A DIAGONALIZATION-BASED PARAREAL ALGORITHM FOR DISSIPATIVE AND WAVE PROPAGATION PROBLEMS

MARTIN J. GANDER* AND SHU-LIN WU†

Abstract. We present a new parareal algorithm based on a *diagonalization* technique proposed recently. The algorithm uses a single implicit Runge-Kutta (IRK) method with the same small step-size for both the $\mathcal{F}$ and $\mathcal{G}$ propagators in parareal, and would thus converge in one iteration when used directly like this, however without any speedup due to the sequential way parareal uses $\mathcal{G}$. We then approximate $\mathcal{G}$ with a *head-tail* coupled condition such that $\mathcal{G}$ can be parallelized using diagonalization in time. We show that with a suitable choice of the parameter in the *head-tail* condition, our new parareal algorithm converges very rapidly, both for parabolic and hyperbolic problems, even in the non-linear case. We illustrate our new algorithm with numerical experiments.

Key words. Parareal algorithm, Diagonalization technique, Parallel coarse propagator, Dissipative problems, Wave propagation problems, Convergence analysis

AMS subject classifications. 65M55, 65M12, 65M15, 65Y05

1. Introduction.

We are interested in using the parareal algorithm [26] for solving initial value problems of the form

$$\dot{u} + f(t,u) = 0,\ u(0) = u_0, \tag{1.1}$$

where $f : (0,T) \times \mathbb{R}^m \to \mathbb{R}^m$, and $u_0 \in \mathbb{R}^m$. The parareal algorithm uses two propagators $\mathcal{F}$ and $\mathcal{G}$, where classically $\mathcal{F}$ uses a small (fine) time step δt and $\mathcal{G}$ a large (coarse) time step Δt. An even larger time interval ΔT is used to partition the time interval $(0,T)$ of interest with $T_n = n\Delta T$, see Figure 1.1. For any initial guess u_n^0 at $t = T_n$, the parareal algorithm computes for iteration index $k = 0, 1, 2, \ldots$ and with $u_0^k = u_0$

$$u_{n+1}^{k+1} = \mathcal{G}_{\Delta t}(T_n, u_n^{k+1}) + \mathcal{F}_{\delta t}(T_n, u_n^k) - \mathcal{G}_{\Delta t}(T_n, u_n^k),\ n = 0, 1, \ldots, N_t - 1, \tag{1.2}$$

where $N_t = \frac{T}{\Delta T}$. The initial guess $\{u_n^0\}_{n\geq 1}$ can be random or generated by the $\mathcal{G}$-propagator. Here and hereafter, $\mathcal{F}_{\delta t}(T_n, u_n^k)$ and $\mathcal{G}_{\Delta t}(T_n, u_n^k)$ denote numerical solutions of (1.1) at $t = T_{n+1}$, with initial value u_n^k at $t = T_n$. Upon convergence, we have from (1.2) that $u_{n+1}^\infty = \mathcal{F}_{\delta t}(T_n, u_n^\infty)$, i.e., the approximation at T_n has the accuracy of the fine propagator $\mathcal{F}$.

Convergence of the parareal algorithm is well studied [16, 19, 39, 40, 43], and parareal is widely used [4, 22, 25, 28, 31, 33, 34, 37]. The parareal algorithm also provides insight for understanding and/or designing new parallel-in-time (PinT) algorithms, e.g., the MGRIT algorithm [2, 9], the PFFAST algorithm [7] and the ParaExp algorithm [15]. For a survey of PinT algorithms, see [20], and for further PinT activities `http://parallel-in-time.org`. The classical parareal algorithm (1.2) works well for *dissipative* ODEs, while for wave propagation problems the algorithm has convergence problems [19]. To illustrate this, we consider

$$\partial_t u - \nu\partial_{xx}u + u_x = 0,\ (x,t) \in (-1,1) \times (0,4), \tag{1.3a}$$

where $\nu > 0$, together with initial value $u(x,0) = e^{-20x^2}$ and periodic boundary conditions

*Martin J. Gander, Section de Mathématiques, University of Geneva, CH-1211 Geneva, Switzerland. E-mail: `martin.gander@unige.ch`

†Shu-Lin Wu (corresponding author), School of Mathematics and Statistics, Northeast Normal University, Changchun 130024, China. E-mail: `wushulin84@hotmail.com`

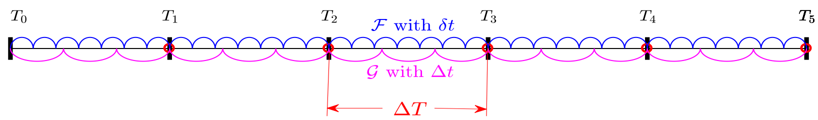

FIG. 1.1. *Grid setting for the classical parareal algorithm: $\Delta T > \Delta t > \delta t$.*

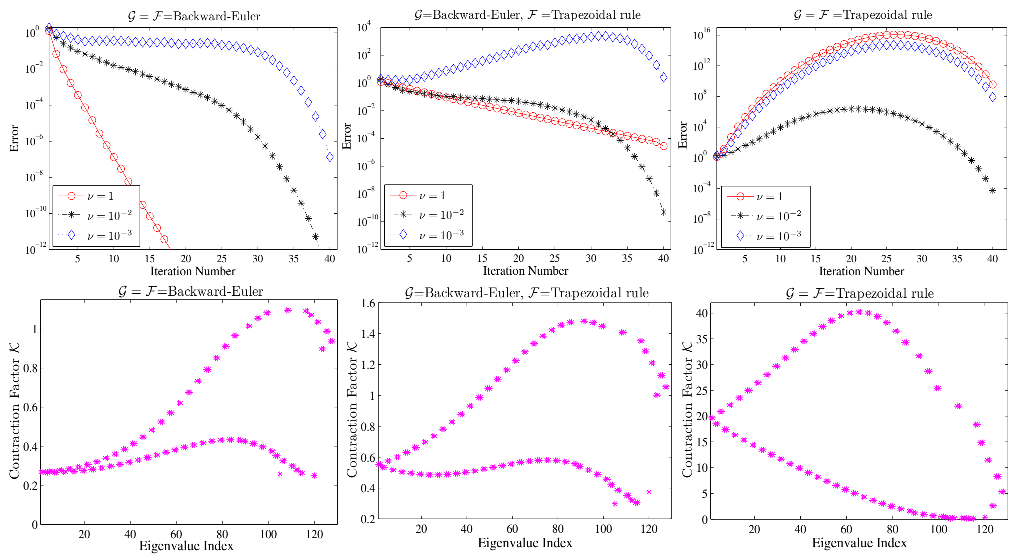

FIG. 1.2. *Top: the error at each parareal iteration with three different values of ν. Bottom: the contraction factor $\mathcal{K}$ of the parareal algorithm for $\nu = 10^{-3}$ at each eigenvalue of the matrix A.*

$u(-1,t) = u(1,t)$. Applying the centered finite difference formula with $\Delta x = \frac{1}{64}$ leads to

$$\dot{\boldsymbol{u}} + A\boldsymbol{u} = 0,\ \ A = \frac{\nu}{\Delta x^2}\begin{bmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ -1 & & & -1 & 2 \end{bmatrix} + \frac{1}{2\Delta x}\begin{bmatrix} 0 & 1 & & & -1 \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ 1 & & & -1 & 0 \end{bmatrix}. \tag{1.3b}$$

The diffusion coefficient ν controls the wave characteristic of the solution $\boldsymbol{u}(t)$. Let $\Delta T = 0.1$, $\Delta t = \Delta T$ and $\delta t = 0.01$. Then for three choices of the propagators $\mathcal{G}$ and $\mathcal{F}$, we show in Figure 1.2 (top) the error $\max_{n=1,2,\ldots,N_t} \|u_n - u_n^k\|_\infty$ of the parareal algorithm at each iteration, where $\{u_n\}$ is the converged solution and $\{u_n^k\}$ is the k-th iterate. We see that as ν becomes small, i.e., the PDE (1.3a) becomes advection dominated, the convergence rate of the parareal algorithm deteriorates.

The disappointing convergence behavior of the parareal algorithm can be understood from its convergence factor [19],

$$\rho = \max_{z\in\sigma(\Delta T A)} \mathcal{K}(J, \widetilde{J}, z), \quad \mathcal{K} := \frac{\left|\mathcal{R}_f^J(\frac{z}{J}) - \mathcal{R}_g^{\widetilde{J}}(\frac{z}{\widetilde{J}})\right|}{1 - |\mathcal{R}_g^{\widetilde{J}}(\frac{z}{\widetilde{J}})|}, \tag{1.4}$$

where $\mathcal{R}_f$ and $\mathcal{R}_g$ are the stability functions of the propagators $\mathcal{F}$ and $\mathcal{G}$, $J\colon = \Delta T/\delta t$ and $\widetilde{J}\colon = \Delta T/\Delta t$, and $\sigma(\cdot)$ denotes the spectrum of the matrix involved. We call $\mathcal{K}$ the 'contraction factor', which is the convergence factor corresponding to a single eigenvalue. For $\nu = 10^{-3}$, we show in Fig. 1.2 on the bottom row the contraction factor $\mathcal{K}$ at each eigenvalue of the matrix A in (1.3b). We see that for all three combinations of $\mathcal{G}$ and $\mathcal{F}$, the convergence factor ρ is larger than 1.

Over the past years, there were many efforts to adapt the parareal algorithm to wave propagation problems; see, e.g., [3,5,6,8,17,18,35]. However, as pointed out by Ruprecht in [36], these existing approaches are "*typically with significant overhead, leading to further degradation of efficiency, or limited applicability*". The main problem in classical parareal is the different phase speeds due to the different grids used by $\mathcal{F}$ and $\mathcal{G}$, and that in the hyperbolic limit, small scale features propagate unchanged over long distance, which a coarse propagator $\mathcal{G}$ cannot do. For such problems, the numerical solutions generated by the propagators $\mathcal{F}$ and $\mathcal{G}$ should thus be as close as possible. A simple idea is to use the same IRK method and discretization step $\delta t = \Delta t$ for both $\mathcal{F}$ and $\mathcal{G}$; then we see from (1.2) that the parareal algorithm converges after 1 iteration, since in this case it holds exactly $\mathcal{F}_{\delta t}(T_n, u_n^k) = \mathcal{G}_{\Delta t}(T_n, u_n^k)$. Unfortunately, there is no speedup then because the implementation of the propagator $\mathcal{G}$ is as expensive as $\mathcal{F}$.

To make this simple idea practical, we need to find a different way to approximate $\mathcal{G}$ that uses the same method and grid as $\mathcal{F}$ to reduce its computational cost. One could parallelize $\mathcal{G}$ in time using diagonalization [12], originally proposed in [29], but this leads to bad convergence of the parareal algorithm, as our numerical results in Section 2.1 show. The new idea we propose here is to solve with $\mathcal{G}$ instead of the original problem (1.1) on each large subinterval $[T_n, T_{n+1}]$ a problem which lends itself to good time parallel solution by diagonalization, namely the *head-tail* coupled problem

$$\dot{u} + f(t, u) = 0, \quad u(T_n) = \alpha u(T_{n+1}) + (1-\alpha)u_n^k, \quad t \in (T_n, T_{n+1}), \tag{1.5}$$

where $\alpha \in (0,1)$ is a parameter. We note that for $\alpha \in (0,1)$ the existence and uniqueness of the solution are different from both the initial-value case ($\alpha = 0$) and the time-periodic case ($\alpha = 1$) and these issues merit further study, which is however beyond the scope of the present paper. The reason for using such a head-tail condition is twofold. First, with a small α this condition is close to $u(T_n) = u_n^k$ and thus as we explained above the new parareal algorithm will converge rapidly. Second, as we will see in Section 2 the head-tail coupled condition results in an α-circulant matrix of the time discretization and this allows us to solve (1.5) efficiently via the diagonalization technique. The new parareal algorithm studied here is

$$u_{n+1}^{k+1} = \mathcal{F}^*_{\delta t}(\alpha, T_n, u_n^{k+1}) + \mathcal{F}_{\delta t}(T_n, u_n^k) - \mathcal{F}^*_{\delta t}(\alpha, T_n, u_n^k), \ n = 0, 1, \ldots, N_t - 1, \tag{1.6a}$$

where the quantity $\mathcal{F}^*_{\delta t}(\alpha, T_n, u_n^k)$ denotes the solution of (1.5) at $t = T_{n+1}$. For a concrete example we consider the linear θ-method, by which this quantity is specified as:

$$\begin{cases} z_0 = \alpha z_J + (1-\alpha)u_n^k, \\ \frac{z_{j+1}-z_j}{\delta t} + \theta f(t_{n,j+1}, z_{j+1}) + (1-\theta) f(t_{n,j}, z_j) = 0, \\ j = 0, 1, \ldots, J-1, \\ \mathcal{F}^*_{\delta t}(\alpha, T_n, u_n^k) := z_J, \end{cases} \tag{1.6b}$$

where $t_{n,j} = T_n + j\delta t$. The key point here is that we use the same time-integrator (denoted by $\mathcal{F}$) with a small step-size δt to define both $\mathcal{F}_{\delta t}(\cdot)$ and $\mathcal{F}^*_{\delta t}(\cdot)$, but for the former the time-integrator is applied to (1.1) and for the latter the time-integrator is applied to a new model (1.5); see Figure 1.3 for an illustration of the idea.

We will show that when applying the diagonalization technique to (1.5), the roundoff error can be bounded by $\mathcal{O}(2\varepsilon J/\alpha)$, where ε is the machine precision, which is often negligible compared to the discretization error. (A precise definition of the roundoff error for the diagonalization technique can be found in [12,21].) We will also prove in the

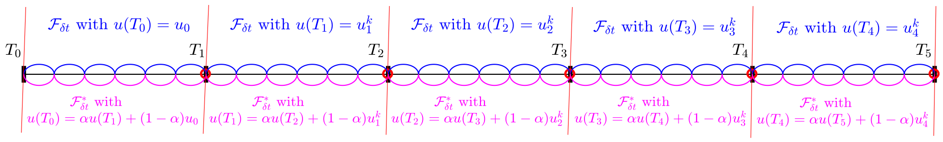

FIG. 1.3. *For the new parareal algorithm* (1.6a)*, both $\mathcal{F}_{\delta t}$ and $\mathcal{F}^*_{\delta t}$ are defined by applying $\mathcal{F}$ with the small step-size δt to the governing ODEs, while for the latter the ODEs are equipped with a 'head-tail' coupled condition $u(T_n) = \alpha u(T_{n+1}) + (1-\alpha)u_n^k$ in each large time-interval $[T_n, T_{n+1}]$.*

dissipative case that the convergence factor of our new parareal algorithm can be bounded as $\rho \le \alpha$. If the system of ODEs has wave propagation characteristic, the convergence factor can be bounded as $\rho \le \frac{2N_t\alpha}{1+\alpha}$. We will also give a convergence analysis of the new parareal algorithm for nonlinear problems, under the assumption that the nonlinear function $f(t,u)$ in (1.1) satisfies suitable Lipschitz conditions with respect to the second variable u.

At the end of this section, we mention that using a parameterized head-tail coupled condition together with the diagonalization technique to design PinT algorithms already appears in [21,40,42]. However, the mechanism and application scope of these algorithms are essentially different from our new parareal algorithm studied here. In particular, these algorithms are not suitable for solving nonlinear problems on long time intervals. Besides this, the algorithms in [40,42] are not suitable for wave propagation problems. As we will show in this paper, the new parareal algorithm however does not have these disadvantages. Recently, similar diagonalization technique also appears in [30,32], which in essence corresponds to a special case of our parameterized head-tail coupled condition, i.e., $\alpha = 1$. The algorithms in [30,32] are also not suitable for wave propagation problems.

2. The diagonalization-based coarse propagator. We now explain how the parallelization of the coarse propagator by diagonalization works for both the original problem and our new one with the *head-tail* condition.

For the original problem with initial value condition, the diagonalization in time relies on the strategy studied in [12,29], for which the main point is to divide each subinterval by a geometric time stepping mesh $\{\delta t_j := \delta t_1\tau^{j-1}\}_{j=1}^J$, where $\tau > 1$ is a constant and δ_1 satisfies $\delta t_1 = \frac{\Delta T(\tau-1)}{\tau^J-1}$ (such that $\sum_{j=1}^J \delta t_j = \Delta T$). For the linear system of ODEs

$$\dot{u} + Au = f(t),\ u(0) = u_0,\ t \in (0,T),\ A \in \mathbb{R}^{m\times m}, \tag{2.1}$$

if we use the Backward-Euler method as the time-integrator this approach leads to the following computation of $\mathcal{F}^*_{\delta t}(\alpha, T_n, u_n^k)$:

$$\left(\underbrace{\begin{bmatrix} \frac{1}{\delta t_1} & & & \\ -\frac{1}{\delta t_2} & \frac{1}{\delta t_2} & & \\ & \ddots & \ddots & \\ & & -\frac{1}{\delta t_J} & \frac{1}{\delta t_J} \end{bmatrix}}_{:=B} \otimes I_x + I_t \otimes A\right) \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_J \end{bmatrix} = \begin{bmatrix} f_{n+\frac{1}{J}} + \frac{u_n^k}{\delta t_1} \\ f_{n+\frac{2}{J}} \\ \vdots \\ f_{n+1} \end{bmatrix}, \tag{2.2}$$

where $z_J = \mathcal{F}^*_{\delta t}(\alpha, T_n, u_n^k)$ is the desired quantity, $f_{n+\frac{j}{J}} = f(t_{n,j})$ and $I_t \in \mathbb{R}^{J\times J}, I_x \in \mathbb{R}^{m\times m}$ are identity matrices. Problems exist for this approach in that the roundoff error arising from diagonalizing the time discretization matrix B is large and increases rapidly as J increases. In particular, from [12] we know that the roundoff error satisfies

$$\text{roundoff error} = \mathcal{O}(2\varepsilon J \text{Cond}_2(V)), \quad \varepsilon \text{ the machine precision.} \tag{2.3}$$

Unfortunately, using the geometric time stepping mesh the condition number of V increases rapidly as J increases and this seriously deteriorates the convergence rate of the parareal algorithm (see numerical illustrations in [40, Section 5]).

We next explain the details for computing $\mathcal{F}^*_{\delta t}(\alpha, T_n, u_n^k)$ using the head-tail coupled condition. We consider the linear and nonlinear cases separately. Here, for simplicity we continue to consider the linear θ-method as in (1.6b) and the case of the general IRK methods can be found in the Appendix.

2.1. The linear case. For the linear ODEs (2.1), the computation of $\mathcal{F}^*_{\delta t}(\alpha, T_n, u_n^k)$ specified by (1.6b) using the θ-method can be represented as

$$\left(\frac{1}{\delta t}\underbrace{\begin{bmatrix} 1 & & & -\alpha \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}}_{:=C_\alpha} \otimes I_x + \underbrace{\begin{bmatrix} \theta & & & \alpha(1-\theta) \\ 1-\theta & \theta & & \\ & \ddots & \ddots & \\ & & 1-\theta & \theta \end{bmatrix}}_{:=\widetilde{C}_{\alpha,\theta}} \otimes A\right)\underbrace{\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_J \end{bmatrix}}_{:=Z} = b_n^k, \tag{2.4}$$

$$b_n^k = \left(g_{n,1}^\top + (1-\alpha)\left(\frac{I_x}{\delta t} + (1-\theta)A\right)(u_n^k)^\top, g_{n,2}^\top, \dots, g_{n,J}^\top\right)^\top,$$

i.e., $(C_\alpha \otimes I_x + \delta t \widetilde{C}_{\alpha,\theta} \otimes A)Z = \delta t b_n^k$ and $\mathcal{F}^*_{\delta t}(\alpha, T_n, u_n^k) := z_J$. Both C_α and $\widetilde{C}_{\alpha,\theta}$ are α-circulant matrices and they can be simultaneously diagonalized as follows.

Lemma 2.1. *The matrices C_α and $\widetilde{C}_{\alpha,\theta} \in \mathbb{R}^{J\times J}$ can be diagonalized as*

$$C_\alpha = V(\alpha)D(\alpha)V^{-1}(\alpha), \ \ \widetilde{C}_{\alpha,\theta} = V(\alpha)\widetilde{D}(\alpha,\theta)V^{-1}(\alpha), \tag{2.5a}$$

where $V(\alpha) = \Lambda(\alpha)F$ and

$$\begin{aligned} &\Lambda(\alpha) = \operatorname{diag}\left(1, \alpha^{-\frac{1}{J}}, \dots, \alpha^{-\frac{J-1}{J}}\right), \\ &F = [\omega_1, \omega_2, \dots, \omega_J], \textit{with } \omega_j = \left[1, e^{i\frac{2(j-1)\pi}{J}}, \dots, e^{i\frac{2(j-1)(J-1)\pi}{J}}\right]^\top, \\ &D(\alpha) = \operatorname{diag}(\lambda_1, \dots, \lambda_J), \text{with } \lambda_j = 1 - \alpha^{\frac{1}{J}} e^{-i\frac{2(j-1)\pi}{J}}, \\ &\widetilde{D}(\alpha,\theta) = \operatorname{diag}(\tilde{\lambda}_1, \dots, \tilde{\lambda}_J), \text{with } \tilde{\lambda}_j = \theta + (1-\theta)\alpha^{\frac{1}{J}} e^{-i\frac{2(j-1)\pi}{J}}. \end{aligned} \tag{2.5b}$$

Proof. The spectral decomposition of an α-circulant matrix is well known, the details can be found for example in [1, Theorem 2.10]. ◻

Substituting (2.5a) into (2.4) gives the following diagonalization procedure for computing $\mathcal{F}^*_{\delta t}(\alpha, T_n, u_n^k)$:

$$\begin{aligned} &\text{Step-(a)}: \quad && ((\Lambda(\alpha)F)^{-1} \otimes I_x)P = \delta t b_n^k, \\ &\text{Step-(b)}: && \left(\lambda_j I_x + \tilde{\lambda}_j(\delta t A)\right) q_j = p_j, \ j = 1, 2, \dots, J, \\ &\text{Step-(c)}: && ((\Lambda(\alpha)F) \otimes I_x)Z = Q, \end{aligned} \tag{2.6}$$

where $P = (p_1^\top, \dots, p_J^\top)^\top, \ Q = (q_1^\top, \dots, q_J^\top)^\top \in \mathbb{C}^{mJ}$.

Remark 2.1 (FFT for Step-(a) and Step-(c) in (2.6)). *In (2.6), since F is a discrete Fourier matrix, Step-(a) and Step-(c) can be done by the Fast Fourier Transform (FFT) at the cost $\mathcal{O}(mJ\log_2 J)$*[1]*. The FFT can also be implemented in parallel*[2]*, see for example*

[1]There is a factor m here, because the vectors b_n^k and W consist of J subvectors and thus during the (inverse) FFT every element of F^{-1} acts on vectors (of length m) instead of scalar complex numbers.

[2]One can download the software of the parallel FFT from http://www.fftw.org/.

[24]. The cost for computing the (inverse) FFT can then be reduced to $\mathcal{O}(m\log_2 J)$. This cost is significantly smaller than that of solving the linear problems in Step-(b) of (2.6), when A is a large scale matrix.

Remark 2.2. *It is natural to ask how to solve each of the J algebraic equations in (2.6). We rewrite the equation as*

$$(\eta_j I_x+\delta t A)q_j=\tilde{p}_j,\ \tilde{p}_j:=p_j/\tilde{\lambda}_j,\ \eta_j:=\lambda_j/\tilde{\lambda}_j. \tag{2.7}$$

For $\theta=1$ and $\theta=\frac{1}{2}$, it holds that $\Re(\eta_j)\geq 0$ for all $j=1,2,\ldots,J$. Hence, if the real parts of the eigenvalues of the matrix A are non-negative[3]*, the complex shift λ_j does not lead to too much difficulty. In particular, as shown in [41], a multigrid method using Richardson iterations with optimized damping parameter as the smoother works very well in this case. For the case that A is an indefinite matrix, we refer the interested reader to [10] for a thorough study of preconditioned GMRES. Note that for $\theta=0$ (i.e., the Forward Euler method), the algorithm proposed here could still be useful, since then instead of solving a system at each time step, a matrix multiplication is needed at each time step, and with the present algorithm, all these matrix multiplications could be executed in parallel.*

Since $\alpha\in(0,1)$ and F is a discrete Fourier matrix, from (2.5b) we see that

$$\mathrm{Cond}_2(V(\alpha))\leq \mathrm{Cond}_2(\Lambda(\alpha))\mathrm{Cond}_2(F)\leq\alpha^{-1}. \tag{2.8}$$

Now, by substituting (2.8) into (2.3) we know that the roundoff error arising from the first and last steps of (2.6) satisfies

$$\text{roundoff error}=\mathcal{O}(2\varepsilon J/\alpha). \tag{2.9}$$

2.2. The nonlinear case. To clearly describe the details of the diagonalization technique for nonlinear $\mathcal{F}^*_{\delta t}(\alpha,T_n,u_n^k)$, we assume again that $\mathcal{F}$ is the linear θ-method; for s-stage IRK methods see the Appendix. From (1.6b), we have

$$\left(\frac{1}{\delta t}C_\alpha\otimes I_x\right)Z+\underbrace{\begin{bmatrix}\theta f(t_{n,1},z_1)+(1-\theta)f(t_{n,0},\alpha z_J+(1-\alpha)u_n^k)\\ \theta f(t_{n,2},z_2)+(1-\theta)f(t_{n,1},z_1)\\ \vdots\\ \theta f(t_{n,J},z_J)+f(t_{n,J-1},z_{J-1})\end{bmatrix}}_{:=\mathbf{f}(Z)}=b_n^k, \tag{2.10}$$

$$b_n^k=\left(\frac{(1-\alpha)(u_n^k)^\top}{\delta t},0,\ldots,0\right)^\top,\ Z=(z_1^\top,\ldots,z_J^\top)^\top,$$

where the matrix C_α is given by (2.4). Applying Newton's method to (2.10) gives

$$Z^{(l+1)}=Z^{(l)}-\left(\frac{1}{\delta t}C_\alpha\otimes I_x+\mathbf{J}(Z^{(l)})\right)^{-1}\left(\left(\frac{1}{\delta t}C_\alpha\otimes I_x\right)Z^{(l)}+\mathbf{f}(Z^{(l)})-b_n^k\right),$$

i.e.,

$$\left(\frac{1}{\delta t}C_\alpha\otimes I_x+\mathbf{J}(Z^{(l)})\right)Z^{(l+1)}=-\mathbf{f}(Z^{(l)})+b_n^k+\mathbf{J}(Z^{(l)})Z^{(l)}, \tag{2.11}$$

where $Z^{(l)}=\left((z_1^{(l)})^\top,(z_2^{(l)})^\top,\ldots,(z_J^{(l)})^\top\right)^\top$ and the Jacobian matrix $\mathbf{J}(Z^{(l)})$ of $\mathbf{f}$ is

$$\mathbf{J}(Z^{(l)})=\begin{bmatrix}\nabla f(t_{n,j+1},z_1^{(l)}) & & & \alpha\nabla f(t_{n,0},\alpha z_J^{(l)}+(1-\alpha)u_n^k)\\ & \nabla f(t_{n,j+1},z_2^{(l)}) & & \\ & & \ddots & \\ & & & \nabla f(t_{n,j+1},z_J^{(l)})\end{bmatrix}.$$

[3] This holds when A is an approximation of en elliptic, self adjoint operator, i.e., $A\approx\sum_{j,l=1}^d\frac{\partial}{\partial x_l}\left(a_{jl}(x)\frac{\partial}{\partial x_j}\right)$.

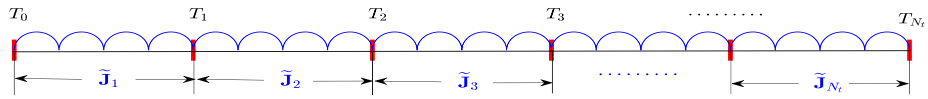

FIG. 2.1. *There are N_t large subintervals for the new parareal algorithm (1.6a) and we use N_t different Jacobian matrices for these subintervals. The n-th subinterval $[T_n, T_{n+1}]$ contains J fine time points and the Jacobian matrix $\widetilde{\mathbf{J}}_n$ is used for all these fine time points.*

To solve the linear system (2.11) by the diagonalization technique, following the idea in [13] we replace the Jacobian matrix $\mathbf{J}(Z^{(l)})$ by $I_J \otimes \widetilde{\mathbf{J}}(Z^{(l)})$, where

$$\widetilde{\mathbf{J}}(Z^{(l)}) := \frac{1}{J}\left(\sum_{j=1}^{J-1} \nabla f(t_{n,j}, z_j^{(l)}) + \nabla f\left(t_{n,0}, \alpha z_J + (1-\alpha)u_n^k\right)\right). \tag{2.12}$$

Substituting (2.12) into (2.11) leads to the simplified Newton iteration

$$\left(\frac{1}{\delta t} C_\alpha \otimes I_x + I_J \otimes \widetilde{\mathbf{J}}(Z^{(l)})\right) Z^{(l+1)} = -\mathbf{f}(Z^{(l)}) + b_n^k + (I_J \otimes \widetilde{\mathbf{J}}(Z^{(l)}))Z^{(l)}. \tag{2.13}$$

With the spectral decomposition of C_α (see Lemma 2.1), we can solve $Z^{(l+1)}$ via the diagonalization technique (cf. (2.6)).

REMARK 2.3. *For nonlinear ODEs, the main idea of applying the new parareal algorithm (1.6a) lies in approximating the J Jacobian matrices (on the fine time points $\{t_{n,j}\}_{j=1}^J$) by a single matrix $\widetilde{\mathbf{J}}$. Since we have N_t large subintervals, there are N_t different approximate Jacobian matrices in total; see Fig. 2.1 for illustration.*

3. Convergence analysis in the linear case. We now analyze the convergence properties of the new parareal algorithm (1.6a). In this section we consider the linear case (2.1); for the nonlinear case see the next section.

3.1. General result. We first consider the linear ODEs (2.1). The following lemma is useful to analyze the convergence factor of the new parareal algorithm (1.6a).

LEMMA 3.1. *Let $\sigma \in [0,1]$. Then, for the matrix*

$$\mathcal{M}(\sigma) = \begin{bmatrix} 1 & & & \\ -\sigma & 1 & & \\ & \ddots & \ddots & \\ & & -\sigma & 1 \end{bmatrix} \in \mathbb{C}^{N_t \times N_t}, \tag{3.1}$$

the inverse $\mathcal{M}^{-1}(\sigma)$ is a non-negative matrix[4] *and $\|\mathcal{M}^{-1}(\sigma)\|_\infty = \frac{1-\sigma^{N_t}}{1-\sigma}$.*

Note. When $\sigma = 1$, the quantity $\frac{1-\sigma^{N_t}}{1-\sigma}$ should be understood in the 'limiting' sense, i.e., $\lim_{\sigma\to 1} \frac{1-\sigma^{N_t}}{1-\sigma} = N_t$.

Proof. The proof is as the proof of [19, Lemma 4.4], so we omit the details. □

THEOREM 3.2. *For the linear ODEs (2.1), assume that A is a diagonalizable matrix with $A = S\mathrm{diag}(\mu_1, \mu_2, \ldots, \mu_m)S^{-1}$ and $\sigma(A) \subseteq \mathbb{C}^+$. Let $\mathcal{F}$ be an A-stable IRK method with stability function $\mathcal{R}(z)$ and $\{u_n\}_{n=1}^{N_t}$ be the solutions generated by $\mathcal{F}$ at the coarse time points $\{T_n\}_{n=1}^{N_t}$, i.e.,*

$$u_{n+1} = \mathcal{F}_{\delta t}(T_n, u_n),\ n = 0, 1, \ldots, N_t - 1. \tag{3.2}$$

[4] A non-negative matrix is a matrix where every element is non-negative.

Let $\{u_n^k\}_{n=1}^{N_t}$ be the k-th iterate generated by the parareal algorithm (1.6a) with parameter $\alpha\in(0,1)$. Then, the error $e_n^k:=u_n-u_n^k$ satisfies

$$\max_{N_t\geq n\geq 1}\|Se_n^k\|_\infty\leq \rho^k(\alpha,J,N_t)\max_{N_t\geq n\geq 1}\|Se_n^0\|_\infty,\ \forall k\geq 1, \tag{3.3a}$$

where $\rho(\alpha,J,N_t)=\max_{z\in\sigma(\Delta TA)}\mathcal{K}(\alpha,J,N_t,z)$ and

$$\begin{aligned}\mathcal{K}(\alpha,J,N_t,z)&=\frac{1-|\mathcal{R}_g(\alpha,J,z)|^{N_t}}{1-|\mathcal{R}_g(\alpha,J,z)|}\left|\mathcal{R}^J\left(\frac{z}{J}\right)-\mathcal{R}_g(\alpha,J,z)\right|,\\ \mathcal{R}_g(\alpha,J,z)&:=\frac{(1-\alpha)\mathcal{R}^J(\frac{z}{J})}{1-\alpha\mathcal{R}^J(\frac{z}{J})}.\end{aligned} \tag{3.3b}$$

Proof. Subtracting (1.6a) from (3.2) gives

$$\begin{aligned}u_{n+1}-u_{n+1}^{k+1}=&\left[\mathcal{F}_{\delta t}^*(\alpha,T_n,u_n)-\mathcal{F}_{\delta t}^*(\alpha,T_n,u_n^{k+1})\right]+\left[\mathcal{F}_{\delta t}(T_n,u_n)-\mathcal{F}_{\delta t}(T_n,u_n^k)\right]-\\ &\left[\mathcal{F}_{\delta t}^*(\alpha,T_n,u_n)-\mathcal{F}_{\delta t}^*(\alpha,T_n,u_n^k)\right].\end{aligned} \tag{3.4}$$

In the following, without loss of generality we assume $g(t)=0$ for the linear ODEs (2.1). Then performing one-step of the IRK method can be expressed as

$$u_{n,j+1}=\mathcal{R}(\delta tA)u_{n,j}+\tilde{g}_{n,j}=\mathcal{R}\left(\Delta TA/J\right)u_{n,j}.$$

Hence,

$$\mathcal{F}_{\delta t}(T_n,u_n)-\mathcal{F}_{\delta t}(T_n,u_n^k)=\mathcal{R}^J\left(\Delta TA/J\right)e_n^k. \tag{3.5a}$$

Moreover, the quantity $\mathcal{F}_{\delta t}^*(\alpha,T_n,u_n^k)$ can be computed as

$$\begin{cases}z_0=\alpha z_J+(1-\alpha)u_n^k,\ z_{j+1}=\mathcal{R}\left(\Delta TA/J\right)z_j,\ j=0,1,\dots,J-1,\\ z_J=\mathcal{F}_{\delta t}^*(\alpha,T_n,u_n^k).\end{cases}$$

From this, we have

$$z_J=\mathcal{R}^J\left(\Delta TA/J\right)z_0=\alpha\mathcal{R}^J\left(\Delta TA/J\right)z_J+(1-\alpha)\mathcal{R}^J\left(\Delta TA/J\right)u_n^k.$$

Therefore, $\mathcal{F}_{\delta t}^*(\alpha,T_n,u_n^k)=(1-\alpha)\left[I_x-\alpha\mathcal{R}^J\left(\Delta TA/J\right)\right]^{-1}\mathcal{R}^J\left(\Delta TA/J\right)u_n^k$. This implies

$$\mathcal{F}_{\delta t}^*(\alpha,T_n,u_n)-\mathcal{F}_{\delta t}^*(\alpha,T_n,u_n^k)=\mathcal{R}_g(\alpha,J,\Delta TA)e_n^k, \tag{3.5b}$$

where $\mathcal{R}_g(\alpha,J,\Delta TA)=(1-\alpha)\left[I_x-\alpha\mathcal{R}^J\left(\frac{\Delta TA}{J}\right)\right]^{-1}\mathcal{R}^J\left(\frac{\Delta TA}{J}\right)$.

Now, substituting (3.5a) and (3.5b) into (3.4) gives

$$e_{n+1}^{k+1}=\mathcal{R}_g(\alpha,J,\Delta TA)e_n^{k+1}+\mathcal{R}^J\left(\Delta TA/J\right)e_n^k-\mathcal{R}_g(\alpha,J,\Delta TA)e_n^k. \tag{3.6}$$

Since $A=S\boldsymbol{\mu}S^{-1}$ with $\boldsymbol{\mu}=\mathrm{diag}(\mu_1,\mu_2,\dots,\mu_m)$, from (3.6) we have

$$Se_{n+1}^{k+1}=S\mathcal{R}_g(\alpha,J,\Delta T\boldsymbol{\mu})S^{-1}(Se_n^{k+1})+S\mathcal{R}^J\left(\Delta T\boldsymbol{\mu}/J\right)S^{-1}(Se_n^k)-S\mathcal{R}_g(\alpha,J,\Delta T\boldsymbol{\mu})S^{-1}(Se_n^k).$$

Let $z=\Delta T\mu$ with $\mu\in\boldsymbol{\mu}$ being an arbitrary eigenvalue of A and $\zeta_n^k(\mu)$ be the corresponding component of Se_n^k. Then, it holds that

$$\begin{aligned}&\zeta_{n+1}^{k+1}(\mu)=\mathcal{R}_g(\alpha,J,z)\zeta_n^{k+1}(\mu)+\mathcal{R}^J\left(z/J\right)\zeta_n^k(\mu)-\mathcal{R}_g(\alpha,J,z)\zeta_n^k(\mu)\\ &\Longrightarrow|\zeta_{n+1}^{k+1}(\mu)|\leq|\mathcal{R}_g(\alpha,J,z)||\zeta_n^{k+1}(\mu)|+\left|\mathcal{R}^J\left(z/J\right)-\mathcal{R}_g(\alpha,J,z)\right||\zeta_n^k(\mu)|,\end{aligned} \tag{3.7}$$

where $n=0,1,\dots,N_t-1$ and $\zeta_0^k(\mu)=0$. From (3.7) we have

$$\mathcal{M}(\sigma)\underbrace{\begin{bmatrix}\zeta_1^{k+1}(\mu)\\ \zeta_2^{k+1}(\mu)\\ \vdots\\ \zeta_{N_t}^{k+1}(\mu)\end{bmatrix}}_{:=\boldsymbol{\zeta}^{k+1}(\mu)}\leq\underbrace{\begin{bmatrix}0&&&\\ \tilde{\sigma}&0&&\\ &\ddots&\ddots&\\ &&\tilde{\sigma}&0\end{bmatrix}}_{:=\mathcal{N}(\tilde{\sigma})}\underbrace{\begin{bmatrix}\zeta_1^{k}(\mu)\\ \zeta_2^{k}(\mu)\\ \vdots\\ \zeta_{N_t}^{k}(\mu)\end{bmatrix}}_{:=\boldsymbol{\zeta}^{k}(\mu)}, \tag{3.8a}$$

where $\mathcal{M}(\sigma)$ is the Toeplitz matrix given by (3.1) and

$$\sigma = |\mathcal{R}_g(\alpha, J, z)|, \ \tilde{\sigma} = \left|\mathcal{R}^J\left(\frac{z}{J}\right) - \mathcal{R}_g(\alpha, J, z)\right|. \tag{3.8b}$$

Since $\sigma(A) \subseteq \mathbb{C}^+ (\Rightarrow z \in \mathbb{C}^+)$ and $\mathcal{F}$ is an A-stable RK method, it holds that $|\mathcal{R}^J(z/J)| < 1, \forall z \in \sigma(\Delta T A)$. Hence, for $\alpha \in (0,1)$ from (3.3b) we have

$$\sigma = \frac{(1-\alpha)|\mathcal{R}^J(\frac{z}{J})|}{|1-\alpha\mathcal{R}^J(\frac{z}{J})|} \leq \frac{(1-\alpha)|\mathcal{R}^J(z/J)|}{1-|\alpha\mathcal{R}^J(z/J)|} < \frac{1-\alpha}{1-|\alpha|} = 1.$$

Hence, by using Lemma 3.1 we know that $\mathcal{M}^{-1}(\sigma)$ is a non-negative matrix and thus from (3.8a) it holds that

$$\boldsymbol{\zeta}^{k+1}(\mu) \leq \mathcal{M}^{-1}(\sigma)\mathcal{N}(\tilde{\sigma})\boldsymbol{\zeta}^k(\mu) \Longrightarrow \|\boldsymbol{\zeta}^{k+1}(\mu)\|_\infty \leq \|\mathcal{M}^{-1}(\sigma)\|_\infty \|\mathcal{N}(\tilde{\sigma})\|_\infty \|\boldsymbol{\zeta}^k(\mu)\|_\infty.$$

Since $\|\mathcal{N}(\tilde{\sigma})\|_\infty = \tilde{\sigma}$ and $\|\mathcal{M}^{-1}(\sigma)\|_\infty = \frac{1-\sigma^{N_t}}{1-\sigma}$ (see Lemma 3.1), we get

$$\|\boldsymbol{\zeta}^{k+1}(\mu)\|_\infty \leq \frac{(1-\sigma^{N_t})\tilde{\sigma}}{1-\sigma}\|\boldsymbol{\zeta}^k(\mu)\|_\infty. \tag{3.9}$$

Because $\boldsymbol{\zeta}^k(\mu) = (\zeta_1^k(\mu), \zeta_2^k(\mu), \ldots, \zeta_{N_t}^k(\mu))^\top$ and $\zeta_n^k(\mu)$ is a component of the error Se_n^k corresponding the eigenvalue μ of A, we have

$$\max_{\mu\in\sigma(A)} \|\boldsymbol{\zeta}^k(\mu)\|_\infty = \max_{\mu\in\sigma(A)} \max_{N_t\geq n\geq 1} |\zeta_n^k(\mu)| = \max_{N_t\geq n\geq 1}\left(\max_{\mu\in\sigma(A)} |\zeta_n^k(\mu)|\right) = \max_{N_t\geq n\geq 1} \|Se_n^k\|_\infty.$$

By substituting this into (3.9) and by using (3.8b), we get (3.3a)-(3.3b). ∎

By using Theorem 3.2, for a given time-integrator $\mathcal{F}$, the ratio $J = \Delta T/\delta t$ and the spectrum $\sigma(A)$, it is easy to calculate the convergence factor ρ. It is however important to analyze ρ qualitatively to see how it depends on these quantities. For an arbitrary coefficient matrix A, we do not know how to make such a qualitative analysis. In the following, we consider two representative cases: the heat equation and the wave equation. For the former, the matrix A is an SPD matrix and therefore the variable $z \in \sigma(\Delta T A)$ satisfies $z \in [0, \infty)$. For the latter, it holds that $\sigma(A) \subseteq i\mathbb{R}$ and thus $z \in i\mathbb{R}$. It is well known that the heat equation represents a class of 'easy' problems for parareal and the wave equation represents a class of 'hard' problems; see [36].

3.2. Qualitative analysis: the heat equation. When semi-discretizing the heat equation, the matrix A in (2.1) is an SPD matrix. Since the eigenvalues of A can be arbitrarily large, we consider in the following $z \in [0, \infty)$ for the contraction factor $\mathcal{K}(\alpha, J, z)$ (cf.(3.3b)). In the proof of Theorem 3.2, we already showed that $|\mathcal{R}_g(\alpha, J, z)| \leq 1$ when $\mathcal{F}$ is an A-stable time-integrator and $z \in \mathbb{C}^+$. Hence, we have

$$\mathcal{K}(\alpha, J, N_t, z) \leq \widetilde{\mathcal{K}}(\alpha, J, z) := \frac{\left|\mathcal{R}^J\left(\frac{z}{J}\right) - \mathcal{R}_g(\alpha, J, z)\right|}{1 - |\mathcal{R}_g(\alpha, J, z)|}. \tag{3.10}$$

In the following, we try to find a sharp upper bound for $\widetilde{\mathcal{K}}(\alpha, J, z)$.

Theorem 3.3. *Let $\mathcal{F}$ be an A-stable IRK method with stability function $\mathcal{R}(z)$ and $J \geq 2$ be an even integer. For the coefficient matrix A of the linear ODE system (2.1), suppose A is diagonalizable and $\sigma(A) \subseteq [0, \infty)$. Then, for $\alpha \in (0,1)$ the convergence factor ρ of the parareal algorithm (1.6a) satisfies*

$$\rho(\alpha, J, N_t) \leq \alpha, \ \forall J \geq 2, \ \forall N_t \geq 2.$$

Note. Using an even $J(= \Delta T/\delta t)$ is not really a restriction for practical computation.

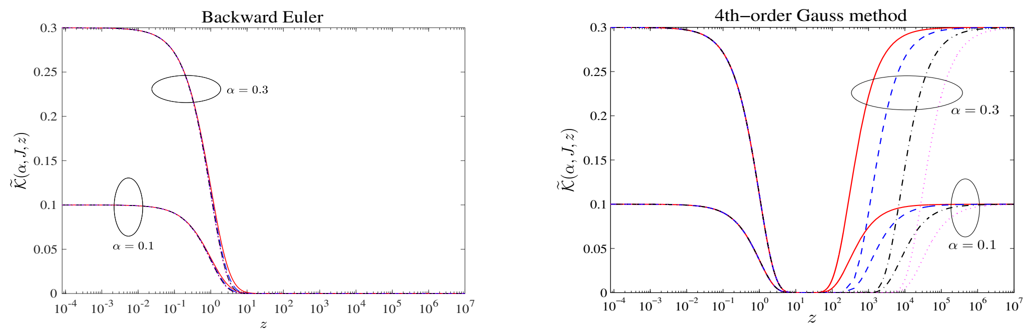

FIG. 3.1. *For two values of α and four values of J, the quantity $\widetilde{\mathcal{K}}(\alpha, J, z)$ as a function of z. Left: $\mathcal{F}$=Backward Euler; Right: $\mathcal{F}$=4th-order Gauss method. For each subfigure, '———' for $J = 5$, '- - - -' for $J = 10$, '$- \cdot - \cdot -$' for $J = 25$ and '$\cdots\cdots$' for $J = 50$.*

Proof. From Theorem 3.2 and (3.10), it is sufficient to prove that

$$\max_{z \geq 0} \widetilde{\mathcal{K}}(\alpha, J, z) \leq \alpha, \ \forall J \geq 2. \tag{3.11}$$

Since $\mathcal{R}_g(\alpha, J, z) = \frac{(1-\alpha)\mathcal{R}^J(\frac{z}{J})}{1-\alpha\mathcal{R}^J(\frac{z}{J})}$ (cf.(3.3b)), we have $\mathcal{R}^J\left(\frac{z}{J}\right) = \frac{\mathcal{R}_g(\alpha,J,z)}{1-\alpha(1-\mathcal{R}_g(\alpha,J,z))}$. Moreover, since $\mathcal{F}$ is an A-stable IRK method and $z \in [0, \infty)$ and $J \geq 2$ is an even integer, it holds that $\mathcal{R}^J(\frac{z}{J}) \in [0, 1]$. Hence, $\mathcal{R}_g(\alpha, J, z) \in (0, 1)$. Let

$$\phi = \mathcal{R}_g(\alpha, J, z) \in (0, 1).$$

Then, it holds that $\mathcal{R}^J\left(\frac{z}{J}\right) = \frac{\phi}{1-\alpha(1-\phi)}$ and that

$$\widetilde{\mathcal{K}}(\alpha, J, z) = \frac{\left|\frac{\phi}{1-\alpha(1-\phi)} - \phi\right|}{1 - |\phi|} = \frac{\frac{\phi}{1-\alpha(1-\phi)} - \phi}{1 - \phi} = \frac{\alpha\phi}{1 - \alpha + \alpha\phi}.$$

For $\alpha \in (0, 1)$, it holds by direct calculation that $\max\limits_{\phi\in(0,1)} \frac{\alpha\phi}{1-\alpha+\alpha\phi} = \left.\frac{\alpha\phi}{1-\alpha+\alpha\phi}\right|_{\phi=1} = \alpha$. ∎

In Fig. 3.1, we show the quantity $\widetilde{\mathcal{K}}(\alpha, J, z)$ as a function of z, where $\alpha = 0.3$ and $\alpha = 0.1$. We consider two choices of $\mathcal{F}$: Backward Euler and the 4th-order Gauss method. Their stability functions are

$$\mathcal{R}(z) = \begin{cases} \frac{1}{1+z}, & \text{Backward Euler}, \\ \frac{1-\frac{z}{2}+\frac{z^2}{12}}{1+\frac{z}{2}+\frac{z^2}{12}}, & \text{4th-order Gauss method}. \end{cases}$$

For these two time-integrators, it is clear that $\mathcal{R}(z) > 0$ for all $z \in [0, \infty)$ and from Fig. 3.1 we see that the ratio J does not affect the upper bound of $\widetilde{\mathcal{K}}$.

However when $\mathcal{F}$ is the Trapezoidal Rule, the ratio J indeed has a significant influence on $\widetilde{\mathcal{K}}$: if J is even, $\widetilde{\mathcal{K}}$ approaches α as $z \to 0$ or $z \to \infty$, while if J is odd, $\widetilde{\mathcal{K}}$ approaches 1 as $z \to \infty$, see Fig. 3.2. We now explain this: because $\mathcal{R}(z) = \frac{1-\frac{z}{2}}{1+\frac{z}{2}}$, we have

$$\lim_{z\to\infty} \mathcal{R}_g(\alpha, J, z) = \lim_{z\to\infty} \frac{(1-\alpha)\mathcal{R}^J(\frac{z}{J})}{1 - \alpha\mathcal{R}^J(\frac{z}{J})} = \begin{cases} 1, & \text{if } J \text{ is even}, \\ \frac{\alpha-1}{\alpha+1}, & \text{if } J \text{ is odd}. \end{cases}$$

This gives

$$\lim_{z\to\infty} \widetilde{\mathcal{K}}(\alpha, J, z) = \lim_{z\to\infty} \frac{\left|\mathcal{R}^J\left(\frac{z}{J}\right) - \mathcal{R}_g(\alpha, J, z)\right|}{1 - |\mathcal{R}_g(\alpha, J, z)|} = \begin{cases} \alpha, & \text{if } J \text{ is even}, \\ \frac{|-1-\frac{\alpha-1}{\alpha+1}|}{1-\frac{1-\alpha}{\alpha+1}} = 1, & \text{if } J \text{ is odd}. \end{cases}$$

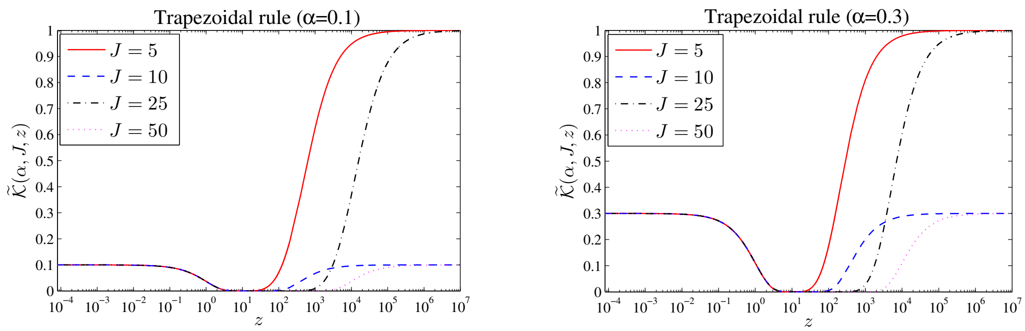

FIG. 3.2. *When $\mathcal{F}$ is the Trapezoidal Rule, the ratio $J = \Delta T/\delta t$ has an important effect on $\widetilde{\mathcal{K}}(\alpha, J, z)$. Left: $\alpha = 0.1$; Right: $\alpha = 0.3$.*

3.3. Quantitative analysis: wave equations. We next consider the case $z \in i\mathbb{R}$ in (3.3b). As shown in Fig. 1.2, the classical parareal algorithm does not converge in this case. For linear wave propagation equations or dispersive wave equations, it is reasonable to consider an *energy preserving* propagator $\mathcal{F}$, i.e., the stability function satisfies $|\mathcal{R}(z)| \equiv 1$ for $z \in i\mathbb{R}$.

THEOREM 3.4. *Assume that the stability function $\mathcal{R}(z)$ of $\mathcal{F}$ satisfies $|\mathcal{R}(z)| \equiv 1$ for $z \in i\mathbb{R}$ and that the coefficient matrix A of the linear ODE system (2.1) is diagonalizable and $\sigma(A) \subseteq i\mathbb{R}$. Then, the convergence factor ρ of the parareal algorithm (1.6a) satisfies*

$$\rho(\alpha, J, N_t) \le \frac{2\alpha}{1+\alpha} N_t, \ \forall J \ge 2. \tag{3.12}$$

Note. The Gauss RK methods, the Lobatto IIIA methods and the Lobatto IIIB methods $|\mathcal{R}(z)| \equiv 1$ ($\forall z \in i\mathbb{R}$); see [23].

Proof. Since $|\mathcal{R}(z)| = 1$ for $z \in i\mathbb{R}$, a routine calculation yields

$$\mathcal{K}(\alpha, J, N_t, z) = \frac{1 - |\mathcal{R}_g(\alpha, J, z)|^{N_t}}{1 - |\mathcal{R}_g(\alpha, J, z)|} \left|1 - \frac{1-\alpha}{1 - \alpha\mathcal{R}^J(\frac{z}{J})}\right| = \alpha \frac{1 - |\mathcal{R}_g(\alpha, J, z)|^{N_t}}{1 - |\mathcal{R}_g(\alpha, J, z)|} \left|\frac{1 - \mathcal{R}^J(\frac{z}{J})}{1 - \alpha\mathcal{R}^J(\frac{z}{J})}\right|.$$

Let $\sigma = |\mathcal{R}_g(\alpha, J, z)|$. Then, it holds that

$$\mathcal{K}(\alpha, J, N_t, z) = \alpha \frac{1 - \sigma^{N_t}}{1 - \sigma} \left|\frac{1 - \mathcal{R}^J(\frac{z}{J})}{1 - \alpha\mathcal{R}^J(\frac{z}{J})}\right|. \tag{3.13}$$

In the proof of Theorem 3.2 we already showed that $\sigma \in (0, 1)$, and hence

$$\max_{\sigma \in (0,1)} \frac{1 - \sigma^{N_t}}{1 - \sigma} = N_t. \tag{3.14}$$

We next analyze the second term $\left|\frac{1-\mathcal{R}^J(\frac{z}{J})}{1-\alpha\mathcal{R}^J(\frac{z}{J})}\right|$ in (3.13). Since $\left|\mathcal{R}^J(\frac{z}{J})\right| = \left|\mathcal{R}(\frac{z}{J})\right|^J \equiv 1$, we can represent $\mathcal{R}^J(\frac{z}{J})$ as $\mathcal{R}^J\left(\frac{z}{J}\right) = e^{i\tilde{z}}$ with some suitable $\tilde{z} \in \mathbb{R}$, and thus

$$\max_{z \in i\mathbb{R}} \left|\frac{1 - \mathcal{R}^J(\frac{z}{J})}{1 - \alpha\mathcal{R}^J(\frac{z}{J})}\right| \le \max_{\tilde{z} \in \mathbb{R}} \left|\frac{1 - e^{i\tilde{z}}}{1 - \alpha e^{i\tilde{z}}}\right|. \tag{3.15a}$$

We have $\left|\frac{1-e^{i\tilde{z}}}{1-\alpha e^{i\tilde{z}}}\right| = \frac{\sqrt{2-2\cos(\tilde{z})}}{\sqrt{1+\alpha^2-2\alpha\cos(\tilde{z})}}$ and therefore $\max_{\tilde{z}\in\mathbb{R}} \left|\frac{1-e^{i\tilde{z}}}{1-\alpha e^{i\tilde{z}}}\right| = \frac{2}{1+\alpha}$. Substituting this into (3.15a) gives

$$\max_{z \in i\mathbb{R}} \left|\frac{1 - \mathcal{R}^J(\frac{z}{J})}{1 - \alpha\mathcal{R}^J(\frac{z}{J})}\right| \le \frac{2}{1+\alpha}. \tag{3.15b}$$

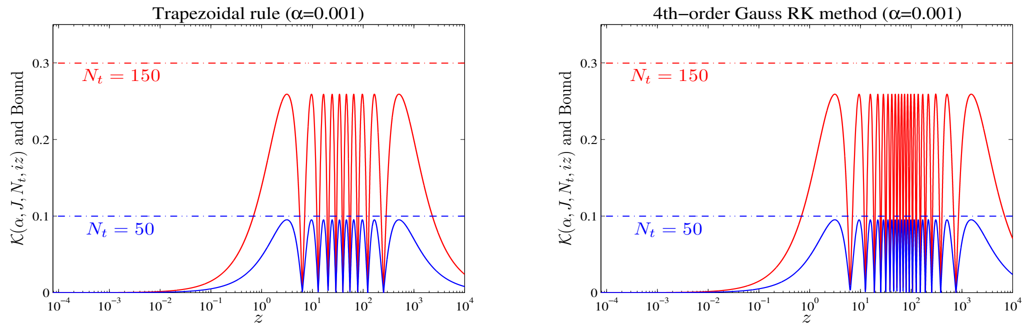

FIG. 3.3. *For the Trapezoidal rule and the 4th-order Gauss RK method, the contraction factor $\mathcal{K}$ as a function of z, when $\alpha = 10^{-3}$ is fixed. Here, $J = 20$; for other J the plots look similar.*

Now, substituting (3.15b) and (3.14) into (3.13) gives the desired result (3.12). $\square$

REMARK 3.1. *For wave equations, the estimate given by (3.12) implies that the number of coarse time points, i.e., $N_t = T/\Delta T$, has an effect on the convergence rate of the new parareal algorithm (1.6a). This is very different from the case of the heat equation, because for the latter Theorem 3.3 implies that the convergence rate does not depend on N_t. For wave propagation problems we can still obtain a good convergence rate by choosing a suitable parameter α. For example, for $N_t = 1000$ we can use $\alpha = 0.1/(2N_t) = 5 \times 10^{-5}$ to get a convergence factor $\rho \approx 0.1$.*

For the Trapezoidal rule and the 4th-order Gauss RK method, we show in Fig. 3.3 the contraction factor $\mathcal{K}(\alpha, J, N_t, iz)$ as a function of z, when $\alpha = 10^{-3}$ is fixed. We consider two values of N_t and for each N_t the bound $\frac{2\alpha}{1+\alpha}N_t$ is also shown by a dash-dotted line. We see that, in contrast to the heat equation, for wave equations the quantity N_t indeed has an effect on the convergence factor of the new parareal algorithm. Moreover, we see that the bound given by Theorem 3.4 is sharp.

The bound (3.12) holds for any IRK method satisfying $|\mathcal{R}(iz)| \equiv 1$ for $z \in \mathbb{R}$ and it is interesting to check whether this also holds for other IRK methods or not. To this end, we consider the 3-stage 4th-order SDIRK method (see [23]),

$$\begin{array}{r|lll} \tilde{r} & r & 0 & 0 \\ 1-2\tilde{r} & \frac{1}{2}-r & r & 0 \\ \tilde{r} & 2r & 1-4r & r \\ \hline & r & \frac{1}{2} & 1-r \end{array}, \quad r = \frac{1}{\sqrt{3}}\cos\left(\frac{\pi}{18}\right) + \frac{1}{2}, \ \tilde{r} = \frac{1}{6(2r-1)^2}. \tag{3.16}$$

In Fig. 3.4 on the left, we plot $|\mathcal{R}(iz)|$ and we see that $|\mathcal{R}(iz)| \neq 1$ for $z > 0$. On the right, we show $\mathcal{K}(\alpha, J, N_t, iz)$ for two values of N_t. Clearly, the estimate (3.12) still holds.

For completeness, we show in Fig. 3.5 the contraction factor $\mathcal{K}(\alpha, J, N_t, iz)$ as a function of z for three other IRK methods when $N_t = 200$ is fixed. We consider two values of α and it is clear that a smaller α reduces the convergence factor ρ. This confirms the estimate (3.12) very well, since the bound $\frac{2\alpha}{1+\alpha}N_t$ diminishes when α becomes smaller.

3.4. Optimal choice of α in practice. Both Theorem 3.3 and Theorem 3.4 suggest that one should choose *the parameter α as small as possible*, since the convergence factor ρ diminishes with α. This is however not true in practice, due to the roundoff error arising from the implementation of $\mathcal{F}^*_{\delta t}(\alpha, T_n, u_n^k)$ via the diagonalization technique (cf.(2.6)). To illustrate this, we apply the new parareal algorithm (1.6a) to the advection-dominated diffusion equation (1.3a) with $\Delta x = \frac{1}{200}$, $\Delta T = 0.04$, $T = 4$ and $\nu = 10^{-6}$. Then, for two values of the ratio J we show in Fig. 3.6 the measured error of the new parareal algorithm at each iteration when α varies. For each α, the quantity $2\varepsilon J/\alpha$ is also shown

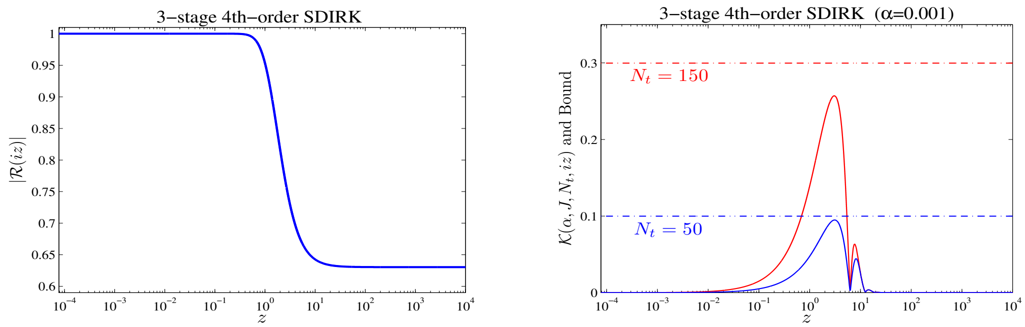

FIG. 3.4. *For the 3-stage 4th-order SDIRK method* (3.16)*, the stability function does not satisfy* $|\mathcal{R}(z)| \equiv 1$ *for* $z \in i\mathbb{R}$*; see the left subfigure. But for the contraction factor* $\mathcal{K}(\alpha, J, N_t, iz)$*, the estimate* (3.12) *still holds. Here,* $J = 20$*; for other* J *the plots look similar.*

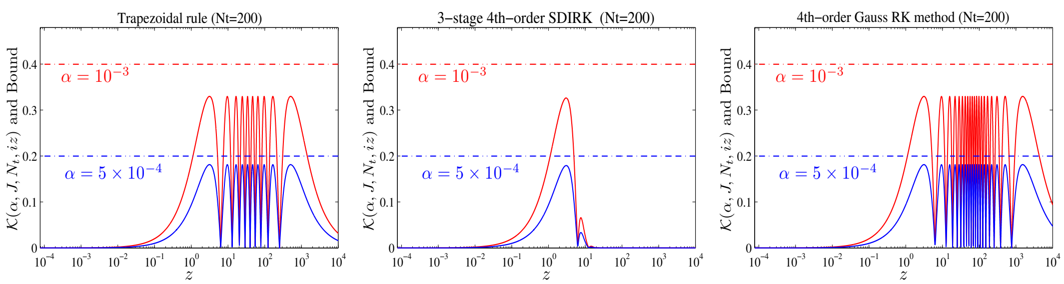

FIG. 3.5. *For the three IRK methods, the contraction factor* $\mathcal{K}(\alpha, J, N_t, iz)$ *as a function of* z *when* $N_t = 200$ *is fixed. Here,* $J = 20$*; for other* J *the plots look similar.*

(the horizontal dash-dotted lines), where ε is the machine precision. This quantity is an estimate of the roundoff error for the diagonalization technique (cf. (2.9)). The horizontal solid lines denote the time discretization error which indicates at which iteration number the parareal algorithm should stop.

From Fig. 3.6 we see that a smaller α leads to faster convergence for the first few iterations, but then the error stagnates at a certain level. The quantity $\frac{2\varepsilon J}{\alpha}$ predicts the ‘stagnation level’ very well[5] and thus gives a principle to select a good parameter $\alpha_{\rm opt}$,

$$\frac{2\varepsilon J}{\alpha_{\rm opt}} = \delta t^p \quad \Longrightarrow \quad \alpha_{\rm opt} = \frac{2\varepsilon J}{\delta t^p}. \tag{3.17}$$

The idea here is to balance the roundoff error and the time discretization error $\mathcal{O}(\delta t^p)$, where p denotes the order of the IRK method.

4. Convergence analysis for the nonlinear case. The convergence analysis for the non-linear case is based on the following assumptions.

[5] We tested many other problems, including some nonlinear parabolic and hyperbolic problems, and it turns out that the quantity $2\varepsilon J/\alpha$ is also a good predictor of the ‘stagnation level’.

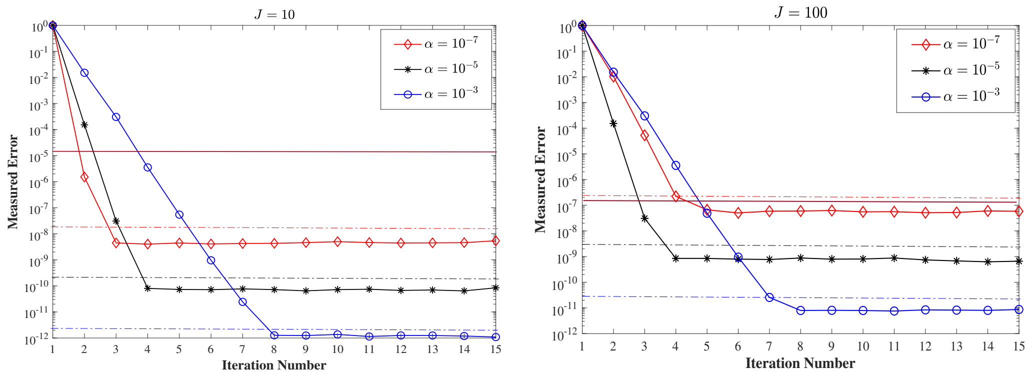

FIG. 3.6. *Measured error of the new parareal algorithm for two choices of the ratio $J = \Delta T/\delta t$, using different head-tail coupling parameters α. A smaller α results in faster convergence for the first few iterations, but then the error stagnates at a certain level.*

ASSUMPTION 1. *For the function $f(t, u)$ appearing in (1.1), suppose there exist constants $L_1 \geq 0$, $L_2 \geq 0$ and $L_3 \geq 0$ such that*

$$\langle f(t, v_1) - f(t, v_2), v_1 - v_2 \rangle \leq -L_1 \|v_1 - v_2\|_2^2, \forall t \in [0, T], \forall v_1, v_2 \in \mathbb{R}^m, \tag{4.1a}$$

$$\langle f(t, v_1) - f(t, v_2) - (f(t, \tilde{v}_1) - f(t, \tilde{v}_2)), v_1 - v_2 - (\tilde{v}_1 - \tilde{v}_2) \rangle \tag{4.1b}$$

$$\leq -L_2 \|v_1 - v_2 - \tilde{v}_1 + \tilde{v}_2\|_2^2, \forall t \in [0, T], \forall v_1, \tilde{v}_1, v_2, \tilde{v}_2 \in \mathbb{R}^m,$$

$$\|f(t, v_1) - f(t, v_2)\|_2 \leq L_3 \|v_1 - v_2\|_2, \ \forall v_1, v_2 \in \mathbb{R}^m, \tag{4.1c}$$

where $\langle \cdot \rangle$ denotes the Euclidean inner product. Moreover, we assume that $\alpha \in (0, 1)$ and that the $\mathcal{F}$-propagator is an exact solver.

LEMMA 4.1. *Under Assumption 1, it holds that*

$$\|\mathcal{F}^*_{\delta t}(\alpha, T_n, w) - \mathcal{F}^*_{\delta t}(\alpha, T_n, \tilde{w})\|_2 \leq \frac{(1-\alpha)e^{-L_1 \Delta T}}{1 - \alpha e^{-L_1 \Delta T}} \|w - \tilde{w}\|_2. \tag{4.2}$$

Proof. Suppose $v_1(t)$ and $v_2(t)$ are the solutions of the two ODEs

$$\begin{cases} v_1'(t) = f(t, v_1(t)), t \in [T_n, T_{n+1}], \\ v_1(T_n) = \alpha v_1(T_{n+1}) + (1-\alpha) w, \end{cases} \qquad \begin{cases} v_2'(t) = f(t, v_2(t)), t \in [T_n, T_{n+1}], \\ v_2(T_n) = \alpha v_2(T_{n+1}) + (1-\alpha) \tilde{w}. \end{cases}$$

Then, it is clear that $\mathcal{F}^*_{\delta t}(\alpha, T_n, w) = v_1(T_{n+1})$ and $\mathcal{F}^*_{\delta t}(\alpha, T_n, \tilde{w}) = v_2(T_{n+1})$. Taking the difference between these two ODEs and by using Assumption 1, we have

$$\begin{aligned} \langle v_1'(t) - v_2'(t), v_1(t) - v_2(t) \rangle &= \langle f(t, v_1(t)) - f(t, v_2(t)), v_1(t) - v_2(t) \rangle \\ &\leq -L_1 \|v_1(t) - v_2(t)\|_2^2. \end{aligned}$$

Because $\langle v_1'(t) - v_2'(t), v_1(t) - v_2(t) \rangle = \|v_1(t) - v_2(t)\|_2 \frac{d\|v_1(t) - v_2(t)\|_2}{dt}$, we have

$$\begin{cases} \frac{d\|v_1(t) - v_2(t)\|_2}{dt} \leq -L_1 \|v_1(t) - v_2(t)\|_2, \ t \in [T_n, T_{n+1}], \\ \|v_1(T_n) - v_2(T_n)\|_2 \leq \alpha \|v_1(T_{n+1}) - v_2(T_{n+1})\|_2 + (1-\alpha) \|w - \tilde{w}\|_2. \end{cases} \tag{4.3}$$

Applying the Gronwall lemma to the governing equation in (4.3) in the interval $[T_n, T_{n+1}]$ leads to $\|v_1(T_{n+1}) - v_2(T_{n+1})\|_2 \leq e^{-L_1 \Delta T} \|v_1(T_n) - v_2(T_n)\|_2$. Inserting the second inequality in (4.3) into this inequality gives

$$\|v_1(T_{n+1}) - v_2(T_{n+1})\|_2 \leq \alpha e^{-L_1 \Delta T} \|v_1(T_{n+1}) - v_2(T_{n+1})\|_2 + (1-\alpha) e^{-L_1 \Delta T} \|w - \tilde{w}\|_2,$$

i.e., $\|v_1(T_{n+1}) - v_2(T_{n+1})\|_2 \leq \frac{(1-\alpha)e^{-L_1\Delta T}}{1-\alpha e^{-L_1\Delta T}}\|w-\tilde{w}\|_2$. $\square$

LEMMA 4.2. *Under Assumption 1, we have*

$$\begin{aligned}
&\|\mathcal{F}_{\delta t}(T_n, w) - \mathcal{F}^*_{\delta t}(\alpha, T_n, w) - (\mathcal{F}_{\delta t}(T_n, \tilde{w}) - \mathcal{F}^*_{\delta t}(\alpha, T_n, \tilde{w}))\|_2 \leq \\
&\alpha \frac{L_3}{L_2 - (1-\alpha)L_1}\left(e^{-L_1\Delta T} - e^{-\frac{L_2}{1-\alpha}\Delta T}\right)\|w-\tilde{w}\|_2.
\end{aligned} \tag{4.4}$$

Proof. Let $v_1(t), v_2(t), \tilde{v}_1(t)$ and $\tilde{v}_2(t)$ be the solutions of the ODEs

$$\begin{aligned}
&\begin{cases} v_1'(t) = f(t, v_1(t)), t \in [T_n, T^*], \\ v_1(T_n) = w, \end{cases} &&\begin{cases} v_2'(t) = f(t, v_2(t)), t \in [T_n, T^*], \\ v_2(T_n) = \alpha v_2(T^*) + (1-\alpha)w, \end{cases} \\
&\begin{cases} \tilde{v}_1'(t) = f(t, \tilde{v}_1(t)), t \in [T_n, T^*], \\ \tilde{v}_1(T_n) = \tilde{w}, \end{cases} &&\begin{cases} \tilde{v}_2'(t) = f(t, \tilde{v}_2(t)), t \in [T_n, T^*], \\ \tilde{v}_2(T_n) = \alpha \tilde{v}_2(T^*) + (1-\alpha)\tilde{w}. \end{cases}
\end{aligned} \tag{4.5}$$

If $T^* = T_{n+1}$, it is clear that $\mathcal{F}_{\delta t}(T_n, w) = v_1(T^*)$, $\mathcal{F}^*_{\delta t}(\alpha, T_n, w) = v_2(T^*)$, $\mathcal{F}_{\delta t}(T_n, \tilde{w}) = \tilde{v}_1(T^*)$ and $\mathcal{F}^*_{\delta t}(\alpha, T_n, \tilde{w}) = \tilde{v}_2(T^*)$. Integrating the first two ODEs gives

$$\begin{cases} v_1(T^*) = w + \int_{T_n}^{T^*} f(t, v_1(t))dt, \\ v_2(T^*) = v_2(T_n) + \int_{T_n}^{T^*} f(t, v_2(t))dt = \alpha v_2(T^*) + (1-\alpha)w + \int_{T_n}^{T^*} f(t, v_2(t))dt, \\ \left(\Longrightarrow v_2(T^*) = w + \frac{1}{1-\alpha}\int_{T_n}^{T^*} f(t, v_2(t))dt\right). \end{cases}$$

The difference $r(T^*) := v_1(T^*) - v_2(T^*)$ satisfies

$$\begin{aligned}
r(T^*) &= \int_{T_n}^{T^*} f(t, v_1(t))dt - \frac{1}{1-\alpha}\int_{T_n}^{T^*} f(t, v_2(t))dt \\
&= \frac{1}{1-\alpha}\int_{T_n}^{T^*} [f(t, v_1(t)) - f(t, v_2(t))]\, dt - \frac{\alpha}{1-\alpha}\int_{T_n}^{T^*} f(t, v_1(t))dt.
\end{aligned}$$

Similarly, by letting $\tilde{r}(T^*) = \tilde{v}_1(T^*) - \tilde{v}_2(T^*)$, we have

$$\tilde{r}(T^*) = \frac{1}{1-\alpha}\int_{T_n}^{T^*} [f(t, \tilde{v}_1(t)) - f(t, \tilde{v}_2(t))]\, dt - \frac{\alpha}{1-\alpha}\int_{T_n}^{T^*} f(t, \tilde{v}_1(t))dt.$$

Let $R(T^*) = r(T^*) - \tilde{r}(T^*)$. Then, it holds that

$$\begin{aligned}
R(T^*) =& \frac{1}{1-\alpha}\int_{T_n}^{T^*} [f(t, v_1(t)) - f(t, v_2(t)) - (f(t, \tilde{v}_1(t)) - f(t, \tilde{v}_2(t)))]\, dt - \\
& \frac{\alpha}{1-\alpha}\int_{T_n}^{T^*} (f(t, v_1(t)) - f(t, \tilde{v}_1(t)))\, dt.
\end{aligned}$$

Now, we regard T^* as a variable in the interval $[T_n, T_{n+1}]$ (i.e., $T^* \in [T_n, T_{n+1}]$). Then, differentiating $R(T^*)$ with respect to T^* gives

$$\begin{cases} R'(T^*) = \frac{1}{1-\alpha}\left([f(t, v_1(T^*)) - f(t, v_2(T^*))] - [f(t, \tilde{v}_1(T^*)) - f(t, \tilde{v}_2(T^*))]\right) - \\ \qquad \frac{\alpha}{1-\alpha}\left[f(T^*, v_1(T^*)) - f(T^*, \tilde{v}_1(T^*))\right], \quad T^* \in [T_n, T_{n+1}], \\ R(T_n) = 0. \end{cases}$$

Similar to the proof of Lemma 4.1, by using (4.1b) we have

$$\begin{cases} \frac{d\|R(T^*)\|_2}{dT^*} \leq -\frac{L_2}{1-\alpha}\|R(T^*)\|_2 + \frac{\alpha}{1-\alpha}\|f(T^*, v_1(T^*)) - f(T^*, \tilde{v}_1(T^*))\|_2, T^* \in [T_n, T_{n+1}], \\ \|R(T_n)\|_2 = 0. \end{cases}$$

Then, by using (4.1c) we have

$$\frac{d\|R(T^*)\|_2}{dT^*} \leq -\frac{L_2}{1-\alpha}\|R(T^*)\|_2 + \frac{\alpha L_3}{1-\alpha}\|v_1(T^*) - \tilde{v}_1(T^*)\|_2, \ T^* \in [T_n, T_{n+1}]. \tag{4.6}$$

We next estimate $\|v_1(T^*) - \tilde{v}_1(T^*)\|_2$. From the first and third ODE in (4.5), similar to (4.3) we have

$$\begin{cases} \frac{d\|v_1(t)-\tilde{v}_1(t)\|_2}{dt} \leq -L_1\|v_1(t) - \tilde{v}_1(t)\|_2, \ t \in [T_n, T^*], \\ \|v_1(T_n) - \tilde{v}_1(T_n)\|_2 = \|w - \tilde{w}\|_2, \end{cases}$$

and this gives $\|v_1(T^*) - \tilde{v}_1(T^*)\|_2 \leq e^{-L_1(T^*-T_n)}\|w - \tilde{w}\|_2$. Inserting this into (4.6) gives

$$\frac{d\|R(T^*)\|_2}{dT^*} \leq -\frac{L_2}{1-\alpha}\|R(T^*)\|_2 + \frac{\alpha L_3 e^{-L_1(T^*-T_n)}}{1-\alpha}\|w - \tilde{w}\|_2, \ T^* \in [T_n, T_{n+1}].$$

From this differential inequality, by the Gronwall lemma we get

$$\|R(T_{n+1})\|_2 \leq \alpha\frac{L_3}{L_2 - (1-\alpha)L_1}\left(e^{-L_1\Delta T} - e^{-\frac{L_2}{1-\alpha}\Delta T}\right)\|w - \tilde{w}\|_2. \tag{4.7}$$

Because $r(T^*) = v_1(T^*) - v_2(T^*)$, $\tilde{r}(T^*) = \tilde{v}_1(T^*) - \tilde{v}_2(T^*)$ and $R(T^*) = r(T^*) - \tilde{r}(T^*)$, the desired result (4.4) follows from (4.7). □

THEOREM 4.3. *Under Assumption 1, for the new parareal algorithm* (1.6a) *we have*

$$\max_{n=1,2,\ldots,N_t}\|u(T_n) - u_n^{k+1}\|_2 \leq \rho\max_{n=1,2,\ldots,N_t}\|u(T_n) - u_n^k\|_2,$$

$$\text{with } \rho = \frac{\alpha}{(1-\alpha)(1-\sigma)}\frac{e^{-L_1\Delta T} - e^{-\frac{L_2}{1-\alpha}\Delta T}}{\frac{L_2}{1-\alpha} - L_1}L_3 \text{ and } \sigma = \frac{(1-\alpha)e^{-L_1\Delta T}}{1-\alpha e^{-L_1\Delta T}}. \tag{4.8}$$

Proof. Let $\xi_n^k = u(T_n) - u_n^k$. Then, for $n \geq 0$ we have

$$\begin{aligned} \xi_{n+1}^{k+1} &= \mathcal{F}_{\delta t}(T_n, u(T_n)) - \left[\mathcal{F}_{\delta t}^*(\alpha, T_n, u_n^{k+1}) + \mathcal{F}_{\delta t}(T_n, u_n^k) - \mathcal{F}_{\delta t}^*(\alpha, T_n, u_n^k)\right] \\ &= \left[\mathcal{F}_{\delta t}(T_n, u_n) - \mathcal{F}_{\delta t}^*(\alpha, T_n, u_n)\right] - \left[\mathcal{F}_{\delta t}(T_n, u_n^k) - \mathcal{F}_{\delta t}^*(\alpha, T_n, u_n^k)\right] + \\ &\quad \left[\mathcal{F}_{\delta t}^*(\alpha, T_n, u_n) - \mathcal{F}_{\delta t}^*(\alpha, T_n, u_n^{k+1})\right]. \end{aligned}$$

By applying Lemma 4.2 to the first two terms and Lemma 4.1 to the last term, we get

$$\|\xi_{n+1}^{k+1}\|_2 \leq \alpha\frac{L_3}{L_2 - (1-\alpha)L_1}\left(e^{-L_1\Delta T} - e^{-\frac{L_2}{1-\alpha}\Delta T}\right)\|\xi_n^k\|_2 + \frac{(1-\alpha)e^{-L_1\Delta T}}{1-\alpha e^{-L_1\Delta T}}\|\xi_n^{k+1}\|_2.$$

Let $\sigma = \frac{(1-\alpha)e^{-L\Delta T}}{1-\alpha e^{-L\Delta T}}$ and $\kappa = \alpha\frac{L_3}{L_2-(1-\alpha)L_1}\left(e^{-L_1\Delta T} - e^{-\frac{L_2}{1-\alpha}\Delta T}\right)$. Then, by noticing that $\xi_0^k = 0$ for all $k \geq 0$ we have

$$\underbrace{\begin{bmatrix} 1 & & & \\ -\sigma & 1 & & \\ & \ddots & \ddots & \\ & & -\sigma & 1 \end{bmatrix}}_{=\mathcal{M}}\begin{bmatrix} \|\xi_1^{k+1}\|_2 \\ \|\xi_2^{k+1}\|_2 \\ \vdots \\ \|\xi_{N_t}^{k+1}\|_2 \end{bmatrix} \leq \underbrace{\begin{bmatrix} 0 & & & \\ \kappa & 0 & & \\ & \ddots & \ddots & \\ & & \kappa & 0 \end{bmatrix}}_{=\mathcal{N}}\begin{bmatrix} \|\xi_1^k\|_2 \\ \|\xi_2^k\|_2 \\ \vdots \\ \|\xi_{N_t}^k\|_2 \end{bmatrix}.$$

Since $L_1 \geq 0$ and $\alpha \in (0,1)$, it holds that $\sigma \in (0,1]$. By using Lemma 3.1, $\|\mathcal{M}^{-1}\|_\infty = \frac{1-\sigma^{N_t}}{1-\sigma} \leq \frac{1}{1-\sigma}$. Considering that $\|\mathcal{N}\|_\infty = \kappa$, we get the desired result (4.8). □

REMARK 4.1. *An important property of the convergence factor ρ given by* (4.8) *is that $\rho \to 0$ as $\alpha \to 0$. This can be seen by noticing that*

$$(1-\alpha)(1-\sigma) \to e^{-L_1\Delta T}, \quad \frac{e^{-L_1\Delta T} - e^{-\frac{L_2}{1-\alpha}\Delta T}}{\frac{L_2}{1-\alpha} - L_1} \to \frac{e^{-L_1\Delta T} - e^{-L_2\Delta T}}{L_2 - L_1},$$

and therefore $\rho = \mathcal{O}(\alpha)$. This implies that for nonlinear ODEs, at least in the asymptotic sense, the new parareal algorithm (1.6a) *converges faster when a smaller α is used.*

5. Numerical results. In this section, we provide numerical results to illustrate our theoretical analysis. For all numerical experiments, the parareal algorithm starts from a random initial guess[6] and stops when the error satisfies

$$\max_{1\le n\le N_t} \|u_n^k - u_n^{\rm ref}\|_\infty \le 10^{-12}, \tag{5.1}$$

where $\{u_n^{\rm ref}\}_{n=1}^{N_t}$ are the reference solutions obtained by directly applying the $\mathcal{F}$-propagator.

5.1. PDEs with fractional Laplacian. For the first set of numerical results, we consider the two-sided fractional diffusion equation

$$\begin{cases} \partial_t u - \left(d_1(x)_0\mathcal{D}_x^{\frac{3}{2}} + d_2(x)_x\mathcal{D}_1^{\frac{3}{2}}\right)u + g(x,t,u) = 0, & (x,t)\in(0,1)\times(0,4],\\ u(x,0)=\sin(4\pi x), & x\in(0,1),\\ u(0,t)=u(1,t)=0, & t\in(0,4), \end{cases} \tag{5.2}$$

where $d_1(x)\ge 0$, $d_2(x)\ge 0$. For any $\gamma\in(1,2]$ the symbols ${}_0\mathcal{D}_x^\gamma u$ and ${}_x\mathcal{D}_1^\gamma u$ are the left and right Riemann-Liouville fractional operators

$$\begin{aligned} {}_0\mathcal{D}_W x^\gamma u(x,t) &= \frac{1}{\Gamma(2-\gamma)}\frac{\partial^2}{\partial x^2}\int_0^x \frac{u(\tilde{x},t)}{(x-\tilde{x})^{\gamma-1}}d\tilde{x},\\ {}_x\mathcal{D}_1^\gamma u(x,t) &= \frac{1}{\Gamma(2-\gamma)}\frac{\partial^2}{\partial x^2}\int_x^1 \frac{u(\tilde{x},t)}{(x-\tilde{x})^{\gamma-1}}d\tilde{x}. \end{aligned}$$

For space discretization, we use the 2nd-order weighted and shifted Grünwald difference (WSGD) formula established in [38]. For a well-defined function $v(x)$ on the bounded interval $[0,1]$, the WSGD formula leads to the following matrix approximation of the left and right Riemann-Liouville fractional derivatives on the uniformly spaced grid points $\{x_j = j\Delta x, \Delta x = 1/m, j=1,2,\ldots,m-1\}$:

$$\left({}_0\mathcal{D}_x^\gamma\right)\left(v(x_1),\ldots,v(x_{m-1})\right)^\top = W_\gamma \mathbf{v}, \left({}_x\mathcal{D}_1^\gamma\right)\left(v(x_1),\ldots,v(x_{m-1})\right)^\top = W_\gamma^\top \mathbf{v},$$

where $\mathbf{v}=(v_1,\ldots,v_{m-1})^\top$, $v_j = v(x_j)+\mathcal{O}(\Delta x^2)$ and

$$W_\gamma = \begin{pmatrix} \omega_1^{(\gamma)} & \omega_0^{(\gamma)} & & & \\ \omega_2^{(\gamma)} & \omega_1^{(\gamma)} & \omega_0^{(\gamma)} & & \\ \vdots & \omega_2^{(\gamma)} & \omega_1^{(\gamma)} & \ddots & \\ \omega_{m-2}^{(\gamma)} & \cdots & \ddots & \ddots & \omega_0^{(\gamma)} \\ \omega_{m-1}^{(\gamma)} & \omega_{m-2}^{(\gamma)} & \cdots & \omega_2^{(\gamma)} & \omega_1^{(\gamma)} \end{pmatrix}, \text{ with } \begin{cases} \omega_0^{(\gamma)} = \frac{\gamma}{2}\eta_0^{(\gamma)}, \\ \omega_l^{(\gamma)} = \frac{\gamma}{2}\eta_l^{(\gamma)} + \frac{2-\gamma}{2}\eta_{l-1}^{(\gamma)} \text{ for } l\ge 1. \end{cases}$$

The quantities $\{\eta_l^{(\gamma)}\}_{l\ge 0}$ are the coefficients of the power series of $(1-z)^\gamma$,

$$\eta_0^{(\gamma)} = 1,\ \eta_l^{(\gamma)} = \left(1-\frac{1+\gamma}{l}\right)\eta_{l-1}^{(\gamma)},\ l=1,2,\ldots.$$

Let $u_j(t)\approx u(x_j,t)$. Then, by applying the WSGD formula to each of the four fractional derivatives in (5.2) we get the ODE system

$$\boldsymbol{u}'(t) + A\boldsymbol{u}(t) + G(t,\boldsymbol{u}) = 0, \tag{5.3a}$$

where $\boldsymbol{u}(t) = (u_1(t),\ldots,u_m(t)))^\top$, $G(t,\boldsymbol{u}) = (g(x_1,t,u_1(t)),\ldots,g(x_m,t,u_m(t)))^\top$ and

$$A = -\frac{1}{(\Delta x)^\gamma}\left(D_1 W_\gamma + D_2 W_\gamma^\top\right). \tag{5.3b}$$

In (5.3b), $D_1 = \mathrm{diag}(d_1(x_1),\ldots,d_1(x_m))$ and $D_2 = \mathrm{diag}(d_2(x_1),\ldots,d_2(x_m))$.

[6]In practice, one usually starts with the coarse propagator, but we use a random initial guess here for the convergence study.

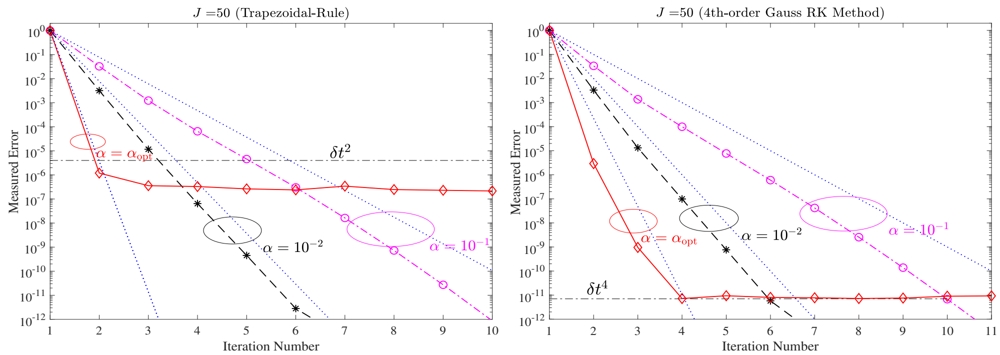

FIG. 5.1. *Convergence of the new parareal algorithm* (1.6a) *using the Trapezoidal Rule (left) and the 4th-order Gauss RK method (right), together with the linear bound from Theorem 3.3. The theoretical and numerical curves belonging together are indicated by the colored ellipses. Here, according to* (3.17) *the optimal parameter is* α_{opt}=*5.5e-9 (left) and* α_{opt}=*1e-3 (right).*

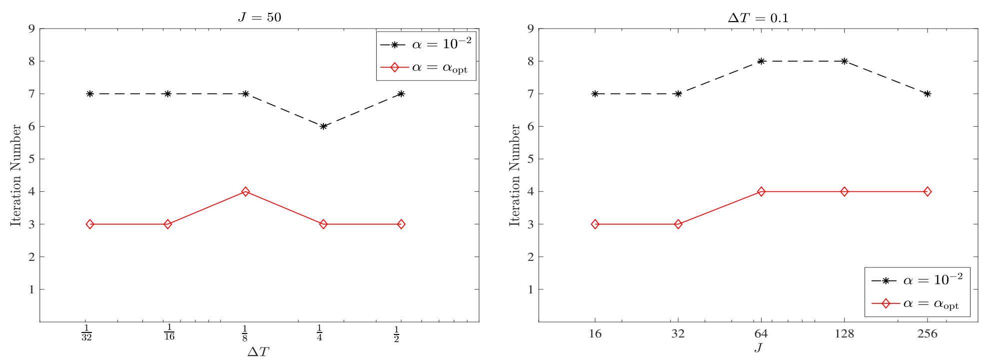

FIG. 5.2. *Dependence of the convergence rate of the new parareal algorithm* (1.6a) *on* ΔT *(left) and* J *(right) in the linear case using the Trapezoidal Rule. Here the parameter* α_{opt} *is specified by* (3.17) *with* $p = 2$.

5.1.1. The linear case: $g(x,t,u) = 0$**.** For the linear case, we consider the coefficient functions $d_1(x) = d_2(x) = 2x(1-x)^5$, which results in an SPD matrix A in (5.3b). Let $\Delta x = \frac{1}{200}$, $\Delta T = 0.1$ and $J = 50$. For the Trapezoidal Rule and the 4th-order Gauss RK method, we show in Figure 5.1 the convergence of the new parareal algorithm (1.6a) together with the linear bound given by Theorem 3.3 (dotted line), for three values $\alpha = 0.1, 0.01, \alpha_{\text{opt}}$ (the optimal parameter from (3.17)). The horizontal line denotes the quantity δt^p, which represents the time discretization error and indicates where the iterations should stop in practice. We see that as α decreases the new parareal algorithm converges faster and that for $\alpha = \alpha_{\text{opt}}$ the convergence curve stagnates at the level of the time discretization error.

In Fig. 5.2, we show the dependence of the new parareal algorithm (1.6a) on the ratio J and the large step-size ΔT when one of these two quantities is fixed and the other varies. We see that the convergence rate is robust with respect to these two parameters.

5.1.2. The nonlinear case: $g(x,t,u) = \sin(\frac{\pi}{2}u)$**.** The coefficient functions $d_1(x)$ and $d_2(x)$ are given by

$$d_1(x) = x^5\sqrt{1-x},\ d_2(x) = \sqrt{x}(1-x)^5/8. \tag{5.4}$$

Since $d_1(x) \neq d_2(x)$, the matrix A in (5.3a) is non-symmetric. In Fig. 5.3, we show the

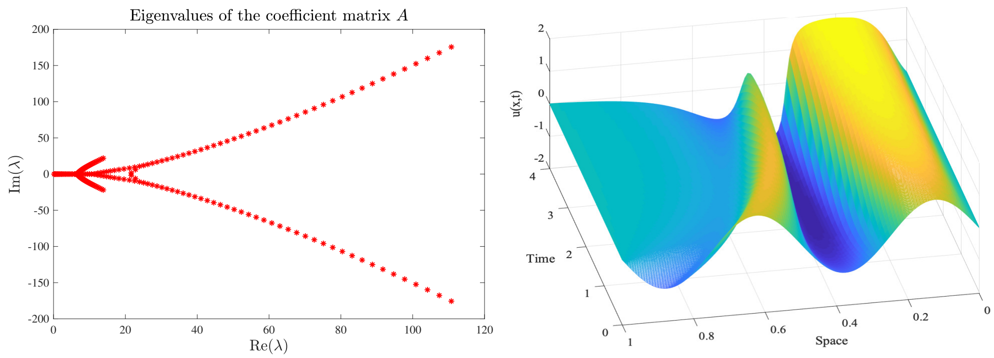

FIG. 5.3. *Left: eigenvalues of the coefficient matrix A with $d_1(x)$ and $d_2(x)$ from (5.4). Right: solution $u(x,t)$ of the fractional PDE for the nonlinear source term $g(x,t,u) = \sin(\frac{\pi}{2}u)$.*

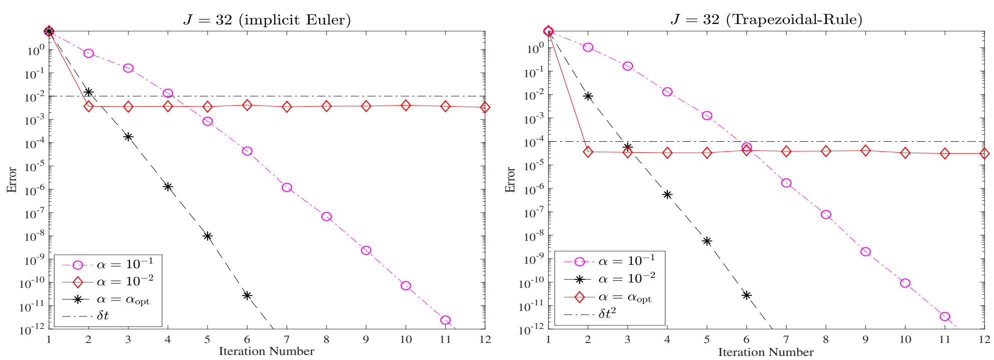

FIG. 5.4. *Convergence of the new parareal algorithm (1.6a) for the nonlinear case $g(x,t,u) = \sin(\frac{\pi}{2}u)$. The horizontal line denotes the time discretization error δt^p with $p = 1$ and $p = 2$. Here, according to (3.17) the optimal parameter is $\alpha_{\rm opt}$=4.5e-12 (left) and $\alpha_{\rm opt}$=1.5e-9 (right).*

eigenvalues of the coefficient matrix A with $\Delta x = \frac{1}{200}$ on the left, and the solution of the nonlinear fractional PDE on the right.

Let $\Delta T = 0.1$ and $J = 32$. We show in Fig. 5.4 the measured convergence rate of the new parareal algorithm (1.6a) for two choices of the time integrator: Backward Euler and the Trapezoidal Rule, with three values of the parameter α. Similar to the linear case, we see that the new parareal algorithm from Fig. 5.4 converges faster when α is smaller. In particular, when $\alpha = \alpha_{\rm opt}$ the error decays rapidly for the first few iterations and then stagnates at the level of the time discretization error.

Similar to Fig. 5.2, we show in Fig. 5.5 the iteration number against the ratio J and the large step-size ΔT and we see that the convergence rate is robust with respect to these two discretization parameters.

For nonlinear ODEs, as explained in Section 2.2, we need to do simplified Newton iterations in each large subinterval $[T_n, T_{n+1}]$. It is important to check how many Newton iterations are needed for the new parareal algorithm (1.6a). To this end, at the k-th iteration of the new parareal algorithm we define the maximal number of Newton iterations by

$$\text{Newton-Iteration}(k) := \max\left\{\max_{1\leq n\leq N_t} l_n^k, \max_{1\leq n\leq N_t} l_n^{k-1}\right\}, \quad k \geq 1, \tag{5.5}$$

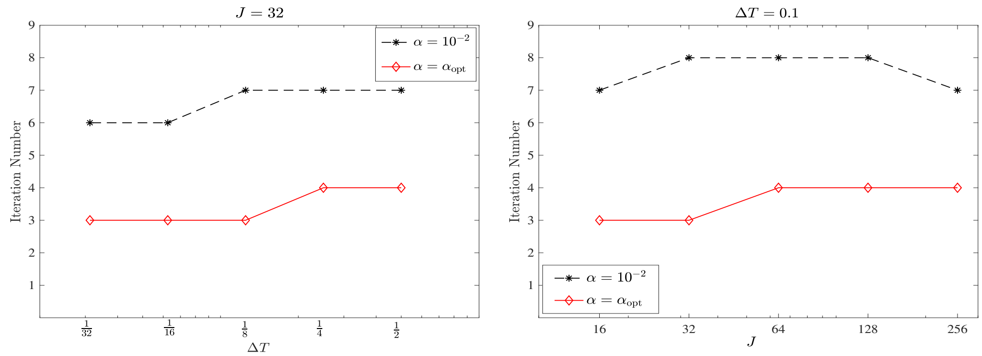

FIG. 5.5. *Dependence of the convergence rate of the new parareal algorithm* (1.6a) *on* ΔT *(left) and* J *(right) for the nonlinear case* $g(x,t,u) = \sin(\frac{\pi}{2}u)$ *using Backward Euler.*

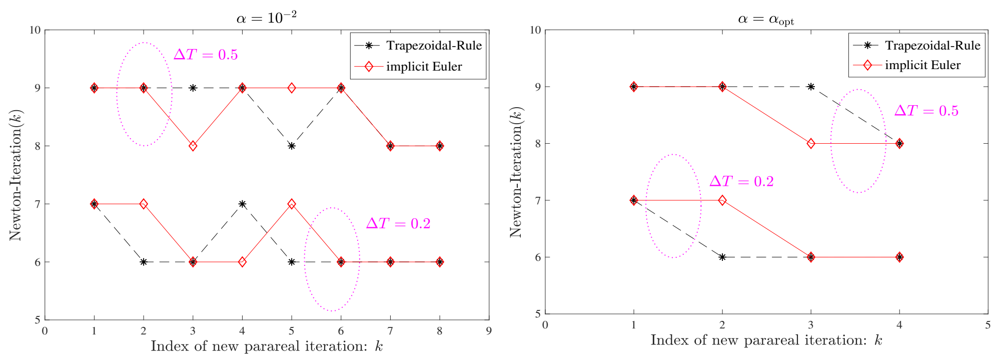

FIG. 5.6. *For the nonlinear fractional PDE, in each parareal iteration (outer loop) we need to do simplified Newton iterations (inner loop) for the* N_t *large subintervals. For two values of* α*, the maximal Newton iteration number defined in* (5.6) *for each parareal iteration.*

where l_n^k (resp. l_n^{k-1}) denotes the number of simplified Newton iterations[7] for computing $\mathcal{F}^*_{\delta t}(\alpha, T_n, u_n^k)$ (resp. $\mathcal{F}^*_{\delta t}(\alpha, T_n, u_n^{k-1})$) in the n-th subinterval $[T_n, T_{n+1}]$. In Fig. 5.6 we show Newton-Iteration(k) for two values of α: $\alpha = 10^{-2}$ (left) and $\alpha = \alpha_{\rm opt}$ (right). We see that the maximal number of simplified Newton iterations is 8−9 during the parareal iteration if $\Delta T = 0.5$. If we use a smaller step-size $\Delta T = 0.2$, the maximal Newton iteration number is reduced to 6−7. We now explain this reduction: as we mentioned in Remark 2.3, for nonlinear ODEs an important idea of applying the new parareal algorithm (1.6a) lies in approximating the J Jacobian matrices (on the fine time points $\{t_{n,j}\}_{j=1}^J$) by a single matrix $\widetilde{\mathbf{J}}_n$, i.e.,

$$\nabla f(\boldsymbol{u}(t_{n,j})) \approx \widetilde{\mathbf{J}}_n,\ j = 1, 2, \ldots, J, \tag{5.6}$$

where $n = 1, 2, \ldots, N_t$ and $f(\boldsymbol{u}(t_{n,j})) = A\boldsymbol{u}(t_{n,j}) + \sin(\frac{\pi}{2}\boldsymbol{u}(t_{n,j}))$. If we keep J fixed and reduce ΔT, the variation of the solution $\boldsymbol{u}(t)$ in the interval $[T_n, T_{n+1}]$ becomes small and thus $\widetilde{\mathbf{J}}_n$ becomes a good approximation of the J Jacobian matrices. This explains why the simplified Newton iteration number diminishes when ΔT becomes small. For the nonlinear fractional PDE (5.2), the variation of the solution $\boldsymbol{u}(t)$ along the whole time interval is not large and thus a smaller ΔT only *slightly* reduces the iteration number. If

[7]The tolerance in the simplified Newton iteration is the same as in the parareal algorithm, i.e., for $\alpha > \alpha_{\rm opt}$ it is 10^{-12} and for $\alpha = \alpha_{\rm opt}$ the tolerance is δt^p, where p is the order of the time-integrator.

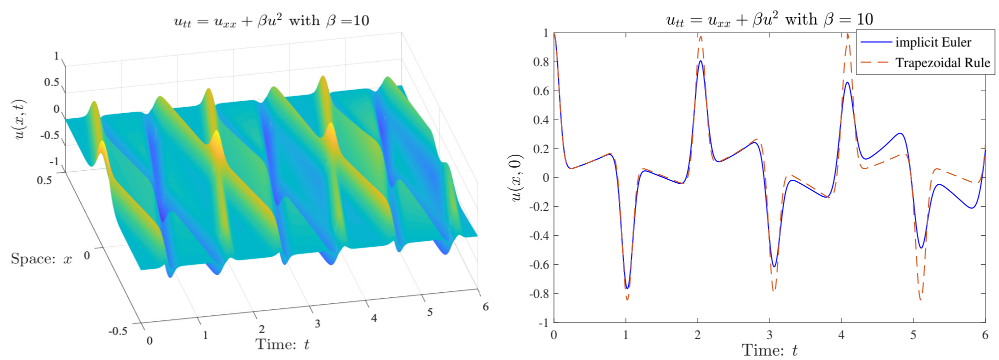

FIG. 5.7. *For the nonlinear wave equation* (5.7) *with* $\beta = 10$*, the solution* $u(x,t)$ *computed by the Trapezoidal Rule (left) and a comparison of the solution at* $x = 0$ *computed by two time-integrators, the Trapezoidal Rule and Backward Euler. The solution computed by the latter is damped as* t *grows.*

the solution $\boldsymbol{u}(t)$ has a large variation, ΔT has a dramatic effect on the Newton iteration number. We will show this for a nonlinear wave equation in the next subsection.

5.2. The nonlinear wave equation. We next consider the nonlinear wave equation from [14],

$$\begin{cases} \partial_{tt}u = \partial_{xx} + \beta u^2, & (x,t) \in (-\frac{1}{2}, \frac{1}{2}) \times (0,6), \\ u(-\frac{1}{2},t) = u(\frac{1}{2},t), & t \in (0,6), \\ u(x,0) = e^{-100x^2}, u_t(x,0) = 0, & x \in (-\frac{1}{2}, \frac{1}{2}). \end{cases} \tag{5.7}$$

The parameter $\beta > 0$ represents the strength of the nonlinearity. The problem is discretized in space by using centered finite differences with a mesh-size $\Delta x = \frac{1}{128}$. This results in the system of ODEs

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix}' = \begin{bmatrix} O & I \\ A & O \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \beta \boldsymbol{u}^2 \end{bmatrix}, \quad A := \frac{1}{\Delta x^2} \begin{bmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{bmatrix},$$

where the operation $\boldsymbol{u}^2$ should be understood component-wise.

For $\beta = 10$, the numerical solution $u(x,t)$ computed by the Trapezoidal Rule is shown in Fig. 5.7 on the left, from which we see clearly the wave propagation character of the solution. If we compute the solution with Backward Euler, which is a numerically dissipative IRK method, the wave is damped as the time t increases. This is illustrated in Fig. 5.7 on the right, where we show the solution at the space point $x = 0$. (The solution is computed by the two time-integrators with $\Delta t = \frac{1}{256}$.) We will thus only consider the Trapezoidal Rule in the following.

With $J = 40$, $\Delta x = \frac{1}{128}$, $\Delta T = 0.2$ and two values of the problem parameter β, we show in Fig. 5.8 the error at each iteration of the new parareal algorithm (1.6a) for three values of α. We see that a smaller α also leads to faster convergence of the new parareal algorithm and that the optimal parameter α_{opt} makes the error stagnate at the level of the time discretization error.

We next show in Fig. 5.9 the measured iteration number of the new parareal algorithm when one of the two discretization parameters ΔT and J varies and the other one is fixed. Similar to the fractional PDE, for the nonlinear wave equation we see that the convergence rate is still insensitive to the change in J. But quite differently, the iteration

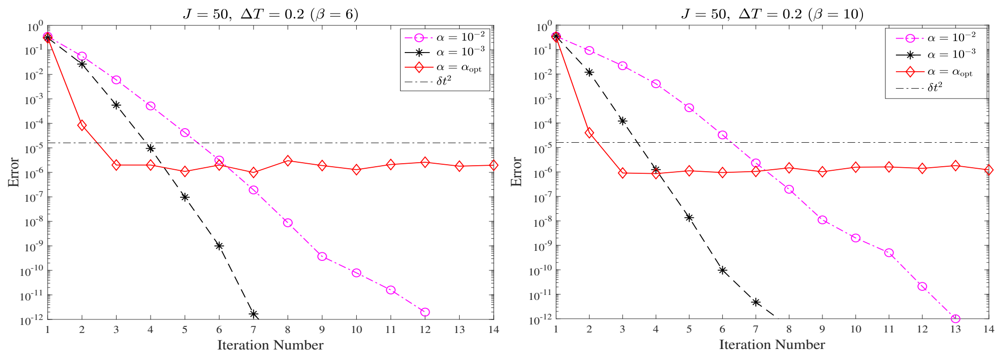

FIG. 5.8. *For the nonlinear wave equation* (5.7) *with* $\beta = 6$ *(left) and* $\beta = 10$ *(right), the measured convergence rate of the new parareal algorithm* (1.6a)*. The quantity* $\alpha_{\rm opt}$ *is the optimal parameter given by* (3.17) *and the horizontal line denotes the time discretization error* δt^2*. Here,* $\alpha_{\rm opt}$*=1.38e-9.*

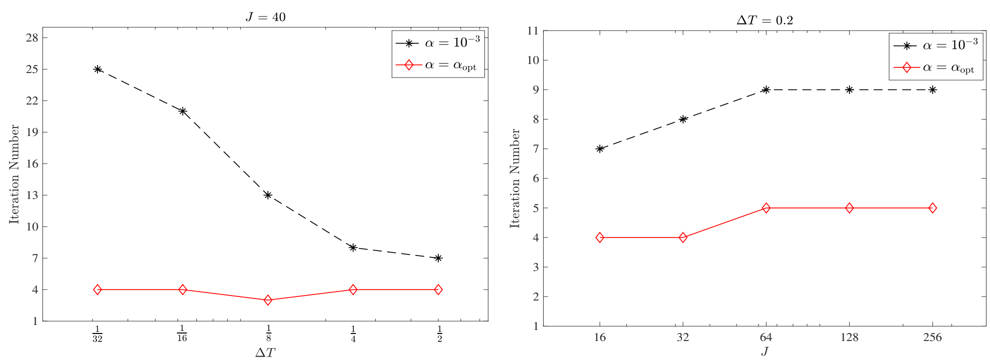

FIG. 5.9. *For the nonlinear wave equation* (5.7) *with* $\beta = 10$*, the dependence of the convergence rate of the new parareal algorithm* (1.6a) *on* ΔT *(left) and* J *(right).*

number increases when ΔT becomes small. This confirms our analysis in the linear case very well, since $\rho \leq \frac{2\alpha N_t}{1+\alpha}$ and N_t (cf. Theorem 3.4) increases as ΔT decreases.

In Fig. 5.10, we show the Newton iteration number (inner loop) for each parareal iteration (outer loop). On the left, we consider $J = 40$ and two values of ΔT. We see that a smaller ΔT obviously reduces the Newton iteration number, especially when $\alpha = 10^{-3}$. The reason for this reduction is the same as we explained for Fig. 5.6. The Newton iteration number is however insensitive to the change of J, as we can see in Figure 5.10 on the right. For the two subfigures, compared to $\alpha = 10^{-3}$ we see that the Newton iteration number is smaller when $\alpha = \alpha_{\rm opt}$. This is because of the tolerance for the Newton iteration: for the latter the tolerance is δt^2, while for the former it is 10^{-12}. If we set δt^2 as the tolerance for both $\alpha = 10^{-3}$ and $\alpha = \alpha_{\rm opt}$, the Newton iteration numbers are similar.

6. Conclusions. We proposed a new variant of the parareal algorithm which allows the use of a coarse propagator that discretizes the underlying problem on the same mesh as the fine propagator. To make the coarse propagator cheaper, we make it quasi time periodic using a head-tail coupling condition, which permits its solution in parallel using the diagonalization technique. This approach removes completely phase and dispersion differences between the coarse and fine propagator, which is important for hyperbolic problems. We also determined the optimal choice of the parameter in the head-tail

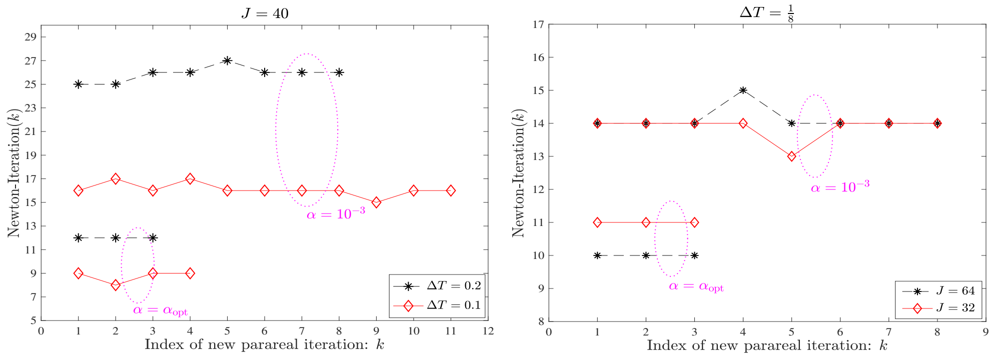

FIG. 5.10. *For the nonlinear wave equation* (5.7) *with* $\beta = 10$*, the maximal Newton iteration number defined in* (5.6) *for each parareal iteration.*

coupling condition.

An interesting observation is that with the optimal choice, the new algorithm sometimes converges in only one iteration to the truncation error of the numerical scheme: this is the case for the linear fractional Laplacian example with the Trapezoidal Rule in Fig. 5.1 on the left, and for the nonlinear fractional Laplacian example in Fig. 5.4, and it also almost holds for the non-linear wave equation example in Fig. 5.8. It does not quite hold for the advection-dominated diffusion equation in Fig. 3.6, where about two iterations are needed, and not at all for the fractional Laplacian example with the 4th-order Gauss RK method in Fig. 5.1 on the right, where the new algorithm takes three iterations. Convergence in one iteration indicates with our random initial guess that one could use the new coarse propagator as a stand alone time parallel solver. We will analyze conditions under which this approach is successful in more detail.

Acknowledgement. The authors are very grateful to the anonymous referees for their careful reading of a preliminary version of the manuscript and their valuable suggestions, which greatly improved the quality of this paper. The second author is supported by the NSF of China (No. 11771313), the NSF of Sichuan province (No. 2018JY0469) and the Science Challenge Project (No. TZ2016002).

REFERENCES

[1] D. Bini, G. Latouche, and B. Meini, *Numerical methods for structured Markov chains*, Oxford University Press: New York, 2005.

[2] V. A. Dobrev, T. Kolev, N. A. Petersson, and J. B. Schroder, *Two-level convergence theory for multigrid reduction in time (MGRIT)*, SIAM J. Sci. Comput., 39 (2017), pp. S501-S527.

[3] F. Chen, J. S. Hesthaven, and X. Zhu, *On the use of reduced basis methods to accelerate and stabilize the Parareal method*, in: Quarteroni, A., Rozza, G. (eds.) Reduced Order Methods for Modeling and Computational Reduction, MS&A–Modeling, Simulation and Applications, vol. 9, pp. 187-214. Springer, Berlin, 2014.

[4] X. H. Du, M. Sarkis, C. F. Schaerer, and D. B. Szyld, *Inexact and truncated parareal-in-time Krylov subspace methods for parabolic optimal control problems*, Electron. Trans. Numer. Anal., 40 (2013), pp. 36-57.

[5] X. Dai and Y. Maday, *Stable parareal in time method for first- and second-order hyperbolic systems*, SIAM J. Sci. Comput., 35 (2013), pp. A52-A78.

[6] A. Eghbal, A. G. Gerber, and E. Aubanel, *Acceleration of unsteady hydrodynamic simulations using the Parareal algorithm*, J. Comput. Sci., 19 (2017), pp. 57-76.

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

[8] C. Farhat, J. Cortial, C. Dastillung, and H. Bavestrello, *Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses*, Int. J. Numer. Methods Eng., 67 (2006), pp. 697-724.

[9] R. D. Falgout, S. Friedhoff, T. Kolev, S. P. MacLachlan, and J. B. Schroder, *Parallel time integration with multigrid*, SIAM J. Sci. Comput., 36 (2014), pp. C635-C661.

[10] M.J. Gander, I.G. Graham, and E.A. Spence, *Applying GMRES to the Helmholtz equation with shifted Laplacian preconditioning: What is the largest shift for which wavenumber-independent convergence is guaranteed ?* Numer. Math., 131 (2015), pp. 567-614.

[11] M. J. Gander, F. Kwok, and H. Zhang, *Multigrid interpretations of the parareal algorithm leading to an overlapping variant and MGRIT*, Comput. Visual Sci., 19 (2018), pp. 59-74.

[12] M.J. Gander, L. Halpern, J. Rannou, and J. Ryan, *A direct time parallel solver by diagonalization for the wave equation*, SIAM J. Sci. Comp., 41 (2019), pp. A220-A245..

[13] M. J. Gander and L. Halpern, *Time parallelization for nonlinear problems based on diagonalization*, Lect. Notes Comput. Sci. Eng., 116 (2017), pp. 163-170.

[14] M. J. Gander and S. Güttel, *A nonlinear ParaExp algorithm*, Lect. Notes Sci. Eng., 125 (2017), pp. 261-270.

[15] M. J. Gander and S. Güttel, *ParaExp: A parallel integrator for linear initial-value problems*, SIAM J. Sci. Comp., 35 (2013), pp. C123-C142.

[16] M. J. Gander, J. L. Jiang, B. Song, and H. Zhang, *Analysis of two parareal algorithms for time-periodic problems*, SIAM J. Sci. Comput., 35 (2013), pp. A2393-A2415.

[17] M. J. Gander and M. Petcu, *Analysis of a modified parareal algorithm for second-order ordinary differential equations*, AIP Conference Proceedings, 936(1) (2007), pp. 233-236.

[18] M. J. Gander and M. Petcu, *Analysis of a Krylov subspace enhanced parareal algorithm for linear problems*, ESAIM: Proceedings, 25 (2008), pp. 114-129.

[19] M. J. Gander and S. Vandewalle, *Analysis of the parareal time-parallel time-integration method*, SIAM J. Sci. Comput., 29 (2007), pp. 556-578.

[20] M. J. Gander, *50 years of time parallel time integration*, in "Multiple Shooting and Time Domain Decomposition", T. Carraro, M. Geiger, S. Körkel, R. Rannacher, editors, Springer Verlag, pp. 69-114, 2015,

[21] 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.

[22] T. Haut and B. Wingate, *An asymptotic parallel-in-time method for highly oscillatory PDEs*, SIAM J. Sci. Comput., 36 (2014), pp. A693-A713.

[23] E. Hairer and G. Wanner, *Solving ordinary differential equations II: Stiff and differential-algebraic problems* (2nd ed.), Berlin, New York: Springer-Verlag, 1996.

[24] M. A. Inda and R. H. Bisseling, *A simple and efficient parallel FFT algorithm using the BSP model*, Parallel Computing, 27 (2001), pp. 1847-1878.

[25] F. Legoll, T. Lelièvre, and G. Samaey, *A micro-macro parareal algorithm: application to singularly perturbed ordinary differential equations*, SIAM J. Sci. Comput., 35 (2013), pp. A1951-A1986.

[26] J.-L. Lions, Y. Maday, and G. Turinici, *A "parareal" in time discretization of PDE's*, C. R. Acad. Sci. Paris Sér. I Math., 332 (2001), pp. 661-668.

[27] A. Kreienbuehl, A. Naegel, D. Ruprecht, R. Speck, G. Wittum, and R. Krause, *Numerical simulation of skin transport using Parareal*, Comput. Visual Sci., 17 (2015), pp. 99-108.

[28] T. R. Mathew, M. Sarkis, and C. E. Schaerer, *Analysis of block parareal preconditioners for parabolic optimal controal problems*, SIAM J. Sci. Comput., 32 (2010), pp. 1180-1200.

[29] Y. Maday and E. M. Rønquist, *Parallelization in time through tensorproduct space-time solvers*, C. R. Math. Acad. Sci. Paris, 346 (2008), pp. 113-118.

[30] 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., 40 (2018), pp. A1012-A1033.

[31] A. S.Nielsen, G. Brunner, and J. S. Hesthaven, *Communication-aware adaptive Parareal with application to a nonlinear hyperbolic system of partial differential equations*, J. Comput. Phys., 371 (2018), pp. 483-505.

[32] M. Neum uller and I. Smears, *Time-parallel iterative solvers for parabolic evolution equations*, SIAM J. Sci. Comput., 41 (2019), pp. C28-C51.

[33] J. M. Reynolds-Barredo, D. E. Newman, R. Sanchez, D. Samaddar, L. A. Berry, and W. R. Elwasif, *Mechanisms for the convergence of time-parallelized, parareal turbulent plasma simulations*, J. Comput. Phys., 231 (2012), pp. 7851-7867.

[34] J. M. Reynolds-Barredo, D. E. Newman, and R. Sanchez, *An analytic model for the convergence of turbulent simulations time-parallelized via the parareal algorithm*, J. Comput. Phys., 255 (2013), pp. 293-315.

[35] D. Ruprecht and R. Krause, *Explicit parallel-in-time integration of a linear acoustic-advection system*, Comput. Fluids, 59 (2012), pp. 72-83.
[36] D. Ruprecht, *Wave propagation characteristics of Parareal*, Comput. Visual Sci., 19 (2018), pp. 1-17.
[37] J. Steiner, D. Ruprecht, R. Speck, and R. Krause, *Convergence of parareal for the Navier-Stokes equations depending on the Reynolds number*, Lect. Notes Comput. Sci. Eng., 103 (2015), pp. 195-202.
[38] W. Tian, H. Zhou, and W. Deng, *A class of second order difference approximations for solving space fractional diffusion equations*, Math. Comput., 84 (2015), pp. 1703-1727.
[39] S. L. Wu and T. Zhou, *Convergence analysis for three parareal solvers*, SIAM J. Sci. Comput., 37 (2015), pp. A970-A992.
[40] S. L. Wu, *Toward parallel coarse grid correction for the parareal algorithm*, SIAM J. Sci. Comput., 40 (2018), pp. A1446-A1472.
[41] S. L. Wu, H. Zhang and T. Zhou, *A diagonalization-based multi-grid method for time-periodic fractional diffusion equations*, Numer. Lin. Algebra Appl., 25 (2018), e2178 (DOI: 10.1002/nla.2178).
[42] S. L. Wu and T. Zhou, *Acceleration of the two-level MGRIT algorithm via the diagonalization technique*, SIAM J. Sci. Comput., 41 (2019), pp. A3421-A3448.
[43] Q. Xu, J. S. Hesthaven, and F. Chen, *A parareal method for time-fractional differential equations*, J. Comput. Phys., 293 (2015), pp. 173-183.

7. Appendix. In this appendix, we address how to generalize the diagonalization described in Section 2.2 to an s-stage IRK method. Suppose the s-stage IRK method is specified by the Butcher tableau

$$\begin{array}{c|c}\boldsymbol{\gamma} & \boldsymbol{\Theta}\\ \hline & \boldsymbol{\omega}\end{array} = \begin{array}{c|cccc}\gamma_1 & \theta_{1,1} & \theta_{1,2} & \dots & \theta_{1,s}\\ \gamma_2 & \theta_{2,1} & \theta_{2,2} & \dots & \theta_{2,s}\\ \vdots & \vdots & \vdots & \dots & \vdots\\ \gamma_s & \theta_{s,1} & \theta_{s,2} & \dots & \theta_{s,s}\\ \hline & \omega_1 & \omega_2 & \dots & \omega_s\end{array}. \tag{7.1}$$

Then, for $\mathcal{F}^*_{\delta t}(\alpha, T_n, u_n^k)$ we only need to replace the middle step in (1.6b) by

$$\begin{aligned}&\begin{bmatrix}r_1\\ \vdots\\ r_s\end{bmatrix} + \delta t(\boldsymbol{\Theta}\otimes I_x)\begin{bmatrix}f(t_{n,j}+\gamma_1\delta t, z_j+r_1)\\ \vdots\\ f(t_{n,j}+\gamma_s\delta t, z_j+r_s)\end{bmatrix} = 0,\\ &\frac{z_{j+1}-z_j}{\delta t} = \mathbf{w}R(z_j),\ \text{with}\ R(z_j) = (r_1^\top, r_2^\top, \dots, r_s^\top)^\top \in \mathbb{R}^{ms},\end{aligned} \tag{7.2}$$

where $z_0 = \alpha z_J + (1-\alpha)u_n^k \in \mathbb{R}^m$, $\mathbf{w} = \frac{1}{\delta t}(\boldsymbol{\omega}\boldsymbol{\Theta}^{-1})\otimes I_x \in \mathbb{R}^{m\times ms}$, $j = 0, \dots, J-1$. The details for this representation of the s-stage IRK method can be found in [23, pp.118-119]. Here, we regard R as a vector function of z_j. We have

$$\frac{1}{\delta t}(C_\alpha\otimes I_x)Z = \mathbf{R}(Z) + b_n^k,\ \mathbf{R}(Z) := \begin{bmatrix}\mathbf{w}R(\alpha z_J + (1-\alpha)u_n^k)\\ \mathbf{w}R(z_1)\\ \vdots\\ \mathbf{w}R(z_{J-1}),\end{bmatrix} \in \mathbb{R}^{Jm},$$

where $Z=(z_1^\top, z_2^\top, \dots, z_J^\top)^\top \in \mathbb{R}^{Jm}$ and C_α and b_n^k are given by (2.4) and (2.10). Applying Newton's iteration to this nonlinear system gives

$$\begin{aligned}&\left(\frac{1}{\delta t}C_\alpha\otimes I_x-\begin{bmatrix}0 & & & & \alpha\mathbf{w}\nabla R_J^{(l)}\\ \mathbf{w}\nabla R_1^{(l)} & 0 & & & \\ & \mathbf{w}\nabla R_2^{(l)} & 0 & & \\ & & \ddots & \ddots & \\ & & & \mathbf{w}\nabla R_{J-1}^{(l)} & 0\end{bmatrix}\right)Z^{(l+1)}=\\ &b_n^k+\mathbf{R}(Z^{(l)})-\begin{bmatrix}0 & & & & \alpha\mathbf{w}\nabla R_J^{(l)}\\ \mathbf{w}\nabla R_1^{(l)} & 0 & & & \\ & \mathbf{w}\nabla R_2^{(l)} & 0 & & \\ & & \ddots & \ddots & \\ & & & \mathbf{w}\nabla R_{J-1}^{(l)} & 0\end{bmatrix}Z^{(l)},\end{aligned}\tag{7.3}$$

where $\nabla R_j^{(l)}=\nabla R(z_j^{(l)})\in\mathbb{R}^{ms\times m}$ for $j=1,2,\dots,J-1$ and $\nabla R_J^{(l)}=\nabla R(\alpha z_J^{(l)}+(1-\alpha)u_n^k)\in\mathbb{R}^{ms\times m}$. Next, similar to the Backward Euler method studied in Section 2.2, we replace all the Jacobian matrices $\nabla R_j^{(l)}$ by the averaged one $\overline{\nabla R}^{(l)}:=\frac{\sum_{j=1}^J\nabla R_j^{(l)}}{J}\in\mathbb{R}^{ms\times m}$. Similar to (2.13), this leads to the *simplified* Newton iteration

$$\left(\frac{1}{\delta t}C_\alpha\otimes I_x-\widetilde{C}_\alpha\otimes\left(\mathbf{w}\overline{\nabla R}^{(l)}\right)\right)Z^{(l+1)}=b_n^k+\mathbf{R}(Z^{(l)})-\widetilde{C}_\alpha\otimes\left(\mathbf{w}\overline{\nabla R}^{(l)}\right)Z^{(l)},\tag{7.4}$$

where $\widetilde{C}_\alpha=I_J-C_\alpha$. Since both C_α and $\widetilde{C}_\alpha$ are α-circulant matrices, we can solve $Z^{(l+1)}$ from (7.4) via the diagonalization technique as in Section 2.2.In particular, the major computation for getting $Z^{(l+1)}$ is to solve the following J independent linear problems

$$\left(\frac{1}{\delta t}\lambda_j I_x-\tilde{\lambda}_j\left(\mathbf{w}\overline{\nabla R}^{(l)}\right)\right)q_j=p_j,\ j=1,2,\dots,J,\tag{7.5}$$

where $\{\lambda_j\}$ and $\{\tilde{\lambda}_j\}$ are the eigenvalues of the α-circulant matrices C_α and $\widetilde{C}_\alpha$. Note that $\mathbf{w}\overline{\nabla R}^{(l)}\in\mathbb{R}^{m\times m}$, and therefore the dimensions of the linear systems in (7.5) are $m\times m$ only, s times smaller than the dimension of the IRK system (7.2), which is $sm\times sm$.

To implement (7.4), with $Z^{(l)}=\left((z_1^{(l)})^\top,(z_2^{(l)})^\top,\dots,(z_J^{(l)})^\top\right)^\top$ obtained from the previous iteration we need to compute the two sets of quantities

$$\begin{aligned}&\left\{R(z_1^{(l)}),\dots,R(z_{J-1}^{(l)}),R(\alpha z_J^{(l)}+(1-\alpha)u_n^k)\right\},\\ &\left\{\nabla R(z_1^{(l)}),\dots,\nabla R(z_{J-1}^{(l)}),\nabla R(\alpha z_J^{(l)}+(1-\alpha)u_n^k)\right\}.\end{aligned}\tag{7.6}$$

The first set of quantities are components of $\mathbf{R}(Z^{(l)})$ and the second set of quantities are components for getting the averaged Jacobian matrix $\overline{\nabla R}^{(l)}$. We can compute $R(z_j^{(l)})\in\mathbb{R}^{ms}$ and $\nabla R(z_j^{(l)})\in\mathbb{R}^{ms\times m}$ using classical techniques for solving the IRK system (7.2) with $z_j=z_j^{(l)}$. For example, an application of Newton's method gives

$$\begin{aligned}&M_j^{(l,v)}\Delta R_j^{(l,v)}=-R_j^{(l,v)}-\delta t(\mathbf{\Theta}\otimes I_x)\begin{bmatrix}f(t_{n,j}+\gamma_1\delta t, z_j^{(l)}+r_1^{(l,v)})\\ \vdots\\ f(t_{n,j}+\gamma_s\delta t, z_j^{(l)}+r_s^{(l,v)})\end{bmatrix},\\ &R_j^{(l,v+1)}:=R_j^{(l,v)}+\Delta R_j^{(l,v)},\end{aligned}\tag{7.7}$$

where $R_j^{(l,v)} = ((r_1^{(l,v)})^\top, \ldots, (r_s^{(l,v)})^\top)^\top$ and the Jacobian matrix $M_j^{(l,v)}$ is given by

$$
\begin{aligned}
M_j^{(l,v)} &= I_s \otimes I_x + \delta t(\boldsymbol{\Theta} \otimes I_x) \begin{bmatrix} \nabla f_{j,1}^{(l,v)} & & \\ & \ddots & \\ & & \nabla f_{j,s}^{(l,v)} \end{bmatrix}, \\
\nabla f_{j,p}^{(l,v)} &:= \nabla f(t_{n,j} + \gamma_p \delta t, z_j^{(l)} + r_p^{(l,v)}) \in \mathbb{R}^{m\times m},\ p = 1, 2, \ldots, s.
\end{aligned}
\tag{7.8}
$$

The converged solution will give our desired $R(z_j^{(l)})$, i.e., after $v_{\max}$ iterations we get $R_j^{(l,v_{\max})} \approx R(z_j^{(l)})$, where $v_{\max}$ is specified by some given tolerance.

After obtaining $R_j^{(l,v_{\max})} = ((r_1^{(l,v_{\max})})^\top, \ldots, (r_s^{(l,v_{\max})})^\top)^\top$ from Newton's iterations (7.7), we can fix the matrix $\nabla R(z_j^{(l,v_{\max})}) \approx \nabla R(z_j^{(l)})$ as follows: from the IRK system in (7.2), by using the *implicit function theorem* it holds that

$$
\begin{bmatrix} \mathrm{d}r_1 \\ \vdots \\ \mathrm{d}r_s \end{bmatrix} + \delta t(\boldsymbol{\Theta} \otimes I_x) \begin{bmatrix} \nabla f(t_{n,j} + \gamma_1 \delta t, z_j + r_1)(\mathrm{d}z_j + \mathrm{d}r_1) \\ \vdots \\ \nabla f(t_{n,j} + \gamma_s \delta t, z_j + r_s)(\mathrm{d}z_j + \mathrm{d}r_s) \end{bmatrix} = 0,
$$

where 'd' denotes the differential operator. Hence, with $z_j = z_j^{(l)}$ the last iterate of (7.7) satisfies

$$
M_j^{(l,v_{\max})} \begin{bmatrix} \frac{\mathrm{d}r_1^{(l,v_{\max})}}{\mathrm{d}z_j} \\ \vdots \\ \frac{\mathrm{d}r_s^{(l,v_{\max})}}{\mathrm{d}z_j} \end{bmatrix} = -\delta t(\boldsymbol{\Theta} \otimes I_x) \begin{bmatrix} \nabla f_{j,1}^{(l,v_{\max})} \\ \vdots \\ \nabla f_{j,s}^{(l,v_{\max})} \end{bmatrix}, \tag{7.9}
$$

where $\nabla f_{j,p}^{(l,v_{\max})}$ and $M_j^{(l,v_{\max})}$ are the matrices given by (7.8) with $v = v_{\max}$. For the last iteration of (7.7), the computation of the increment $\Delta R_j^{(l,v_{\max})}$ uses a technique to invert the matrix $M_j^{(l,v_{\max})}$ and this same technique (e.g., the LU decomposition of $M_j^{(l,v_{\max})}$) can be reused to perform step (7.9).

From the above description, we see that the computational cost of the diagonalization technique at each time point is slightly larger than that of a single IRK system: the former is approximately $1 + \frac{1}{v_{\max}} + \frac{1}{s v_{\max}}$ times larger than the latter, where the quantity $\frac{1}{v_{\max}}$ reflects that the computation of $\nabla R(z_j^l)$ is only carried out at the last iteration of Newton's method applied to the IRK system, and the quantity $\frac{1}{s v_{\max}}$ is due to the fact that the dimension of the linear system in (7.5) is s times smaller than that of (7.9).

Moreover, if a diagonal IRK method is used, i.e., the matrix $\boldsymbol{\Theta}$ is a lower triangular matrix, $M_j^{(l,v)}$ is a block lower triangular matrix and we can solve each sub-vector of $\Delta R_j^{(l,v)}$ from (7.7) one-by-one through a forward substitution and there is no need to do matrix inversion with the full matrix $\boldsymbol{\Theta}$ at once. This holds for (7.9) as well.