Parareal algorithm via Chebyshev-Gauss spectral collocation method

Authors: Zhou, Quan, Liu, Yicheng, Wu Shu-Lin, Wu Shu-Lin

Date: 2023-04-20T00:00:00+00:00

Abstract

We present the Parareal-CG algorithm for time-dependent differential equations in this work. The algorithm is a parallel in time iteration algorithm utilizes Chebyshev-Gauss spectral collocation method for fine propagator F and backward Euler method for coarse propagator G. As far as we know, this is the first time that the spectral method used as the F propagator of the parareal algorithm. By constructing the stable function of the Chebyshev-Gauss spectral collocation method for the symmetric positive definite (SPD) problem, we find out that the Parareal-CG algorithm and the Parareal-TR algorithm, whose F propagator is chosen to be a trapezoidal ruler, converge similarly, i.e., the Parareal-CG algorithm converge as fast as Parareal-Euler algorithm with sufficient Chebyhsev-Gauss points in every coarse grid. Numerical examples including ordinary differential equations and time-dependent partial differential equations are given to illustrate the high efficiency and accuracy of the proposed algorithm.

Full Text

Preamble

Parareal Algorithm via Chebyshev-Gauss Spectral Collocation Method

Quan Zhou∗, Yicheng Liu†, Shu-Lin Wu‡

Abstract. We present the Parareal-CG algorithm for time-dependent differential equations in this work. The algorithm is a parallel-in-time iteration method that utilizes the Chebyshev-Gauss spectral collocation method for the fine propagator F and the backward Euler method for the coarse propagator G. To our knowledge, this is the first time that a spectral method has been used as the F propagator in the parareal algorithm. By constructing the stability function of the Chebyshev-Gauss spectral collocation method for symmetric positive definite (SPD) problems, we find that the Parareal-CG algorithm and the Parareal-TR algorithm, whose F propagator is chosen to be the trapezoidal rule, converge similarly. That is, the Parareal-CG algorithm converges as fast as the Parareal-Euler algorithm with sufficient Chebyshev-Gauss points in every coarse grid. Numerical examples including ordinary differential equations and time-dependent partial differential equations are given to illustrate the high efficiency and accuracy of the proposed algorithm.

Key words. Parareal algorithm, Chebyshev-Gauss spectral collocation method, nonlinear ODEs, SPD problems, time-dependent PDEs.

MSC classification. 65L05, 65L20, 65L60, 68Q60.

Introduction

We consider utilizing the parareal algorithm for initial-value problems of the form:

$$u'(t) = f(t, u), \quad u(0) = u_0, \quad t \in [0, T],$$

where $f : (0, T) \times \mathbb{R}^m \to \mathbb{R}^m$ and $u_0 \in \mathbb{R}^m$. Parareal is a well-studied parallel-in-time algorithm developed by Lions, Maday, and Turinici in 2001 [?]. The algorithm obtains the solution in a limited number of predictor-corrector iterations utilizing random initial values at each temporal subinterval, stopping when a tolerance is reached. The global error produced by this iterative method is equivalent to that obtained by the serial fine propagator. Due to advantages of the parareal algorithm including but not limited to efficiency and convergence, many relevant methods have emerged in recent years [?, ?, ?, ?, ?]. Together with these methods, the parareal algorithm has been applied in various fields of research, including optimal control problems [?, ?, ?, ?], wave equations [?, ?], stochastic differential equations [?, ?, ?], Hamiltonian systems [?, ?], incompressible flows [?, ?], heat equations [?, ?, ?], algebraic equations [?, ?], molecular dynamics [?, ?], and other partial differential equations [?, ?, ?].

The parareal algorithm combines two numerical methods: the coarse propagator G and fine propagator F, associated with large time step ΔT and small time step Δt, respectively. The ratio $J = \Delta T/\Delta t$ is assumed to be greater than 1. The G propagator is often chosen to be the backward Euler method, which is inexpensive and strongly stable, making it suitable for large time step ΔT computations. Additionally, numerical methods based on Taylor expansion and quadrature formulas, which are much cheaper, are frequently chosen as F propagators, and the convergence of various F propagators has been thoroughly investigated for symmetric positive definite (SPD) problems:

$$u'(t) + Au(t) = g(t), \quad A \in \mathbb{R}^{m\times m} \text{ symmetric positive definite (SPD).}$$

Gander and Vandewalle [?] proposed the convergence theorem of the parareal algorithm and illustrated numerically that the Parareal-Euler algorithm using backward Euler for F converges rapidly. Based on this, Mathew, Sarkis, and Schaerer [?] proved theoretically that the parareal-Euler algorithm converges robustly with convergence factor 0.298 for $J \geq 2$. Wu [?] demonstrated that robust convergence also holds when the F propagator is chosen to be the second-order diagonal implicit Runge-Kutta (DIRK2) method and TR/BDF2 method (the ode23tb solver for ODEs in MATLAB). For $J \geq 2$, the convergence factors of Parareal-2sDIRK and Parareal-TR/BDF2 are 0.316 and 0.333, respectively. Wu and Zhou [?] also showed analysis for the F propagator using the third-order diagonal implicit Runge-Kutta method with convergence factor 0.333 for $J \geq 4$; they also proved that for the trapezoidal formula and fourth-order Gauss-Runge-Kutta integrator chosen as F propagator, there exists a $J^*_{\min}$ depending on both the spectral radius of A and the step size Δt which makes the convergence factor 0.333 for $J \geq J^*_{\min}$. Recently, Yang, Yuan, and Zhou [?] gave a more general result stating that if the F propagator is strongly stable single-step integrators, there must exist a positive $J^*_{\min}$ (independent of step sizes ΔT, Δt, terminal time T, problem data u_0 and f, as well as the spectral radius of A) such that the parareal algorithm converges linearly with convergence factor close to 0.3 for all $J \geq J^*_{\min}$.

In this work, we choose the Chebyshev-Gauss spectral collocation method [?] for the fine propagator and present the Parareal-CG algorithm. The Chebyshev-Gauss spectral collocation method is an overall iteration method with M Chebyshev-Gauss points in the computation interval, combining the advantages of spectral accuracy and computational efficiency. Also, it is unnecessary to solve implicit equations for every fine time step Δt as in classical methods via Taylor expansion mentioned in [?, ?, ?, ?]. We briefly introduce spectral collocation methods for ODEs. Clenshaw and Norton first presented the Chebyshev-Picard method for solving nonlinear ordinary differential equations in 1963, in which the nonlinear term was approximated by Chebyshev series, so that the method was collocated at Chebyshev-Gauss-Lobatto points and implemented by Picard iteration. Later, a matrix-vector form of the method was introduced by Feagin and Nacozy [?], greatly increasing computational efficiency. Yang and Wang [?] proposed the Chebyshev-Gauss spectral collocation method via Chebyshev-Gauss points for ODEs in a single interval and analyzed convergence by the hp version. The method demonstrated significant advantages in astrodynamics simulations [?, ?, ?, ?] due to its spectral accuracy and computational efficiency. Additionally, some relevant spectral collocation methods for solving ODEs have also been proposed in recent years [?, ?, ?].

We present the matrix-vector form of the Chebyshev-Gauss spectral collocation method using the approach in [?]. Based on this, we construct its stability function and illustrate the convergence of the Parareal-CG method numerically. We find there exists an $M^*_{\min}$ depending on the spectral radius of A and the step size ΔT which makes the convergence factor 0.333 for $M \geq M^*_{\min}$. The proposed method has the same convergence as Parareal-TR and Parareal-Gauss4 methods.

The rest of the paper is organized as follows: in Section 2 we recall the Chebyshev-Gauss spectral collocation method and its convergence theorem. The Parareal-CG algorithm is proposed in Section 3. We provide the stability function of the Chebyshev-Gauss spectral collocation method and observe the convergence of the Parareal-CG algorithm in Section 4. Several numerical experiments are carried out in Section 5 to demonstrate the high accuracy and convergence of the proposed method. We finally give some conclusions in Section 6.

2 Revisit of Chebyshev-Gauss Spectral Collocation Method

In this section, we revisit the Chebyshev-Gauss spectral collocation method proposed by Yang and Wang [?] in a general interval $[a, b]$, $(b > a > 0)$ with the initial condition $u(a) = u_a$.

Let $T_l(\tau) = \cos(l \arccos(\tau))$ be the standard Chebyshev polynomials of degree l, $(l = 0, 1, \cdots)$ with $\tau \in [-1, 1]$. Then by using the affine transformation we can define the shifted Chebyshev polynomials:

$$\tilde{T}_l(t) = T_l\left(\frac{2(t-a)}{b-a}\right), \quad t \in [a, b], \quad l = 0, 1, \cdots$$

According to the definition, one can obtain the following shifted Chebyshev derivative relationship directly:

$$\tilde{T}_0'(t) = 0, \quad \tilde{T}_1'(t) = \frac{2}{b-a}\tilde{T}_0(t), \quad \tilde{T}_{l+1}'(t) = \frac{2(l+1)}{b-a}\tilde{T}_l(t) + \frac{l-1}{l+1}\tilde{T}_{l-1}'(t), \quad l \geq 2.$$

Let τ_m denote the standard Chebyshev-Gauss (CG) points in $(-1, 1)$:

$$\tau_m = -\cos\frac{(2m+1)\pi}{2M+2}, \quad m = 0, 1, \cdots, M,$$

the corresponding shifted Chebyshev-Gauss (CG) points t_m have the form:

$$t_m = \frac{b-a}{2}\tau_m + \frac{a+b}{2}, \quad m = 0, 1, \cdots, M,$$

which are the zeros of $\tilde{T}_{M+1}(t)$.

Denote $P_{M+1}(a, b)$ as the set of polynomials of degree at most $M + 1$ in (a, b). The Chebyshev-Gauss spectral collocation method seeks $u_M(t) \in P_{M+1}(a, b)$ defined by:

$$u_M(t) = \sum_{m=0}^{M+1} \hat{u}_m \tilde{T}_m(t),$$

such that:

$$u_M'(t_m) = I_M f(t_m, u_M(t_m)), \quad u_M(a) = u_a, \quad m = 0, 1, \cdots, M,$$

where $I_M f(t,u(t)) : C(a,b) \to P_M(a,b)$ is the Chebyshev interpolation of $f(t,u(t))$ defined by:

$$I_M f(t,u_M(t)) = \sum_{m=0}^{M} \hat{f}_m \tilde{T}_m(t),$$

the coefficients $\{\hat{f}_m\}_{m=0}^{M}$ are determined by the forward discrete Chebyshev transform:

$$\hat{f}_m = \frac{c_m}{M+1} \sum_{l=0}^{M} f(t_l, u_M(t_l)) \tilde{T}_m(t_l),$$

where $c_0 = 2$, $c_m = 1$ for $m \geq 1$. Then by the shifted Chebyshev derivative relationship, we can derive the coefficients $\{\hat{u}_m\}_{m=0}^{M+1}$:

$$\hat{u}_{M+1} = \frac{b-a}{4(M+1)} \hat{f}_M, \quad \hat{u}_M = \frac{b-a}{2(M+1)} \hat{f}_{M-1}, \quad \hat{u}_m = \frac{b-a}{2m} (c_{m-1}\hat{f}_{m-1} - \hat{f}_{m+1}), \quad \hat{u}_0 = u_a - \sum_{k=1}^{M+1} (-1)^k \hat{u}_k.$$

Yang and Wang [**?**] presented the error estimate for the hp-version of the single interval Chebyshev-Gauss spectral collocation method. Before introducing the theorem, we first present some notations used throughout the error estimate: - $H^r_\omega(a,b)$, $(r \geq 0)$ denotes the weighted Sobolev space with weight function $\omega = (t-a)(b-t)$ in (a,b); in particular, $H^0_\omega(a,b) = L^2_\omega(a,b)$. - $\|v\|_\omega$ denotes the norm of the space $L^2_{\omega^{-1/2}}(a,b)$.

The error estimate theorem is stated as follows.

Theorem 2.1 Assume that $f(t,z)$ fulfills the Lipschitz condition:

$$|f(z_1,t) - f(z_2,t)| \leq L|z_1 - z_2|, \quad L > 0,$$

and $0 < L(b-a) < \beta < \frac{1}{4}$ holds (where β is a certain constant). Then for any $u \in H^r_{\omega^{r-3/2}}(a,b)$ with integers $2 \leq r \leq N+1$, we have:

$$\|u-u_M\|^2_{L^2(a,b)} \leq (b-a)\|u-u_M\|^2_\omega \leq C_\beta (b-a)^3 M^{4-2r} \int_a^b \omega^{r-3/2}(t) \left(\frac{d^r}{dt^r}u(t)\right)^2 dt,$$

$$|u(b) - u_M(b)| \leq C_\beta (b-a)^2 M^{2-r} \left(\int_a^b \omega^{r-3/2}(t) \left(\frac{d^r}{dt^r}u(t)\right)^2 dt\right)^{1/2},$$

where C_β is a positive constant depending only on β.

In actual computation, we use a Picard iteration procedure to compute the coefficients $\{\hat{u}_m\}_{m=0}^{M}$. The approach is simple to implement, especially for complex nonlinear problems. The p-th $(p = 1, 2, \cdots)$ Picard iteration form of the method is:

$$u_M^{p+1}(t_m) = I_M f(t_m, u_M^p(t_m)).$$

If the equation satisfies the convergence condition in Theorem 2.1, the iteration solution $u_M^p(t)$ will converge to the numerical solution $u_M(t)$ with sufficiently large p, and the convergence is of order one. That is, there exists a constant $0 < C_p < 1$ such that $\|e^{p+1}\|_\infty \le C_p\|e^p\|_\infty$ with the definition $e^p := u_M^p(t) - u_M(t)$.

Algorithm 1 Chebyshev-Gauss Spectral Collocation Algorithm

Input: Provide the initial guess of $\{u_M^0(t_m)\}_{m=0}^M$, the tolerance ε.

For $p = 0, 1, \cdots$: - **Step 1.** Evaluate the values of $\{f(t_m, u_M^p(t_m))\}_{m=0}^M$. - **Step 2.** Compute the coefficients $\{\hat{f}_m^p\}_{m=0}^M$ by (2.8). - **Step 3.** Compute the coefficients $\{\hat{u}_m^{p+1}\}_{m=0}^{M+1}$ by (2.9). - **Step 4.** Update the data of $\{u_M^{p+1}(t_m)\}_{m=0}^M$ by (2.5). - **Step 5.** If the iteration error satisfies the stopping criterion $\|u_M^{p+1}(t) - u_M^p(t)\|_\infty < \varepsilon$, terminate the iteration; otherwise go back to Step 1. - **Step 6.** Compute $u_M^{p+1}(b) = \sum_{m=0}^{M+1} \hat{u}_m^{p+1}\tilde{T}_m(b)$.

We designate F_{CG} as the numerical propagator defined by the Chebyshev-Gauss spectral collocation method. The numerical output of Algorithm 1 with M points in the interval $[a, b]$ is represented as $F_{CG}(a, u_a, M, b - a)$. That is:

$$u_M(b) = F_{CG}(a, u_a, M, b - a).$$

3 Parareal-CG Algorithm

The parareal algorithm introduced by Gander and Vandewalle [?] is revisited in this section, using the Chebyshev-Gauss spectral collocation method as the fine propagator. First, we divide the whole time interval $[0, T]$ uniformly by $0 = T_0 < T_1 < \cdots < T_N = T$ and define $\Delta T = T/N$. Second, we divide each (T_n, T_{n+1}) by $M(\ge 2)$ shifted Chebyshev-Gauss points (2.4). Then, the low-order and inexpensive numerical method G propagator is applied to the coarse time grids, while the Chebyshev-Gauss spectral collocation method F_{CG} propagator having spectral accuracy is utilized in the fine grids. The time-sequential and time-parallel parts of the algorithm are denoted by the symbols (cid:9) and $\oplus$, respectively. The following is the parareal method employing F_{CG} as the fine propagator.

Algorithm 2 Parareal-CG Algorithm

(cid:9) **Initialization:** Compute sequentially $u_{n+1}^0 = G(T_n, u_n^0, \Delta T)$ with $u_0^0 = u_0$.

For $k = 0, 1, \cdots$: $\oplus$ **Step 1.** On each subinterval $[T_n, T_{n+1}]$, compute $\tilde{u}_{n+1}^k = F_{CG}(T_n, u_n^k, M, \Delta T)$. (cid:9) **Step 2.** Perform sequential corrections:

$$u_{n+1}^{k+1} = G(T_n, u_n^{k+1}, \Delta T) + \tilde{u}_{n+1}^k - G(T_n, u_n^k, \Delta T),$$

where u_{n+1}^{k+1} is the $(k + 1)$-th iteration value. (cid:9) **Step 3.** If $\{u_n^{k+1}\}_{n=1}^N$ satisfies the stopping criterion, terminate the iteration and output $\{u_n^{k+1}\}_{n=1}^N$; otherwise go back to Step 1.

Given that the implicit Euler method is L-stable and feasible for high coarse time step sizes ΔT required by the parareal algorithm, using it as the coarse propagator makes sense. Other reliable implicit numerical methods, such as implicit Runge-Kutta methods, are also available for use as the G propagator, although they are all significantly more costly than the implicit Euler method. In this paper, we apply the Chebyshev-Gauss spectral collocation method to the fine propagator, and the compact form of the Parareal-CG method may be expressed as:

$$u_{n+1}^{k+1} = G(T_n, u_n^{k+1}, \Delta T) + F_{CG}(T_n, u_n^k, M, \Delta T) - G(T_n, u_n^k, \Delta T).$$

For comparison, we also take into account other high-order and expensive numerical methods like the trapezoidal formula. Accordingly, each interval $[T_n, T_{n+1}]$ should be divided into $J(\geq 2)$ small time-intervals $[T_{n+j/J}, T_{n+(j+1)/J}]$, $j = 0, 1, \cdots, J-1$. We assume the intervals are of uniform size, therefore $\Delta t = \Delta T/J$, $T_{n+j} = T_n + j\Delta T/J$. The parareal algorithm can be derived by:

Algorithm 3 Parareal Algorithm

(cid:9) **Initialization:** Compute sequentially $u_{n+1}^0 = G(T_n, u_n^0, \Delta T)$ with $u_0^0 = u_0$.

For $k = 0, 1, \cdots$: $\oplus$ **Step 1.** On each subinterval $[T_n, T_{n+1}]$, compute $\tilde{u}_{n+j+1}^k = F(T_{n+j}, \tilde{u}_{n+j}^k, \Delta t)$ with initial value $\tilde{u}_n^k = u_n^k$ and $j = 0, 1, \cdots, J-1$. (cid:9) **Step 2.** Perform sequential corrections:

$$u_{n+1}^{k+1} = G(T_n, u_n^{k+1}, \Delta T) + \tilde{u}_{n+1}^k - G(T_n, u_n^k, \Delta T),$$

where u_{n+1}^{k+1} is the $(k+1)$-th iteration value. (cid:9) **Step 3.** If $\{u_n^{k+1}\}_{n=1}^N$ satisfies the stopping criterion, terminate the iteration and output $\{u_n^{k+1}\}_{n=1}^N$; otherwise go back to Step 1.

The compact form of the parareal algorithm can be derived by:

$$u_{n+1}^{k+1} = G(T_n, u_n^{k+1}, \Delta T) + F^J(T_n, u_n^k, \Delta t) - G(T_n, u_n^k, \Delta T),$$

where $F^J(T_n, u_n^k, \Delta t)$ stands for the result of running J steps of the fine propagator F with initial value u_n^k and the small step-size Δt.

4 Convergence Analysis

Based on the parareal convergence analysis given by Gander and Vandewalle [**?**], the convergence of the Parareal-CG method is analyzed by constructing the stability function of the Chebyshev-Gauss spectral collocation method in this section.

4.1 Stability Function of Chebyshev-Gauss Spectral Method

In order to obtain the convergence factor $K(z,M)$ of the Parareal-CG algorithm in Algorithm 2, in this subsection we present the stability function of the Chebyshev-Gauss spectral collocation method $R_{CG}(z,M)$.

Lemma 4.1 For given M and $z := \lambda\Delta T$, the stability function $R_{CG}(z,M)$ of the Chebyshev-Gauss spectral collocation method in $[T_n, T_{n+1}]$, $(n = 1, 2, \cdots, N)$ is:

$$R_{CG}(z,M) = T(I_2 - zC_\alpha(I_1 + zT_1C_\alpha)^{-1}T_1)E,$$

where T_1 and C_α are the coefficient matrices defined in (4.7) and (4.9), respectively; I_1 and I_2 are two identity matrices of size $(M+1)$ and $(M+2)$, respectively; $T = [1, 1, \cdots, 1]_{1\times(M+2)}$ and $E = [1, 0, \cdots, 0]^T_{(M+2)\times 1}$ are two constant vectors.

Proof. We first introduce the matrix-vector form of the Chebyshev-Gauss spectral collocation method to be able to express R_{CG} explicitly. Denote the evaluations of the approximated polynomial for the p-th $(p = 1, 2, \cdots)$ iteration $u^p_M(t)$ at the shifted Chebyshev-Gauss points $\{t_m\}^M_{m=0}$, $(n = 0, 1, \cdots, N-1)$ by:

$$u^p = [u^p_M(t_0), u^p_M(t_1), \cdots, u^p_M(t_M)]^T_{(M+1)\times 1}.$$

Equation (2.9) indicates that the vector u^p can be expressed as a consequence of the fact $T_l(t_m) = T_l(\tau_m)$, $(l = 1, 2, \cdots; m = 0, 1, \cdots, M)$ where T_1 is a coefficient matrix defined by:

$$u^p = T_1\hat{u}^p, \quad (T_1)_{(M+1)\times(M+2)} = \begin{bmatrix} T_0(\tau_0) & T_1(\tau_0) & \cdots & T_{M+1}(\tau_0) \\ T_0(\tau_1) & T_1(\tau_1) & \cdots & T_{M+1}(\tau_1) \\ \vdots & \vdots & \ddots & \vdots \\ T_0(\tau_M) & T_1(\tau_M) & \cdots & T_{M+1}(\tau_M) \end{bmatrix}.$$

Denote $\hat{f}(t_m, u^p_M(t_m)) = \hat{f}^p_m$ and suppose the initial value be $u(T_n) = u_{T_n}$. Then the coefficient vector $\hat{u}^p$ can be expressed by:

$$\hat{u}^p = \begin{bmatrix} u_{T_n} + \sum_{m=2}^{M+1}(-1)^{m-1}\hat{u}^p_m \\ \frac{\Delta T}{2}\hat{f}^{p-1}_0 \\ \frac{\Delta T}{4}(\hat{f}^{p-1}_1 - \hat{f}^{p-1}_3) \\ \vdots \\ \frac{\Delta T}{2(M-1)}(\hat{f}^{p-1}_{M-2} - \hat{f}^{p-1}_M) \\ \frac{\Delta T}{4(M+1)}\hat{f}^{p-1}_{M-1} \end{bmatrix} = U_0 + RS\hat{f}^p,$$

where R and S are coefficient matrices and the first line of S satisfies $s_m = (-1)^m\frac{1}{m+1}$ for $m = 2, 3, \cdots, M$.

U_0 and $\hat{f}^p$ can be defined by:

$$U_0 = [u_{T_n}, 0, 0, \cdots, 0]^T_{(M+2)\times 1}, \quad \hat{f}^p = [\hat{f}^{p-1}_0, \hat{f}^{p-1}_1, \cdots, \hat{f}^{p-1}_M]^T_{(M+1)\times 1}.$$

For simplicity, we set the values of the function $f(t, u^{p-1}(t))$ on the Chebyshev-Gauss points by the notation $f_m^{p-1} = f(t_m, u_M^{p-1}(t_m))$, $m = 0, 1, \cdots, M$. Then (2.8) can be expressed as:

$$\hat{f}^{p-1} = \frac{2}{M+1} \begin{bmatrix} \frac{1}{2} f_0^{p-1} T_0(\tau_0) + f_1^{p-1} T_0(\tau_1) + \cdots + f_M^{p-1} T_0(\tau_M) \\ f_0^{p-1} T_1(\tau_0) + f_1^{p-1} T_1(\tau_1) + \cdots + f_M^{p-1} T_1(\tau_M) \\ \vdots \\ f_0^{p-1} T_M(\tau_0) + f_1^{p-1} T_M(\tau_1) + \cdots + f_M^{p-1} T_M(\tau_M) \end{bmatrix} = V T_2 f^{p-1},$$

where V and T_2 are coefficient matrices and the vector f^{p-1} is defined by:

$$f^{p-1} = [f(t_0, u_M^{p-1}(t_0)), f(t_1, u_M^{p-1}(t_1)), \cdots, f(t_M, u_M^{p-1}(t_M))]^T_{(M+1)\times 1}.$$

To this end, given the initial guess u^0, the p-th matrix-vector iteration version of the Chebyshev-Gauss spectral collocation method for solving (2.6) is:

$$\hat{u}^p = U_0 - \Delta T C_\alpha f^{p-1}, \quad u^p = T_1 \hat{u}^p,$$

where $C_\alpha = RSVT_2$.

We develop the matrix-vector form of the method in order to acquire the stability function on the one hand, and to considerably improve computing efficiency so that it may be employed in our numerical experiments on the other. The matrix-vector form of the method improves computational efficiency sufficiently.

Following this, we establish the linear stability function for the Chebyshev-Gauss spectral collocation method. For this, we initially derive the special form of the method for solving the linear equation:

$$u^p = T_1 \hat{u}^p = T_1(U_0 - \Delta T C_\alpha f^{p-1}) = T_1 U_0 - \lambda \Delta T T_1 C_\alpha u^{p-1}.$$

By assuming $u^* = [u^*(t_0), u^*(t_1), \cdots, u^*(t_M)]^T$ to be the convergence solution after a sufficient number of iterations, we may rewrite (4.10) for given $z := \lambda \Delta T$ as:

$$u^* = T_1 U_0 - z T_1 C_\alpha u^* = (I_1 + z T_1 C_\alpha)^{-1} T_1 U_0,$$

where I_1 is an identity matrix of size $(M+1)$. Then we can derive the corresponding coefficient vector $\hat{u}^* = [u_0^*, u_1^*, \cdots, u_{M+1}^*]^T$ directly:

$$\hat{u}^* = U_0 - \Delta T C_\alpha(\lambda u^*) = U_0 - z C_\alpha (I_1 + z T_1 C_\alpha)^{-1} T_1 U_0.$$

Finally, the stability function of the Chebyshev-Gauss spectral collocation method $R_{CG}(z, M)$ could be acquired by the compression relations:

$$R_{CG}(z, M) = \frac{u^*(T_{n+1})}{u_{T_n}} = T \hat{u}^* = T(I_2 - z C_\alpha (I_1 + z T_1 C_\alpha)^{-1} T_1) E,$$

where I_2 is an $(M+2)$-dimensional identity matrix, T and E are two vectors defined by $T = [1, \cdots, 1, 1]_{1\times(M+2)}$ and $E = [1, 0, \cdots, 0]^T_{(M+2)\times 1}$.

We can obtain the stability functions R_{CG} particularly when $M = 0$ and $M = 1$:

$$R_{CG}(z, 0) = \frac{2 - z}{2 + z}, \quad R_{CG}(z, 1) = \frac{z^2 - 8z + 16}{z^2 + 8z + 16}.$$

For $M = 0, 1, 2, 4, 20$, we display $|R_{CG}(z, M)|$ as functions of $z_{\max} := \Delta T \lambda_{\max}$ in Figure 4.1, which shows that for various M we have $|R_{CG}(z)| \leq 1$. As a result, we can obtain the contraction factor of the Parareal-CG method and derive the convergence analysis.

4.2 Convergence Analysis of Parareal-CG Method

The following conclusion regarding the convergence of the Parareal-CG Algorithm on sufficiently long time intervals can be deduced directly from the analysis provided by Gander and Vandewalle [?] for the linear system of ODEs (1.2) with $A \in \mathbb{R}^{m \times m}$ a symmetric positive definite matrix (which therefore can be diagonalized and all eigenvalues are positive real numbers).

Theorem 4.1 ([17]) Let F_{CG} be the Chebyshev-Gauss spectral collocation propagator with stability function $R_{CG}(z)$ (4.13), and let $\sigma(A) = \{\lambda_1, \cdots, \lambda_m\}$ be the set of eigenvalues of matrix A in (1.2). Then the errors $\{e_n^k\}$ of the parareal-CG algorithm at the k-th iteration using the backward-Euler method as the coarse propagator satisfy:

$$\|Ve_n^k\|_\infty \leq \rho^k \sup_n \|Ve_0^k\|_\infty, \quad \rho = \max_{\lambda \in \sigma(A)} K(\Delta T \lambda, M),$$

where ρ is the convergence factor of the parareal algorithm, $k \geq 1$ is the iteration index, and $V \in \mathbb{R}^{m \times m}$ consists of the eigenvectors of A (i.e., $V^{-1}AV = \mathrm{diag}(\lambda_1, \cdots, \lambda_m)$). The argument K_{CG}, which is the convergence factor corresponding to a single eigenvalue (or "contraction factor" hereafter), is defined as:

$$K_{CG}(z, M) = \left| \frac{R_{CG}(z, M) - \frac{1}{1+z}}{1 - \frac{1}{1+z}} \right|.$$

The convergence theorem 4.1 states that the behavior of the Parareal-CG algorithm over a long time interval is determined by the convergence factor $\rho(M)$ defined by $\rho(M) := \max_{z \in [0, z_{\max}]} K(z, M)$ where $z_{\max} = \lambda_{\max} \Delta T$ and $\lambda_{\max}$ denotes the maximal eigenvalue (or an upper bound) of the coefficient matrix A in (1.2). We can infer from equation (4.16) that the smaller $\rho(M)$ is, the faster the algorithm converges. To maintain the convergence rate, the convergence factor $\rho(M)$ prefers to be around $\frac{1}{3}$.

The convergence factor of the Parareal-CG algorithm $K_{CG}(z, M)$ as functions of $z_{\max} := \Delta T \lambda_{\max}$ for various M are shown in Figure 4.2. We may infer from the behavior of $K_{CG}(z, M)$, which is similar to the convergence factor of Parareal-TR and Parareal-Gauss4 algorithms discussed in [?], that:

$$\lim_{z \to 0} K_{CG}(z) = 0, \quad \lim_{z \to \infty} K_{CG}(z) = 1.$$

This implies that the Parareal-CG algorithm cannot keep the convergence rate for arbitrarily large z; however, by choosing an appropriately large M, it is still possible to get a feasible parareal solver. In further detail, as M grows, the convergence factor $\rho(M) := \max_{z\in[0,z_{\max}]} K_{CG}(z, M)$ tends to be around $\frac{1}{3}$ for some given z.

The convergence analysis can be derived from the convergence theorem 4.1 of the Parareal-CG algorithm and the stability function (4.1) of the Chebyshev-Gauss spectral collocation method.

Theorem 4.2 Given a fixed constant $z_{\max} = \Delta T \lambda_{\max} > 0$, there exists some positive integer $M^*_{\min}$ such that:

$$\max_{z\in[0,z_{\max}]} K_{CG}(z, M) \leq \frac{1}{3} \quad \text{if } M \geq M^*_{\min},$$

where K_{CG} is the contraction factor of the Parareal-CG algorithm defined by (4.17) with the stability function of the Chebyshev-Gauss spectral collocation method R_{CG} given in (4.13).

The lower bound $M^*_{\min}$ of the Parareal-CG algorithm is provided by:

$$M^*_{\min} = \begin{cases} 0 & \text{if } z_{\max} \leq z^*_0, \\ 1 & \text{if } 0 < z_{\max} \leq z^*_1, \\ M^*_{CG} & \text{otherwise,} \end{cases}$$

where $M^*_{CG} > 1$ depends on $z_{\max}$ and is the minimum positive integer which satisfies:

$$|R_{CG}(z_{\max}, M)| \leq \frac{3 + z_{\max}}{3(1 + z_{\max})}.$$

Moreover, the quantity z^*_0 is the unique positive root of $K_{CG}(z, 0) = \frac{1}{3}$, that is, $z^*_0 = 1$, and z^*_1 is the maximum positive root of $K_{CG}(z, 1) = 1$.

Proof. When $M = 0$, we can derive $R_{CG}(z, 0)$ and $K_{CG}(z, 0)$:

$$R_{CG}(z, 0) = \frac{2 - z}{2 + z}, \quad K_{CG}(z, 0) = \frac{2z}{(2 + z)(1 + z)} \quad (z > 0).$$

It is obvious that $K_{CG}(z, 0)$ is an increasing function with respect to $z > 0$ and $K_{CG}(z, 0) = \frac{1}{3}$ has the unique root $z^*_0 = 1$. That is to say:

$$\max_{z\in[0,z_{\max}]} K_{CG}(z, 0) \leq \frac{1}{3} \quad \text{for } z_{\max} \in (0, z^*_0], \quad \max_{z\in[0,z_{\max}]} K_{CG}(z, 0) \leq \frac{2z_{\max}}{(2 + z_{\max})(1 + z_{\max})} \quad \text{for } z_{\max} \in (z^*_0, \infty).$$

When $M = 1$, we can derive $R_{CG}(z, 1)$ and $K_{CG}(z, 1)$:

$$R_{CG}(z, 1) = \frac{(4 - z)^2}{(4 + z)^2}, \quad K_{CG}(z, 1) = \frac{|z^2 - 8z|}{(4 + z)^2(1 + z)} \quad (z > 0).$$

There are three roots z_{11}^*, z_{12}^*, and z_{13}^* of $K_{CG}(z,1) = \frac{1}{3}$, respectively, where $z_{12}^* = 2$ is also the maximum value point of $K_{CG}(z,1)$. Moreover, $K_{CG}(z,1)$ is an increasing function for $z \geq 8 + 6\sqrt{2}$. In conclusion, we may obtain:

$$\max_{z\in[0,z_{\max}]} K_{CG}(z,1) \leq \frac{1}{3} \quad \text{for } z_{\max} \in (0, z_1^*], \quad \max_{z\in[0,z_{\max}]} K_{CG}(z,1) \leq \frac{|z_{\max}^2 - 8z_{\max}|}{(4+z_{\max})^2(1+z_{\max})} \quad \text{for } z_{\max} \in (z_1^*, \infty).$$

When $M \geq 2$, we can infer from Figure 4.2 that there exists only one root $z^*(M) > z_1^*$ of $K_{CG}(z,M) = \frac{1}{3}$ and $K_{CG}(z,M)$ is an increasing function for $z \geq z^*(M)$. Then we have:

$$\max_{z\in[0,z_{\max}]} K_{CG}(z,M) = \frac{1}{3} \quad \text{for } z_{\max} \in (0, z^*(M)], \quad \max_{z\in[0,z_{\max}]} K_{CG}(z,M) = K_{CG}(z_{\max},M) \quad \text{for } z_{\max} \in (z^*(M),$$

Therefore, we can derive that Theorem 4.2 holds.

We can infer from Theorem 4.2 that M has a significant impact on the convergence rates because the lower bound $M_{\min}^*$ increases as $z_{\max} = \Delta T \lambda_{\max}$ grows. Using while loops in MATLAB, we can derive $M_{\min}^*$ for a given $z_{\max}$ since $M_{\min}^*$ is a natural number.

Remark 4.1 (Comparison to other Parareal algorithms) Researchers have proved the following proposition for the Parareal-Euler [**?**] and Parareal-TR/BDF2 [**?**] algorithms utilizing backward Euler and TR/BDF2 (i.e., ode23 solver for ODEs in MATLAB), respectively, as the F propagators in earlier papers:

$$\max_{z\in[0,z_{\max}]} K_{\text{Euler, TR/BDF2}}(z,J) < \frac{1}{3} \quad \forall z_{\max} > 0,$$

which implies that for all $z_{\max} > 0$, the convergence rates of these parareal algorithms satisfy $\rho(J) < \frac{1}{3}$. Unfortunately, not all methods yield such a uniform result. The following conclusion holds for the Parareal-TR and Parareal-Gauss4 algorithms [**?**] when utilizing the trapezoidal rule and fourth-order Gauss Runge-Kutta method as F-propagator:

$$\max_{z\in[0,z_{\max}]} K_{\text{TR, Gauss4}}(z,J) < \frac{1}{3} \quad \text{if } J \text{ is even and } J \geq J_{\min}^*.$$

This indicates that if the mesh ratio J is large enough, the two parareal algorithms will converge as fast as the Parareal-Euler method. Our Parareal-CG method behaves similarly to Parareal-TR and Parareal-Gauss4 algorithms.

For the explicit F-propagators forward Euler and fourth-order explicit Runge-Kutta algorithms used in the Parareal-fEuler and Parareal-4ERK algorithms, respectively, the convergence factor K begins to suddenly trend to infinity at some $z_{\max}$. Therefore, these parareal algorithms can be used to solve only a very limited number of equations:

$$K_{\text{fEuler, 4ERK}}(z,J) = \infty, \quad z > z^*.$$

The convergence factors of the three kinds of parareal algorithms are shown in Figure 4.4.

5 Numerical Experiment

In this section we verify the convergence and efficiency of the Parareal-CG algorithm. In all computations, the initial iteration for the parareal algorithm is chosen randomly and the iteration process stops when the following tolerance is obtained:

$$\|u_n^{k+1} - u_n^k\|_\infty \leq 10^{-10}.$$

Moreover, the absolute error and iteration error in all experiments are defined by:

$$\text{Absolute error} : \max_n \|u_n^{k+1} - u(T_n)\|_\infty,$$

$$\text{Iteration error} : \max_n \|u_n^{k+1} - u_n^k\|_\infty.$$

Example 5.1 (Near Earth satellite motion integration problem) We take a two-body motion in low Earth orbit (LEO) with just mutual gravitational attraction as the first example. The system of second-order three-dimensional equations is:

$$\begin{cases} x''(t) = -\frac{\mu}{r(t,x,y,z)^3}x(t), & t \in [0, 50], \\ y''(t) = -\frac{\mu}{r(t,x,y,z)^3}y(t), & t \in [0, 50], \\ z''(t) = -\frac{\mu}{r(t,x,y,z)^3}z(t), & t \in [0, 50], \end{cases}$$

where x, y and z are the three coordinates in some Earth-centered inertial reference frame; r is the distance between the two bodies defined by $r(t, x, y, z) = \sqrt{x(t)^2 + y(t)^2 + z(t)^2}$; and $\mu = 3.986 \times 10^5 \text{km}^3/\text{s}^2$ is the Earth gravitational constant. We formulate the equations as a first-order system of six-dimensional differential equations, where the solution is uniquely determined by the initial position and velocity:

$$[x(0), y(0), z(0)] = [464.856, 6667.880, 574.231] \text{ km},$$

$$[x'(0), y'(0), z'(0)] = [-2.8381188, -0.7871898, 7.0830275] \text{ km/s}.$$

We obtain the exact solution of equation (5.3) by solving the two-body Keplerian motion using the F and G approach in [**?**].

We compare the accuracy of Parareal-CG, Parareal-Euler, Parareal-TR/BDF2, and Parareal-Gauss4 algorithms revisited in Remark 4.1 for solving this example. In each coarse grid, we fix $\Delta T = 0.25$ with $M = 6$ for the Parareal-CG algorithm and $J = 6$ for other parareal algorithms. Then we compute the absolute errors and iteration errors of the parareal algorithms, shown in Figure 5.5. From the convergence behaviors in the figures, we can draw the following conclusions: - In comparison to Parareal-Euler and Parareal-TR, the Parareal-CG algorithm utilizes fewer iterations, while the Parareal-Gauss4 algorithm uses the same number of iterations as the Parareal-CG algorithm. - Using the same number of points $M = J = 6$ in every coarse grid, the Parareal-CG algorithm has higher accuracy than the other three methods.

Example 5.2 (Burgers' equation) Consider the 1D Burgers' equation with initial and boundary conditions:

$$\begin{cases} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, & (x,t) \in [0,2) \times [0,4], \\ u(x,0) = \frac{2\nu\pi\sin(\pi x)}{\alpha+\cos(\pi x)}, & x \in [0,2), \\ u(0,t) = u(2,t) = 0, & t \in [0,4], \end{cases}$$

In our computations, we choose $\alpha = 2$ and test $\nu = 0.05, 0.005$. The exact solution of the equation is:

$$u(x,t) = \frac{2\nu\pi\exp(-\pi^2\nu t)\sin(\pi x)}{\alpha + \exp(-\pi^2\nu t)\cos(\pi x)}.$$

We divide the spatial domain $x \in [0,2)$ into an N_x mesh uniformly with $\Delta x = 2/N_x$, giving $x_j = j\Delta x$, $j = 0, 1, \cdots, N_x - 1$. Applying the fourth-order compact finite difference scheme to approximate $\frac{\partial u}{\partial x}$ and $\frac{\partial^2 u}{\partial x^2}$, we have:

$$\frac{\partial u}{\partial x}(x_j,t) = \frac{1}{12\Delta x}(u(x_{j+2},t) + 4u(x_{j+1},t) - 4u(x_{j-1},t) - u(x_{j-2},t)),$$

$$\frac{\partial^2 u}{\partial x^2}(x_j,t) = \frac{1}{\Delta x^2}(u(x_{j+2},t) - 2u(x_{j+1},t) + u(x_j,t)).$$

Define the vectors:

$$u = [u(x_0,t), u(x_1,t), \cdots, u(x_{N_x-1},t)]^T, \quad \frac{\partial u}{\partial x} = \left[\frac{\partial u(x_0,t)}{\partial x}, \frac{\partial u(x_1,t)}{\partial x}, \cdots, \frac{\partial u(x_{N_x-1},t)}{\partial x}\right]^T.$$

Then combining the compact finite difference scheme with the periodic boundary condition, the Burgers' equation (5.4) has the following spatial semidiscrete scheme:

$$\dot{u} + A_1 u + u \cdot (A_2 u) = 0,$$

where the coefficient matrices A_1 and A_2 are defined accordingly.

The three figures in Figure 5.6 show the dependence of the convergence rate of the Parareal-CG algorithm on ΔT, Δx, and M, respectively. From the figure we can draw the following conclusions: - Each ν possesses robust convergence rate with respect to changes in ΔT. The convergence factor decreases as ΔT increases. - Each ν possesses robust convergence rate with respect to changes in Δx. The convergence factor increases as Δx varies from small to large since the eigenvalues of A_1 and A_2 increase as Δx reduces. - The convergence rate is insensitive to the choice of M with $\Delta T = 1/32$, which implies the high accuracy of the Chebyshev-Gauss spectral collocation method. - For all experiments, $\nu = 0.005$ needs fewer iterations than $\nu = 0.05$.

6 Conclusion

In this paper, we propose the Parareal-CG method for solving time-dependent differential equations, where the coarse propagator G is fixed as the backward Euler method and the fine propagator F is chosen to be the Chebyshev-Gauss spectral collocation method. The algorithm does have a convergence factor around 0.333, although the number of Chebyshev-Gauss points M needs to be somewhat large. The spectral radius of matrix A and ΔT provide the lower bound of M that guarantees such an expected convergence factor. Numerical experiments illustrate the accuracy and convergence of the presented algorithm.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable suggestions, which helped us improve this article greatly. This work was partially supported by the Postgraduate Scientific Research Innovation Project of Hunan Province (No. CX20210012).

References

[1] X. Bai, *Modified Chebyshev-Picard iteration methods for solution of initial value and boundary value problems* (Ph.D. dissertation), Texas A&M University, College Station, TX, 2010.

[2] L. Baffico, S. Bernard, Y. Maday, G. Turinici, and G. Zérah, Parallel-in-time molecular-dynamics simulations, *Phys. Rev. E* 66 (3) (2002) 057701.

[3] X. Bai, J. L. Junkins, Modified Chebyshev-Picard iteration methods for orbit propagation, *J. Astronaut. Sci.* 58 (4) (2011) 583–613.

[4] D. Q. Bui, C. Japhet, Y. Maday, and P. Omnes, Coupling parareal with optimized Schwarz waveform relaxation for parabolic problems, *SIAM J. Numer. Anal.* 60 (3) (2022) 913–939.

[5] C.-E. Brehier, X. Wang, On parareal algorithms for semilinear parabolic stochastic PDEs, *SIAM J. Numer. Anal.* 58 (1) (2020) 254–278.

[6] J. D. Cole, On a quasi-linear parabolic equation occurring in aerodynamics, *Quart. Appl. Math.* 9 (1951) 225–236.

[7] C. W. Clenshaw, H. J. Norton, The solution of nonlinear ordinary differential equations in Chebyshev series, *Comput. J.* 6 (1963/64) 88–92.

[8] S. Costanzo, T. Sayadi, M. F. de Pando, P. J. Schmid, and P. Frey, Parallel-in-time adjoint-based optimization-application to unsteady incompressible flows, *J. Comput. Phys.* 471 (2022) 111664.

[9] X. Dai, C. Bris, F. Legoll and Y. Maday, Symmetric parareal algorithms for Hamiltonian systems, *ESAIM Math. Model. Numer. Anal.* 47 (3) (2013) 717–742.

[10] F. Danieli, B. S. Southworth and A. J. Wathen, Space-time block preconditioning for incompressible flow, *SIAM J. Sci. Comput.* 44 (1) (2022) A337–A363.

[11] F. Danieli, A. Wathen, All-at-once solution of linear wave equations, *Numer. Linear Algebra Appl.* 28 (6) (2021) e2386.

[12] T. Feagin, P. Nacozy, Matrix formulation of the Picard method for parallel computation, *Celestial Mech.* 29 (2) (1983) 107–115.

[13] L. Fang, S. Vandewalle and J. Meyers, A parallel-in-time multiple shooting algorithm for large-scale PDE-constrained optimal control problems, *J. Comput. Phys.* 452 (2022) 110926.

[14] M. J. Gander, 50 years of time parallel time integration, in *Multiple Shooting and Time Domain Decomposition*, T. Carraro, M. Geiger, S. Körkel and R. Rannacher, eds., Springer-Verlag, Heidelberg, 2015.

[15] M. J. Gander, E. Hairer, Analysis for parareal algorithms applied to Hamiltonian differential equations, *J. Comput. Appl. Math.* 259 (2014) 2–13.

[16] I. C. Garcia, I. Kulchytska-Ruchka and S. Schöps, Parareal for index two differential algebraic equations, *Numer. Algorithms* 91 (1) (2022) 389–412.

[17] M. J. Gander, S. Vandewalle, Analysis of the parareal time-parallel time-integration method, *SIAM J. Sci. Comput.* 29 (2) (2007) 556–578.

[18] B. Guo, Z. Wang, Legendre-Gauss collocation methods for ordinary differential equations, *Adv. Comput. Math.* 30 (3) (2009) 249–280.

[19] B. Guo, Z. Wang, A spectral collocation method for solving initial value problems of first order ordinary differential equations, *Discrete Contin. Dyn. Syst. Ser. B* 14 (3) (2010) 1029–1054.

[20] M. J. Gander, S. Wu, A diagonalization-based parareal algorithm for dissipative and wave propagation problems, *SIAM J. Numer. Anal.* 58 (5) (2020) 2981–3009.

[21] E. Hairer, G. Wanner, *Solving Ordinary Differential Equation II: Stiff and Differential-Algebraic Problems*, Springer-Verlag, Berlin, 1991.

[22] J. Hong, X. Wang and L. Zhang, Parareal exponential -scheme for longtime simulation of stochastic Schrödinger equations with weak damping, *SIAM J. Sci. Comput.* 41 (6) (2019) B1155–B1177.

[23] Y. Jiang, J. Liu, Fast parallel-in-time quasi-boundary value methods for backward heat conduction problems, *Appl. Numer. Math.* 184 (2023) 325–339.

[24] M. La Scala, A. Bose, Relaxation/Newton methods for concurrent time step solution of differential-algebraic equations in power system dynamic simulations, *IEEE Trans. Circuits Systems I Fund. Theory Appl.* 40 (5) (1993) 317–330.

[25] C. Lohmann, J. Dünnebacke and S. Turek, Fourier analysis of a time-simultaneous two-grid algorithm using a damped Jacobi waveform relaxation smoother for the one-dimensional heat equation, *J. Numer. Math.* 30 (3) (2022), 173–207.

[26] F. Legoll, T. Lelièvre, and U. Sharma, An Adaptive Parareal Algorithm: Application to the Simulation of Molecular Dynamics Trajectories, *SIAM J. Sci. Comput.* 44 (1) (2022) B146–B176.

[27] J. Lions, Y. Maday and G. Turinici, A "parareal" in time discretization of PDE's, *C. R. Acad. Sci. Paris Sér. I Math.* 332 (7) (2001) 661–668.

[28] J. Liu, Z. Wang, A ROM-accelerated parallel-in-time preconditioner for solving all-at-once systems in unsteady convection-diffusion PDEs, *Appl. Math. Comput.* 416 (2022) 126750.

[29] J. Liu, X. Wang, S. Wu and T. Zhou, A well-conditioned direct PinT algorithm for first- and second-order evolutionary equations, *Adv. Comput. Math.* 48 (3) (2022) 1–29.

[30] T. P. Mathew, M. Sarkis and C. E. Schaerer, Analysis of block parareal preconditioners for parabolic optimal control problems, *SIAM J. Sci. Comput.* 32 (3) (2010) 1180–1200.

[31] K. Pentland, M. Tamborrino, D. Samaddar, and L. C. Appel, Stochastic Parareal: An Application of Probabilistic Methods to Time-Parallelization, *SIAM J. Sci. Comput.* (2022) S82–S102.

[32] J. Shen, T. Tang and L. Wang, *Spectral Methods: Algorithms, Analysis and Applications*, Springer, Heidelberg, 2011.

[33] H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems*, Reston, VA: American Institute of Aeronautics and Astronautics, Inc, 1st edition, 2003.

[34] G. A. Staff, E. M. Rønquist, Stability of the parareal algorithm, in *Domain Decomposition Methods in Science and Engineering*, Lecture Notes Comput. Sci. Eng. 40 (2005) 449–456.

[35] S. Wu, Convergence analysis of some second-order parareal algorithms, *IMA J. Numer. Anal.* 35 (3) (2015) 1315–1341.

[36] S. Wu, T. Huang, A fast second-order parareal solver for fractional optimal control problems, *J. Vib. Control* 24 (15) (2018) 3418–3433.

[37] R. Woollands, J. L. Junkins, Nonlinear differential equation solvers via adaptive Picard-Chebyshev iteration: applications in astrodynamics, *J. Guid. Control Dyn.* 42 (5) (2019) 1007–1022.

[38] S. Wu, J. Liu, A parallel-in-time block-circulant preconditioner for optimal control of wave equations, *SIAM J. Sci. Comput.* 42 (3) (2020) A1510–A1540.

[39] Z. Wang, J. Mu, A multiple interval Chebyshev-Gauss-Lobatto collocation method for ordinary differential equations, *Numer. Math. Theor. Meth. Appl.* 9 (4) (2016) 619–639.

[40] R. Watschinger, M. Merta, G. Of and J. Zapletal, A parallel fast multipole method for a space-time boundary element method for the heat equation, *SIAM J. Sci. Comput.* 44 (4) (2022) C320–C345.

[41] Y. Wang, G. Ni and Y. Liu, Multistep Newton-Picard method for nonlinear differential equations, *J. Guid. Control Dyn.* 43 (11) (2020) 2148–2155.

[42] S. Wu, T. Zhou, Convergence analysis for three parareal solvers, *SIAM J. Sci. Comput.* 37 (2) (2015) A970–A992.

[43] S. Wu, T. Zhou, Parallel implementation for the two-stage SDIRK methods via diagonalization, *J. Comput. Phys.* 428 (2021) 110076.

[44] X. Yang, Z. Wang, A Chebyshev-Gauss spectral collocation method for ordinary differential equations, *J. Comput. Math.* 33 (1) (2015) 59–85.

[45] J. Yang, Z. Yuan and Z. Zhou, Robust convergence of parareal algorithms with arbitrarily high-order fine propagators, *SSRN Electronic Journal*, 2022.

[46] L. Zhang, J. Wang, W. Zhou, L. Liu and L. Zhang, Convergence analysis of parareal algorithm based on Milstein scheme for stochastic differential equations, *J. Comput. Math.* 38 (3) (2020) 487–501.