psi_2(0) = 0$, $\psi_2(\sigma) \geq 0$ which implies that

$\psi_1'(\sigma) > 0$. And, since $\psi_1(0) = 0$, $\psi_1(\sigma) \leq 0$ and then $\psi'(\sigma) \leq 0$. Hence, ψ is decreasing in $(-\infty, 0)$ 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 $M\widehat{\mathcal{M}}^{-1}$ associated to 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. \square

Remark 1. Note that $\widehat{\mathcal{M}}^{-1}$ is derived from the continuous form of 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 also easily prove that (18) is efficient for (12) when the boundary condition of the Laplacian in (13) is Dirichlet-Neumann in 1D or Dirichlet-Dirichlet boundary condition 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 $y_{tg}(x) = \frac{1}{2} \exp(-100(x-1/4)^2) + \frac{1}{2} \exp(-100(x-3/4)^2)$ with Dirichlet boundary conditions. We use Matlab R2021b as our numerical environment. The operator \mathcal{P}_ℓ is computed by the Matlab function `expm`. As \mathcal{L} is full rank, `expm` computes the approximations \tilde{Q} of Q and \tilde{D} of D then approximates $\exp(\mathcal{L})$ by $\tilde{Q} \exp(\tilde{D}) \tilde{Q}^T$. Moreover, to get \mathcal{R}_ℓ , we use two quadrature formulas. The first one is Implicit Euler and the second is 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 (19) and the maximal eigenvalue σ_{max} of MP^{-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, wherein the upper bound in (19) follows efficiently the evolution of σ_{max} for various δt . So, (19) provides a efficient estimation of the upper bound of the eigenvalues of MP^{-1} .

Next, we investigate the convergence speed of the linear system (12) 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 1. We observe that for solving (12), GMRES without preconditioner reaches 500 iterations without satisfying the tolerance for both quadrature formulas. Moreover, the eigenvalues of M for both quadrature formulas lie in $(1, 500)$. But, for the preconditioned case, when SDIRK is used, GMRES reaches the tolerance with only 2 iterations, and 9 iterations for Implicit Euler case.

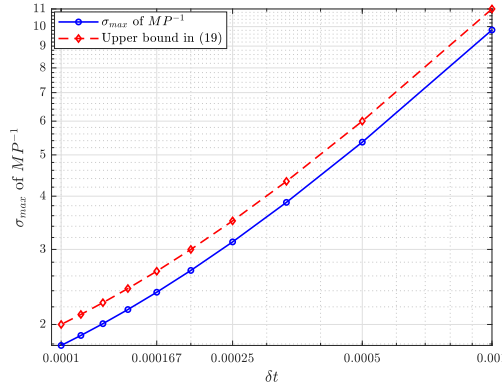


Figure 2: The maximal eigenvalue of $M\widehat{\mathcal{M}}^{-1}$ and the upper bound in (19) for various δt when $\alpha = 10^{-4}$.

For $\alpha = 10^{-6}$, the result is shown in Table 2. We observe that GMRES without preconditioner reaches 500 iterations without satisfying the tolerance for both quadrature formulas. Moreover, the eigenvalues of M increase compare to the ones in $\alpha = 10^{-4}$ case. Besides, for the preconditioned system when SDIRK is used GMRES convergence in 4 iterations whereas for Implicit Euler, 44 iterations.

Furthermore, for both Tables 1 and 2, we observe that GMRES for the preconditioned system for SDIRK case convergences faster than the Implicit Euler case. This follows from Remark 1 because the order of SDIRK is 3 higher than the order 1 of Implicit Euler. We also observe that the maximal eigenvalue of $M\widehat{\mathcal{M}}^{-1}$ for Implicit Euler method is less than for SDIRK, meaning that the upper bound in (19) can be lower for high order methods than for Implicit Euler.

Table 1: GMRES for solving (12) for $\alpha = 10^{-4}$.

Matrix M				
Schemes	σ_{min}	σ_{max}	Nbr Iters	Residuals
Euler	2.0	4.9e2	500	1.3e-8
SDIRK	1.08	4.9e2	500	4.08e-7

Preconditioned matrix MP^{-1}				
Schemes	σ_{min}	σ_{max}	Nbr Iters	Residuals
Euler	1.0	1.78	9	7.7e-9
SDIRK	0.97	1.0	2	9.64e-9

Table 2: GMRES for solving (12) for $\alpha = 10^{-6}$.

Matrix M				
Schemes	σ_{min}	σ_{max}	Nbr Iters	Residuals
Euler	1e2	4.97e4	500	1.08e-6
SDIRK	9.68	4.96e4	500	4.2e-5

Preconditioned matrix MP^{-1}				
Schemes	σ_{min}	σ_{max}	Nbr Iters	Residuals
Euler	1.0	7.76	44	8.37e-9
SDIRK	0.74	1.0	4	7.31e-9

Finally, one can use an explicit method as quadrature formula to approximate \mathcal{R}_ℓ as long as a CFL condition is satisfied, just like in the implicit method case.

4.2 Wave equation case

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

$$u_{tt} = \Delta u + \nu, \text{ on } (0, 1) \times (0, T), \quad (22)$$

with $u(x, 0) = u_0(x), u_t(x, 0) = 0, x \in (0, 1)$. We can rewrite (22) as the ODE system

$$\dot{y} = \mathcal{L}y + \mathcal{B}\nu, \quad (23)$$

with

$$y = \begin{bmatrix} u \\ u_t \end{bmatrix}, \mathcal{L} = \begin{bmatrix} 0 & I \\ \Delta & 0 \end{bmatrix} \text{ and } \mathcal{B} = \begin{bmatrix} 0 \\ I \end{bmatrix}.$$

For simplicity, we will consider homogeneous Dirichlet boundary conditions in space. In order to derive a preconditioner for the system (12), we will again look to the continuous problem. Since \mathcal{L} is no longer symmetric, the continuous analogue of M takes the form

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

Let us evaluate this integral. First note that

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

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

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

is a symmetric negative definite matrix, so we can write

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

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

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

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

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

where

$$\begin{aligned} C_{11}(s) &= A^{-1}\sin^2(sA^{1/2}), \\ C_{12}(s) &= A^{-1/2}\cos(sA^{1/2})\sin(sA^{1/2}), \\ C_{21}(s) &= C_{12}(s), \\ C_{22}(s) &= \cos^2(sA^{1/2}). \end{aligned}$$

We can now integrate these coefficients to obtain

$$I + \frac{1}{\alpha} \int_0^T e^{s\mathcal{L}}\mathcal{B}\mathcal{B}^T e^{s\mathcal{L}^T} ds = \mathcal{M} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},$$

where

$$\begin{aligned} M_{11} &= I + \frac{T}{2\alpha}A^{-1} - \frac{1}{4\alpha}A^{-3/2}\sin(2TA^{1/2}), \\ M_{12} &= \frac{1}{4\alpha}A^{-1}(I - \cos(2TA^{1/2})), \\ M_{21} &= M_{12}, \\ M_{22} &= (1 + \frac{T}{2\alpha})I + \frac{1}{4\alpha}A^{-1/2}\sin(2TA^{1/2}). \end{aligned}$$

To use \mathcal{M} as a preconditioner, we must calculate \mathcal{M}^{-1} . Since all the M_{ij} are square and they all commute, we have

$$\mathcal{M}^{-1} = \begin{bmatrix} D^{-1}M_{22} & -D^{-1}M_{12} \\ -D^{-1}M_{21} & D^{-1}M_{11} \end{bmatrix},$$

with $D = M_{11}M_{22} - M_{12}M_{21}$. Of course we do not want to compute square roots and trigonometric functions of A ; to approximate the blocks of \mathcal{M}^{-1} , we again study their symbols, i.e., the functions obtained by replacing A by a scalar $\sigma > 0$ (this is equivalent to diagonalizing the blocks and studying their eigenvalues as a function of the eigenvalues of A). Since σ is an approximation of $k^2\pi^2$, $k = 1, 2, \dots$, we observe that for highest frequencies σ ,

- $D \approx \frac{1}{4\alpha^2}(2\alpha + T)A^{-1}(TI + 2\alpha A)$;
- $D^{-1}M_{11} \approx \left[1 - \frac{1}{(2\alpha + T)}\right] I$;
- $D^{-1}M_{12} = D^{-1}M_{21} \approx 0$;
- $D^{-1}M_{22} \approx I - (TI + 2\alpha A)^{-1}$.

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

$$\widehat{\mathcal{M}}^{-1} = I - \begin{bmatrix} (aI + bA)^{-1} & 0 \\ 0 & cI \end{bmatrix}, \quad (24)$$

where $a = T$, $b = 2\alpha$ and finally $c = 1/(2\alpha + T)$. Each application of the preconditioner therefore involves only one solution of the elliptic problem $(aI + bA)v = \mathbf{f}$, just like for the heat equation.

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 $y_{tg}(x) = [(\frac{1}{2}\exp(-100(x - 1/4)^2) + \frac{1}{2}(-100(x - 3/4)^2), 0]$. We set $L = 10$, $r = 100$, and we use the same Matlab functions for computing \mathcal{R}_ℓ and $\widehat{\mathcal{P}}_\ell$.

Let us investigate the convergence speed of the linear system (12) in GMRES for $N = 1000$. 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

Table 3: GMRES for solving (12) involved (23) for $\alpha = 10^{-6}$.

Matrix M			
Schemes	cond(M)	Nbr Iters	Residuals
Euler	3.8e4	60	9.85e-10
SDIRK	3.8e4	60	9.85e-10
Preconditioned matrix MP^{-1}			
Schemes	cond(MP^{-1})	Nbr Iters	Residuals
Euler	2.59	4	3.33e-14
SDIRK	2.59	4	3.31e-14

3, where we observe that (12) converges in 84 iterations for both quadrature formulas and $\alpha = 10^{-6}$ whereas the preconditioned one converges in only 4 iterations.

Unlike for the heat equation, the high order of SDIRK does not improve the convergence speed for preconditioned and unpreconditioned system compare to Implicit Euler. Nonetheless, even for $N \in \{100, \dots, 1000\}$ the number of iterations in GMRES for the preconditioned system remains the same whether for SDIRK or Implicit Euler.

These results show that for fixed r , the discretization parameters in time do not really change the behaviour of the convergence of (12) in GMRES. We also observe that when r increases, the number of iterations of the unpreconditioned

Table 4: GMRES for solving (12) involved (23) with Implicit Euler with $\alpha = 10^{-6}$ for various r .

r	cond(M)	Nbr Iters	cond(MP^{-1})	Nbr Iters
10	4.83e2	10	2.47	5
50	1.03e4	41	2.59	4
150	7.75e4	76	2.81	3
250	1.68e5	92	2.53	3
350	2.48e5	104	2.49	3

system increases whereas the number of iterations of the preconditioned system decreases. The results are shown in Table 4.

Table 5: GMRES for solving (12) involved (23) with Implicit Euler for various α .

α	cond(M)	Nbr Iters	cond(MP^{-1})	Nbr Iters
1e-5	2.26e4	52	2.41	3
1e-4	4.48e3	34	1.64	3
1e-3	4.98e2	20	1.09	3
1e-2	5.12e1	13	1.01	3
1e-1	6.03	9	1.01	3
1e0	1.5	6	1.0	3
1e1	1.05	4	1.0	2

Finally, in Table 5, we observe that when α increases, our preconditioner remains efficient.

5 Conclusion

We introduced a new time parallel algorithm for time dependent linear quadratic optimal control problem when a cheap exponential integrator is available. We proposed two preconditioners to solves efficiently the linear equation that comes from the algorithm on the particular cases of heat and wave equations respectively.

We are currently working the behavior of the preconditioner when an explicit method is used to obtain M , while respecting the CFL condition to better understand the impact of the use of explicit and implicit methods for solving (23). Finally, we are studying the optimal control problem involving the wave equation with boundary control in one and several dimensions.

Acknowledgements

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 ALLOWAPP, grant ANR-19-CE46-0013-01 and Projet DEEPNUM, 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] Albrecht Böttcher and Sergei M. Grudsky. *Spectral Properties of Banded Toeplitz Matrices*. Society for Industrial and Applied Mathematics, 2005.
- [3] M. Gander, F. Kwok, and G. Wanner. Constrained optimization: From Lagrangian mechanics to optimal control and PDE constraints. in *Optimization with PDE Constraints*, Springer, pages 151–202, 2014.
- [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] Andrea Toselli and Olof Widlund. *Domain Decomposition Methods – Algorithms and Theory*. Springer-Verlag, Berlin, 2005.
- [18] Colin S.C. Tsang. Preconditioners for linear parabolic optimal control problems. Master’s thesis, Hong Kong Baptist University, Hong Kong, China, August 2017.
- [19] 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.