

# Manifold Parareal–A Parareal Algorithm For Gradient Flow

Martin J. Gander<sup>[0000–0001–8450–9223]</sup>,  
Yafei Sun<sup>[0009–0008–7512–9550]</sup>,  
Yingxiang Xu<sup>[0000–0001–7994–5274]</sup>

## 1 Introduction and motivation

We are interested in solving matrix least squares spectrally constrained approximation problems (MLSSCAP) [7, 5], which are a special type of inverse eigenvalue problems for the reconstruction of a structured matrix from prescribed spectral data [6, 4], with important applications such as control design and system identification [8, 5, 4]. An illustrative example for an MLSSCAP is (see [7, Problem A]):

**Example A.** For a symmetric matrix  $A \in \mathbb{R}^{N \times N}$  and  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$  with distinct  $\lambda_i \in \mathbb{R}$ , find a symmetric matrix  $X^* \in \mathbb{R}^{N \times N}$ , such that  $X^*$  has the same eigenvalues as  $\Lambda$  and approximates  $A$  with minimal Frobenius norm error.

The MLSSCAP for this example can be written as a least squares problem (LSP),

$$X^* = \arg \min_{X \in \{Q^T \Lambda Q \mid Q \in O(N)\}} F(X) = \arg \min_{X \in \{Q^T \Lambda Q \mid Q \in O(N)\}} \frac{1}{2} \|X - A\|_F^2, \quad (1)$$

where  $O(N) := \{Q \in \mathbb{R}^{N \times N} \mid Q^T Q = I\}$  is called Stiefel manifold, the set of all orthogonal matrices [2]. Using the spectral constraint condition  $X = Q^T \Lambda Q$ ,  $Q \in O(N)$ , the MLSSCAP (1) can be equivalently written as

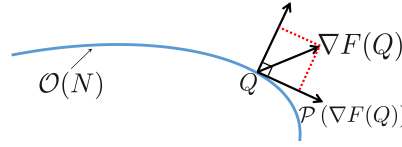
$$\arg \min_{Q \in O(N)} F(Q) := \arg \min_{Q \in O(N)} \frac{1}{2} \|Q^T \Lambda Q - A\|_F^2. \quad (2)$$

---

Martin J. Gander  
University of Geneva, Department of Mathematics, Switzerland, e-mail: martin.gander@unige.ch

Yafei Sun  
Jilin University, School of Mathematics, China, e-mail: yafeisun@jlu.edu.cn

Yingxiang Xu  
Northeast Normal University, School of Mathematics and Statistics, China, e-mail: yxxu@nenu.edu.cn



**Fig. 1** Schematic illustration of the projected gradient flow.

From the solution  $Q_* \in \mathcal{O}(N)$  to (2), we obtain with  $X^* = Q_*^\top \Lambda Q_*$  the solution of (1), and thus focus here on (2), to which we also refer to as MLSSCAP.

To solve MLSSCAP (2), consider gradient descent:  $(Q_{n+1} - Q_n) / \alpha = -\nabla F(Q_n)$  with  $Q_0 \in \mathcal{O}(N)$  and  $\alpha > 0$ . Taking the limit  $\alpha \rightarrow 0$ , we obtain a continuous counterpart of the gradient descent method, namely, the gradient flow [14],

$$\frac{dQ(t)}{dt} = -\nabla F(Q(t)), \quad (3)$$

which is a first-order matrix-valued ODE. By integrating this initial value problem (IVP) with the initial condition  $Q(0) = Q_0 \in \mathcal{O}(N)$ , the cost function  $F(Q)$  is guaranteed to be non-increasing along the forward flow  $Q(t)$ , because

$$\frac{dF(Q(t))}{dt} = \sum_{i=1}^N \sum_{j=1}^N \frac{\partial F(Q)}{\partial Q_{ij}} \frac{dQ_{ij}}{dt} = \langle \nabla F(Q), \frac{dQ(t)}{dt} \rangle = -\|\nabla F(Q)\|_F^2 \leq 0. \quad (4)$$

Further assumptions on  $F(Q(t))$ , such as Lipschitz continuity, allows one to prove convergence of  $Q(t)$  to a stationary point  $Q_*$  as  $t \rightarrow \infty$  [1, 14].

Following the trajectory of the gradient flow system of ODEs (3) constitutes an important class of techniques in optimization [6]. There is however an important omission in the gradient flow (3) for solving MLSSCAP (2): the orthogonality constraint on the solution  $Q$  is neglected. As a result, the trajectory  $Q(t)$  is not guaranteed to remain on the Stiefel manifold  $\mathcal{O}(N)$ , and the numerical solution  $Q_*$  may lose orthogonality, thereby destroying the isospectral property of  $X^* = Q_*^\top \Lambda Q_*$ .

To properly account for the orthogonality constraint on the solution  $Q(t)$ , a corrected gradient flow was introduced in [7, 2] by projecting the gradient  $\nabla F(Q)$  onto the tangent space at the current point  $Q$  on  $\mathcal{O}(N)$ ,

$$\frac{dQ(t)}{dt} = -\mathcal{P}(\nabla F(Q(t))), \quad (5)$$

where  $\mathcal{P}(\nabla F(Q(t)))$  denotes the projection of the gradient of  $F(Q)$  onto the tangent space of  $\mathcal{O}(N)$ . An intuitive illustration of this projected gradient is shown in Fig. 1. Since the projected gradient flow (5) defines a matrix-valued ODE on the Stiefel manifold  $\mathcal{O}(N)$ , and the term  $-\mathcal{P}(\nabla F(Q(t)))$  represents the negative Riemannian gradient of  $F(Q(t))$  on  $\mathcal{O}(N)$ , integrating the IVP (5) with  $Q_0$  amounts to searching continuously a minimizer of MLSSCAP (2) on  $\mathcal{O}(N)$ . In this way, the orthogonality constraint is well considered.

To demonstrate the enhanced performance of the projected gradient flow (5) in comparison with the classical gradient flow (3), we consider **Example A** with

$$A = \begin{bmatrix} 1.4096 & 0.9432 & 0.6802 & 0.6224 & 1.0136 & 0.6420 & 0.5767 & 1.3365 & 0.5802 & 1.5269 \\ 0.9432 & 0.1795 & 0.4110 & 0.1623 & 0.9871 & 1.0296 & 0.8267 & 1.2512 & 1.1203 & 1.5517 \\ 0.6802 & 0.4110 & 1.4846 & 1.0656 & 1.0864 & 0.6946 & 1.4645 & 1.2591 & 1.0640 & 0.5390 \\ 0.6224 & 0.1623 & 1.0656 & 0.0671 & 0.8135 & 0.6334 & 1.6968 & 1.8564 & 1.1832 & 0.7695 \\ 1.0136 & 0.9871 & 1.0864 & 0.8135 & 0.0557 & 0.9542 & 0.6078 & 1.2076 & 1.1080 & 0.9857 \\ 0.6420 & 1.0296 & 0.6946 & 0.6334 & 0.9542 & 0.0312 & 0.8811 & 1.0510 & 1.3304 & 0.7840 \\ 0.5767 & 0.8267 & 1.4645 & 1.6968 & 0.6078 & 0.8811 & 1.5464 & 0.8840 & 0.6552 & 0.4812 \\ 1.3365 & 1.2512 & 1.2591 & 1.8564 & 1.2076 & 1.0510 & 0.8840 & 1.9671 & 1.1637 & 0.9480 \\ 0.5802 & 1.1203 & 1.0640 & 1.1832 & 1.1080 & 1.3304 & 0.6552 & 1.1637 & 0.5162 & 1.2659 \\ 1.5269 & 1.5517 & 0.5390 & 0.7695 & 0.9857 & 0.7840 & 0.4812 & 0.9480 & 1.2659 & 1.2618 \end{bmatrix},$$

$$\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N) = \text{diag}(10.7432, 1.9428, 1.3788, 0.9681, 0.8285, 0.0378, -0.2702, -0.8675, -1.6232, -1.8710).$$

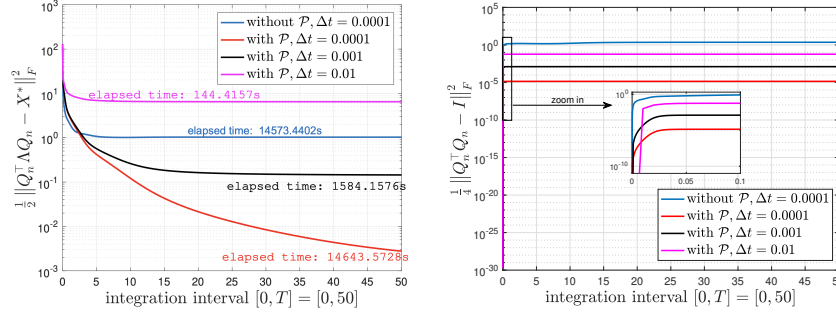
We first provide the gradient  $\nabla F(Q)$  and the projected gradient  $\mathcal{P}(\nabla F(Q))$ . By using the properties of the Frobenius product  $\langle \cdot, \cdot \rangle$  and the fact that  $\Lambda$  and  $Q^\top \Lambda Q - A$  are symmetric matrices, the differential of  $F(Q)$  with respect to  $Q$  is  $dF(Q) = d\frac{1}{2}\langle Q^\top \Lambda Q - A, Q^\top \Lambda Q - A \rangle = \langle 2\Lambda Q(Q^\top \Lambda Q - A), dQ \rangle$ . The relation  $dF(Q) = \langle \nabla F(Q), dQ \rangle$  (see Eq. (4)) gives the gradient of  $F(Q)$  as  $\nabla F(Q) = 2\Lambda Q(Q^\top \Lambda Q - A)$ . Accordingly, the projected gradient  $\mathcal{P}(\nabla F(Q))$  can be computed following the procedure described in [7, Section 3], [1, Section 2]:  $\mathcal{P}(\nabla F(Q)) = Q [Q^\top \Lambda Q(Q^\top \Lambda Q - A) - (Q^\top \Lambda Q - A)Q^\top \Lambda Q]$ .

We now use the classical gradient flow (3) and the projected gradient flow (5) to solve this concrete example. This can be achieved by integrating the corresponding IVPs with the same initial condition  $Q(0) = Q_0 \in \mathcal{O}(N)$ ,

$$\begin{cases} \frac{dQ(t)}{dt} = -\nabla F(Q(t)), & t \in (0, T], \\ Q(0) = Q_0, & t = 0. \end{cases} \quad \begin{cases} \frac{dQ(t)}{dt} = -\mathcal{P}(\nabla F(Q(t))), & t \in (0, T], \\ Q(0) = Q_0, & t = 0. \end{cases} \tag{6}$$

For the numerical discretization, we use the Backward Euler method (BE). Since the RHS  $-\nabla F(Q(t))$  and  $-\mathcal{P}(\nabla F(Q(t)))$  are nonlinear functions of  $Q$ , we need to solve a nonlinear equation at each step. The initial condition  $Q_0 \in \mathcal{O}(N)$  for the problems in (6) is given as

$$\begin{bmatrix} -0.3140 & 0.3838 & -0.4861 & 0.3858 & -0.4778 & 0.2095 & -0.0911 & -0.1463 & -0.0628 & 0.2513 \\ -0.3254 & -0.2401 & -0.2667 & 0.4809 & 0.4037 & -0.3459 & 0.0540 & 0.0472 & -0.4016 & -0.2916 \\ -0.2795 & -0.2853 & 0.3780 & 0.2488 & -0.5903 & -0.0599 & 0.1998 & 0.2367 & 0.2255 & -0.3701 \\ -0.2504 & 0.5228 & 0.0446 & -0.3719 & -0.0478 & -0.3133 & 0.2988 & -0.3920 & -0.0640 & -0.4197 \\ -0.3025 & -0.5137 & -0.1628 & -0.4032 & -0.1060 & 0.3014 & 0.4130 & -0.2123 & -0.3083 & 0.2069 \\ -0.3397 & -0.0968 & -0.1956 & -0.3413 & -0.0912 & -0.6203 & -0.2064 & 0.3022 & 0.2343 & 0.3723 \\ -0.4054 & 0.0327 & 0.2719 & -0.2313 & 0.0081 & 0.2783 & -0.6943 & 0.0589 & -0.3277 & -0.1915 \\ -0.3057 & 0.3614 & 0.0327 & -0.0539 & 0.3102 & 0.3307 & 0.3745 & 0.6478 & 0.0595 & 0.0540 \\ -0.3426 & -0.1612 & -0.1447 & 0.0499 & 0.3393 & 0.2377 & -0.1088 & -0.3350 & 0.7184 & -0.1463 \\ -0.2694 & 0.0971 & 0.6221 & 0.2860 & 0.1645 & -0.1256 & 0.1107 & -0.3080 & -0.0534 & 0.5444 \end{bmatrix}.$$



**Fig. 2** Left: convergence behavior of the discretized gradient flow with and without projection  $\mathcal{P}$  for different integration step sizes  $\Delta t$ . The elapsed time is also recorded for the discretized gradient flow with  $\mathcal{P}$ . Right: orthogonality of the numerical approximation  $Q^n$ . Integration interval  $[0, T] = [0, 50]$ .

We introduce several quantities to assess the performance of the gradient flows (5) and (3) when discretized using BE. Note that the matrix  $A$  has distinct eigenvalues, it thus has a unique eigen-decomposition (ignoring the sign of the eigenvectors):  $A = Q_A^T \Lambda_A Q_A$  with  $\Lambda_A = \text{diag}(\lambda_1^A, \dots, \lambda_N^A)$ . Furthermore, if the eigenvalues are sorted in the same order, for example,  $\lambda_1 > \dots > \lambda_N$  and  $\lambda_1^A > \dots > \lambda_N^A$ , MLSSCAP (1) has a theoretically optimal solution  $X^* = Q_A^T \Lambda Q_A$  (see [3, Theorem 3.2] and [12, Theorem 6.3.5]). This implies that for any optimal solution  $Q_* \in \mathcal{O}(N)$  to MLSSCAP (2), it should satisfy  $\frac{1}{2} \|Q_*^T \Lambda Q_* - A\|_F^2 = \frac{1}{2} \|X^* - A\|_F^2 = \frac{1}{2} \|Q_A^T \Lambda Q_A - A\|_F^2$ . Since  $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$  with distinct  $\lambda_i, i = 1, \dots, N$ , we have  $X^* = Q_A^T \Lambda Q_A = Q_*^T \Lambda Q_*$ , indicating that changing the sign of different rows in  $Q_A$  yields different  $Q_*$ . Hence, when  $Q_n$  (the numerical approximation at  $t_n = n\Delta t$ ) tends to one of the optimal solutions  $Q_*$  as  $n \rightarrow \infty$ , we have  $Q_n^T \Lambda Q_n \rightarrow X^*$ . Based on this fact, we use the quantity  $\frac{1}{2} \|Q_n^T \Lambda Q_n - X^*\|_F^2$  to characterize the convergence behavior of the gradient flows. To evaluate how well  $Q_n$  preserves orthogonality, we use the orthogonality measure  $\frac{1}{4} \|Q_n^T Q_n - I\|_F^2$  [1].

The corresponding numerical results are shown in Fig. 2. We see that the classical gradient flow (3) fails to converge to an optimal solution. The quantity  $\frac{1}{2} \|Q_n^T \Lambda Q_n - X^*\|_F^2$  cannot be reduced and the orthogonality condition is also violated, even though a small step  $\Delta t = 0.0001$  is used. In contrast, the projected gradient flow (5) significantly improves both the search for optimal solutions and the preservation of orthogonality. Even for larger integration steps, it still yields superior results in terms of optimality (cf.  $\Delta t = 0.001$ ) and orthogonality (cf.  $\Delta t = 0.001, 0.01$ ). Moreover, this superiority can be further enhanced by using a smaller step size, and/or a larger integration interval, as the quantity  $\frac{1}{2} \|Q_n^T \Lambda Q_n - X^*\|_F^2$  still exhibits a decreasing trend when the integration reaches  $T$ . However, the computational time increases rapidly, as shown in Fig. 2, where the elapsed times were recorded using MATLAB's build-in functions `tic` and `toc`. We see that for the projected gradient flow with  $\Delta t = 0.0001$ , the computation time is almost 14644 seconds. This issue is further exacerbated by the fact that the integration must be performed over long time intervals to converge

to a stationary point. Moreover, in practical applications, additional factors such as the problem size (i.e., the matrix dimension) and the complexity of the gradient evaluation can further contribute to a substantial increase in computational cost.

To address the high computational cost that arises when very small integration steps and large integration intervals are required to solve the projected gradient flow for MLSSCAP using this approach<sup>1</sup>, we propose in the next section a new parallel algorithm—*Manifold Parareal*, which greatly accelerates the numerical integration and ensures orthogonality.

## 2 Manifold Parareal

Since the gradient flow can be viewed as a time-dependent matrix-valued ODE, the Parareal algorithm [13] provides an efficient parallel strategy for accelerating its numerical solution. The classical parareal algorithm uses two propagators,  $\mathcal{F}_{\delta t}$  and  $\mathcal{G}_{\Delta t}$ , where classically  $\mathcal{F}_{\delta t}$  uses a fine time step  $\delta t$  and  $\mathcal{G}_{\Delta t}$  a coarse step  $\Delta t$ . Usually an even larger time interval  $\Delta T$  is used to partition the time interval  $[0, T]$  of interest with  $T_n = n\Delta T$ , see e.g. [10, Fig. 1]. For any initial guess  $Q_n^0$  at  $t = T_n$ , the classical Parareal algorithm computes for iteration index  $k = 0, 1, 2, \dots$  and with  $Q_0^k = Q_0$

$$Q_{n+1}^{k+1} = \mathcal{G}_{\Delta t} \left( T_n, Q_n^{k+1} \right) + \mathcal{F}_{\delta t} \left( T_n, Q_n^k \right) - \mathcal{G}_{\Delta t} \left( T_n, Q_n^k \right), \quad n = 0, 1, \dots, N_t - 1, \quad (7)$$

where  $N_t = T/\Delta T$ . The initial guess  $\{Q_n^0\}_{n \geq 1}$  can be random or generated by the  $\mathcal{G}_{\Delta t}$  propagator. Here and hereafter,  $\mathcal{F}_{\delta t} (T_n, Q_n^k)$  and  $\mathcal{G}_{\Delta t} (T_n, Q_n^k)$  denote numerical approximations of the projected gradient flow (5) at  $t = T_{n+1}$ , with initial value  $Q_n^k$  at  $t = T_n$ . At convergence, we have from (7) that  $Q_{n+1}^\infty = \mathcal{F}_{\delta t} (T_n, Q_n^\infty)$ , i.e., the approximation at  $T_n$  has the accuracy of the fine propagator. This shows that the final numerical approximation  $Q_n^\infty$  may still not be (approximately) optimal and its orthogonality may remain unsatisfactory when a standard numerical scheme (e.g., BE) is used for  $\mathcal{F}_{\delta t}$  and the time step  $\delta t$  is not sufficiently small.

To overcome this, we adopt the idea of the *Standard Projection Method* introduced in [11, Chapter 4, Algorithm 4.2]. In particular, we incorporate a *retraction* strategy into the standard numerical scheme, and the resulting new scheme is then used as the propagators  $\mathcal{F}_{\delta t}$  and  $\mathcal{G}_{\Delta t}$ . The retraction  $\mathcal{R}$  refers to an operation that pulls the current computed approximation  $Q_{n+1}$ , which may not lie on the Stiefel manifold  $O(N)$ , back onto  $O(N)$ . In practice, for instance, when using BE with step size  $\Delta t$  to integrate the projected gradient flow (5), the retraction  $\mathcal{R}$  is applied as follows:

$$\text{BE}_{\mathcal{P}\mathcal{R}} : \hat{Q}_{n+1} = Q_n - \Delta t \mathcal{P}(\nabla F(\hat{Q}_{n+1})), \quad Q_{n+1} = \mathcal{R}(\hat{Q}_{n+1}) \in O(N). \quad (8)$$

<sup>1</sup> We will later use retractions in the context of Manifold Parareal, but one could also use orthogonality-preserving methods.

An illustrative diagram for this kind of method can be found in [11, Chapter 4, Figure 4.2]. Although a variety of retractions  $\mathcal{R}$  could be used, we restrict our attention here to the one based on the singular value decomposition (SVD). As shown in [11, Chapter 4, Section 9.1], if the SVD of  $\hat{Q}_{n+1}$  is  $\hat{Q}_{n+1} = U\Sigma V^\top$ , then the optimal retraction is given by  $Q_{n+1} = \mathcal{R}(\hat{Q}_{n+1}) = UV^\top \in \mathcal{O}(N)$ , in the sense that  $Q_{n+1} = UV^\top$  solves the optimization problem  $\arg \min_{Q \in \mathcal{O}(N)} \|Q - \hat{Q}_{n+1}\|_F$ . Note that an explicit scheme with retraction  $\mathcal{R}$  could also be used to solve the MLSSCAP, provided the step size  $\Delta t$  is sufficiently small. However, due to the strong nonlinearity of the gradient term  $\mathcal{P}(\nabla F(Q))$ , the gradient flow system is generally stiff, making implicit time integration preferable. As a result, the dominant computational cost in each step arises from solving the nonlinear system. For the moderate size  $N = 10$  considered here, this system can be solved efficiently using a generic nonlinear solver, and the associated cost is modest (on the order of  $O(N^6)$ ). Moreover, the SVD-based retraction is performed only once per step and requires only  $O(N^3)$  operations, which remains small compared to the cost of the nonlinear solve. For larger  $N$ , more scalable nonlinear solvers (e.g., matrix-free Newton-Krylov methods) would be needed, but such modifications do not affect the parareal framework, which operates at the time-integration level.

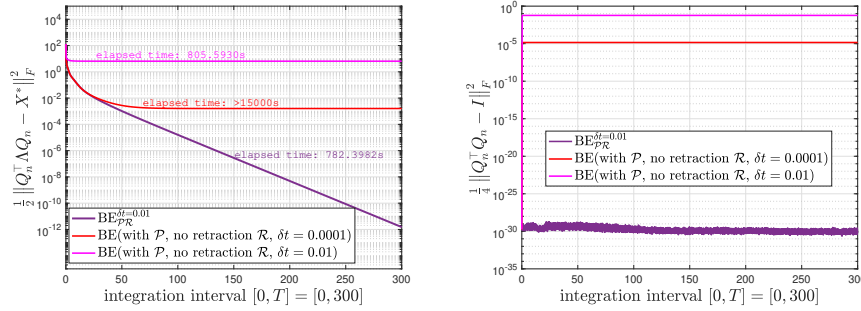
Using the modified BE scheme (8) as the fine and coarse propagators leads to our new Manifold Parareal algorithm for solving the projected gradient flow (5) defined on the Stiefel manifold  $\mathcal{O}(N)$ ,

$$Q_{n+1}^{k+1} = \mathcal{G}_{\mathcal{P}\mathcal{R}}^{\Delta t} \left( T_n, Q_n^{k+1} \right) + \mathcal{F}_{\mathcal{P}\mathcal{R}}^{\delta t} \left( T_n, Q_n^k \right) - \mathcal{G}_{\mathcal{P}\mathcal{R}}^{\Delta t} \left( T_n, Q_n^k \right), \quad (9)$$

where we use the notations  $\mathcal{F}_{\mathcal{P}\mathcal{R}}^{\delta t}$  and  $\mathcal{G}_{\mathcal{P}\mathcal{R}}^{\Delta t}$  to distinguish these new propagators from those in the classical algorithm (7). We note that Parareal has previously been extended to problems with geometric constraints, such as Hamiltonian systems on symplectic manifolds, where structure-preserving variants have been developed to maintain symplecticity [9]. In contrast, the proposed Manifold Parareal (9) here focuses on dissipative gradient flows on the orthogonal manifold, where the solution must preserve orthogonality throughout the evolution. The performance of this algorithm will be investigated in the next section.

### 3 Numerical Experiments

We reconsider **Example A**, using the data specified in Section 1. The projected gradient flow (5) is first solved sequentially using the new fine propagator  $\mathcal{F}_{\mathcal{P}\mathcal{R}}^{\delta t}$  with time step  $\delta t = 0.01$  on the integration interval  $[0, T] = [0, 300]$ . For comparison, the results in Fig. 2 for BE (with  $\mathcal{P}$ ,  $\delta t = 0.01$  and  $\delta t = 0.0001$ ) without retraction are also included. The corresponding convergence results are shown in Fig. 3. We observe on the left that the new scheme  $\text{BE}_{\mathcal{P}\mathcal{R}}^{\delta t}$ , with retractions implemented via SVD, yields significantly more accurate results than the standard BE, even though the time step used,  $\delta t = 0.01$ , is much larger than  $\delta t = 0.0001$ . Moreover, the new



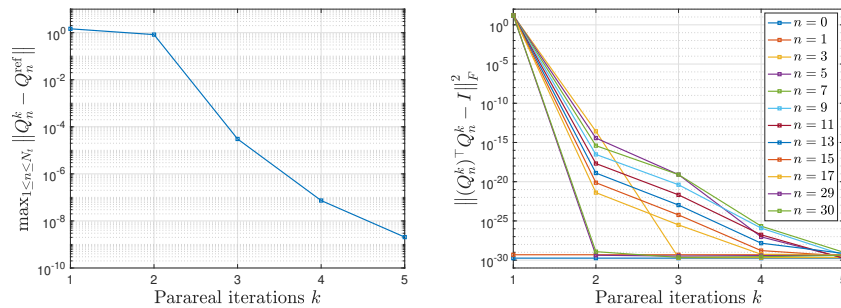
**Fig. 3** Comparison of the new retraction-based Euler scheme  $\text{FE}_{\mathcal{P}\mathcal{R}}^{\delta t}$  (8) and the standard Euler method for **Example A**: convergence (left) and orthogonality preservation (right).

scheme  $\text{BE}_{\mathcal{P}\mathcal{R}}^{\delta t}$  requires substantially less computational time (approximately 782 seconds), compared with the more than 15000 seconds needed by standard BE with  $\delta t = 0.0001$ . In addition, the orthogonality of the numerical solutions is very well preserved, as shown in the right plot.

We next investigate the performance of the Manifold Parareal algorithm (9), where the new integrator  $\text{BE}_{\mathcal{P}\mathcal{R}}^{\delta t}$  (8) is used as the fine and coarse propagators  $\mathcal{F}_{\mathcal{P}\mathcal{R}}^{\delta t}, \mathcal{G}_{\mathcal{P}\mathcal{R}}^{\Delta t}$ . The integration interval  $[0, T] = [0, 300]$  is partitioned into 30 subintervals of length  $\Delta T = 10$ , giving  $N_t = T/\Delta T = 30$ . The fine and coarse time steps are set to  $\delta t = 0.01, \Delta t = 0.1$ . For all numerical experiments, the Manifold Parareal algorithm starts with a random guess and stops when  $\max_{1 \leq n \leq N_t} \|Q_n^k - Q_n^{\text{ref}}\| < \epsilon$ , with  $\epsilon = 10^{-8}$  chosen to match the tolerance of the nested nonlinear solver. This prevents numerical noise from affecting the convergence study and ensures that the observed behavior is due solely to the Parareal iteration. Here  $\{Q_n^{\text{ref}}\}_{n=1}^{N_t}$  denotes the reference solution obtained by direct application of the fine propagator. In practical applications, since the reference solution is unavailable, the iteration can be terminated when the difference of iterates or the residual is small. Moreover, although a fixed final time  $T$  is used here for comparison, gradient flows are typically stopped adaptively near steady state, based on decay of the energy, the norm of the projected gradient, or successive step differences. The effective final time is thus determined dynamically. These stopping criteria can be combined with a multiple-time-window strategy to further improve efficiency, though this is not pursued here.

The convergence results in Fig. 4 show that for the given data setting, Manifold Parareal achieves convergence within 5 iterations, and numerical solutions at all time points satisfy orthogonality. This indicates that Manifold Parareal not only yields numerical solutions whose accuracy matches that of the fine-grid serial computation (the numerical approximation  $Q_T$  at the final time  $t = T$  also satisfies  $\frac{1}{2} \|(Q_T)^T \Lambda Q_T - X^*\|_F^2 \approx 10^{-12}$  (see Fig. 3)) but also achieves pronounced parallel acceleration, effectively reducing the computational cost of the entire solution (optimization) procedure.

To conclude, our new Manifold Parareal algorithm for gradient flows arising from spectrally constrained matrix least-squares problems incorporates retractions



**Fig. 4** Left: Convergence behavior of Manifold Parareal (9). Right: Changes in the orthogonality of numerical solutions at different time points during parareal iterations.

into the fine and coarse propagators to achieve rapid convergence and computational efficiency while preserving orthogonality. Retractions may be applied at each step or adaptively to reduce unnecessary matrix operations; a theoretical analysis of this strategy is ongoing. Future work includes large-scale performance studies and applications where the solution trajectory is of interest.

## References

1. Ablin, P., Peyré, G.: Fast and accurate optimization on the orthogonal manifold without retraction. *Proc. Mach. Learn. Res.* **151**, 5636–5657 (2022)
2. Boumal, N.: *An Introduction to Optimization on Smooth Manifolds*. Cambridge University Press, Cambridge (2023)
3. Chen, X., Chu, M.T.: On the least squares solution of inverse eigenvalue problems. *SIAM J. Numer. Anal.* **33**(6), 2417–2430 (1996)
4. Chu, M.T.: A list of matrix flows with applications. In: *Proceedings of the Workshop on Hamiltonian and Gradient Flows, Algorithms and Control, Fields Institute Communications*, vol. 3, pp. 87–97. Fields Institute, Fields Institute, Canada (1994)
5. Chu, M.T.: Inverse eigenvalue problems. *SIAM Rev.* **40**, 1–39 (1998)
6. Chu, M.T.: Linear algebra algorithms as dynamical systems. *Acta Numer.* **17**, 1–86 (2008)
7. Chu, M.T., Driessel, K.R.: The projected gradient method for least squares matrix approximations with spectral constraints. *SIAM J. Sci. Comput.* **27**(4), 1050–1060 (1990)
8. Chu, M.T., Golub, G.H.: Structured inverse eigenvalue problems. *Acta Numer.* **11**, 1–71 (2002)
9. Dai, X., Bris, C.L., Legoll, F., Maday, Y.: Symmetric parareal algorithms for hamiltonian systems. *ESAIM Math. Model. Numer. Anal.* **47**(3), 717–742 (2013)
10. Gander, M.J., Wu, S.L.: A diagonalization-based parareal algorithm for dissipative and wave propagation problems. *SIAM J. Numer. Anal.* **58**(5), 2981–3009 (2020)
11. Hairer, E., Wanner, G., Lubich, C.: *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, 2 edn. Springer Ser. Comput. Math. Springer Berlin, Heidelberg (2006)
12. Horn, R.A., Johnson, C.R.: *Matrix Analysis*. Cambridge University Press, Cambridge (2012)
13. Lions, J.L., Maday, Y., Turinici, G.: A “parareal” in time discretization of PDE’s. *C. R. Acad. Sci. Paris Sér. I Math.* **332**, 661–668 (2001)
14. Tarmoun, S., Franca, G., Haeffele, B.D., Vidal, R.: Understanding the dynamics of gradient flow in overparameterized linear models. *Proc. Mach. Learn. Res.* **139**, 10153–10161 (2021)