arXiv:2302.06406v1 [math.NA] 13 Feb 2023

ON GENERALIZED PRECONDITIONERS FOR TIME-PARALLEL PARABOLIC OPTIMAL CONTROL

ARNE BOUILLON*†, GIOVANNI SAMAEY*, AND KARL MEERBERGEN*

Abstract. The ParaDiag family of algorithms solves differential equations by using preconditioners that can be inverted in parallel through diagonalization. In the context of optimal control of linear parabolic PDEs, the state-of-the-art ParaDiag method is limited to solving self-adjoint problems with a tracking objective. We propose three improvements to the ParaDiag method: the use of alpha-circulant matrices to construct an alternative preconditioner, a generalization of the algorithm for solving non-self-adjoint equations, and the formulation of an algorithm for terminal-cost objectives. We present novel analytic results about the eigenvalues of the preconditioned systems for all discussed ParaDiag algorithms in the case of self-adjoint equations, which proves the favorable properties the alpha-circulant preconditioner. We use these results to perform a theoretical parallel-scaling analysis of ParaDiag for self-adjoint problems. Numerical tests confirm our findings and suggest that the self-adjoint behavior, which is backed by theory, generalizes to the non-self-adjoint case. We provide a sequential, open-source reference solver in MATLAB for all discussed algorithms.

Key words. Optimal control, ParaDiag algorithm, preconditioning, parallel-in-time

MSC codes. 49M05, 65F08, 65K10, 65Y05

1. Introduction.

We are interested in optimal-control problems of the form

$$\min_{y,u} J(y,u) \quad \text{s.t.} \quad y_t = g(y) + u, \quad y(0) = y_{\text{init}} \tag{1.1}$$

over time $[0,T]$. Here, y represents a space- and time-dependent state variable with initial condition y_{init} evolving under the influence of a (potentially non-linear) function g, while u is a control input with which y is steered. We want to choose u to minimize J, which is either a *tracking* or a *terminal-cost* objective function

$$J(y,u) = \begin{cases} \text{Tracking:} & \frac{1}{2}\int_0^T \left\|y(t) - y_{\text{d}}(t)\right\|_2^2 \mathrm{d}t + \frac{\gamma}{2}\int_0^T \left\|u(t)\right\|_2^2 \mathrm{d}t \\ \text{Terminal cost:} & \frac{1}{2}\left\|y(T) - y_{\text{target}}\right\|_2^2 + \frac{\gamma}{2}\int_0^T \left\|u(t)\right\|_2^2 \mathrm{d}t. \end{cases} \tag{1.2}$$

Tracking objectives aim to keep y as close as possible to a trajectory y_{d}, while terminal cost only requires the final position to be close to some y_{target}. The factor $\gamma > 0$ regularizes the control term and may also model the practical cost of control. We space-discretize this problem (using bold-faced vectors) and consider only linear functions $\boldsymbol{g}(\boldsymbol{y}) = -K\boldsymbol{y}$ for some $K \in \mathbb{C}^{M\times M}$. A solution to (1.1) satisfies the boundary value problem (BVP)

$$\boldsymbol{y}'(t) = -K\boldsymbol{y}(t) - \boldsymbol{\lambda}(t)/\gamma, \quad \boldsymbol{y}(0) = \boldsymbol{y_{\text{init}}} \tag{1.3a}$$

$$\begin{cases} \text{Tracking:} & \boldsymbol{\lambda}'(t) = K^*\boldsymbol{\lambda}(t) + \boldsymbol{y_{\text{d}}}(t) - \boldsymbol{y}(t), \quad \boldsymbol{\lambda}(T) = \boldsymbol{0} \\ \text{Terminal cost:} & \boldsymbol{\lambda}'(t) = K^*\boldsymbol{\lambda}(t), \quad \boldsymbol{\lambda}(T) = \boldsymbol{y}(T) - \boldsymbol{y_{\text{target}}} \end{cases} \tag{1.3b}$$

consisting of coupled equations in the state $\boldsymbol{y}(t)$ and the *adjoint* state $\boldsymbol{\lambda}(t) \coloneqq -\gamma\boldsymbol{u}(t)$, with one initial and one terminal condition [10, 33, 17]. To solve (1.3), we consider *parallel-in-time* methods for optimal control. These are inspired by time-parallel initial-value problem (IVP) solvers, which overcome the inherently serial nature of time integration. Leveraging these techniques enables the construction of algorithms for optimal control which scale well in parallel when increasing the amount of work in the time dimension.

*NUMA, Department of Computer Science, KU Leuven, Leuven, Belgium
†Corresponding author: arne.bouillon@kuleuven.be

Some parallel-in-time approaches for the optimal-control problem (1.1) use the *direct-adjoint* optimization loop [14, 28], where all embedded IVP solves are tackled using time-parallel methods such as PFASST [7] or the well-known Parareal algorithm [20]. Others use the system (1.3); an example is ParaOpt [10], inspired by Parareal. In this paper, we will expand on the time-parallel algorithm proposed for self-adjoint tracking problems in [33]. Belonging to the ParaDiag family [23, 12, 22, 11, 32], this algorithm constructs a discretized *all-at-once* system of (1.3) and solves it iteratively, using a preconditioner that is invertible in parallel.

This paper is organized as follows. We start from the method in [33], whose current preconditioner P is limited as we will see; generalizations to new situations are found in sections 2 and 3. Section 2 examines the tracking case, containing an updated *alpha-circulant* preconditioner $P(\alpha)$, analytic expressions for the preconditioned eigenvalues of ParaDiag and an extension of the method to non-self-adjoint problems. Section 3 introduces a novel ParaDiag method for terminal-cost objective functions, again featuring an analytic eigenvalue analysis. In section 4, we use these results as a theoretical basis to predict weak scalability of both ParaDiag methods for self-adjoint, dissipative equations. The numerical results in section 5 confirm this scalability for both self-adjoint and non-self-adjoint equations. In section 6, we conclude and propose further research directions.

As a final note, we mention the very recent paper [19], which also constructs alpha-circulant preconditioners for self-adjoint tracking problems. Our approach is very different, offering a more direct generalization of [33]. The analysis presented here results in exact analytical eigenvalues (both for our method and for [33]) instead of a bound and our method straightforwardly generalizes to non-self-adjoint equations.

2. ParaDiag for tracking objectives. This section considers the tracking objective in (1.2), for which a ParaDiag procedure (limited to problems with self-adjoint $K = K^*$) is described in [33]. We review this method in subsection 2.1. Subsequently, subsection 2.2 looks at the limiting case $T \to 0$, in which ParaDiag is discovered to lack robustness. We counteract this with an improvement to the preconditioner, using novel analytic results in subsections 2.3 and 2.4 to prove its more favorable properties. Subsection 2.5 concludes by proposing a generalization to problems where $K \neq K^*$.

For ease of exposition, we use an implicit-Euler time discretization with time step τ throughout this paper, although other discretizations can be treated similarly.

2.1. Existing method. The existing algorithm requires a self-adjoint K – that is, $K = K^*$. The all-at-once system for (1.3) then reads [33]

$$A \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{\lambda} \end{bmatrix} \coloneqq \left(\begin{bmatrix} B & \frac{\tau I}{\gamma} \\ -\tau I & B^\top \end{bmatrix} \otimes I + \tau \begin{bmatrix} I & \\ & I \end{bmatrix} \otimes K \right) \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b_1} \\ \boldsymbol{b_2} \end{bmatrix}, \tag{2.1}$$

where I is used to denote identity matrices – each of appropriate size – and

$$B = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}, \quad \boldsymbol{b_1} = \begin{bmatrix} \boldsymbol{y_{\text{init}}}^\top & 0 & \dots & 0 \end{bmatrix}^\top, \quad \text{and} \quad \boldsymbol{b_2} = -\tau \boldsymbol{y_d}. \tag{2.2}$$

In (2.1), we have grouped the discretized unknowns $\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1^\top & \boldsymbol{y}_2^\top & \cdots & \boldsymbol{y}_{L-1}^\top \end{bmatrix}^\top$ and $\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_1^\top & \boldsymbol{\lambda}_2^\top & \cdots & \boldsymbol{\lambda}_{L-1}^\top \end{bmatrix}^\top$, where $\boldsymbol{y}_l$ and $\boldsymbol{\lambda}_l$ are approximations to $\boldsymbol{y}(t = l\tau)$ and $\boldsymbol{\lambda}(t = l\tau)$. The relation of $\boldsymbol{y_d}$ to $\boldsymbol{y_d}(t)$ is analogous. $\boldsymbol{y}_0$ and $\boldsymbol{\lambda}_L$ are known from (1.3), while $\boldsymbol{y}_L$ and $\boldsymbol{\lambda}_0$ are not required to compute any other quantities in this discretization – hence they are left out of (2.1). We introduce $\widehat{L} \coloneqq L - 1$ such that B is $\widehat{L} \times \widehat{L}$.

Next, a rescaling is applied[1]: $\widehat{\boldsymbol{\lambda}} \coloneqq \boldsymbol{\lambda}/\sqrt{\gamma}$ and $\widehat{\boldsymbol{b}}_{\boldsymbol{2}} \coloneqq \boldsymbol{b}_{\boldsymbol{2}}/\sqrt{\gamma}$. Solving

$$\widehat{A}\begin{bmatrix}\boldsymbol{y}\\ \widehat{\boldsymbol{\lambda}}\end{bmatrix} \coloneqq \left(\begin{bmatrix} B & \frac{\tau I}{\sqrt{\gamma}} \\ -\frac{\tau I}{\sqrt{\gamma}} & B^\top\end{bmatrix} \otimes I + \tau \begin{bmatrix} I & \\ & I\end{bmatrix} \otimes K\right)\begin{bmatrix}\boldsymbol{y}\\ \widehat{\boldsymbol{\lambda}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{b_1}\\ \widehat{\boldsymbol{b}}_{\boldsymbol{2}}\end{bmatrix} \tag{2.3}$$

for $\boldsymbol{y}$ and $\widehat{\boldsymbol{\lambda}}$ is done iteratively, with a solver such as GMRES [25]. By replacing B with a circulant approximation C, we construct the preconditioner

$$P = \begin{bmatrix} C & \frac{\tau I}{\sqrt{\gamma}} \\ -\frac{\tau I}{\sqrt{\gamma}} & C^\top\end{bmatrix} \otimes I + \tau \begin{bmatrix} I & \\ & I\end{bmatrix} \otimes K \quad \text{with} \quad C = \begin{bmatrix} 1 & & & -1 \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}. \tag{2.4}$$

As shown later, P is invertible in parallel. Any circulant $C \in \mathbb{R}^{\widehat{L}\times\widehat{L}}$ diagonalizes as

$$C = \mathbb{F}^* D \mathbb{F} \quad \text{with} \quad D = \operatorname{diag}(\sqrt{\widehat{L}}\mathbb{F}\boldsymbol{c}_1), \tag{2.5}$$

as can be found in [3]. Here, $\mathbb{F} = \{\mathrm{e}^{2\pi \mathrm{i} jk/\widehat{L}}/\sqrt{\widehat{L}}\}_{j,k=0}^{\widehat{L}-1}$ is the discrete Fourier matrix and $\boldsymbol{c}_1$ is C's first column. The authors in [33] first factorize

$$P = \left(\begin{bmatrix}\mathbb{F}^* & \\ & \mathbb{F}^*\end{bmatrix} \otimes I\right)\left(\begin{bmatrix} D & \frac{\tau I}{\sqrt{\gamma}} \\ -\frac{\tau I}{\sqrt{\gamma}} & D^*\end{bmatrix} \otimes I + \tau \begin{bmatrix} I & \\ & I\end{bmatrix} \otimes K\right)\left(\begin{bmatrix}\mathbb{F} & \\ & \mathbb{F}\end{bmatrix} \otimes I\right) \tag{2.6}$$

and then argue for a diagonalization $\left[\begin{smallmatrix} D & \tau I/\sqrt{\gamma} \\ -\tau I/\sqrt{\gamma} & D^*\end{smallmatrix}\right] = WHW^{-1}$ with $W = \left[\begin{smallmatrix} I & S_2 \\ S_1 & I\end{smallmatrix}\right]$, where H and $S_{\{1,2\}}$ are diagonal (see [33] for details). Defining $V \coloneqq \left[\begin{smallmatrix}\mathbb{F}^* & \\ & \mathbb{F}^*\end{smallmatrix}\right]W$,

$$P^{-1} = (V \otimes I)(H \otimes I + \tau I \otimes K)^{-1}(V^{-1} \otimes I). \tag{2.7}$$

Algorithm 2.1 summarizes the procedure for solving (2.1), including how to multiply by P^{-1} in parallel.

Algorithm 2.1 ParaDiag for solving the tracking problem (2.1), based on [33]

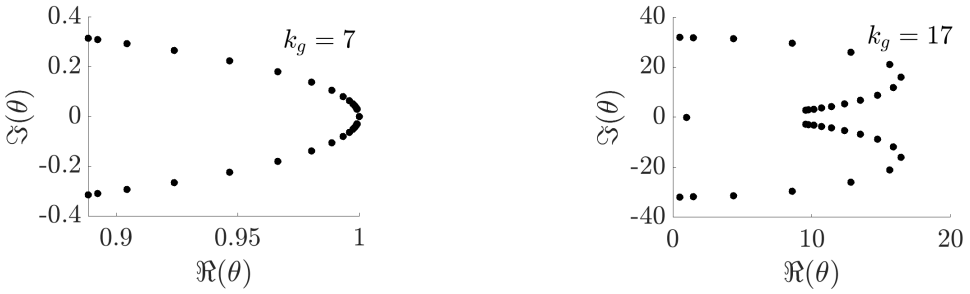
Input: Vectors $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$ defined by (2.2)
Self-adjoint matrix K characterising the problem by (1.3)
Matrices H and W following from the time discretization
Output: The vectors $\boldsymbol{y}$ and $\boldsymbol{\lambda}$ that solve (2.1)

1: Rescale $\widehat{\boldsymbol{b}}_{\boldsymbol{2}} = \boldsymbol{b}_{\boldsymbol{2}}/\sqrt{\gamma}$.
2: Solve (2.3) for $\boldsymbol{y}$ and $\widehat{\boldsymbol{\lambda}}$ using an iterative method, with preconditioner P from (2.4). When asked to compute $\left[\begin{smallmatrix}\boldsymbol{x}\\ \boldsymbol{z}\end{smallmatrix}\right] = P^{-1}\left[\begin{smallmatrix}\boldsymbol{v}\\ \boldsymbol{w}\end{smallmatrix}\right]$:
3: Calculate $\boldsymbol{r_1} \coloneqq (\mathbb{F} \otimes I)\boldsymbol{v}$ and $\boldsymbol{s_1} \coloneqq (\mathbb{F} \otimes I)\boldsymbol{w}$ using the (parallel) FFT.
4: Calculate $\boldsymbol{q_2} \coloneqq (W^{-1} \otimes I)\left[\begin{smallmatrix}\boldsymbol{r_1}\\ \boldsymbol{s_1}\end{smallmatrix}\right]$.
5: For $l = \{1, \ldots, 2\widehat{L}\}$, solve (in parallel)

$$\boldsymbol{q}_{\boldsymbol{3},l} \coloneqq (h_{l,l}I + \tau K)^{-1}\boldsymbol{q}_{\boldsymbol{2},l} \tag{2.8}$$

and partition the variables as $\left[\begin{smallmatrix}\boldsymbol{r}_3\\ \boldsymbol{s}_3\end{smallmatrix}\right] \coloneqq \boldsymbol{q_3}$.
6: Calculate $\left[\begin{smallmatrix}\boldsymbol{r}_4\\ \boldsymbol{s}_4\end{smallmatrix}\right] \coloneqq (W \otimes I)\left[\begin{smallmatrix}\boldsymbol{r}_3\\ \boldsymbol{s}_3\end{smallmatrix}\right]$.
7: Calculate $\boldsymbol{x} = (\mathbb{F}^* \otimes I)\boldsymbol{r_4}$ and $\boldsymbol{z} = (\mathbb{F}^* \otimes I)\boldsymbol{s_4}$ using the (parallel) FFT.
8: Rescale $\boldsymbol{\lambda} = \sqrt{\gamma}\widehat{\boldsymbol{\lambda}}$.

[1] In [33], $\boldsymbol{y}$ and $\boldsymbol{b_1}$ are rescaled, but this is completely equivalent.



(a) $T = 1$ (b) $T = 10^{-4}$

Fig. 2.1: Eigenvalues θ of $P^{-1}\widehat{A}$ and iteration count k_g of ParaDiag for the example in subsection 2.2, using GMRES with relative tolerance 10^{-6}. Figure 2.1a mimics [33, Figure 6], but Figure 2.1b discovers issues when T is small.

2.2. The small-T limit and alpha-circulants. When using iterative linear-system solvers, convergence speed often depends substantially on the distribution of the eigenvalues of the preconditioned matrix [29] – in our case, of $P^{-1}\widehat{A}$ (while there are exceptions such as CGN, which relies on singular values instead, the rest of this paper will assume an iterative solver whose behavior is mainly determined by the eigenvalues). Specifically, eigenvalues that are clustered together and lie far enough from 0 are beneficial. While we stress that convergence is not exclusively determined by eigenvalues (for an extreme example, see [15]), they do play an important role, and studying them to make an educated guess about convergence is commonplace [24]. In particular, the authors of [33] performed an empirical eigenvalue study for the method in subsection 2.1 and also compared the GMRES and BiCGStab [30] iterative solvers, with the former proving to be faster. To illustrate the importance of eigenvalues, we choose GMRES and follow [33] in considering the discretized Laplacian on spatial domain $\Omega = [0, 1]$ with isolated boundary

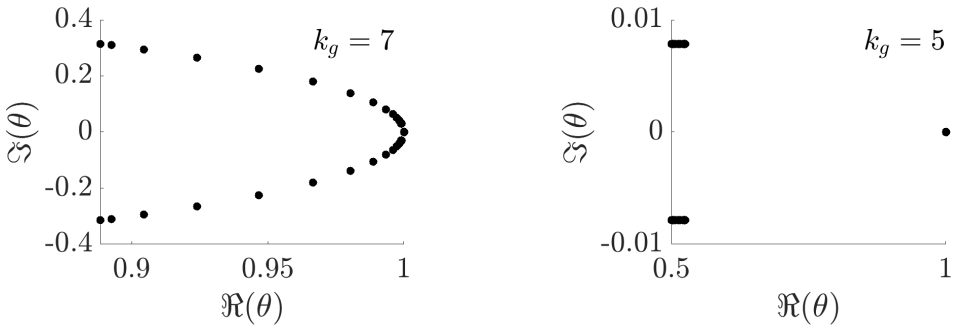
$$K = \frac{1}{\Delta x^2}\begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 2 & -1 \\ & & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{M\times M}, \tag{2.9}$$

where $\Delta x = 1/M$. We use $M = 16$, $L = 128$ and $\gamma = 10^{-5}$ as in [33] and set $y_{\mathrm{d}}(x, t) = y_{\mathrm{init}}(x) = \exp(-100(x - 0.5)^2)$ – these do not impact the preconditioned eigenvalues, but may still influence the iteration count. We study this in two regimes. Figure 2.1a uses [33]'s time horizon $T = 1$. The eigenvalues are clustered around unity and the GMRES[2] iteration count k_g is low. When reducing the time interval by setting $T = 10^{-4}$, however, Figure 2.1b reveals large variations in the eigenvalues and an increased iteration count. All this happens even though this case is well-behaved: since the time steps are smaller, two subsequent discretization points barely differ.

Later, in subsection 2.3, we discover the origin of this behavior for different Ts. We first propose an altered preconditioner $P(\alpha)$ with a parameter $\alpha \in \mathbb{C}$. Let

$$P(\alpha) = \begin{bmatrix} C(\alpha) & \tau\frac{I}{\sqrt{\gamma}} \\ -\tau\frac{I}{\sqrt{\gamma}} & C^*(\alpha) \end{bmatrix} \otimes I + \tau \begin{bmatrix} I & \\ & I \end{bmatrix} \otimes K \quad \text{with} \quad C(\alpha) = \begin{bmatrix} 1 & & & -\alpha \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \tag{2.10}$$

[2]We use a relative GMRES tolerance of 10^{-6}, which is MATLAB's default, throughout this paper.



(a) $T = 1$ (b) $T = 10^{-4}$

Fig. 2.2: Eigenvalues θ of $P^{-1}(-1)\widehat{A}$ and iteration count k_g of ParaDiag for the example in subsection 2.2, using GMRES with relative tolerance 10^{-6}

where $C(\alpha)$ is not circulant, but *alpha-circulant* (circulant except that super-diagonal entries have been multiplied by some $\alpha \neq 0$). Alpha-circulants diagonalize as [3]

$$C(\alpha) = VD(\alpha)V^{-1} \quad \text{with} \quad V = \Gamma_\alpha^{-1}\mathbb{F}^* \quad \text{and} \quad D(\alpha) = \operatorname{diag}(\sqrt{\widehat{L}}\mathbb{F}\Gamma_\alpha \boldsymbol{c}_1(\alpha)), \tag{2.11}$$

where $\Gamma_\alpha = \operatorname{diag}(1, \alpha^{1/\widehat{L}}, \cdots, \alpha^{(\widehat{L}-1)/\widehat{L}})$. The idea of using alpha-circulants occurs in the IVP literature [12, 22], but while IVPs can use $\alpha \approx 0$ such that $P(\alpha) \approx \widehat{A}$, our $P(\alpha)$ comes with severe limitations. To successfully perform a factorization like (2.6), $C(\alpha)$ and $C^*(\alpha)$ must be simultaneously diagonalizable. This is only the case when

$$\Gamma_\alpha^{-1} = \Gamma_\alpha^* \Leftrightarrow |\alpha| = 1. \tag{2.12}$$

Contrary to the IVP situation, under the constraint (2.12) it is far less clear that setting $\alpha \neq 1$ is advantageous. We choose[3] $\alpha = -1$ to reiterate our previous experiment. Figure 2.2 shows that $P(-1)$ does not display the same defects for small T as did $P = P(1)$: the case $T = 1$ looks identical, but for $T = 10^{-4}$ the eigenvalues cluster instead of dispersing and the GMRES iteration count remains low. In the next subsection, we unravel this fact analytically.

2.3. Analytic eigenvalue expressions. We study the preconditioner $P(\alpha)$ for the significant cases $\alpha = \pm 1$. We start by rescaling the system (2.3) by multiplying by $I \otimes (b_{0,0}I + \tau K)^{-1}$, where $b_{0,0}$ is B's top-left element. This yields the system

$$\widehat{A}_{\mathrm{p}} \begin{bmatrix} \boldsymbol{y} \\ \widehat{\boldsymbol{\lambda}} \end{bmatrix} := \left[\begin{array}{cccc|ccc} I & & & & \Psi & & \\ -\Phi & I & & & & \Psi & \\ & \ddots & \ddots & & & & \ddots \\ & & -\Phi & I & & & \Psi \\ \hline -\Psi & & & & I & -\Phi & \\ & \ddots & & & & \ddots & \ddots \\ & & -\Psi & & & & I \\ & & & -\Psi & & & -\Phi \end{array}\right] \begin{bmatrix} \boldsymbol{y} \\ \widehat{\boldsymbol{\lambda}} \end{bmatrix} = \widehat{\boldsymbol{b}}_{\mathbf{p}}, \tag{2.13}$$

with $\widehat{b}_{\mathrm{p}}$ a rescaled version of $\widehat{b}$. For this implicit-Euler discretization, $\Phi = (I + \tau K)^{-1}$ and $\Phi = \tau/\sqrt{\gamma}(I + \tau K)^{-1}$. We perform the same rescaling on $P(\alpha)$, forming $P_{\mathrm{p}}(\alpha)$. The preconditioned matrix $P^{-1}(\alpha)\widehat{A}$ is then equal to $P_{\mathrm{p}}^{-1}(\alpha)\widehat{A}_{\mathrm{p}}$.

[3] (-1)-circulants have also been called *skew-circulant* or *negacyclic* matrices [6].

We make use of the self-adjointness of K (and, thus, of Φ and Ψ) to drastically reduce the number of parameters that determine the eigenvalues of $P_{\mathrm{p}}^{-1}(\alpha)\widehat{A}_{\mathrm{p}}$. Write $K = V^{-1}\Sigma V$ with diagonal Σ. Φ and Ψ are also diagonalizable by V, so $P_{\mathrm{p}}^{-1}(\alpha)\widehat{A}_{\mathrm{p}}$ can be decomposed into *separate* versions for each of K's eigenvalues $\sigma \in \operatorname{diag}\Sigma$. The eigenvalues of $P_{\mathrm{p}}^{-1}(\alpha)\widehat{A}_{\mathrm{p}}$ are the union of those of the matrices

$$P_\sigma^{-1}(\alpha)\widehat{A}_\sigma \coloneqq \left[\begin{array}{ccc|ccc} 1 & & -\alpha\varphi & \psi & & \\ -\varphi & \ddots & & & \ddots & \\ & \ddots & 1 & & & \psi \\ \hline -\psi & & & 1 & -\varphi & \\ & \ddots & & & \ddots & \ddots \\ & & -\psi & & & -\varphi \end{array}\right]^{-1} \left[\begin{array}{ccc|ccc} 1 & & & \psi & & \\ -\varphi & \ddots & & & \ddots & \\ & \ddots & 1 & & & \psi \\ \hline -\psi & & & 1 & -\varphi & \\ & \ddots & & & \ddots & \ddots \\ & & -\psi & & & -\varphi \end{array}\right] \tag{2.14}$$

for all σ, with $\varphi = (1+\tau\sigma)^{-1}$ and $\psi = \frac{\tau}{\sqrt{\gamma}}\varphi$. Then, by writing $\widehat{\sigma} \coloneqq \tau\sigma$ and $\widehat{\gamma} \coloneqq \frac{\tau}{\sqrt{\gamma}}$,

$$\varphi = (1+\widehat{\sigma})^{-1} \quad \text{and} \quad \psi = \widehat{\gamma}(1+\widehat{\sigma})^{-1}, \tag{2.15}$$

which eliminates τ as an independent variable. The only parameters left that are relevant to the eigenvalues of $P^{-1}(\alpha)\widehat{A}$ are the different $\widehat{\sigma}$s (time step–rescaled eigenvalues of K), $\widehat{\gamma}$ (a time step–rescaled value indicating "how much control" is present) and L (the number of time steps, dictating the size $\widehat{L} = L-1$ of the matrices).

In summary, each eigenvalue σ of K defines a preconditioned matrix $P_\sigma^{-1}(\alpha)\widehat{A}_\sigma$, whose eigenvalues are also eigenvalues of $P^{-1}(\alpha)\widehat{A}$. These eigenvalues influence the iterative solver's convergence, as explained in subsection 2.2. The matrices $P_\sigma^{-1}(\alpha)\widehat{A}_\sigma$ can be rewritten as the identity plus a low-rank term:

$$\begin{aligned} P_\sigma^{-1}(\alpha)\widehat{A}_\sigma &= P_\sigma^{-1}(\alpha)(P_\sigma(\alpha) + (\widehat{A}_\sigma - P_\sigma(\alpha))) \\ &= I + P_\sigma^{-1}(\alpha)(\widehat{A}_\sigma - P_\sigma(\alpha)) =: I + P_\sigma^{-1}(\alpha)R_\sigma, \end{aligned} \tag{2.16}$$

from which it follows that the eigenvalues θ of $P_\sigma^{-1}(\alpha)\widehat{A}_\sigma$ are equal to one plus the eigenvalues ω of $P_\sigma^{-1}(\alpha)R_\sigma$. The latter are characterized by Theorem 2.1.

THEOREM 2.1. *Let $\widehat{L} > 3$, $\alpha = \pm 1$ and $\varphi, \psi \in \mathbb{R}\backslash\{0\}$. The $2\widehat{L} \times 2\widehat{L}$ matrix*

$$M = \overbrace{\left[\begin{array}{ccc|ccc} 1 & & -\alpha\varphi & \psi & & \\ -\varphi & \ddots & & & \ddots & \\ & \ddots & 1 & & & \psi \\ \hline -\psi & & & 1 & -\varphi & \\ & \ddots & & & \ddots & \ddots \\ & & -\psi & & & -\varphi \end{array}\right]^{-1}}^{=P_\sigma^{-1}(\alpha)} \overbrace{\left[\begin{array}{ccc|ccc} & & & & & \alpha\varphi \\ & & & & & \\ & & & & & \\ \hline & & & & & \\ & & & & & \\ & & & & & \alpha\varphi \end{array}\right]}^{=R_\sigma} \tag{2.17}$$

has only two potentially non-zero eigenvalues $\omega_{\{1,2\}}$. Specifically,

$$\omega_{\{1,2\}} = \frac{1}{z_2 - z_1}\left(\frac{z_1 - \varphi \pm \psi\mathrm{i}}{1-\alpha z_1^{\widehat{L}}} - \frac{z_2 - \varphi \pm \psi\mathrm{i}}{1-\alpha z_2^{\widehat{L}}}\right) \tag{2.18}$$

where

$$z_{\{1,2\}} = \frac{1+\varphi^2+\psi^2 \pm \sqrt{(1+\varphi^2+\psi^2)^2 - 4\varphi^2}}{2\varphi}. \tag{2.19}$$

Proof. The proof of this theorem is given in Appendix A. ∎

COROLLARY 2.2. *Consider a tracking-type all-at-once system with implicit-Euler time discretization* (2.3) *where* $L > 4$ *and* K *is self-adjoint with eigenvalues* $\{\sigma_m\}_{m=1}^{M}$. *When using ParaDiag with preconditioner* $P(\alpha = \pm 1)$ *(see* (2.10)*), the eigenvalues of the preconditioned system matrix* $P(\alpha)^{-1}\widehat{A}$ *are all either unity or equal to*

$$\theta_{m,\{1,2\}} = 1 + \omega_{m,\{1,2\}} \tag{2.20}$$

where $\omega_{m,\{1,2\}}$ *are given by the formula* (2.18)*, having filled in* $\varphi = (1 + \tau\sigma_m)^{-1}$ *and* $\psi = \frac{\tau}{\sqrt{\gamma}}(1 + \tau\sigma_m)^{-1}$.

COROLLARY 2.3. *Consider a tracking-type all-at-once system with implicit-Euler time discretization* (2.3) *where* $L > 4$ *and* K *is self-adjoint. Denote by* $\mathcal{D}_{0.5,+}$ *the right half of a disk in the complex plane, centered at* 0.5 *and with radius* 0.5. *When using ParaDiag with preconditioner* $P(\alpha = -1)$ *(see* (2.10)*), if* $0 < \varphi < 1$ *for all* φ *(which occurs whenever* K *is positive definite), all eigenvalues of the preconditioned matrix* $P^{-1}(\alpha)\widehat{A}$ *lie within* $\mathcal{D}_{0.5,+}$.

Proof. Lemma C.2(c) shows that the real parts of these eigenvalues are larger than 0.5, while Lemma C.2(d) proves that their distances from the point 0.5 are less than 0.5. Together, these bounds delineate the region $\mathcal{D}_{0.5,+}$. ∎

2.4. Interpreting the eigenvalue results. It is possible to visualize Corollary 2.2 to gain more insight into how the two preconditioners (that is, $\alpha = 1$ and $\alpha = -1$) perform, as well as how they compare. Recall from subsection 2.3 that every eigenvalue σ_m of K corresponds to two non-unity eigenvalues $\theta_{m,\{1,2\}}$ of the preconditioned system matrix, which are complex conjugates of each other. Figure 2.3 plots the θ with positive imaginary part for the cases $\alpha = \pm 1$ in two distinct ways. This section considers dissipative problems, where K is positive definite and hence $\sigma > 0$.

Figures 2.3a and 2.3b are based on the view that $\widehat{\gamma}$ is typically known (one can set the regularization parameter γ and the time step τ), while the eigenvalues σ (and thus $\widehat{\sigma} = \tau\sigma$) could lie anywhere. For each $\widehat{\gamma}$ value, these figures mark the preconditioned eigenvalues for a whole range of $\widehat{\sigma} > 0$ options. We can see Corollary 2.3 in action: for $\alpha = -1$ the eigenvalues lie inside $\mathcal{D}_{0.5,+}$ while, for $\alpha = 1$, they lie outside it. Figures 2.3c and 2.3d add $\widehat{\sigma}$ as a dimension to gain more insight into its influence.

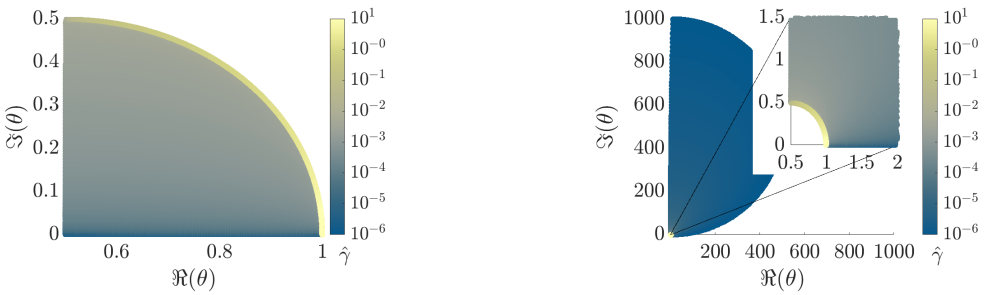
We stress that from these figures, little if anything can be said about how ParaDiag scales when increasing L, the topic of section 4. Instead, we conclude that *for a fixed number of time steps*, ParaDiag can be expected to converge quickly unless

- the equation in the absence of control evolves slowly *relatively to the size of the time interval* ($\widehat{\sigma} \approx 0$); and
- there is little control *relatively to the size of the time interval* ($\widehat{\gamma} \approx 0$).

If these conditions for potentially slow convergence are met, the difference between α values becomes important.

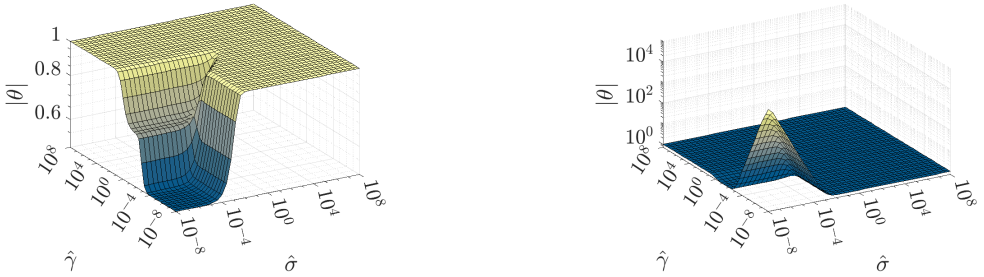
- When $\alpha = 1$, the smaller $\widehat{\sigma}$ and $\widehat{\gamma}$ become, the more θ blows up. This will lead to preconditioned eigenvalues that lie far away from each other, resulting in slow convergence.
- When $\alpha = -1$, small values of $\widehat{\sigma}$ and $\widehat{\gamma}$ slightly pull the eigenvalues away from unity. However, they always stay relatively close due to Corollary 2.3. In addition, clustering may even become better for very small $\widehat{\sigma}$ and $\widehat{\gamma}$: Figure 2.3a shows that the worst clustering occurs at intermediate $\widehat{\gamma}$s.

This analysis explains our observations in Figures 2.1 and 2.2. When $T = 1$, $\widehat{\sigma}$ and $\widehat{\gamma}$ are large enough that the eigenvalues lie close to the edge of $\mathcal{D}_{0.5,+}$ for both $\alpha = \pm 1$. When $T = 10^{-4}$, $\widehat{\sigma}$ and $\widehat{\gamma}$ are very small and $\alpha = -1$ clusters, while $\alpha = 1$ disperses.



(a) θ when $\alpha = -1$

(b) θ when $\alpha = 1$



(c) $|\theta|$ when $\alpha = -1$

(d) $|\theta|$ when $\alpha = 1$

Fig. 2.3: Non-unity eigenvalue θ of $P^{-1}(\alpha)\widehat{A}$ with $\Im(\theta) \geq 0$ for $L = 1000$. Note the scale of the axes in the figures on the right, due to the effects of $\alpha = 1$. The color maps used throughout this text are designed by [5] to be color-vision–deficiency friendly.

Specifically for the GMRES method, it is possible to harness Corollary 2.3 into an upper bound on the convergence of the iterative method.

THEOREM 2.4. *Consider the tracking-type all-at-once system with implicit-Euler time discretization* (2.3) *where* $L > 4$ *and* K *is self-adjoint. When using ParaDiag with* GMRES *preconditioned by* $P(-1)$ *(see* (2.10)*) and if* $0 < \varphi < 1$ *for all* φ *(which occurs whenever* K *is positive definite), the following holds. For any* $0 < \rho < 2$*, there exists a* $\kappa_\rho > 0$ *such that the residual* $\boldsymbol{r}^k$ *at* GMRES *iteration* k *satisfies*

$$\left\|\boldsymbol{r}^k\right\|_2 / \left\|\boldsymbol{r}^0\right\|_2 \leq \kappa(V)\kappa_\rho \rho^{-k}. \tag{2.21}$$

Here, $\kappa(V)$ *is the condition number of the eigenvector matrix* V *of* $P(-1)^{-1}\widehat{A}$*. The* GMRES *residual decreases exponentially with a mesh- and problem-independent factor.*

Proof. From [29], we get the formula

$$\left\|\boldsymbol{r^k}\right\|_2 / \left\|\boldsymbol{r^0}\right\|_2 \leq \kappa(V) \inf_{p_k \in \mathbb{P}_k} \sup_{\sigma \in \Sigma} |p_k(\sigma)| \tag{2.22}$$

which asks us to solve a polynomial-approximation problem: find a degree-k polynomial that takes the value 1 at the origin and, yet, is as small as possible on all the eigenvalues of the preconditioned matrix. However, if we want a generally applicable bound, we do not know these eigenvalues. Luckily, we have Corollary 2.3: if we can find a polynomial that is small on the *entire* semi-disk $\mathcal{D}_{0.5,+}$, then a fortiori, it must also be small on whatever eigenvalues a specific problem happens to generate.

To eliminate the explicit condition $p(0) = 1$, we take the following steps. If we can find a degree-k polynomial $\widehat{p}$ that satisfies $\widehat{p}(0) = 0$ and approximates 1 on $\mathcal{D}_{0.5,+}$, it has the same error as $p(z) = 1 - \widehat{p}(z)$. Since $\widehat{p}(0) = 0$, it must be that $\widehat{p}(z) = zq(z)$, where q is of degree $k-1$. If we find a polynomial q that approximates $1/z$ on $\mathcal{D}_{0.5,+}$, we have a function $\widehat{p}(z) = zq(z)$ which approximates 1 with no higher error than that of q in approximating $1/z$ (this follows easily from the fact that $|z| \leq 1$).

In summary, (2.22)'s rightmost factor is bounded by the best degree-$(k-1)$ polynomial-approximation error to $1/z$ on $\mathcal{D}_{0.5,+}$, which is bounded by Lemma C.3. ∎

2.5. Generalizing past self-adjoint problems. We now extend ParaDiag to the more general, non-self-adjoint setting where K and K^* are not necessarily equal, resulting in the optimality system (1.3).

When self-adjointness is not assumed, (2.1) becomes

$$A\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{\lambda}\end{bmatrix} \coloneqq \left(\begin{bmatrix} B & \tau\frac{I}{\gamma} \\ -\tau I & B^\top\end{bmatrix} \otimes I + \tau \begin{bmatrix} I\otimes K & \\ & I\otimes K^*\end{bmatrix}\right)\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{\lambda}\end{bmatrix} = \begin{bmatrix}\boldsymbol{b_1}\\ \boldsymbol{b_2}\end{bmatrix} \tag{2.23}$$

and, after rescaling,

$$\widehat{A}\begin{bmatrix}\boldsymbol{y}\\ \widehat{\boldsymbol{\lambda}}\end{bmatrix} \coloneqq \left(\begin{bmatrix} B & \tau\frac{I}{\sqrt{\gamma}} \\ -\tau\frac{I}{\sqrt{\gamma}} & B^\top\end{bmatrix} \otimes I + \tau \begin{bmatrix} I\otimes K & \\ & I\otimes K^*\end{bmatrix}\right)\begin{bmatrix}\boldsymbol{y}\\ \widehat{\boldsymbol{\lambda}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{b_1}\\ \widehat{\boldsymbol{b_2}}\end{bmatrix}. \tag{2.24}$$

By a similar procedure as before, we suggest an alpha-circulant preconditioner

$$P(\alpha) = \begin{bmatrix} C(\alpha) & \frac{\tau I}{\sqrt{\gamma}} \\ -\frac{\tau I}{\sqrt{\gamma}} & C^*(\alpha)\end{bmatrix} \otimes I + \tau \begin{bmatrix} I\otimes K & \\ & I\otimes K^*\end{bmatrix}. \tag{2.25}$$

As $K \neq K^*$ in this generalized case, the step from (2.6) to (2.7) cannot be reproduced with this preconditioner. Luckily, it does not need to be! The middle factor in (2.6) can already be inverted in parallel since quantities at different time indices have been decoupled. The further diagonalization in the self-adjoint case also decoupled the state and adjoint equations; skipping this step here only decreases the available parallelism by a factor of 2.

Algorithm 2.2 incorporates the alpha-circulant improvement from subsection 2.2, as well as the above generalization. It can be compared to Algorithm 2.1.

3. ParaDiag for terminal-cost objectives. Both the literature on ParaDiag and this paper have thus far focused on the tracking objective in (1.2). We next develop a ParaDiag-type preconditioner for problems with the terminal-cost objective function, without requiring self-adjointness. The method is designed in subsection 3.1, after which it is analyzed for self-adjoint problems in subsections 3.2 and 3.3.

3.1. A new preconditioner. The optimality system in the terminal-cost case can be discretized with time step τ to form the all-at-once system

$$A\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{\lambda}\end{bmatrix} \coloneqq \left(\begin{bmatrix} B & \frac{\tau}{\gamma}I \\ -b_{0,0}E & B^\top\end{bmatrix} \otimes I + \tau \begin{bmatrix} I\otimes K & \\ -E\otimes K^* & I\otimes K^*\end{bmatrix}\right)\begin{bmatrix}\boldsymbol{y}\\ \boldsymbol{\lambda}\end{bmatrix} = \boldsymbol{b} \tag{3.1}$$

where E is a matrix with as only non-zero a one in the bottom right corner and $b_{0,0}$ is B's top-left element. Focusing once again on implicit Euler, we have that

$$B = \begin{bmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1\end{bmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{bmatrix}\boldsymbol{y_{\text{init}}}^\top & 0 & \dots & 0 & -((I+\tau K^*)\boldsymbol{y_{\text{target}}})^\top\end{bmatrix}^\top. \tag{3.2}$$

Algorithm 2.2 ParaDiag for solving the generalized tracking problem (2.23)

Input: Vectors $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$ defined by (2.2)
Arbitrary matrix K characterising the problem by (1.3)
Matrix $D(\alpha)$ following from the time discretization by (2.11) ($|\alpha| = 1$)
Output: The vectors $\boldsymbol{y}$ and $\boldsymbol{\lambda}$ that solve (2.23)

1: Rescale $\widehat{\boldsymbol{b}}_{\boldsymbol{2}} = \boldsymbol{b_2}/\sqrt{\gamma}$.
2: Solve (2.24) for $\boldsymbol{y}$ and $\widehat{\boldsymbol{\lambda}}$ using an iterative method, with preconditioner $P(\alpha)$ from (2.25). When asked to compute $\left[\begin{smallmatrix}\boldsymbol{x}\\ \boldsymbol{z}\end{smallmatrix}\right] = P^{-1}(\alpha)\left[\begin{smallmatrix}\boldsymbol{v}\\ \boldsymbol{w}\end{smallmatrix}\right]$:
3: Calculate $\boldsymbol{r_1} \coloneqq (\mathbb{F}\Gamma_\alpha \otimes I)\boldsymbol{v}$, $\boldsymbol{s_1} \coloneqq (\mathbb{F}\Gamma_\alpha \otimes I)\boldsymbol{w}$ using the (parallel) FFT.
4: For $l = \{1, \dots, \widehat{L}\}$, solve (in parallel)

$$\begin{bmatrix}\boldsymbol{r_{2,l}}\\ \boldsymbol{s_{2,l}}\end{bmatrix} \coloneqq \begin{bmatrix} d_{l,l}(\alpha)I + \tau K & \frac{\tau}{\sqrt{\gamma}}I \\ -\frac{\tau}{\sqrt{\gamma}}I & d^*_{l,l}(\alpha)I + \tau K^* \end{bmatrix}^{-1} \begin{bmatrix}\boldsymbol{r_{1,l}}\\ \boldsymbol{s_{1,l}}\end{bmatrix}. \tag{2.26}$$

5: Calculate $\boldsymbol{x} = (\Gamma_\alpha^{-1}\mathbb{F}^* \otimes I)\boldsymbol{r_2}$, $\boldsymbol{z} = (\Gamma_\alpha^{-1}\mathbb{F}^* \otimes I)\boldsymbol{s_2}$ using the (parallel) FFT.
6: Rescale $\boldsymbol{\lambda} = \sqrt{\gamma}\widehat{\boldsymbol{\lambda}}$.

In contrast to the tracking situation, the discretization point at time $t = T$ cannot be eliminated due to the more complex terminal condition in (1.3). Thus B is $L \times L$.

ParaDiag methods are fully reliant on the presence of good preconditioners, preferably with a mesh-independent convergence rate. Such a preconditioner must be invertible efficiently and in parallel. Leaving the bottom-left block of (3.1) out of the preconditioner makes this task significantly easier. Indeed, it allows replacing the B blocks by alpha-circulant $C(\alpha)$ blocks to form the preconditioner

$$P(\alpha) = \begin{bmatrix} C(\alpha) & \frac{\tau}{\gamma}I \\ & C^*(\alpha) \end{bmatrix} \otimes I + \tau \begin{bmatrix} I \otimes K & \\ & I \otimes K^* \end{bmatrix}, \tag{3.3}$$

which is block-triangular. Thus multiplication by $P^{-1}(\alpha)$ is possible by first inverting the bottom-right block of (3.3) (which pertains to the adjoint variable λ) and only then solving a second system to find the state y. Due to this procedure, $C(\alpha)$ and $C^*(\alpha)$ no longer need to be simultaneously diagonalizable, and $|\alpha|$ can be smaller than 1, in contrast to the tracking method. Algorithm 3.1 spells out how to solve (3.1) using the ParaDiag method this subsection proposes.

3.2. Analytic eigenvalue expressions. As was the case for tracking, we will formulate analytic eigenvalue results for the special case of a self-adjoint matrix $K = K^*$. The preparatory steps from subsection 2.3 are straightforward to repeat: we perform the same rescaling, resulting in

$$A_{\mathrm{p}} \coloneqq \left[\begin{array}{ccc|ccc} I & & & \Psi & & \\ -\Phi & I & & & \Psi & \\ & \ddots & \ddots & & & \ddots \\ & & -\Phi \;\; I & & & \Psi \\ \hline & & & I & -\Phi & \\ & & & & \ddots & \ddots \\ & & & & & I \;\; -\Phi \\ & & -I & & & I \end{array}\right], P_{\mathrm{p}}(\alpha) \coloneqq \left[\begin{array}{ccc|ccc} I & & -\alpha\Phi & \Psi & & \\ -\Phi & I & & & \Psi & \\ & \ddots & \ddots & & & \ddots \\ & & -\Phi \;\; I & & & \Psi \\ \hline & & & I & -\Phi & \\ & & & & \ddots & \ddots \\ & & & & & I \;\; -\Phi \\ & & & -\alpha\Phi & & I \end{array}\right] \tag{3.6}$$

where $\alpha \in \mathbb{R}$ was assumed. This time, $\Phi = (I + \tau K)^{-1}$ and $\Psi = \frac{\tau}{\gamma}(I + \tau K)^{-1}$. We again perform a decomposition to the scalar case, such that the eigenvalues of

Algorithm 3.1 ParaDiag procedure for solving the terminal-cost problem (3.1)

Input: Vector $\boldsymbol{b}$ defined by (3.2)
Arbitrary matrix K characterising the problem by (1.3)
Matrix $D(\alpha)$ following from the time discretization by (2.11) ($\alpha \neq 0$)

Output: The vectors $\boldsymbol{y}$ and $\boldsymbol{\lambda}$ that solve (3.1)

1: Solve (3.1) for $\boldsymbol{y}$ and $\boldsymbol{\lambda}$ using an iterative method, with preconditioner $P(\alpha)$ from (3.3). When asked to compute $\left[\begin{smallmatrix}\boldsymbol{x}\\ \boldsymbol{z}\end{smallmatrix}\right] = P^{-1}(\alpha)\left[\begin{smallmatrix}\boldsymbol{v}\\ \boldsymbol{w}\end{smallmatrix}\right]$:

▷ Phase 1: invert the bottom-right block

2: Calculate $((\boldsymbol{s_{0,0}})^\top, \dots, (\boldsymbol{s_{1,L}})^\top)^\top := (\mathbb{F}\Gamma_\alpha^{-*} \otimes I)\boldsymbol{w}$ using the (parallel) FFT.

3: For $l = \{1, \dots, L\}$, solve (in parallel)

$$\boldsymbol{s_{2,l}} := (d_{l,l}^*(\alpha)I + \tau K^*)^{-1}\boldsymbol{s_{1,l}} \tag{3.4}$$

and assemble $\boldsymbol{s_2} := ((\boldsymbol{s_{2,1}})^\top, \dots, (\boldsymbol{s_{2,L}})^\top)^\top$.

4: Calculate $\boldsymbol{z} = (\Gamma_\alpha^*\mathbb{F}^* \otimes I)\boldsymbol{s_2}$ using the (parallel) FFT.

▷ Phase 2: invert the rest of the matrix

5: Set $\boldsymbol{r_1} = \boldsymbol{v} - \frac{\tau}{\gamma}\boldsymbol{z}$.

6: Calculate $((\boldsymbol{r_{2,1}})^\top, \dots, (\boldsymbol{r_{2,L}})^\top)^\top := (\mathbb{F}\Gamma_\alpha \otimes I)\boldsymbol{r_1}$ using the (parallel) FFT.

7: For $l = \{1, \dots, L\}$, solve (in parallel)

$$\boldsymbol{r_{3,l}} := (d_{l,l}(\alpha)I + \tau K)^{-1}\boldsymbol{r_{2,l}} \tag{3.5}$$

and assemble $\boldsymbol{r_3} := ((\boldsymbol{r_{3,1}})^\top, \dots, (\boldsymbol{r_{3,L}})^\top)^\top$.

8: Calculate $\boldsymbol{x} = (\Gamma_\alpha^{-1}\mathbb{F}^* \otimes I)\boldsymbol{r_3}$ using the (parallel) FFT.

$P^{-1}(\alpha)A = P_{\mathrm{p}}^{-1}(\alpha)A_{\mathrm{p}}$ are the union of those of

$$P_\sigma^{-1}(\alpha)A_\sigma = \left[\begin{array}{cc|cc} 1 & & & -\alpha\varphi \\ -\varphi & 1 & & \\ & \ddots & \ddots & \\ & & -\varphi & 1 \\ \hline & & & \end{array}\right]^{-1} \left[\begin{array}{cc|cc} \psi & & & \\ & \psi & & \\ & & \ddots & \\ & & & \psi \\ \hline 1 & -\varphi & & \\ & \ddots & \ddots & \\ & & 1 & -\varphi \\ -\alpha\varphi & & & 1 \end{array}\right] \left[\begin{array}{cc|cc} 1 & & & \\ -\varphi & 1 & & \\ & \ddots & \ddots & \\ & & -\varphi & 1 \\ \hline & & & \end{array}\right]^{-1} \left[\begin{array}{cc|cc} \psi & & & \\ & \psi & & \\ & & \ddots & \\ & & & \psi \\ \hline 1 & -\varphi & & \\ & \ddots & \ddots & \\ & & 1 & -\varphi \\ & & & 1 \end{array}\right]. \tag{3.7}$$

for all eigenvalues σ of K, where $\varphi = (1 + \tau\sigma)^{-1}$ and $\psi = \frac{\tau}{\gamma}(1 + \tau\sigma)^{-1}$. We can eliminate τ by defining $\widehat{\sigma} := \tau\sigma$ and $\widehat{\gamma} := \frac{\tau}{\gamma}$, leading to

$$\varphi = (1 + \widehat{\sigma})^{-1} \quad \text{and} \quad \psi = \widehat{\gamma}(1 + \widehat{\sigma})^{-1}. \tag{3.8}$$

These equations are identical to (2.15), but notice that the definition of $\widehat{\gamma}$ is different.

The problem has been reduced to finding the eigenvalues θ of

$$P_\sigma^{-1}(\alpha)A_\sigma = I + P_\sigma^{-1}(\alpha)(A_\sigma - P_\sigma(\alpha)) =: I + P_\sigma^{-1}(\alpha)R_\sigma. \tag{3.9}$$

We get $\theta = 1 + \omega$, where the ωs are eigenvalues of $P_\sigma^{-1}(\alpha)R_\sigma$, studied in Theorem 3.1.

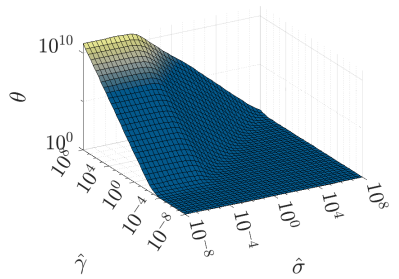


Fig. 3.1: The non-unity preconditioned eigenvalues θ of $P^{-1}(0)A$ with $L = 1000$

THEOREM 3.1. *Let $L > 3$ and $\alpha, \varphi, \psi \in \mathbb{R}$ with $\alpha \neq 0$. The $2L \times 2L$ matrix*

$$M = \left[\begin{array}{cc|cc} 1 & -\alpha\varphi & \psi & \\ -\varphi & \ddots & & \ddots \\ \ddots & 1 & \ddots & \psi \\ & -\varphi \quad 1 & & \\ \hline & & 1 \quad -\varphi & \\ & & \ddots \quad \ddots & \\ & & 1 \quad -\varphi & \\ -\alpha\varphi & & & 1 \end{array}\right]^{-1} \left[\begin{array}{cc} & \alpha\varphi \\ & \\ & \\ \hline & \\ & \\ -1 & \alpha\varphi \end{array}\right] \tag{3.10}$$

where $\varphi \neq \pm 1$ has two potentially non-zero eigenvalues $\omega_{\{1,2\}}$: those of the matrix

$$M_{\text{red}} = \begin{bmatrix} \frac{\alpha\varphi^L}{1-\alpha\varphi^L} + \frac{\psi}{(1-\alpha\varphi^L)^2}\frac{1-\varphi^{2L}}{1-\varphi^2} & -\frac{\alpha\varphi\psi}{(1-\alpha\varphi^L)^2}\frac{1-\varphi^{2L}}{1-\varphi^2} \\ -\frac{\varphi^{L-1}}{1-\alpha\varphi^L} & \frac{\alpha\varphi^L}{1-\alpha\varphi^L} \end{bmatrix}. \tag{3.11}$$

When α goes to zero, this simplifies to

$$\lim_{\alpha\to 0} \omega_1 = \psi\frac{1-\varphi^{2L}}{1-\varphi^2} \quad \text{and} \quad \lim_{\alpha\to 0} \omega_2 = 0. \tag{3.12}$$

Proof. The proof is given in Appendix B. ∎

COROLLARY 3.2. *Consider a terminal-cost all-at-once system with implicit-Euler time discretization (3.1) where $L > 3$ and K is self-adjoint with eigenvalues $\{\sigma_m\}_{m=1}^M$. When using ParaDiag with preconditioner $P(\alpha)$ (see (3.3)), the eigenvalues of the preconditioned system matrix are all either unity or equal to*

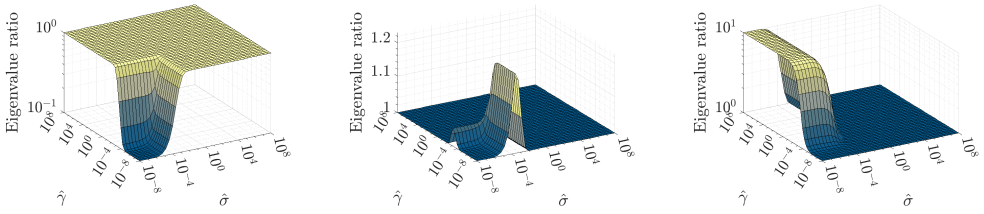
$$\theta_{m,\{1,2\}} = 1 + \omega_{m,\{1,2\}} \tag{3.13}$$

where $\omega_{m,\{1,2\}}$ are given as the eigenvalues of (3.11), having filled in $\varphi = (1+\tau\sigma_m)^{-1}$ and $\psi = \frac{\tau}{\gamma}(1+\tau\sigma_m)^{-1}$ (on the condition that $\varphi \neq \pm 1$).

3.3. Interpreting the eigenvalue results. In practice, α in (3.3) can usually be taken small enough such that the limit (3.12) is valid (the only constraint is rounding errors occurring for very small α [9, 31]). Given a rescaled eigenvalue $\widehat{\sigma}$ of K, Figure 3.1 plots the non-unity preconditioned eigenvalue θ in this limit. Based on this figure, *for a fixed number of time steps*, poor convergence can be expected when

- the equation in the absence of control evolves slowly *relatively to the size of the time interval* ($\widehat{\sigma} \approx 0$); and
- there is a lot of control *relatively to the size of the time interval* ($\widehat{\gamma} \gg 0$).

The first condition is the same as for the tracking preconditioner, but the second is different. Interestingly, the ParaOpt algorithm [10], which also treats terminal-cost objectives, struggles in the high-$\widehat{\gamma}$ regime as well.



(a) Tracking ($\alpha = 1$) (b) Tracking ($\alpha = -1$) (c) Terminal cost ($\alpha \to 0$)

Fig. 4.1: Ratio $\left|\theta(L = 10^4)\right| / \left|\theta(L = 10^3)\right|$ of the preconditioned-eigenvalue magnitudes when scaling L from 10^3 to 10^4 through T, for different preconditioners

4. Parallel-scaling analysis for self-adjoint problems. An oft-used metric in the context of parallel algorithms is *weak scalability* (for time-parallel methods, it was studied in e.g. [1, 4]). The aim is that a program's execution time stays constant when increasing the problem size (in our case the number of time steps L, as we investigate *time*-parallelism) in tandem with the number of processors, keeping their ratio constant. For our optimal-control problem (1.1), we identify two regimes [10].

- If we increase the time horizon T together with L, the time step τ stays constant. In this regime, $\widehat{\sigma}$ and $\widehat{\gamma}$ (from (2.15) or (3.8), depending on the objective function) do not change. The amount of work is increased by an expanding time scope, not by using a more accurate discretization.
- We can also keep T constant but instead increase the amount of time steps L by lowering τ. Then $\widehat{\sigma}$ and $\widehat{\gamma}$ increase with it. The amount of work is increased by using a more fine-grained mesh for the same problem.

This section will use the analytic results from subsections 2.3 and 3.2 to perform a theoretical analysis of ParaDiag's weak scaling, which section 5 later verifies in practice. The approach is to assume the inversion of our preconditioners scales well in all regimes, as attested to by previous ParaDiag algorithms that use similar preconditioners [13, 33, 32]. Then, all that needs to be analyzed is the number of such inversions: if the iterative solver's iteration count stays constant when increasing the problem size, we have achieved good weak scalability. As a proxy for the actual iteration count, we will use the distribution of the preconditioned eigenvalues – if they converge when increasing time parallelism, we will assume for the iteration count to do the same.

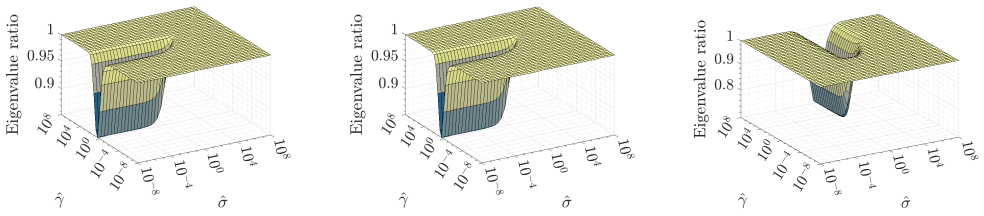
This section is limited to implicit Euler and self-adjoint, dissipative equations. We thus have $\sigma > 0$ and the obvious $\gamma, T > 0$. We aim to show that each of the eigenvalues θ converges to some finite, non-zero value in the relevant scaling limit.

4.1. Increasing the time horizon. Increasing T while keeping τ constant does not affect $\widehat{\sigma}$ or $\widehat{\gamma}$. As a result, the only change in (2.18) and (3.12) is that of the variable L (and, therefore, of $\widehat{L} = L - 1$).

Tracking. In (2.18), we have $0 < z_2 < 1 < z_1$ (see Lemma C.1(a)). As a result, in the limit for large T, $z_1^{\widehat{L}} \to \infty$ and $z_2^{\widehat{L}} \to 0$. That means that, for both $\alpha = \pm 1$, the eigenvalues of the preconditioned matrix converge to

$$\lim_{L\to\infty, T\to\infty, T=\tau L} \theta_{\{1,2\}} = 1 + \frac{1}{z_2 - z_1}(-(z_2 - \varphi \pm \psi \mathrm{i})) = \frac{z_1 - \varphi \pm \psi \mathrm{i}}{z_1 - z_2}. \tag{4.1}$$

This is finite and non-zero; as assumed in the intro to section 4, weak scalability can be expected. Figures 4.1a and 4.1b start from finite $L = 10^3$ and show that $|\theta|$ does not increase significantly when scaling L to 10^4. It even decreases when $\widehat{\sigma}, \widehat{\gamma} \gtrsim 0$, where Figure 2.3b shows $|\theta|$ is high to start with.



(a) Tracking ($\alpha = 1$) (b) Tracking ($\alpha = -1$) (c) Terminal cost ($\alpha \to 0$)

Fig. 4.2: Ratio $\left|\theta(L = 10^4)\right| / \left|\theta(L = 10^3)\right|$ of the preconditioned-eigenvalue magnitudes when scaling L from 10^3 to 10^4 through τ, for different preconditioners

Terminal cost. Something very similar occurs in (3.12). From $0 < \varphi < 1$, it follows that $\varphi^{2L} \to 0$ and the non-zero eigenvalue θ_1 approaches

$$\lim_{L\to\infty, T\to\infty, T=\tau L, \alpha\to 0} \theta_1 = \psi/(1-\varphi^2), \tag{4.2}$$

which is finite and non-zero since $\sigma, \gamma, T > 0$. In the limit $L \to \infty$, weak scalability is expected. Figure 4.1c confirms that θ scales well even when L is finite, except for very low $\widehat{\sigma}$ values, where the asymptotic region is not yet reached.

4.2. Decreasing the time step. Keeping T constant and scaling τ instead slightly complicates matters, as it changes not only L but also $\widehat{\sigma}$ and $\widehat{\gamma}$.

Tracking. Using MATLAB's symbolic toolbox allows us to solve the limit

$$\lim_{L\to\infty, \tau\to 0, \tau L = T, \alpha = \pm 1} \theta_{\{1,2\}} = \frac{1}{2} + \frac{\tanh\left(\frac{T\sqrt{\gamma\sigma^2+1}}{2\sqrt{\gamma}}\right)^{-\alpha}(\sqrt{\gamma}\sigma \pm \mathrm{i})}{2\sqrt{\gamma\sigma^2+1}}. \tag{4.3}$$

This is a finite expression for both $\alpha = \pm 1$ (the denominator cannot reach zero) and is non-zero as well (the real part of the numerator is always positive). Figures 4.2a and 4.2b show that the eigenvalues stay almost constant when scaling a finite L from 10^3 to 10^4, which we assumed implies weak scalability.

Terminal cost. This case is slightly simpler and can be computed by hand.

$$\lim_{L\to\infty, \tau\to 0, \tau L = T, \alpha\to 0} \theta_1 = 1 + (1 - \exp(-2\sigma T))/(\gamma\sigma), \tag{4.4}$$

which is again finite. It also cannot reach zero: the exponential has a negative argument (because $\sigma > 0$), so both terms of the sum are positive. Figure 4.2c illustrates that the scaling translates well to finite L values, again implying weak scalability.

5. Numerical results. This section presents the results of numerical tests assessing the performance of the ParaDiag methods discussed or designed in this paper. Subsection 5.1 first discusses tracking ParaDiag (Algorithm 2.2), considering both $\alpha = 1$ and the novel $\alpha = -1$ variant. Subsection 5.2 then moves on to the new terminal-cost method (Algorithm 3.1).

To perform these tests, we have implemented our ParaDiag algorithms in a MATLAB code we call `pintopt`. The implementation is sequential and not optimized for speed, but rather serves as a readable and well-documented reference implementation that can be used to study iteration counts. The code is publicly available[4].

[4]The `pintopt` MATLAB package is located at https://gitlab.kuleuven.be/numa/public/pintopt.

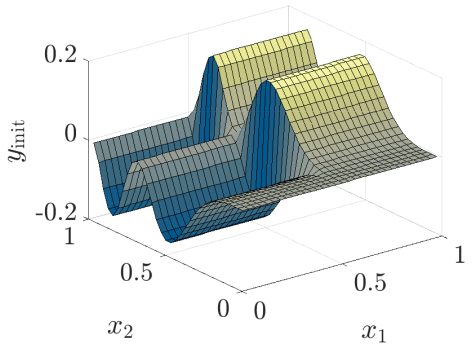


Fig. 5.1: Initial condition y_{init} from (5.4)

All results use GMRES as the iterative solver and are displayed in tables detailing the iteration counts for different parameter configurations. In each table, the rows investigate weak scaling, while the columns vary a different parameter such as the end time T or the regularization parameter γ. The tables at the left perform scaling of L by increasing T – those on the right by decreasing τ.

As a base problem, we study a parabolic diffusion equation, which is self-adjoint and dissipative such that our theoretical results are directly applicable. The problem involves a heat equation in two dimensions on the spatial domain $\Omega = [0,1]^2$. It reads

$$\partial_t y = \Delta y + u \tag{5.1}$$

with periodic boundary conditions and, in the case of tracking, a target trajectory

$$y_{\text{d}}(t,x) = \left(\left(12\pi^2 + \frac{1}{12\pi^2\gamma}\right)(t-T) - \left(1 + \frac{1}{(12\pi^2)^2\gamma}\right)\right)\sin(2\pi x_1)\sin(2\pi x_2) \tag{5.2}$$

or, in the case of terminal cost, a target state

$$y_{\text{target}}(x) = \sin(2\pi x_1)\sin(2\pi x_2). \tag{5.3}$$

This is the two-dimensional version of a problem studied in [14]. In contrast to that paper, we use a non-smooth initial condition

$$y_{\text{init}}(x) = \frac{1}{12\pi^2\gamma}(1-T)\text{sign}(\sin(2\pi x_1))\sin^2(2\pi x_2), \tag{5.4}$$

shown in Figure 5.1. The choice for a non-smooth y_{init} is important, as a smooth initial condition leads to very fast convergence, as noticed in [13, 32]. We want to test our algorithms with a more challenging, non-smooth case.

Next to the self-adjoint equation (5.1) covered fully by this paper's analysis, we also consider a non-self-adjoint advection-diffusion equation that our new algorithms can solve, but for which we do not have theoretical results. Extending the previous equation with an advection term, consider

$$\partial_t y = d\Delta y - \partial_{x_1} y - \partial_{x_2} y + u \tag{5.5}$$

where $d \in \mathbb{R}$ controls the amount of diffusion and may vary. For this equation, we use the same y_{d}, y_{target} and y_{init}, given in (5.2)–(5.4), as for the diffusion equation. Both (5.1) and (5.5) are discretized with $M = 32 \times 32$ points in space and all spatial derivatives are discretized with central differences.

$L\backslash T_{\text{ref}}$	2e0	2e-1	2e-2	2e-3	2e-4
30	3/3	4/4	6/6	14/**8**	∅/**7**
100	3/3	3/3	5/5	10/**7**	19/**8**
300	3/3	3/3	4/4	6/6	14/**8**
1000	3/3	3/3	3/3	5/5	10/**7**

(a) Scaling T, diffusion

$L\backslash T$	2e0	2e-1	2e-2	2e-3	2e-4
30	3/3	4/4	6/6	14/**8**	∅/**7**
100	3/3	4/4	6/6	14/**8**	24/**7**
300	3/3	4/4	6/6	14/**8**	∅/**7**
1000	3/3	4/4	6/6	14/**8**	∅/**7**

(b) Scaling τ, diffusion

$L\backslash\gamma$	5e-8	5e-5	5e-2	5e1	5e4
30	2/2	4/4	3/3	3/3	2/2
100	2/2	4/4	3/3	3/3	2/2
300	2/2	4/4	3/3	3/3	2/2
1000	2/2	3/3	3/3	3/3	2/2

(c) Scaling T, diffusion

$L\backslash\gamma$	5e-8	5e-5	5e-2	5e1	5e4
30	2/2	4/4	3/3	3/3	2/2
100	2/2	5/5	3/3	3/3	3/3
300	3/3	7/7	3/3	3/3	3/3
1000	3/3	8/8	3/3	3/3	3/3

(d) Scaling τ, diffusion

$L\backslash T_{\text{ref}}$	2e0	2e-1	2e-2	2e-3	2e-4
30	5/5	**7**/8	19/**10**	∅/**7**	∅/**4**
100	5/5	7/7	11/11	∅/**9**	∅/**5**
300	5/5	6/6	8/8	19/**10**	∅/**7**
1000	5/5	6/6	7/7	11/11	∅/**9**

(e) Scaling T, advection-diffusion

$L\backslash T$	2e0	2e-1	2e-2	2e-3	2e-4
30	5/5	**7**/8	19/**10**	∅/**7**	∅/**4**
100	6/6	**7**/8	19/**10**	∅/**7**	∅/**4**
300	6/6	8/8	19/**10**	∅/**7**	∅/**4**
1000	6/6	8/8	19/**10**	∅/**7**	∅/**4**

(f) Scaling τ, advection-diffusion

$L\backslash\gamma$	5e-8	5e-5	5e-2	5e1	5e4
30	2/2	4/4	5/5	4/4	3/3
100	2/2	4/4	5/5	4/4	3/3
300	2/2	3/3	5/5	3/3	3/3
1000	2/2	3/3	5/5	3/3	3/3

(g) Scaling T, advection-diffusion

$L\backslash\gamma$	5e-8	5e-5	5e-2	5e1	5e4
30	2/2	4/4	5/5	4/4	3/3
100	2/2	6/6	6/6	4/4	3/3
300	3/3	9/9	6/6	4/4	3/3
1000	3/3	11/11	6/6	3/3	3/3

(h) Scaling τ, advection-diffusion

$L\backslash d$	1e-3	1e-2	1e-1	1e0	1e1
30	7/7	7/7	5/5	3/3	2/2
100	7/7	7/7	5/5	3/3	2/2
300	7/7	7/7	5/5	3/3	2/2
1000	7/7	6/6	5/5	3/3	2/2

(i) Scaling T, advection-diffusion

$L\backslash d$	1e-3	1e-2	1e-1	1e0	1e1
30	7/7	7/7	5/5	3/3	2/2
100	8/8	7/7	6/6	3/3	3/3
300	8/8	7/7	6/6	3/3	3/3
1000	8/8	8/8	6/6	3/3	3/3

(j) Scaling τ, advection-diffusion

Table 5.1: GMRES iteration counts ($\alpha = 1/\alpha = -1$) for tracking ParaDiag applied to the diffusion equation (5.1) or the advection-diffusion equation (5.5). The symbol ∅ indicates a failure to converge within 25 iterations. When one α value outperforms the other, it is bold-faced. By default, $T_{\text{ref}} = 2$, $\gamma = 0.05$, and $d = 0.1$ when applicable.

5.1. Tracking. The results from applying ParaDiag to the (advection-)diffusion example are listed in Table 5.1. The case $\alpha = -1$ outperforms $\alpha = 1$ when T is small and does not make much difference otherwise, as observed in subsection 2.2. A wide variety of γ values is tested, all resulting in very reasonable iteration counts.

There are two more significant observations. Firstly, the iteration count mostly stays constant when increasing L, which is the weak scalability theorized in section 4. Secondly, the advection-diffusion case is remarkably comparable to the pure diffusion equation. The scaling is very similar and the α choices manifest in the same way – decreasing the amount of diffusion d does not change this, as observed in Table 5.1j. This is excellent news, as it suggests the conclusions from our self-adjoint study may apply more broadly to non-self-adjoint problems as well.

$L\backslash T_{\text{ref}}$	2e0	2e-1	2e-2	2e-3	2e-4
30	2	3	4	4	3
100	2	2	3	4	3
300	3	2	3	4	4
1000	3	2	2	3	4

(a) Scaling T, diffusion

$L\backslash T$	2e0	2e-1	2e-2	2e-3	2e-4
30	2	3	4	4	3
100	2	3	4	4	3
300	2	3	4	4	3
1000	2	3	4	4	3

(b) Scaling τ, diffusion

$L\backslash\gamma$	5e-8	5e-5	5e-2	5e1	5e4
30	3	3	2	1	1
100	12	3	2	1	1
300	12	3	3	1	1
1000	∅	3	3	1	1

(c) Scaling T, diffusion

$L\backslash\gamma$	5e-8	5e-5	5e-2	5e1	5e4
30	3	3	2	1	1
100	11	3	2	1	1
300	11	3	2	1	1
1000	10	3	2	1	1

(d) Scaling τ, diffusion

$L\backslash T_{\text{ref}}$	2e0	2e-1	2e-2	2e-3	2e-4
30	7	7	5	4	3
100	5	6	6	4	3
300	5	5	6	5	4
1000	5	5	6	6	4

(e) Scaling T, advection-diffusion

$L\backslash T$	2e0	2e-1	2e-2	2e-3	2e-4
30	7	7	5	4	3
100	6	7	5	4	3
300	5	6	5	4	3
1000	6	6	5	4	3

(f) Scaling τ, advection-diffusion

$L\backslash\gamma$	5e-8	5e-5	5e-2	5e1	5e4
30	7	7	7	2	1
100	∅	5	5	2	1
300	∅	5	5	2	1
1000	∅	5	5	2	1

(g) Scaling T, advection-diffusion

$L\backslash\gamma$	5e-8	5e-5	5e-2	5e1	5e4
30	7	7	7	2	1
100	7	7	6	2	1
300	7	7	5	2	1
1000	7	7	6	2	1

(h) Scaling τ, advection-diffusion

$L\backslash d$	1e-3	1e-2	1e-1	1e0	1e1
30	9	8	7	5	4
100	8	7	5	5	4
300	5	5	5	5	4
1000	5	5	5	5	4

(i) Scaling T, advection-diffusion

$L\backslash d$	1e-3	1e-2	1e-1	1e0	1e1
30	9	8	7	5	4
100	9	7	6	5	4
300	11	7	5	5	4
1000	11	8	6	3	3

(j) Scaling τ, advection-diffusion

Table 5.2: GMRES iteration counts for terminal-cost ParaDiag applied to the diffusion equation (5.1) or the advection-diffusion equation (5.5). The symbol ∅ indicates a failure to converge within 25 iterations. By default, $T_{\text{ref}} = 2$, $\gamma = 0.05$, and $d = 0.1$ when applicable. All results use $\alpha = 10^{-4}$.

5.2. Terminal cost. The same experiments were done for terminal-cost objectives in Table 5.2, although $\alpha = 10^{-4}$ was chosen here. The results are very promising. For the diffusion equation, iteration counts are low across the board, with no scenario surpassing 4 iterations. The only exception is that of a *very* small γ (that is, a large $\widehat{\gamma}$), which was indeed theorized to work poorly in subsection 3.3.

When adding advection, slightly more iterations are needed, but the increase stays reasonable. Again, the qualitative insights from the self-adjoint case carry over: a small regularization parameter γ can cause slow convergence. Small d values also seem to result in an increased iteration count. However, scaling is excellent in all scenarios, confirming the theoretical conclusions from section 4.

6. Conclusions. This paper has extended optimization ParaDiag in three ways. For the existing algorithm [33], aimed at tracking objectives, we proposed an alpha-circulant extension to improve the small-T regime and a generalization to non-self-adjoint problems. We also designed a new algorithm to treat terminal-cost objectives. In doing so, we greatly expanded the range of problems for which efficient ParaDiag algorithms are available.

Secondly, we were able to formulate a precise expression for the preconditioned eigenvalues of all optimization ParaDiag methods, in the self-adjoint case. This significantly improves our understanding of these algorithms, for which very little theory was available before. We used this knowledge for two purposes.

- For dissipative, self-adjoint equations with a tracking objective, and when using the new parameter $\alpha = -1$ to construct a preconditioner, we were able to prove a guaranteed GMRES convergence factor of $1/2$.
- In a theoretical parallel-scaling analysis, we conjectured good weak scalability of all ParaDiag variants in the limit for many time steps.

This scalability was confirmed by numerical experiments that used GMRES iteration counts as an indicator of performance. In addition, these tests suggested the theoretical conclusions carry over to the non-self-adjoint case, even though our analysis does not apply there.

As a third contribution, our progress clears the way for exciting research in the future. With a more general method, into which some theoretical insight is available, potential next steps include non-linear ParaDiag algorithms, such as those already proposed for IVP ParaDiag [8, 22].

Other interesting avenues for future work include the study of different time-discretization methods – especially if they can be written as (2.13) or (3.6) since then, our theoretical results apply – or improvements to ParaDiag for hyperbolic equations comparable to those in this paper. Different techniques have also been applied to solve the smaller systems in the inversion procedure for IVP ParaDiag more efficiently [21, 16]. Adapting these methods to optimization ParaDiag could substantially improve performance. Lastly, a very interesting recent result [18] in the domain of IVP ParaDiag suggests using alpha-circulant approximations, but not as preconditioners. Instead, it is noted that in the IVP situation, the exact system matrix is $P(\alpha)$ with $\alpha = 0$ and its inversion is seen as an interpolation problem, with as data points several inversions with $\alpha_j \neq 0$. Our terminal-cost preconditioner (3.3) is not suitable for this, as it has $P(0) \neq A$. However, for the tracking preconditioner (2.10), $P(0) = \widehat{A}$ does hold. As this text makes $\alpha \neq 1$ feasible for tracking, we can use different α_j with magnitude 1 as data points and [18]'s technique could now apply to tracking-type optimal-control ParaDiag.

Appendix A. Proof of Theorem 2.1. We denote by $C(\alpha)$ the top-left block of the matrix inverted in (2.17). We start by switching the top and bottom halves of the rows of both R and the inverted matrix – which does not change M – and applying the block matrix inversion property from [2, page 44], giving

$$
\begin{aligned}
M &= \begin{bmatrix} -\psi I & C^\top(\alpha) \\ C(\alpha) & \psi I \end{bmatrix}^{-1} \left[\begin{array}{c|c} & \alpha\varphi \\ \hline \alpha\varphi & \end{array}\right] \\
&= \begin{bmatrix} (-\psi I - \frac{1}{\psi} C^\top(\alpha) C(\alpha))^{-1} & \\ & (\psi I + \frac{1}{\psi} C(\alpha) C^\top(\alpha))^{-1} \end{bmatrix} \begin{bmatrix} I & -\frac{1}{\psi} C^\top(\alpha) \\ \frac{1}{\psi} C(\alpha) & I \end{bmatrix} \left[\begin{array}{c|c} & \alpha\varphi \\ \hline \alpha\varphi & \end{array}\right] \\
&= \begin{bmatrix} \overbrace{(-\psi I - \frac{1}{\psi} C^\top(\alpha) C(\alpha))^{-1}}^{=-H} & \underbrace{(\psi I + \frac{1}{\psi} C(\alpha) C^\top(\alpha))^{-1}}_{=:H} \end{bmatrix} \left[\begin{array}{c|c} -\frac{\alpha\varphi}{\psi} & \\ \hline \frac{\varphi^2}{\psi}\alpha^2 & \frac{\alpha\varphi}{\psi} \\ \hline \alpha\varphi & -\frac{\varphi^2}{\psi}\alpha^2 \\ & \frac{\alpha\varphi}{\psi} \end{array}\right],
\end{aligned}
$$

where we know $\alpha^2 = 1$. The fact that M has only two potentially non-zero eigenvalues is clear: the second matrix in the product above has rank 2, such that the result of a multiplication by it cannot have any higher rank.

To find out more about these non-zero eigenvalues, first observe that

$$
C^\top(\alpha) C(\alpha) = C(\alpha) C^\top(\alpha), \tag{A.1}
$$

as can be easily checked. This justifies the use of the variable H for both blocks above. We will try to calculate H later, but in the spirit of not doing excess work, let us first see which parts of H we need at all.

$$
M = \begin{bmatrix} -H & \\ & H \end{bmatrix} \left[\begin{array}{c|c} -\frac{\alpha\varphi}{\psi} & \\ \hline \frac{\varphi^2}{\psi} & \frac{\alpha\varphi}{\psi} \\ \hline \alpha\varphi & -\frac{\varphi^2}{\psi} \\ & \frac{\alpha\varphi}{\psi} \end{array}\right] =: \left[\begin{array}{c|c} a_1 & b_1 \\ \vdots & \vdots \\ a_{\widehat{L}} & b_{\widehat{L}} \\ \hline a_{\widehat{L}+1} & b_{\widehat{L}+1} \\ \vdots & \vdots \\ a_{2\widehat{L}} & b_{2\widehat{L}} \end{array}\right], \tag{A.2}
$$

which means that M's non-zero eigenvalues are the same as those of its middle block

$$
M_{\text{red}} = \begin{bmatrix} a_{\widehat{L}} & b_{\widehat{L}} \\ a_{\widehat{L}+1} & b_{\widehat{L}+1} \end{bmatrix} = \begin{bmatrix} \frac{\alpha\varphi}{\psi} h_{\text{end},0} - \frac{\varphi^2}{\psi} h_{\text{end,end}} & -\alpha\varphi h_{\text{end,end}} \\ \alpha\varphi h_{0,0} & -\frac{\varphi^2}{\psi} h_{0,0} + \frac{\alpha\varphi}{\psi} h_{0,\text{end}} \end{bmatrix}. \tag{A.3}
$$

Thus, it suffices to find the corner values of

$$H = (\psi I + \frac{1}{\psi}C(\alpha)C^\top(\alpha))^{-1} = \psi(\underbrace{\psi^2 I + C(\alpha)C^\top(\alpha)}_{=:G})^{-1}$$

$$= \psi \begin{bmatrix} 1+\varphi^2+\psi^2 & -\varphi & & -\alpha\varphi \\ -\varphi & 1+\varphi^2+\psi^2 & \ddots & \\ & \ddots & \ddots & -\varphi \\ -\alpha\varphi & & -\varphi & 1+\varphi^2+\psi^2 \end{bmatrix}^{-1}. \tag{A.4}$$

The matrix G being inverted is α-circulant and symmetric – qualities that are maintained by the inversion. Then $h_{0,0} = h_{\text{end,end}}$ and $h_{0,\text{end}} = h_{\text{end},0}$. Hence

$$a_{\widehat{L}} = b_{\widehat{L}+1} = -\frac{\varphi^2}{\psi}h_{0,0} + \frac{\alpha\varphi}{\psi}h_{0,\text{end}} \quad \text{and} \quad a_{\widehat{L}+1} = -b_{\widehat{L}} = \alpha\varphi h_{0,0}. \tag{A.5}$$

This means that

$$\text{eig}(M_{\text{red}}) = a_{\widehat{L}} \pm b_{\widehat{L}}\mathrm{i}. \tag{A.6}$$

All this assumes we have inverted G. Alpha-circulant matrices can be inverted using their spectral decomposition (2.11). Up until now, we used this as a computational tool; extracting useful analytical expressions is not a trivial feat. The diagonalization reads $G = \Gamma_\alpha^{-1}\mathbb{F}^* D \mathbb{F}\Gamma_\alpha \Leftrightarrow G^{-1} = \Gamma_\alpha^{-1}\mathbb{F}^* D^{-1}\mathbb{F}\Gamma_\alpha$. Here,

$$\begin{aligned} D &= \text{diag}(\sqrt{\widehat{L}}\mathbb{F}\Gamma_\alpha \boldsymbol{g_1}) \\ &= \text{diag}\{1+\varphi^2+\psi^2 - \mathrm{e}^{(j/\widehat{L})2\pi\mathrm{i}}\alpha^{1/\widehat{L}}\varphi - \mathrm{e}^{-(j/\widehat{L})2\pi\mathrm{i}}\alpha^{(\widehat{L}-1)/\widehat{L}}\alpha\varphi\}_{j=0}^{\widehat{L}-1} \\ &= \text{diag}\left\{d(\beta_j(\alpha,\widehat{L}))\right\}_{j=0}^{\widehat{L}-1} \end{aligned} \tag{A.7}$$

where $\boldsymbol{g_1}$ denotes G's first column and where we defined

$$d(\beta) = 1+\varphi^2+\psi^2 - 2\varphi\cos\beta \quad \text{and} \quad \beta_j(\alpha,\widehat{L}) = \begin{cases} 2j\pi/\widehat{L} & \text{if } \alpha = 1 \\ 2(j+\frac{1}{2})\pi/\widehat{L} & \text{if } \alpha = -1. \end{cases} \tag{A.8}$$

Then from $H = \psi G^{-1}$ follows, noting the definitions of Γ_α and $\mathbb{F}$,

$$h_{0,0}(\alpha,\widehat{L}) = h_{\text{end,end}}(\alpha,\widehat{L}) = \frac{\psi}{\widehat{L}}\sum_{j=0}^{\widehat{L}-1} d(\beta_j(\alpha,\widehat{L}))^{-1}, \tag{A.9a}$$

$$\begin{aligned} h_{0,\text{end}}(\alpha,\widehat{L}) = h_{\text{end},0}(\alpha,\widehat{L}) &= \frac{\psi}{\widehat{L}}\sum_{j=0}^{\widehat{L}-1} \alpha^{(\widehat{L}-1)/\widehat{L}}\mathrm{e}^{-(j/\widehat{L})2\pi\mathrm{i}} d(\beta_j(\alpha,\widehat{L}))^{-1} \\ &= \alpha\frac{\psi}{\widehat{L}}\sum_{j=0}^{\widehat{L}-1} \mathrm{e}^{-\beta_j(\alpha,\widehat{L})\mathrm{i}} d(\beta_j(\alpha,\widehat{L}))^{-1}. \end{aligned} \tag{A.9b}$$

These analytic expressions are not insightful. Luckily, we have yet another avenue to find $h_{0,0}$ and $h_{0,\text{end}}$: [27, Theorem 1(a)] offers explicit formulas for inverting certain

three-element circulant matrices. For $\alpha = 1$, those formulas mean that

$$h_{0,0}(1,\widehat{L}) = h_{\text{end,end}}(1,\widehat{L}) = \psi\frac{z_1 z_2}{\varphi(z_2-z_1)}\left(\frac{1}{1-z_1^{\widehat{L}}} - \frac{1}{1-z_2^{\widehat{L}}}\right), \tag{A.10a}$$

$$h_{0,\text{end}}(1,\widehat{L}) = h_{\text{end},0}(1,\widehat{L}) = \psi\frac{z_1 z_2}{\varphi(z_2-z_1)}\left(\frac{z_1}{1-z_1^{\widehat{L}}} - \frac{z_2}{1-z_2^{\widehat{L}}}\right) \tag{A.10b}$$

with $z_{\{1,2\}} = (1+\varphi^2+\psi^2 \pm \sqrt{(1+\varphi^2+\psi^2)^2-4\varphi^2})/(2\varphi)$. However, [27] tells us nothing about the case $\alpha = -1$.

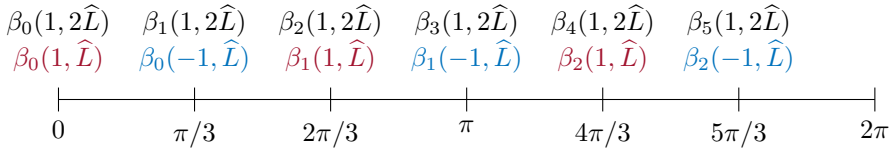


Fig. A.1: $\beta_j(\cdot,\cdot)$ for different parameters when $\widehat{L} = 3$

Now, we can utilize the expressions (A.9). Figure A.1 shows the spacing of the β_js when $\widehat{L} = 3$ for $\alpha = 1$ (red) and $\alpha = -1$ (blue). Combined with (A.9a), it is clear that $h_{0,0}(1,2\widehat{L})$ sums over the same βs as $h_{0,0}(1,\widehat{L})$ and $h_{0,0}(-1,\widehat{L})$ combined. After correcting for the scaling by $\widehat{L}$ in (A.9a), we get

$$h_{0,0}(-1,\widehat{L}) = 2h_{0,0}(1,2\widehat{L}) - h_{0,0}(1,\widehat{L}) = \psi\frac{z_1 z_2}{\varphi(z_2-z_1)}\left(\frac{1}{1+z_1^{\widehat{L}}} - \frac{1}{1+z_2^{\widehat{L}}}\right). \tag{A.11}$$

A similar technique can be used for $h_{0,\text{end}}$, yielding

$$\begin{aligned} h_{0,\text{end}}(-1,\widehat{L}) &= h_{0,\text{end}}(1,\widehat{L}) - 2h_{0,\text{end}}(1,2\widehat{L}) \\ &= -\psi\frac{z_1 z_2}{\varphi(z_2-z_1)}\left(\frac{z_1}{1+z_1^{\widehat{L}}} - \frac{z_2}{1+z_2^{\widehat{L}}}\right). \end{aligned} \tag{A.12}$$

As a last step, we have that $z_1 z_2 = 1$. To see this, note that

$$\begin{aligned} z_1 z_2 &= \frac{1}{4\varphi^2}\left((1+\varphi^2+\psi^2)^2 - (\sqrt{(1+\varphi^2+\psi^2)^2-4\varphi^2})^2\right) \\ &= \frac{1}{4\varphi^2}\left((1+\varphi^2+\psi^2)^2 - (1+\varphi^2+\psi^2)^2 + 4\varphi^2\right) = 1. \end{aligned} \tag{A.13}$$

Eliminating the square root is allowed due to its contents always being non-negative. Indeed, $(1+\varphi^2+\psi^2)^2 - 4\varphi^2$ reaches a minimum for $\psi = 0$, where we get $1 + 2\varphi^2 + \varphi^4 - 4\varphi^2 = (1-\varphi^2)^2$, which cannot be negative. Filling the expressions for the hs into (A.5) and (A.6) proves the theorem.

Appendix B. Proof of Theorem 3.1. We use the notation $C(\alpha)$ for the top-left block of the matrix inverted in (3.10). We start the proof similarly to Appendix A.

Finding M_{red}. The inverse of the block-triangular matrix can be rewritten as

$$M = \begin{bmatrix} \overbrace{C^{-1}(\alpha)}^{=:H} & -\psi \overbrace{C^{-1}(\alpha)C^{-\top}(\alpha)}^{=:G} \\ & \underbrace{C^{-\top}(\alpha)}_{=H^\top} \end{bmatrix} \left[\begin{array}{c|c} \alpha\varphi & \\ \hline & \\ -1 & \alpha\varphi \end{array}\right] =: \left[\begin{array}{c|c} a_1 & b_1 \\ \vdots & \vdots \\ \underline{a_L} & \underline{b_L} \\ a_{L+1} & b_{L+1} \\ \vdots & \vdots \\ a_{2L} & b_{2L} \end{array}\right] \tag{B.1}$$

such that the potentially non-zero eigenvalues of M are the same as those of

$$M_{\text{red}} = \begin{bmatrix} a_L & b_L \\ a_{L+1} & b_{L+1} \end{bmatrix} = \begin{bmatrix} \alpha\varphi h_{\text{end},0} + \psi g_{\text{end,end}} & -\alpha\varphi\psi g_{\text{end,end}} \\ -h_{\text{end},0} & \alpha\varphi h_{\text{end},0} \end{bmatrix}. \tag{B.2}$$

Reducing the unknowns to H. It seems that we need the values $h_{\text{end},0}$ and $g_{\text{end,end}}$ to make further progress. First, let us consider

$$G = C^{-1}(\alpha)C^{-\top}(\alpha) = (C^\top(\alpha)C(\alpha))^{-1}. \tag{B.3}$$

To solve a similar problem in Theorem 2.1's proof, we noted that the matrix being inverted was alpha-circulant and acted on that knowledge. However, if $|\alpha| \neq 1$, this is not the case anymore (as can easily be checked), so another method needs to be found. We can first express $g_{\text{end,end}}$ in terms of H as

$$g_{\text{end,end}} = \boldsymbol{H}_{\mathbf{end},:}(\boldsymbol{H}^\top)_{:,\mathbf{end}} = \left\|\boldsymbol{H}_{\mathbf{end},:}\right\|_2^2 = \sum\nolimits_{j=0}^{L-1} h_{\text{end},j}^2. \tag{B.4}$$

Inverting $C(\alpha)$. Let us now work on the problem of finding $H = C^{-1}(\alpha)$. If $C(\alpha)$ were fully circulant, [27, Theorem 1(d)] would offer a relatively simple analytical expression for its inverse; unfortunately, it is alpha-circulant. Even the technique to invert (-1)-circulant matrices from Theorem 2.1's proof does not suffice here. Luckily, we have yet another trick up our sleeves.

Again, the key is inverting the diagonalization in (2.11). Consider doing so for $C(\alpha)$, as well as for an *actually* circulant matrix $\widehat{C}$ – defined later – giving

$$C^{-1}(\alpha) = \Gamma_\alpha^{-1}\mathbb{F}^* \operatorname{diag}(\sqrt{L}\mathbb{F}\Gamma_\alpha \boldsymbol{c_1}(\alpha))^{-1}\mathbb{F}\Gamma_\alpha, \tag{B.5a}$$

$$\widehat{C}^{-1} = \mathbb{F}^* \operatorname{diag}(\sqrt{L}\mathbb{F}\widehat{\boldsymbol{c}}_{\mathbf{1}})^{-1}\mathbb{F}. \tag{B.5b}$$

If we now require $\widehat{\boldsymbol{c}}_{\mathbf{1}} = \Gamma_\alpha \boldsymbol{c_1}(\alpha)$, this fully defines $\widehat{C}$. But then (B.5) means that $H = C^{-1}(\alpha) = \Gamma_\alpha^{-1}\widehat{C}^{-1}\Gamma_\alpha$, which allows computing H. By [27, Theorem 1(d)],

$$h_{\text{end},j} = \alpha^{(j-(L-1))/L}\alpha^{(L-j-1)/L}\frac{\varphi^{L-j-1}}{1-\alpha\varphi^L} = \frac{\varphi^{L-j-1}}{1-\alpha\varphi^L} \tag{B.6}$$

from which immediately follow $h_{\text{end},0} = \frac{\varphi^{L-1}}{1-\alpha\varphi^L}$ and

$$\begin{aligned} g_{\text{end,end}} &= \sum_{j=0}^{L-1} h_{\text{end},j}^2 = \sum_{j=0}^{L-1}\left(\frac{\varphi^j}{1-\alpha\varphi^L}\right)^2 = \frac{1}{(1-\alpha\varphi^L)^2}\sum_{j=0}^{L-1}(\varphi^2)^j \\ &= \frac{1}{(1-\alpha\varphi^L)^2}\frac{1-\varphi^{2L}}{1-\varphi^2}. \end{aligned}$$

Filling these into (B.2) gives (3.11), while the limits (3.12) are then trivial as the entire second column of M_{red} goes to zero when $\alpha \to 0$.

Appendix C. Proofs of auxiliary lemmas.

LEMMA C.1 (Some properties of Theorem 2.1's z_1 and z_2). *Defining*

$$z_{\{1,2\}} = (1 + \varphi^2 + \psi^2 \pm \sqrt{(1 + \varphi^2 + \psi^2)^2 - 4\varphi^2})/(2\varphi), \tag{C.1}$$

the following properties hold.

(a) If $0 < \varphi < 1$, both z_1 and z_2 are real-valued and it holds that $0 < z_2 < 1 < z_1$.
(b) If $0 < \varphi < 1$, it holds that $z_2 \leq \varphi$.

Proof. We prove these claims one by one.

(a) The quantity in (C.1)'s square root reads

$$(1 + \varphi^2 + \psi^2)^2 - 4\varphi^2 = (1 + \varphi^2 + \psi^2 + 2\varphi) \underbrace{(1 + \varphi^2 + \psi^2 - 2\varphi)}_{=(1-\varphi)^2+\psi^2}, \tag{C.2}$$

which is positive, such that the zs are real numbers. They are also positive, as follows from $4\varphi^2 > 0$ and $\varphi > 0$. Furthermore, $z_1 > z_2$. Since their product is 1 (see Appendix A), z_1 must be larger than 1 while z_2 is smaller.

(b) We write

$$\begin{aligned}
& z_2 \leq \varphi \\
\Leftrightarrow \quad & 1 + \varphi^2 + \psi^2 - \sqrt{(1 + \varphi^2 + \psi^2)^2 - 4\varphi^2} \leq 2\varphi^2 \\
\Leftrightarrow \quad & 1 - \varphi^2 + \psi^2 \leq \sqrt{(1 + \varphi^2 + \psi^2)^2 - 4\varphi^2} \\
\Leftrightarrow \quad & (1 - \varphi^2 + \psi^2)^2 \leq (1 + \varphi^2 + \psi^2)^2 - 4\varphi^2 \\
\Leftrightarrow \quad & 4\varphi^2 \leq 4\varphi^2(1 + \psi^2),
\end{aligned}$$

which is clearly true. ∎

LEMMA C.2 (Some properties of Theorem 2.1's ω_1 and ω_2). *Define*

$$\omega_{\{1,2\}} = \frac{1}{z_2 - z_1} \left(\frac{z_1 - \varphi \pm \psi \mathrm{i}}{1 + z_1^{\widehat{L}}} - \frac{z_2 - \varphi \pm \psi \mathrm{i}}{1 + z_2^{\widehat{L}}} \right) \tag{C.3}$$

with z_1 and z_2 as in (C.1)*, where $\widehat{L} \geq 1$ and $0 < \varphi < 1$. Then denote by $\Re(\omega) = \Re(\omega_1) = \Re(\omega_2)$ the real part characterising the ωs and by $\Im(\omega) = \Im(\omega_1) = -\Im(\omega_2)$ the imaginary part.*

(a) It holds that $\Re(\omega) < 0$ increases monotonically with increasing $\widehat{L}$.
(b) It holds that $\Im(\Omega) > 0$ increases monotonically with increasing $\widehat{L}$.
(c) Following (a), it holds that $-\frac{1}{2} < \Re(\omega_{\{1,2\}})$.
(d) Following (a) and (b), it holds that $\left| \frac{1}{2} + \omega_{\{1,2\}} \right| < \frac{1}{2}$.

Proof. Once again, the claims are addressed one by one.

(a) We rewrite $\Re(\omega) = -\frac{1}{z_1 - z_2} \left(\frac{z_1 - \varphi}{1 + z_1^{\widehat{L}}} + \frac{\varphi - z_2}{1 + z_2^{\widehat{L}}} \right)$ where, due to Lemma C.1(a) and Lemma C.1(b), all numerators and denominators are positive. Appendix A showed $z_1 z_2 = 1 \Leftrightarrow z_2 = 1/z_1$ – filling this in, we get

$$\frac{\mathrm{d}}{\mathrm{d}\widehat{L}} \Re(\omega) = \frac{1}{z_1 - 1/z_1} \frac{z_1^{\widehat{L}-1}(-2\varphi z_1 + z_1^2 + 1) \log z_1}{(z_1^{\widehat{L}} + 1)^2}. \tag{C.4}$$

This is always positive (recall that $z_1 > 1 > \varphi$), such that the claim holds.

(b) A similar technique works for $\Im(\omega) = \frac{\psi}{z_1 - 1/z_1}\left(\frac{1}{1+1/z_1^{\widehat{L}}} - \frac{1}{1+z_1^{\widehat{L}}}\right)$. We find

$$\frac{\mathrm{d}}{\mathrm{d}\widehat{L}}\Im(\omega) = \frac{\psi}{z_1 - 1/z_1}\frac{2z_1^{\widehat{L}}\log z_1}{(z_1^{\widehat{L}}+1)^2}, \tag{C.5}$$

which is a positive quantity, confirming the claim.

(c) From Lemma C.2(a), it follows that

$$\begin{aligned}\Re(\omega) &\geq -\frac{1}{z_1 - z_2}\left(\frac{z_1-\varphi}{1+z_1} + \frac{\varphi - z_2}{1+z_2}\right) = \frac{(z_1-\varphi)(1+z_2)+(\varphi-z_2)(1+z_1)}{(z_2-z_1)(1+z_1)(1+z_2)} \\ &= -\frac{z_1 - z_2 + \varphi(z_1-z_2)}{(z_1-z_2)(2+z_1+z_2)} = -\frac{1+\varphi}{2+z_1+z_2} \\ &= -\frac{1+\varphi}{2+(1+\varphi^2+\psi^2)/\varphi} = -\frac{\varphi(\varphi+1)}{(\varphi+1)^2+\psi^2}.\end{aligned}$$

Thus

$$-\frac{1}{2} < \Re(\omega) \Leftarrow \frac{\varphi(\varphi+1)}{(\varphi+1)^2+\psi^2} < \frac{1}{2} \Leftrightarrow \varphi^2 + \varphi < \frac{\varphi^2}{2} + \varphi + \frac{1}{2} + \frac{\psi^2}{2},$$

the latter of which is true from the condition $0 < \varphi < 1$.

(d) Since $-1/2 < \Re(\omega) < 0$, it holds that $\left|1/2 + \omega_{\{1,2\}}\right|^2$ is bounded above by the squares of $\Re(1/2+\omega_{\{1,2\}})$ maximized over $\widehat{L}$ and $\Im(1/2+\omega)$ maximized over $\widehat{L}$. Thus, using the fact that these maxima are attained for $\widehat{L} \to \infty$ (where $z_1^{\widehat{L}} \to \infty$ and $z_2^{\widehat{L}} \to 0$),

$$\begin{aligned}\left|\frac{1}{2} + \omega_{\{1,2\}}\right|^2 &\leq \left(\frac{1}{2} - \frac{\varphi - z_2}{z_1 - z_2}\right)^2 + \left(\frac{\psi}{z_1 - z_2}\right)^2 \\ &= \left(\frac{(1+\varphi^2+\psi^2)/(2\varphi) - \varphi}{z_1 - z_2}\right)^2 + \left(\frac{\psi}{z_1 - z_2}\right)^2 \\ &= \frac{(1-\varphi^2+\psi^2)^2 + (2\varphi\psi)^2}{\left(2\sqrt{(1+\varphi^2+\psi^2)^2 - 4\varphi^2}\right)^2} = \frac{1}{4}.\end{aligned}$$

This proves the claim. ∎

LEMMA C.3 (Approximation of $1/z$ on a semi-disk). *Denote by $\mathcal{D}_{0.5,+}$ the right half of a disk in the complex plane, centered at 0.5 and with radius 0.5. Define $R = 2$. Then, for any $0 < \rho < R$, there exists some constant κ_ρ such that, for any integer $k \geq 0$, there exists a degree-k polynomial that approximates $f(z) = 1/z$ on $\mathcal{D}_{0.5,+}$ with an infinity-norm error of at most $\kappa_\rho \rho^{-k}$.*

Proof. f is analytic in the complex plane, except for the origin $z = 0$. According to [26, Theorem 4.1], we must find the unique Riemann (conformal) mapping $z \to w(z)$ of the exterior of $\mathcal{D}_{0.5,+}$ to the exterior of the unit disk $\mathcal{D}$ for which $w(\infty) = \infty$ and $w'(\infty) > 0$. The lemma then holds for any R for which f can be analytically extended to the interior of the w-preimage Γ_R of the radius-R origin-centered circle.

In essence, we must find a conformal mapping w from $\mathbb{C}\backslash\mathcal{D}_{0.5,+}$ to $\mathbb{C}\backslash\mathcal{D}$ for which $w(\infty) = \infty$ and $w'(\infty) > 0$, checking how far $w(0)$ is from the origin. We construct

$$w(z) = w_5(w_4(w_3(w_2(w_1(z))))). \tag{C.6}$$

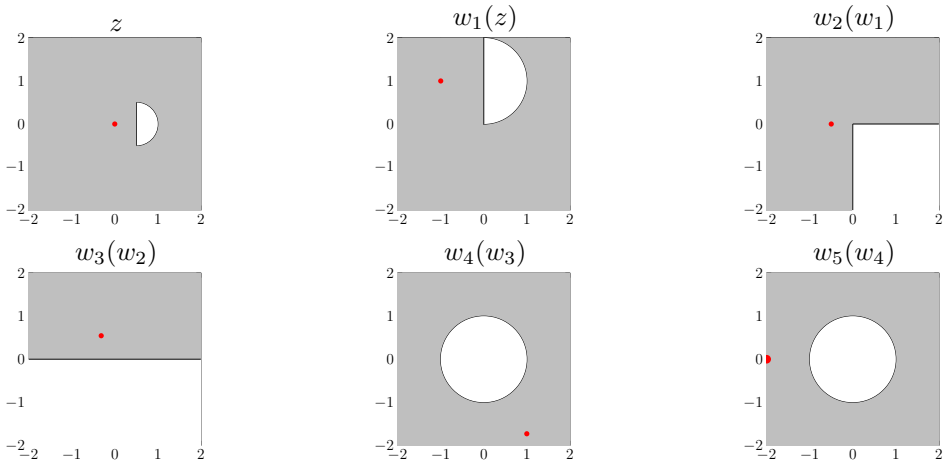


Fig. C.1: Conformal maps forming w. The red dots follow f's pole from $\mathcal{D}_{0.5,+}$ to $\mathcal{D}$. The horizontal and vertical axes denote the real and imaginary parts, respectively.

First, w_1 takes $\mathcal{D}_{0.5,+}$, moves it with its bottom corner to the origin and magnifies it by a factor of two. The mapping that accomplishes this is $w_1 = 2z - 1 + \mathrm{i}$. Then w_2 maps the semi-disk onto a quadrant – or, more to the point, the exterior of the semi-disk into three quadrants. This can be done by the mapping $w_2 = 1/w_1 + \mathrm{i}/2$. Next, w_3 collapses three quadrants into a half-plane with the mapping $w_3 = w_2^{2/3}$. We can then turn a half-plane into the exterior of the unit disk through a Möbius transformation of the form $w_4 = \frac{w_3 - \beta^*}{w_3 - \beta}$ for some β. Recall that w should map ∞ to ∞; this can be done by taking β to be the image of ∞ up until now. If $z = \infty$, we get $w_1 = \infty$, $w_2 = \mathrm{i}/2$ and $w_3 = (\mathrm{i}/2)^{2/3}$. So setting $\beta = (\mathrm{i}/2)^{2/3}$, w_4 is now determined. Finally, we find $w_4'(z = \infty) = (-3\sqrt{3} + 9\mathrm{i})/4$, so with $w_5 = \exp(-2\pi\mathrm{i}/3)w_4$ we get $w'(\infty) = (3\sqrt{3})/2 > 0$.

Figure C.1 illustrates the mapping w. The pole at $z = 0$ maps to $w(0) = -2$, which is at distance $R = 2$ from the origin. This concludes the proof. ∎

Acknowledgments. The authors are grateful to Ignace Bossuyt, Giovanni Conni, Toon Ingelaere, and Vince Maes for their thorough reviews and helpful comments. This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No. 955701. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Belgium, France, Germany, and Switzerland. The work by Karl Meerbergen is partly supported by the Research Foundation Flanders (FWO) grants G0B7818N and G088622N, as well as by the KULeuven Research Council.

REFERENCES

[1] P. Benedusi, M. L. Minion, and R. Krause, *An experimental comparison of a space-time multigrid method with PFASST for a reaction-diffusion problem*, Comput. Math. Appl., 99 (2021), pp. 162–170.

[2] D. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas with Application to Linear Systems Theory*, (2005).

[3] D. A. Bini, G. Latouche, and B. Meini, *Numerical Methods for Structured Markov Chains*, Numer. Math. Sci. Comput., Oxford University Press, 2005.

[4] J. J. Caceres Silva, B. Barán, and C. Schaerer, *Parallel-in-time Parareal implementation using PETSc*, in 2014 XL Latin American Computing Conference (CLEI), 2014, pp. 1–12.

[5] F. Crameri, *Scientific colour maps*. Zenodo, 2021.
[6] P. J. Davis, *Circulant Matrices*, Wiley, 1979.
[7] M. Emmett and M. Minion, *Toward an efficient parallel in time method for partial differential equations*, Commun. Appl. Math. Comput. Sci., 7 (2012), pp. 105–132.
[8] M. J. Gander and L. Halpern, *Time Parallelization for Nonlinear Problems Based on Diagonalization*, in Domain Decomposition Methods in Science and Engineering XXIII, C.-O. Lee, X.-C. Cai, D. E. Keyes, H. H. Kim, A. Klawonn, E.-J. Park, and O. B. Widlund, eds., vol. 116, Springer International Publishing, 2017, pp. 163–170.
[9] M. J. Gander, L. Halpern, J. Rannou, and J. Ryan, *A Direct Time Parallel Solver by Diagonalization for the Wave Equation*, SIAM J. Sci. Comput., 41 (2019), pp. A220–A245.
[10] M. J. Gander, F. Kwok, and J. Salomon, *PARAOPT: A Parareal Algorithm for Optimality Systems*, SIAM J. Sci. Comput., 42 (2020), pp. A2773–A2802.
[11] M. J. Gander, J. Liu, S.-L. Wu, X. Yue, and T. Zhou, *ParaDiag: Parallel-in-time algorithms based on the diagonalization technique*, arXiv:2005.09158 [cs, math], (2021).
[12] M. J. Gander and S.-L. Wu, *Convergence analysis of a periodic-like waveform relaxation method for initial-value problems via the diagonalization technique*, Numer. Math., 143 (2019), pp. 489–527.
[13] A. Goddard and A. Wathen, *A note on parallel preconditioning for all-at-once evolutionary PDEs*, Electron. Trans. Numer. Anal., 51 (2019), pp. 135–150.
[14] S. Götschel and M. L. Minion, *An Efficient Parallel-in-Time Method for Optimization with Parabolic PDEs*, SIAM J. Sci. Comput., 41 (2019), pp. C603–C626.
[15] A. Greenbaum, V. Pták, and Z. Strakoš, *Any Nonincreasing Convergence Curve is Possible for GMRES*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 465–469.
[16] Y. He and J. Liu, *A Vanka-type multigrid solver for complex-shifted Laplacian systems from diagonalization-based parallel-in-time algorithms*, Appl. Math. Lett., 132 (2022), p. 108125.
[17] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, eds., *Optimization with PDE Constraints*, no. 23 in Mathematical Modelling: Theory and Applications, Springer, 2009.
[18] D. Kressner, S. Massei, and J. Zhu, *Improved parallel-in-time integration via low-rank updates and interpolation*, arXiv:2204.03073 [cs, math], (2022).
[19] X.-L. Lin, *A parallel-in-time preconditioner for the Schur complement of parabolic optimal control problems*, (2022). arXiv:2109.12524 [cs, math].
[20] J.-L. Lions, Y. Maday, and G. Turinici, *Résolution d'EDP par un schéma en temps «pararéel»*, C.R. Acad. Sci. Paris Sér. I Math., 332 (2001), pp. 661–668.
[21] J. Liu and Z. Wang, *A ROM-accelerated Parallel-in-Time Preconditioner for Solving All-at-Once Systems from Evolutionary PDEs*, 2020.
[22] J. Liu and S.-L. Wu, *A fast block α-circulant preconditioner for all-at-once system from wave equations*, SIAM J. Matrix Anal. Appl., (2020).
[23] E. McDonald, J. Pestana, and A. Wathen, *Preconditioning and iterative solution of all-at-once systems for evolutionary partial differential equations*, SIAM J. Sci. Comput.
[24] J. W. Pearson, M. Stoll, and A. J. Wathen, *Regularization-Robust Preconditioners for Time-Dependent PDE-Constrained Optimization Problems*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 1126–1152.
[25] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[26] E. Saff, *Logarithmic Potential Theory with Applications to Approximation Theory*, Surv. Approx. Theory, 5 (2010).
[27] S. R. Searle, *On inverting circulant matrices*, Linear Algebra Appl., 25 (1979), pp. 77–89.
[28] C. S. Skene, M. F. Eggl, and P. J. Schmid, *A parallel-in-time approach for accelerating direct-adjoint studies*, J. Comput. Phys., 429 (2021).
[29] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, vol. 50, Siam, 1997.
[30] H. A. van der Vorst, *Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems*, SIAM J. Sci. Statist. Comput., 13 (1992).
[31] S.-L. Wu, *Toward Parallel Coarse Grid Correction for the Parareal Algorithm*, SIAM J. Sci. Comput., 40 (2018), pp. A1446–A1472.
[32] S.-L. Wu and J. Liu, *A Parallel-In-Time Block-Circulant Preconditioner for Optimal Control of Wave Equations*, SIAM J. Sci. Comput., (2020).
[33] S.-L. Wu and T. Zhou, *Diagonalization-based Parallel-in-time algorithms for parabolic PDE-constrained optimization problems*, ESAIM Control Optim. Calc. Var., (2020).