

## ANALYSIS OF TWO PARAREAL ALGORITHMS FOR TIME-PERIODIC PROBLEMS\*

MARTIN J. GANDER<sup>†</sup>, YAO-LIN JIANG<sup>‡</sup>, BO SONG<sup>§</sup>, AND HUI ZHANG<sup>†</sup>

**Abstract.** The parareal algorithm, which permits us to solve evolution problems in a time parallel fashion, has created a lot of attention over the past decade. The algorithm has its roots in the multiple shooting method for boundary value problems, which in the parareal algorithm is applied to initial value problems, with a particular coarse approximation of the Jacobian matrix. It is therefore of interest to formulate parareal-type algorithms for time-periodic problems, which also couple the end of the time interval with the beginning, and to analyze their performance in this context. We present and analyze two parareal algorithms for time-periodic problems: one with a periodic coarse problem and one with a nonperiodic coarse problem. An interesting advantage of the algorithm with the nonperiodic coarse problem is that no time-periodic problems need to be solved during the iteration, since on the time subdomains, the problems are not time-periodic either. We prove for both linear and nonlinear problems convergence of the new algorithms, with linear bounds on the convergence. We also extend these results to evolution partial differential equations using Fourier techniques. We illustrate our analysis with numerical experiments, both for model problems and the realistic application of a nonlinear cooled reverse-flow reactor system of partial differential equations.

**Key words.** parareal algorithm, time-periodic problems, linear convergence, multiple shooting

**AMS subject classifications.** 65L20, 65L70, 65F15

**DOI.** 10.1137/130909172

**1. Introduction.** Time-periodic problems appear typically in special physical situations, for example, in eddy current simulations [1], or when periodic forcing is used, like for periodically forced reactors, see [28, 30]. The numerical simulation of time-periodic problems is a special area of research, since the time-periodicity modifies the problem structure and solution methods significantly; see, for example, [4, 25, 27]. When the scale of the problems increases, it is desirable to use parallel methods to solve such problems, and various kinds of parallel methods have been proposed to solve time-periodic problems in the literature, such as multigrid methods [15] and waveform relaxation methods [16, 18, 19, 20, 29], which have proved to be quite effective for time-periodic problems.

All these methods, however, use spatial parallelism as an essential ingredient. Over the last few years, model order reduction methods have been developed in order to solve systems of lower dimension instead of the original large-scale systems of differential equations, and in such models often the parallelization in space saturates

---

\*Submitted to the journal's Methods and Algorithms for Scientific Computing section February 11, 2013; accepted for publication (in revised form) August 29, 2013; published electronically October 24, 2013. This work was supported by the Natural Science Foundation of China (NSFC) under grant 11071192, 11371287 and the International Science and Technology Cooperation Program of China under grant 2010DFA14700.

<http://www.siam.org/journals/sisc/35-5/90917.html>

<sup>†</sup>Section of Mathematics, University of Geneva, 1211 Geneva 4, Switzerland (Martin.Gander@unige.ch, mike.hui.zhang@hotmail.com).

<sup>‡</sup>College of Mathematics and System Sciences, Xinjiang University, Urumqi 830046, China; School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China (yljiang@mail.xjtu.edu.cn).

<sup>§</sup>Corresponding author. School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China (songxjtu@gmail.com).

rapidly [17]. In this paper, we are interested in the time direction for parallelization of time-periodic problems.

The parareal algorithm is such a time-parallel method that was proposed by Lions, Maday, and Turinici in the context of virtual control to solve evolution problems in parallel; see [21]. In this algorithm, initial value problems are solved on subintervals in time, and through an iteration, the initial values on each subinterval are corrected to converge to the correct values of the overall solution. The parareal algorithm has been used in many application areas; see [3, 6, 11, 22, 23, 24, 13] and references therein. For a precise convergence analysis for linear ordinary and partial differential equations, see [9], and for the nonlinear case, see [10, 32]. We develop in this paper two parareal-type algorithms for time-periodic problems and present a complete convergence analysis for the case of linear and nonlinear problems. It is interesting to note that the parareal algorithms for time-periodic problems do not seem to have a superlinear convergence regime, in contrast to the classical parareal algorithm for initial value problems.

Our paper is organized as follows. In section 2, we present our two parareal algorithms for scalar time-periodic ordinary differential equations (ODEs), using the interpretation of multiple shooting [9]. In section 3, we prove linear convergence results for both algorithms applied to ODEs. In section 4, we extend these convergence results to partial differential equations (PDEs), namely, the heat equation. In section 5, we present a convergence analysis of our two new algorithms for nonlinear ODEs. We illustrate our analysis with numerical experiments in section 6 and present our conclusions in section 7.

**2. Two parareal algorithms for time-periodic problems.** In order to describe the two new algorithms, we consider the time-periodic model problem

$$(2.1) \quad \begin{aligned} \frac{d\mathbf{u}}{dt} &= \mathbf{f}(\mathbf{u}, t), \quad t \in [0, T], \\ \mathbf{u}(0) &= \mathbf{u}(T), \end{aligned}$$

where the nonlinear function  $\mathbf{f} : \mathbb{R}^n \times [0, T] \rightarrow \mathbb{R}^n$  satisfies  $\mathbf{f}(\cdot, 0) = \mathbf{f}(\cdot, T)$ , and  $\mathbf{u} : \mathbb{R} \rightarrow \mathbb{R}^n$  is the solution we want to compute.

To derive our two parareal variants for time-periodic problems, we use the relation of the parareal algorithm to the multiple shooting method [9]: we decompose the time interval  $[0, T]$  into  $N$  subintervals  $[T_n, T_{n+1}]$ ,  $n = 0, 1, \dots, N-1$ , with  $0 = T_0 < T_1 < \dots < T_{N-1} < T_N = T$ , and consider the  $N$  separate initial value problems

$$(2.2) \quad \begin{aligned} \frac{d\mathbf{u}_0}{dt} &= \mathbf{f}(\mathbf{u}_0, t), & \mathbf{u}_0(0) &= \mathbf{U}_0, & t &\in [0, T_1], \\ \frac{d\mathbf{u}_1}{dt} &= \mathbf{f}(\mathbf{u}_1, t), & \mathbf{u}_1(T_1) &= \mathbf{U}_1, & t &\in [T_1, T_2], \\ &\vdots & &\vdots & & \\ \frac{d\mathbf{u}_{N-1}}{dt} &= \mathbf{f}(\mathbf{u}_{N-1}, t), & \mathbf{u}_{N-1}(T_{N-1}) &= \mathbf{U}_{N-1}, & t &\in [T_{N-1}, T_N], \end{aligned}$$

together with the matching conditions satisfied by the solution  $\mathbf{u}$ ,

$$(2.3) \quad \mathbf{U}_0 - \mathbf{U}_N = 0, \quad \mathbf{U}_1 - \mathbf{u}_0(T_1, \mathbf{U}_0) = 0, \quad \dots, \quad \mathbf{U}_N - \mathbf{u}_{N-1}(T, \mathbf{U}_{N-1}) = 0,$$

where  $\mathbf{u}_n(T_{n+1}, \mathbf{U}_n)$ ,  $n = 0, 1, \dots, N-1$ , denotes the solution at  $T_{n+1}$  of the corresponding problem in (2.2). Substituting the first equation in (2.3) into the last one yields



we modify (2.10) by a relaxation, which leads to the *periodic parareal algorithm with initial value coarse problem (PP-IC)*,

$$(2.12) \quad \mathbf{U}_0^{k+1} = \mathbf{U}_N^k,$$

$$(2.13) \quad \mathbf{U}_{n+1}^{k+1} = F(T_{n+1}, T_n, \mathbf{U}_n^k) + G(T_{n+1}, T_n, \mathbf{U}_n^{k+1}) - G(T_{n+1}, T_n, \mathbf{U}_n^k), \\ n = 0, \dots, N - 1.$$

*Remark 2.1.* Comparing with the PP-PC algorithm, we observe that each iterate is not time-periodic in the PP-IC algorithm after a relaxation on the periodic constraint. Instead, the time-periodicity will be recovered at convergence. In this way, no time-periodic problems need to be solved during the iteration.

**3. Convergence analysis for linear scalar ODEs.** We now study the convergence of the two parareal algorithms PP-PC and PP-IC for a linear scalar ODE of the form

$$(3.1) \quad \frac{du}{dt} + au = f(t), \quad u(0) = u(T), \quad t \in [0, T],$$

where  $f : [0, T] \rightarrow \mathbb{R}$  satisfies  $f(0) = f(T)$ ,  $u : [0, T] \rightarrow \mathbb{R}$ , and  $a \in \mathbb{C}$  with  $\Re(a) > 0$ .

For simplicity, we assume that the  $F$ -propagator represents the exact solution,  $F(T_{n+1}, T_n, \mathbf{U}_n^k) = e^{-a\Delta T_{n+1}} \mathbf{U}_n^k + \int_{T_n}^{T_{n+1}} e^{-a(T_{n+1}-s)} f(s) ds$ , where  $\Delta T_{n+1} = T_{n+1} - T_n$ , and we also assume that the time interval  $[0, T]$  is divided into  $N$  subintervals of equal size, i.e.,  $\Delta T_n = \Delta T = \frac{T}{N}$ . For the  $G$ -propagator, we use a one-step method  $G(T_{n+1}, T_n, \mathbf{U}_n^k) = R(a\Delta T) \mathbf{U}_n^k + \phi_n(f, a\Delta T)$ , where  $R(a\Delta T)$  is the stability function of the one-step method. We finally assume that  $G$  satisfies the condition

$$(3.2) \quad |e^{-a\Delta T} - R(a\Delta T)| + |R(a\Delta T)| < 1.$$

LEMMA 3.1. *If  $a$  is real and positive, condition (3.2) is equivalent to the stability condition in [2, 9, 26], namely,*

$$(3.3) \quad |2R(a\Delta T) - e^{-a\Delta T}| < 1.$$

*Proof.* Inequality (3.2) implies (3.3) by the triangle inequality (even for general  $a \in \mathbb{C}$ ):

$$|2R(a\Delta T) - e^{-a\Delta T}| = |R(a\Delta T) + R(a\Delta T) - e^{-a\Delta T}| \leq |R(a\Delta T)| \\ + |e^{-a\Delta T} - R(a\Delta T)| < 1.$$

To show that (3.3) implies (3.2), we treat two cases: if  $\text{sign}(e^{-a\Delta T} - R(a\Delta T)) \neq \text{sign}(R(a\Delta T))$ , then

$$|e^{-a\Delta T} - R(a\Delta T)| + |R(a\Delta T)| = |2R(a\Delta T) - e^{-a\Delta T}| < 1$$

using (3.3), and otherwise

$$|e^{-a\Delta T} - R(a\Delta T)| + |R(a\Delta T)| = |e^{-a\Delta T}| < 1$$

under the assumption on  $a$ , which completes the proof.  $\square$

**3.1. Convergence analysis of the PP-PC algorithm.** We denote the error at iteration step  $k$  of the parareal algorithm at time  $T_n$  by  $e_n^k := u(T_n) - U_n^k$  and define  $\mathbf{e}^k := (e_0^k, e_1^k, \dots, e_{N-1}^k)^T$ . Then we have the following lemma.

LEMMA 3.2. *The errors of the PP-PC algorithm satisfy  $\mathbf{e}^{k+1} = M_1^{-1}N_1\mathbf{e}^k$ , where*

$$M_1 = \begin{bmatrix} 1 & & & -R(a\Delta T) \\ -R(a\Delta T) & 1 & & \\ & & \ddots & \\ & & & -R(a\Delta T) & 1 \end{bmatrix}$$

and

$$N_1 = \begin{bmatrix} 0 & & & e^{-a\Delta T} - R(a\Delta T) \\ e^{-a\Delta T} - R(a\Delta T) & 0 & & \\ & & \ddots & \\ & & & e^{-a\Delta T} - R(a\Delta T) & 0 \end{bmatrix}.$$

*Proof.* From the PP-PC algorithm (2.10), and the assumptions on the  $F$ -propagator and the  $G$ -propagator, we get

$$\begin{aligned} U_n^{k+1} &= G(T_n, T_{n-1}, U_{n-1}^{k+1}) + F(T_n, T_{n-1}, U_{n-1}^k) - G(T_n, T_{n-1}, U_{n-1}^k) \\ &= (R(a\Delta T)U_{n-1}^{k+1} + \phi_{n-1}(f, a\Delta T)) + \left( e^{-a\Delta T}U_{n-1}^k + \int_{T_{n-1}}^{T_n} e^{-a(T_n-s)}f(s)ds \right) \\ &\quad - (R(a\Delta T)U_{n-1}^k + \phi_{n-1}(f, a\Delta T)) \\ &= R(a\Delta T)(U_{n-1}^{k+1} - U_{n-1}^k) + \left( e^{-a\Delta T}U_{n-1}^k + \int_{T_{n-1}}^{T_n} e^{-a(T_n-s)}f(s)ds \right). \end{aligned}$$

By calculating, we find that the errors  $e_n^{k+1} = u(T_n) - U_n^{k+1}$  satisfy

$$\begin{aligned} (3.4) \quad e_n^{k+1} &= \left( e^{-a\Delta T}u(T_{n-1}) + \int_{T_{n-1}}^{T_n} e^{-a(T_n-s)}f(s)ds \right) \\ &\quad - R(a\Delta T)(U_{n-1}^{k+1} - U_{n-1}^k) - \left( e^{-a\Delta T}U_{n-1}^k + \int_{T_{n-1}}^{T_n} e^{-a(T_n-s)}f(s)ds \right) \\ &= R(a\Delta T)e_{n-1}^{k+1} + (e^{-a\Delta T} - R(a\Delta T))e_{n-1}^k \end{aligned}$$

for  $n = 1, 2, \dots, N - 1$ . Similarly, for  $n = 0$ , we get

$$(3.5) \quad e_0^{k+1} = R(a\Delta T)e_{N-1}^{k+1} + (e^{-a\Delta T} - R(a\Delta T))e_{N-1}^k.$$

Writing the relations (3.4) and (3.5) in matrix form concludes the proof. □

From Lemma 3.2 we can see the algorithm PP-PC will not converge in a finite number of steps, which is in contrast to the classical parareal algorithm applied to initial value problems.

Noting that the matrix  $M_1$  and the matrix  $N_1$  are circulant matrices, it is easy to verify that  $M_1^{-1}N_1$  is normal (see [14]). Therefore, from spectral theory, we obtain  $\|M_1^{-1}N_1\|_2 = \rho(M_1^{-1}N_1)$ . Then we denote by the spectral radius  $\rho(M_1^{-1}N_1)$  of the matrix  $M_1^{-1}N_1$  the convergence factor of the PP-PC algorithm. In the following theorem, we provide a bound on the convergence factor of the PP-PC algorithm.

**THEOREM 3.3.** *Let  $\Delta T$  be given, and  $T_n = n\Delta T$  with  $n = 0, 1, \dots, N$ . Assume  $F(T_{n+1}, T_n, U_n^k)$  is the exact solution of problem (3.1) with  $u(T_n) = U_n^k$ , and  $G(T_{n+1}, T_n, U_n^k) = R(a\Delta T)U_n^k + \phi(f, a\Delta T)$  is a one-step method satisfying the stability condition (3.2). Then the convergence factor of the PP-PC algorithm is bounded by*

$$(3.6) \quad \rho(M_1^{-1}N_1) \leq \frac{|e^{-a\Delta T} - R(a\Delta T)|}{1 - |R(a\Delta T)|}.$$

*In the special case when  $a \in \mathbb{R}$  and  $a > 0$ , the convergence factor of the PP-PC algorithm is*

$$(3.7) \quad \rho(M_1^{-1}N_1) = \begin{cases} \frac{|e^{a\Delta T} - R(a\Delta T)|}{\sqrt{R(a\Delta T)^2 - 2R(a\Delta T) + 1}} & \text{if } R(a\Delta T) > 0, \text{ or } R(a\Delta T) < 0 \text{ with } N \text{ even,} \\ \frac{|e^{a\Delta T} - R(a\Delta T)|}{\sqrt{R(a\Delta T)^2 - 2R(a\Delta T) \cos \frac{N-1}{N}\pi + 1}} & \text{if } R(a\Delta T) < 0 \text{ with } N \text{ odd.} \end{cases}$$

*Proof.* We first consider the general case with  $a \in \mathbb{C}$  and  $\Re(a) > 0$ . Assume that  $\lambda_n, n = 0, 1, \dots, N - 1$ , are the eigenvalues of  $M_1^{-1}N_1$ . Using Fourier analysis (see [14]), we get

$$\lambda_n = \frac{\beta e^{-i2\pi n/N}}{1 - \alpha e^{-i2\pi n/N}},$$

where  $\alpha = R(a\Delta T)$  and  $\beta = e^{a\Delta T} - R(a\Delta T)$ . Then the spectral radius of  $M_1^{-1}N_1$  is

$$(3.8) \quad \rho(M_1^{-1}N_1) = \max_{0 \leq n \leq N-1} |\lambda_n| = \max_{0 \leq n \leq N-1} \left| \frac{\beta e^{-i2\pi n/N}}{1 - \alpha e^{-i2\pi n/N}} \right|.$$

Using  $|e^{i2\pi n/N} - \alpha| \geq |e^{i2\pi n/N}| - |\alpha| = 1 - |\alpha| > 0$ , we obtain (3.6) as follows:

$$\rho(M_1^{-1}N_1) = \max_{0 \leq n \leq N-1} |\beta| \left| \frac{1}{e^{i2\pi n/N} - \alpha} \right| \leq \frac{|\beta|}{1 - |\alpha|} = \frac{|e^{a\Delta T} - R(a\Delta T)|}{1 - |R(a\Delta T)|}.$$

We now consider the special case with  $a \in \mathbb{R}$  and  $a > 0$ . From the identity in (3.8), we get

$$\begin{aligned} |\lambda_n| &= |\beta| \left| \frac{e^{-i2\pi n/N}}{1 - \alpha e^{-i2\pi n/N}} \right| = |\beta| \left| \frac{1}{e^{i2\pi n/N} - \alpha} \right| \\ &= \frac{|\beta|}{\sqrt{\alpha^2 - 2\alpha \cos 2\pi n/N + 1}}. \end{aligned}$$

When  $\alpha > 0$ ,  $|\lambda_n|$  is largest when  $n = 0$ , and then

$$\rho(M_1^{-1}N_1) = \frac{|\beta|}{\sqrt{\alpha^2 - 2\alpha + 1}}.$$

When  $\alpha < 0$ , if  $N$  is even, then  $|\lambda_n|$  is largest when  $n = N/2$ , and

$$\rho(M_1^{-1}N_1) = \frac{|\beta|}{1 - |\alpha|};$$

otherwise, if  $N$  is odd, then  $|\lambda_n|$  is largest when  $n = (N - 1)/2$ , and

$$\rho(M_1^{-1}N_1) = \frac{|\beta|}{\sqrt{\alpha^2 - 2\alpha \cos \frac{N-1}{N}\pi + 1}},$$

which completes the proof of (3.7). □

Note that under the stability condition (3.2) on the  $G$ -propagator,  $\rho$  is smaller than 1, and thus the algorithm converges at least linearly. This result is similar to the long time interval case of the parareal algorithm for initial value problems in [9].

*Remark 3.1.* For later use, we note here that if we replace in Lemma 3.2 in the matrix  $M_1$  the term  $R(a\Delta T)$  by  $\alpha$  and in the matrix  $N_1$  the term  $e^{-a\Delta T} - R(a\Delta T)$  by  $\beta$ , and if  $|\alpha| + |\beta| < 1$ , we obtain  $\rho(M_1^{-1}N_1) \leq \frac{|\beta|}{1-|\alpha|}$ .

**3.2. Convergence analysis of the PP-IC algorithm.** As for the PP-PC algorithm, we obtain the following lemma for the relation of the errors in the PP-IC algorithm, albeit now with an error vector which is one longer, i.e.,  $\mathbf{e}^k = (e_0^k, e_1^k, \dots, e_N^k)^T$ .

LEMMA 3.4. *The errors of the PP-IC algorithm satisfy  $\mathbf{e}^{k+1} = M_2^{-1}N_2\mathbf{e}^k$ , where*

$$M_2 = \begin{bmatrix} 1 & & & & \\ -R(a\Delta T) & 1 & & & \\ & & \ddots & & \\ & & & -R(a\Delta T) & 1 \end{bmatrix}$$

and

$$N_2 = \begin{bmatrix} 0 & & & & 1 \\ e^{-a\Delta T} - R(a\Delta T) & 0 & & & \\ & & \ddots & & \\ & & & e^{-a\Delta T} - R(a\Delta T) & 0 \end{bmatrix}.$$

*Proof.* Analogously to the PP-PC algorithm, we get

$$(3.9) \quad e_n^{k+1} = R(a\Delta T)e_{n-1}^{k+1} + (e^{-a\Delta T} - R(a\Delta T))e_{n-1}^k,$$

where  $n = 1, 2, \dots, N$ , and for  $n = 0$ , we get  $e_0^{k+1} = R(a\Delta T)e_{N-1}^k + (e^{-a\Delta T} - R(a\Delta T))e_{N-1}^{k-1} = e_N^k$ . Writing these relations in matrix form concludes the proof. □

LEMMA 3.5. *Assume the  $F$ -propagator is the exact solver of the problem (3.1), and the  $G$ -propagator satisfies the stability condition (3.2), and let  $\lambda$  be an eigenvalue of  $M_2^{-1}N_2$ ,  $\lambda \in \mathbb{C}$ . Then  $|\lambda| < 1$ , and furthermore, if  $\lambda \neq 0$ ,  $\lambda$  satisfies the equation*

$$(3.10) \quad \lambda^{N+1} = (R(a\Delta T)\lambda + e^{-a\Delta T} - R(a\Delta T))^N.$$

*Proof.* Since  $\lambda$  is an eigenvalue of  $M_2^{-1}N_2$ , there exists a nonzero eigenvector  $\boldsymbol{\xi} = (\xi_0, \xi_1, \dots, \xi_N)^T$  with  $M_2^{-1}N_2\boldsymbol{\xi} = \lambda\boldsymbol{\xi}$ , or equivalently  $N_2\boldsymbol{\xi} = \lambda M_2\boldsymbol{\xi}$ , which implies

$$(3.11) \quad \xi_N = \lambda\xi_0, \quad \beta\xi_{n-1} = \lambda(-\alpha\xi_{n-1} + \xi_n), \quad n = 1, 2, \dots, N,$$

where  $\alpha := R(a\Delta T)$ , and  $\beta := e^{-a\Delta T} - R(a\Delta T)$ . Then, from the second relation in (3.11), it follows that

$$(3.12) \quad (\lambda\alpha + \beta)\xi_{n-1} = \lambda\xi_n, \quad n = 1, 2, \dots, N.$$

By induction on  $n$ , since  $\lambda \neq 0$ , we get

$$(3.13) \quad \xi_N = \left( \alpha + \frac{\beta}{\lambda} \right)^N \xi_0.$$

Substituting now the first equation from (3.11) into (3.13), we get

$$(3.14) \quad \lambda \xi_0 = \left( \alpha + \frac{\beta}{\lambda} \right)^N \xi_0.$$

If  $\lambda = -\frac{\beta}{\alpha} \neq 0$ , then we obtain from (3.14) that  $\xi_0 = 0$ , and with (3.13) the entire vector  $\xi = 0$ , a contradiction. Hence we must have  $\lambda \neq -\frac{\beta}{\alpha}$ , and therefore we get

$$\lambda^{N+1} = (\alpha\lambda + \beta)^N,$$

which proves (3.10). Taking the  $N$ th root and the modulus, we obtain further the estimate

$$(3.15) \quad |\lambda|^{\frac{N+1}{N}} = |\alpha\lambda + \beta| \leq |\alpha||\lambda| + |\beta|.$$

Now we prove that  $|\lambda| < 1$  by contradiction. Assume  $|\lambda| \geq 1$ ; then

$$|\lambda|^{\frac{N+1}{N}} \leq (|\alpha| + |\beta|)|\lambda|,$$

which means, using the stability condition (3.2),

$$|\lambda| \leq (|\alpha| + |\beta|)^N < 1,$$

a contradiction to the assumption that  $|\lambda| \geq 1$ . Therefore we must have  $|\lambda| < 1$ .  $\square$

From Lemma 3.5, we can conclude that the PP-IC algorithm is convergent. Since the matrix  $M_2^{-1}N_2$  is not normal, we cannot denote by the spectral radius  $\rho(M_2^{-1}N_2)$  the convergence factor of the PP-IC algorithm any more. However, from spectral theory, we can define the asymptotic convergence factor of the PP-IC algorithm by  $\rho_{asym}(M_2^{-1}N_2) = \lim_{k \rightarrow \infty} (\|e^k\|/\|e^0\|)^{1/k}$ . From [31], we can easily obtain the following lemma.

**LEMMA 3.6.** *Assume the  $F$ -propagator is the exact solver of the problem (3.1) and the  $G$ -propagator satisfies the stability condition (3.2). Then the asymptotic convergence factor of the PP-IC algorithm is bounded by*

$$\rho_{asym}(M_2^{-1}N_2) \leq \rho(M_2^{-1}N_2).$$

In the following theorem, we provide a bound on the asymptotic convergence factor of the PP-IC algorithm based on the bound of spectral radius of the matrix  $M_2^{-1}N_2$ .

**THEOREM 3.7.** *Let  $\Delta T$  be given, and  $T_n = n\Delta T$  with  $n = 0, 1, \dots, N$ . Assume  $F(T_{n+1}, T_n, U_n^k)$  is the exact solution of (3.1) with  $u(T_n) = U_n^k$ , and  $G(T_{n+1}, T_n, U_n^k) = R(a\Delta T)U_n^k + \phi(f, a\Delta T)$  is a one-step method satisfying the stability condition (3.2). Then the asymptotic convergence factor of the PP-IC algorithm is bounded by*

$$\rho_{asym}(M_2^{-1}N_2) \leq \rho(M_2^{-1}N_2) \leq \bar{x},$$

where  $\bar{x}$  is the unique solution in  $(0, 1)$  of the equation  $(|R(a\Delta T)|x + |e^{-a\Delta T} - R(a\Delta T)|)^{\frac{N}{N+1}} - x = 0$ . Furthermore, consider a sequence  $x_l = (|R(a\Delta T)|x_{l-1} +$



$|e^{-a\Delta T} - R(a\Delta T)|)^{\frac{N}{N+1}}$ ,  $x_0 = 1$ . Then  $\rho_{asym}(M_2^{-1}N_2) \leq \rho(M_2^{-1}N_2) < x_l \forall l \geq 1$ , and in particular, when  $l = 1$ , we get

$$\rho_{asym}(M_2^{-1}N_2) \leq \rho(M_2^{-1}N_2) < (|R(a\Delta T)| + |e^{-a\Delta T} - R(a\Delta T)|)^{\frac{N}{N+1}}.$$

*Proof.* We define the auxiliary function  $f(x)$  by

$$(3.16) \quad f(x) := (|\alpha|x + |\beta|)^{\frac{N}{N+1}} - x.$$

Since  $f(1) = (|\alpha| + |\beta|)^{\frac{N}{N+1}} - 1 < 0$  and  $f(0) = |\beta|^{\frac{N}{N+1}} > 0$ , there exists at least one solution of the equation  $f(x) = 0$  in  $(0, 1)$ , and we denote it by  $\bar{x} \in (0, 1)$ . To show uniqueness, we compute the derivative of  $f(x)$ ,

$$(3.17) \quad f'(x) = |\alpha| \frac{N}{N+1} (|\alpha|x + |\beta|)^{-\frac{1}{N+1}} - 1,$$

which has a unique root at  $x^* = |\alpha|^N \left(\frac{N}{N+1}\right)^{N+1} - \frac{|\beta|}{|\alpha|}$ . Since  $f'(x) > 0$  when  $x < x^*$  and  $f'(x) < 0$  when  $x^* < x$ , the function  $f$  is concave and therefore  $f(0) > 0$  and  $f(1) < 0$  implies that the root  $\bar{x}$  of the equation  $f(x) = 0$  in  $(0, 1)$  must be unique.

Now we prove that  $x^* < \bar{x}$  by contradiction: if  $x^* > \bar{x} > 0$ , then  $0 < |\beta|^{\frac{N}{N+1}} = f(0) < f(\bar{x}) = 0$ , since  $f'(x) > 0$  for  $0 < x < x^*$ , a contradiction. So we must have  $x^* < \bar{x}$ .

Next, we show by induction that  $x_l > \bar{x}$ . When  $l = 0$ ,  $1 = x_0 > \bar{x}$ . Assuming that for  $l = s$ ,  $x_s > \bar{x}$ , we obtain the induction step

$$(3.18) \quad x_{s+1} = (|\alpha|x_s + |\beta|)^{\frac{N}{N+1}} > (|\alpha|\bar{x} + |\beta|)^{\frac{N}{N+1}} = \bar{x},$$

and thus  $x_l > \bar{x}$ , for  $l = 0, 1, \dots$ . Since  $f'(x) < 0$  for  $\bar{x} < x < 1$ , we have  $f(x_l) < f(\bar{x}) = 0$ , which implies  $x_{l+1} - x_l = f(x_l) < 0$ , and the sequence is decreasing:  $x_{l+1} < x_l$ .

Finally, let  $\lambda$  be an eigenvalue of  $M^{-1}N$ . From Lemma 3.5, we get the inequality

$$(3.19) \quad f(|\lambda|) = (|\alpha||\lambda| + |\beta|)^{\frac{N}{N+1}} - |\lambda| \geq 0.$$

As we have seen above,  $f$  on  $[0, 1]$  is positive only in the subinterval  $[0, \bar{x})$ , and from Lemma 3.5  $|\lambda| < 1$ , so we must have  $|\lambda| \leq \bar{x} < x_l$ . □

*Remark 3.2.* Again for later use, we note that if we replace in Lemma 3.4 in the matrix  $M_2$  the term  $R(a\Delta T)$  by  $\alpha$  and in the matrix  $N_2$  the term  $e^{-a\Delta T} - R(a\Delta T)$  by  $\beta$ , and if  $|\alpha| + |\beta| < 1$ , we obtain  $\rho(M_2^{-1}N_2) < x_l$ , where  $x_l = (|\alpha|x_{l-1} + |\beta|)^{\frac{N}{N+1}}$ .

**4. Convergence analysis for linear PDEs.** We now use the results derived in section 3 to investigate the performance of the parareal algorithms for time-periodic PDEs. We consider the time-periodic diffusion equation in one dimension,

$$(4.1) \quad \begin{aligned} \frac{\partial u(x, t)}{\partial t} - \frac{\partial^2 u(x, t)}{\partial x^2} &= f(x, t), & (x, t) \in \mathbb{R} \times [0, T], \\ u(x, 0) &= u(x, T), \end{aligned}$$

where  $f : \mathbb{R} \times [0, T] \rightarrow \mathbb{R}$  and satisfies  $f(x, 0) = f(x, T)$  for any  $x \in \Omega$ .

Using a Fourier transform in space with Fourier variable  $\omega$ , we obtain the system of ODEs

$$(4.2) \quad \frac{d\hat{u}}{dt} + \omega^2 \hat{u} = \hat{f}(\omega, t), \quad \hat{u}(0) = \hat{u}(T), \quad t \in [0, T].$$

We assume that the  $G$ -propagator for this problem is a one-step method satisfying the stability condition

$$(4.3) \quad |e^{-z} - R(z)| + |R(z)| < 1, \quad \forall z > 0, \quad z := \omega^2 \Delta T.$$

From the convergence results of the linear scalar problem, we obtain the following corollary.

**COROLLARY 4.1.** *Let  $\Delta T$  be given, and  $T_n = n\Delta T$  with  $n = 0, 1, \dots, N$ . Assume  $F(T_{n+1}, T_n, U_n^k)$  is the exact solution of (4.2) with  $u(T_n) = U_n^k$ , and  $G(T_{n+1}, T_n, U_n^k) = R(z)U_n^k + \phi(f, z)$  is a one-step method satisfying the stability condition (4.3). Then the convergence factor of the PP-PC algorithm is bounded by*

$$(4.4) \quad \rho(M_1^{-1}N_1) \leq \sup_{z \in \mathbb{R}^+} \frac{|e^{-z} - R(z)|}{1 - |R(z)|},$$

and the asymptotic convergence factor of the PP-IC algorithm is bounded by

$$(4.5) \quad \rho_{asym}(M_2^{-1}N_2) \leq \rho(M_2^{-1}N_2) < x_l,$$

where  $x_l$  satisfies  $x_l = \max_{z \in \mathbb{R}^+} (|R(z)|x_{l-1} + |e^{-z} - R(z)|)^{\frac{N}{N+1}}$  with  $x_0 = 1$ . In particular, for the PP-PC algorithm, if the  $G$ -propagator is the backward Euler method, then we obtain the concrete convergence factor estimate

$$(4.6) \quad \rho(M_1^{-1}N_1) \leq \frac{1}{e}.$$

*Proof.* The first two results (4.4) and (4.5) can be easily proved using Theorems 3.3 and 3.7. We thus focus on the proof of (4.6). If we choose the backward Euler method for the  $G$ -propagator, we get  $R(z) = \frac{1}{1+z}$ . Let  $\varphi(z) = \frac{|e^{-z} - R(z)|}{1 - |R(z)|}$ . Noting that  $\frac{1}{1+z} > e^{-z}$ , we can rewrite  $\varphi(z)$  as

$$(4.7) \quad \varphi(z) = \frac{\frac{1}{1+z} - e^{-z}}{1 - \frac{1}{1+z}} = \frac{e^z - z - 1}{ze^z}.$$

In order to find the maximum of  $\varphi(z)$ , we set  $\varphi'(z) = 0$  and find

$$(4.8) \quad e^z = z^2 + z + 1.$$

Now to estimate the maximum, we insert the identity (4.8) into (4.7) and obtain  $\varphi_0(z) = \frac{z}{e^z}$ , whose maximum is easily found to be at  $z = 1$ , where  $\varphi_0(1) = \frac{1}{e}$ . Since  $\rho(M_1^{-1}N_1) \leq \sup_{z \in \mathbb{R}^+} \varphi(z) \leq \sup_{z \in \mathbb{R}^+} \varphi_0(z)$ , we obtain estimate (4.6).  $\square$

*Remark 4.1.* Note that the bound  $\frac{1}{e}$  is not sharp: solving (4.8) numerically, we find  $z_0 = 1.7933$ , which gives  $\varphi(z_0) = 0.2984 < 0.3679 = \frac{1}{e}$ .

**5. Convergence analysis for nonlinear scalar problems.** In this section, we generalize the convergence analysis for nonlinear initial value problems from [10] to nonlinear time-periodic problems. We consider the scalar model problem

$$(5.1) \quad \begin{aligned} \frac{du}{dt} &= f(u, t), \quad t \in [0, T], \\ u(0) &= u(T), \end{aligned}$$

where the function  $f : [0, T] \rightarrow \mathbb{R}$  satisfies  $f(u, 0) = f(u, T)$  and  $u : [0, T] \rightarrow \mathbb{R}$ . We assume that the  $F$ -propagator is an exact solver and the  $G$ -propagator is the backward Euler method. We will further need the following assumptions.

*Assumption 5.1.* For the function  $f(\cdot, t)$ , there exists a nonnegative constant  $L$  such that

$$\langle f(u, t) - f(v, t), u - v \rangle \leq -L|u - v|^2, \quad u, v \in \mathbb{R}, \quad \forall t \in [0, T],$$

where we denote by  $\langle \cdot, \cdot \rangle$  the inner product, with  $\langle u, u \rangle = |u|^2 \quad \forall u \in \mathbb{R}$ .

*Assumption 5.2.* The  $F$ -propagator and the  $G$ -propagator satisfy

$$F(t_2, t_1, x) - G(t_2, t_1, x) = c_{p+1}(x)\Delta t^{p+1} + c_{p+2}(x)\Delta t^{p+2} + \dots, x \in \mathbb{R},$$

where  $\Delta t = t_2 - t_1$ ,  $p$  is the order of the  $G$ -propagator and  $c_{p+j}(j = 1, 2, \dots)$  satisfy a uniform Lipschitz condition, which implies that for sufficiently small  $\Delta t$  there exists a nonnegative constant  $C_1$  such that

$$|(F(t_2, t_1, x) - G(t_2, t_1, x)) - (F(t_2, t_1, y) - G(t_2, t_1, y))| \leq C_1\Delta t^{p+1}|x - y|, \quad x, y \in \mathbb{R}.$$

LEMMA 5.1. *Under Assumption 5.1, we get for the  $F$ -propagator*

$$(5.2) \quad |F(t_2, t_1, x) - F(t_2, t_1, y)| \leq e^{-L\Delta t} |x - y|, \quad x, y \in \mathbb{R}, \quad \Delta t = t_2 - t_1.$$

*For the  $G$ -propagator chosen as the backward Euler method, we get*

$$(5.3) \quad |G(t_2, t_1, x) - G(t_2, t_1, y)| \leq \frac{1}{1 + L\Delta t} |x - y|, \quad x, y \in \mathbb{R}, \quad \Delta t = t_2 - t_1.$$

*Proof.* We first prove (5.2). Suppose the functions  $u_1(t)$  and  $v_1(t)$  satisfy the ODEs

$$(5.4) \quad \begin{aligned} \frac{du_1}{dt} &= f(u_1, t), \quad t \in [t_1, t_2], \\ u_1(t_1) &= x, \end{aligned}$$

and

$$(5.5) \quad \begin{aligned} \frac{dv_1}{dt} &= f(v_1, t), \quad t \in [t_1, t_2], \\ v_1(t_1) &= y. \end{aligned}$$

Then  $F(t_2, t_1, x) = u_1(t_2)$  and  $F(t_2, t_1, y) = v_1(t_2)$ . Taking the difference of (5.4) and (5.5) and using Assumption 5.1, we obtain

$$\begin{aligned} \left\langle \frac{d(u_1 - v_1)}{dt}, u_1 - v_1 \right\rangle &= \langle f(u_1, t) - f(v_1, t), u_1 - v_1 \rangle \\ &\leq -L|u_1 - v_1|^2. \end{aligned}$$

Because

$$\left\langle \frac{d(u_1 - v_1)}{dt}, u_1 - v_1 \right\rangle = |u_1 - v_1| \frac{d|u_1 - v_1|}{dt},$$

we get

$$\frac{d|u_1 - v_1|}{dt} \leq -L|u_1 - v_1|, \quad |u_1 - v_1|(t_1) = |x - y|.$$

Therefore, using the Gronwall lemma, we obtain  $|u_1 - v_1|(t) \leq e^{-L(t-t_1)} |x - y| \quad \forall t \geq t_1$ , which leads with  $\Delta t := t_2 - t_1$  to

$$|F(t_2, t_1, x) - F(t_2, t_1, y)| \leq |u_1(t_2) - v_1(t_2)| \leq e^{-L\Delta t} |x - y|.$$

Next, we prove (5.3): since the  $G$ -propagator is the backward Euler method, the functions  $u_2(t)$  and  $v_2(t)$  satisfy the equations

$$(5.6) \quad \begin{aligned} \frac{u_2(t) - u_2(t_1)}{t - t_1} &= f(u_2(t), t), \quad t \in [t_1, t_2], \\ u_2(t_1) &= x, \end{aligned}$$

and

$$(5.7) \quad \begin{aligned} \frac{v_2(t) - v_2(t_1)}{t - t_1} &= f(v_2(t), t), \quad t \in [t_1, t_2], \\ v_2(t_1) &= y, \end{aligned}$$

and we have  $G(t, t_1, x) = u_2(t)$  and  $G(t, t_1, y) = v_2(t)$ . Taking the difference of (5.6) and (5.7), and using Assumption 5.1, we obtain

$$\begin{aligned} &\langle u_2(t) - v_2(t), u_2(t) - v_2(t) \rangle \\ &= \langle x - y, u_2(t) - v_2(t) \rangle + (t - t_1) \langle f(u_2(t), t) - f(v_2(t), t), u_2(t) - v_2(t) \rangle \\ &\leq |x - y| |u_2(t) - v_2(t)| - L(t - t_1) |u_2(t) - v_2(t)|^2. \end{aligned}$$

Because

$$\langle u_2(t) - v_2(t), u_2(t) - v_2(t) \rangle = |u_2(t) - v_2(t)|^2,$$

we get

$$|u_2(t) - v_2(t)| \leq \frac{1}{1 + L(t - t_1)} |x - y|,$$

and therefore with  $\Delta t := t_2 - t_1$

$$|G(t_2, t_1, x) - G(t_2, t_1, y)| = |u_2(t_2) - v_2(t_2)| \leq \frac{1}{1 + L\Delta t} |x - y|. \quad \square$$

**5.1. Convergence analysis of the PP-PC algorithm.** We present two different estimates for the convergence factor of the PP-PC algorithm for nonlinear problems. Our analysis shows that there are different convergence regimes, and our estimates are adapted to this.

**THEOREM 5.2.** *Under Assumptions 5.1 and 5.2, for sufficiently small  $\Delta T$ , the errors of the PP-PC algorithm satisfy*

$$(5.8) \quad \max_{n=0, \dots, N} |u(T_n) - U_n^{k+1}| \leq \frac{(1 + L\Delta T)C_1}{L} \Delta T^p \max_{n=0, \dots, N} |u(T_n) - U_n^k| \quad \forall k \geq 0.$$

*Proof.* As for the PP-PC algorithm in the linear scalar case, we have

$$(5.9) \quad \begin{aligned} u(T_n) - U_n^{k+1} &= F(T_n, T_{n-1}, u(T_{n-1})) - F(T_n, T_{n-1}, U_{n-1}^k) - G(T_n, T_{n-1}, U_{n-1}^{k+1}) \\ &\quad + G(T_n, T_{n-1}, U_{n-1}^k) \\ &= ((F(T_n, T_{n-1}, u(T_{n-1})) - G(T_n, T_{n-1}, u(T_{n-1}))) \\ &\quad - (F(T_n, T_{n-1}, U_{n-1}^k) - G(T_n, T_{n-1}, U_{n-1}^k))) \\ &\quad + (G(T_n, T_{n-1}, u(T_{n-1})) - G(T_n, T_{n-1}, U_{n-1}^{k+1})). \end{aligned}$$

Denoting by  $e_n^{k+1} = |u(T_n) - U_n^{k+1}|$ , we get

$$(5.10) \quad e_n^{k+1} = |u(T_n) - U_n^{k+1}| \leq |(F(T_n, T_{n-1}, u(T_{n-1})) - G(T_n, T_{n-1}, u(T_{n-1}))) - (F(T_n, T_{n-1}, U_{n-1}^k) - G(T_n, T_{n-1}, U_{n-1}^k))| + |G(T_n, T_{n-1}, u(T_{n-1})) - G(T_n, T_{n-1}, U_{n-1}^k)|.$$

Note that from Assumption 5.2,

$$(5.11) \quad |(F(T_n, T_{n-1}, u(T_{n-1})) - G(T_n, T_{n-1}, u(T_{n-1}))) - (F(T_n, T_{n-1}, U_{n-1}^k) - G(T_n, T_{n-1}, U_{n-1}^k))| \leq C_1 \Delta T^{p+1} |u(T_{n-1}) - U_{n-1}^k|,$$

and from Lemma 5.1,

$$(5.12) \quad |G(T_n, T_{n-1}, u(T_{n-1})) - G(T_n, T_{n-1}, U_{n-1}^k)| \leq \frac{1}{1 + L\Delta T} |u(T_{n-1}) - U_{n-1}^k|.$$

Therefore, we have the bound

$$(5.13) \quad e_n^{k+1} \leq C_1 \Delta T^{p+1} e_{n-1}^k + \frac{1}{1 + L\Delta T} e_{n-1}^{k+1} := \beta e_{n-1}^k + \alpha e_{n-1}^{k+1}, \quad n = 1, 2, \dots, N,$$

and  $e_0^{k+1} = e_N^{k+1}$ . In matrix form, this means

$$M_1 e^{k+1} \leq N_1 e^k$$

with  $M_1, N_1$  defined as in Remark 3.1 and  $e^k := (e_0^k, \dots, e_{N-1}^k)^T$ . Using [31, Thm. 3.18, Def. 3.22, pp. 90–91] we see that  $M_1$  is a nonsingular  $M$ -matrix with nonpositive off-diagonal entries, and therefore the entries of its inverse are all nonnegative. So we have

$$e^{k+1} \leq M_1^{-1} N_1 e^k.$$

We now consider  $\gamma^{k+1} := M_1^{-1} N_1 e^k \geq e^{k+1}$ , or by periodic extension,

$$\begin{aligned} \gamma_n^{k+1} &= \beta e_{n-1}^k + \alpha \gamma_{n-1}^{k+1}, & n = 1, 2, \dots, N, \\ \gamma_n^{k+1} &= \gamma_{n+N}^{k+1}, & n \geq 0. \end{aligned}$$

By induction on  $n$  for the first relation in the above equation, we get

$$\begin{aligned} \gamma_1^{k+1} &= \beta e_0^k + \alpha \gamma_0^{k+1}, \\ \gamma_2^{k+1} &= \beta e_1^k + \alpha(\beta e_0^k + \alpha \gamma_0^{k+1}), \\ &\dots, \\ \gamma_n^{k+1} &= \beta \left( \sum_{j=0}^{n-1} \alpha^j e_{n-j-1}^k \right) + \alpha^n \gamma_0^{k+1}, \\ &\dots, \\ \gamma_N^{k+1} &= \beta \left( \sum_{j=0}^{N-1} \alpha^j e_{N-j-1}^k \right) + \alpha^N \gamma_0^{k+1}. \end{aligned}$$

Substituting the periodic condition  $\gamma_N^{k+1} = \gamma_0^{k+1}$  into the last identity, we can solve,

$$\gamma_0^{k+1} = \frac{\beta}{1 - \alpha^N} \sum_{j=0}^{N-1} \alpha^j e_{N-j-1}^k.$$

Therefore, denoting by  $E_k := \max_{n=0, \dots, N} e_n^k$ , we obtain

$$\begin{aligned} (5.14) \quad e_n^{k+1} &\leq \gamma_n^{k+1} = \beta \left( \sum_{j=0}^{n-1} \alpha^j e_{n-j-1}^k \right) + \frac{\beta \alpha^n}{1 - \alpha^N} \sum_{j=0}^{N-1} \alpha^j e_{N-j-1}^k \\ &\leq \beta \frac{1 - \alpha^n}{1 - \alpha} E_k + \frac{\beta \alpha^n}{1 - \alpha} E_k = \frac{\beta}{1 - \alpha} E_k. \end{aligned}$$

The proof is completed by substituting  $\alpha, \beta$  of (5.13) into the above inequality.  $\square$

**THEOREM 5.3.** *Under Assumption 5.1, the errors of the PP-PC algorithm satisfy*

$$(5.15) \quad \max_{n=0, \dots, N} |u(T_n) - U_n^{k+1}| \leq \frac{e^{-L\Delta T} + \frac{1}{1+L\Delta T}}{1 - \frac{1}{1+L\Delta T}} \max_{n=0, \dots, N} |u(T_n) - U_n^k| \quad \forall k \geq 0.$$

*Proof.* Using (5.9) from Theorem 5.2, and letting  $e_n^{k+1} = |u(T_n) - U_n^{k+1}|$ , we obtain

$$\begin{aligned} (5.16) \quad e_n^{k+1} = |u(T_n) - U_n^{k+1}| &\leq |F(T_n, T_{n-1}, u(T_{n-1})) - F(T_n, T_{n-1}, U_{n-1}^k)| \\ &\quad + |G(T_n, T_{n-1}, u(T_{n-1})) - G(T_n, T_{n-1}, U_{n-1}^k)| \\ &\quad + |G(T_n, T_{n-1}, u(T_{n-1})) - G(T_n, T_{n-1}, U_{n-1}^{k+1})|. \end{aligned}$$

Using Lemma 5.1, we get

$$(5.17) \quad |F(T_n, T_{n-1}, u(T_{n-1})) - F(T_n, T_{n-1}, U_{n-1}^k)| \leq e^{-L\Delta T} |u(T_{n-1}) - U_{n-1}^k|,$$

$$(5.18) \quad |G(T_n, T_{n-1}, u(T_{n-1})) - G(T_n, T_{n-1}, U_{n-1}^k)| \leq \frac{1}{1 + L\Delta T} |u(T_{n-1}) - U_{n-1}^k|,$$

and

$$(5.19) \quad |G(T_n, T_{n-1}, u(T_{n-1})) - G(T_n, T_{n-1}, U_{n-1}^{k+1})| \leq \frac{1}{1 + L\Delta T} |u(T_{n-1}) - U_{n-1}^{k+1}|.$$

Therefore, we obtain

$$\begin{aligned} (5.20) \quad e_n^{k+1} &\leq \left( \frac{1}{1 + L\Delta T} + e^{-L\Delta T} \right) e_{n-1}^k \\ &\quad + \frac{1}{1 + L\Delta T} e_{n-1}^{k+1} := \beta e_{n-1}^k + \alpha e_{n-1}^{k+1}, \quad n = 1, 2, \dots, N, \end{aligned}$$

and  $e_0^{k+1} = e_N^{k+1}$ . Note that (5.20) is of the same form as (5.13), so that we can use (5.14) to finish the proof.  $\square$

**5.2. Convergence analysis of the PP-IC algorithm.** We now show that also the PP-IC algorithm for the nonlinear scalar case has two convergence regimes. Although it is still possible to use generating functions to estimate the component-wise errors  $e_n^{k+1}$ , the analysis gets much more involved in this case. We therefore use

a different technique based on the spectral radius of a dominating linear iteration for the PP-IC algorithm.

THEOREM 5.4. *Under Assumptions 5.1 and 5.2 and the condition*

$$(5.21) \quad \frac{1}{1 + L\Delta T} + C_1\Delta T^{p+1} < 1,$$

the errors of the PP-IC algorithm are dominated by a linear iteration  $S$ , i.e.,

$$(5.22) \quad e^{k+1} \leq S e^k$$

holds componentwise for  $e^k := (|u(T_0) - U_0^k|, |u(T_1) - U_1^k|, \dots, |u(T_N) - U_N^k|)^T$  and the spectral radius of  $S$  is bounded by

$$(5.23) \quad \rho(S) \leq x_l, \quad l \geq 1,$$

where  $x_l = \left(\frac{1}{1+L\Delta T}x_{l-1} + C_1\Delta T^{p+1}\right)^{\frac{N}{l}}$  with  $x_0 = 1$ .

*Proof.* As for (5.13), we obtain

$$e_n^{k+1} \leq C_1\Delta T^{p+1}e_{n-1}^k + \frac{1}{1 + L\Delta T}e_{n-1}^{k+1} := \beta e_{n-1}^k + \alpha e_{n-1}^{k+1}, \quad n = 1, 2, \dots, N,$$

and  $e_0^{k+1} = e_N^k$ , or in matrix form  $M_2 e^{k+1} \leq N_2 e^k$ , with  $M_2, N_2$  defined as in Remark 3.2 and  $e^k := (e_0^k, \dots, e_N^k)^T$ . Using the property of  $M$ -matrices [31, Thm. 3.18, Def. 3.22, pp. 90–91], we have

$$e^{k+1} \leq M_2^{-1} N_2 e^k,$$

and the proof can be completed by using Remark 3.2. □

THEOREM 5.5. *Under Assumption 5.1 and the condition*

$$(5.24) \quad e^{-L\Delta T} + \frac{2}{1 + L\Delta T} < 1,$$

the errors of the PP-IC algorithm are dominated by a linear iteration  $S$ , i.e., (5.22) holds componentwise for  $e^k := (|u(T_0) - U_0^k|, |u(T_1) - U_1^k|, \dots, |u(T_N) - U_N^k|)^T$  and the spectral radius of  $S$  is bounded by

$$(5.25) \quad \rho(S) \leq x_l, \quad l \geq 1,$$

where  $x_l = \left(\frac{1}{1+L\Delta T}x_{l-1} + e^{-L\Delta T} + \frac{1}{1+L\Delta T}\right)^{\frac{N}{l}}$  with  $x_0 = 1$ .

*Proof.* As for (5.20), we obtain

$$e_n^{k+1} \leq \left(\frac{1}{1 + L\Delta T} + e^{-L\Delta T}\right) e_{n-1}^k + \frac{1}{1 + L\Delta T} e_{n-1}^{k+1}, \quad n = 1, 2, \dots, N,$$

and  $e_0^{k+1} = e_N^k$ . The remainder of the proof is similar to the proof of Theorem 5.4. □

**6. Numerical experiments.** We now present numerical experiments to illustrate our analysis. We use for the  $G$ -propagator backward Euler with  $\Delta T = T/N$ , and in all experiments except for the realistic application at the end, we compute the initial guess  $\mathbf{U}^0$  by solving a time-periodic coarse problem. All the errors are measured in the infinity norm.

**6.1. Linear scalar problem.** We first show some numerical experiments for the linear scalar problem (3.1) in Figure 6.1 with an exact solution  $u_{exact} = t^2(1-t)^2$ . Here we use for the  $F$ -propagator a fourth-order Runge–Kutta method with time step  $\Delta t$  on each subinterval. The red line denotes the discretization error level, which means the  $L^\infty$  norm of the error between the discrete solution by the  $F$ -propagator with time step  $\Delta t$  and the exact solution. Here we denote by  $(e^k/e^0)^{1/k}$  the numerical convergence factor, where  $e^k$  is the  $L^\infty$  norm of the error between the iterative solution at the  $k$ th parareal iteration and the discrete solution by the  $F$ -propagator with time step  $\Delta t$ . In addition, in the PP-PC algorithm, the linear system arising from discretization of the periodic coarse problem is solved by a direct solver.

From the figure at the bottom right of Figure 6.1, by fixing  $\Delta T = 1/10$ ,  $\Delta t = 1/200$  and varying  $a$ , we observe that when  $a\Delta T$  is small, the PP-PC algorithm converges much better than the PP-IC algorithm. Especially, when  $a\Delta T$  goes to zero, the convergence factor of the PP-IC algorithm goes to 1, while the convergence factor of the PP-PC algorithm goes to 0. When  $a\Delta T$  is large, however, from Theorem 3.3 and (3.16), the convergence factors of the two algorithms go to 0. Notice that for the PP-PC algorithm, a periodic problem has to be solved in each parareal iteration, so the PP-PC algorithm is substantially more costly than the PP-IC algorithm. Therefore, when  $a\Delta T$  is large, it is better to use the PP-IC algorithm than the PP-PC algorithm.

Then, we consider the dependence of convergence behavior on  $\Delta T$  in Figure 6.2. We see that both algorithms converge faster as  $\Delta T$  decreases. From the figure at the bottom right of Figure 6.2, by fixing  $a = 1$ ,  $\Delta t = 1/512$  and varying  $\Delta T$ , we observe

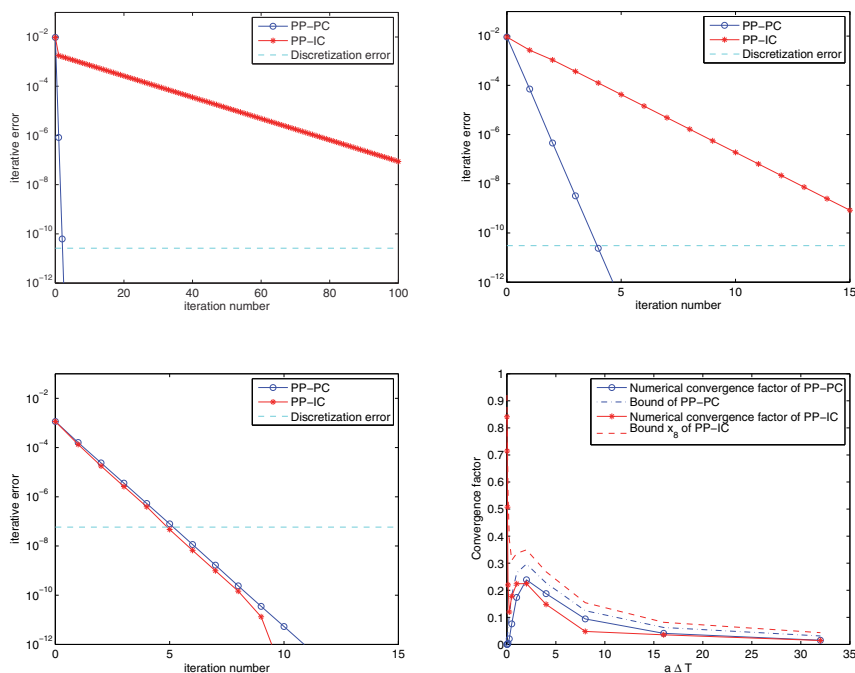


FIG. 6.1. Parareal convergence for the linear scalar case (3.1) with  $T = 1$ ,  $\Delta T = 1/10$ , and  $\Delta t = 1/200$ : comparison of the convergence behavior for  $a = 0.1$  (top left),  $a = 1$  (top right),  $a = 64$  (bottom left), and dependence of the convergence factor on  $a\Delta T$  (bottom right), as described in subsection 6.1.



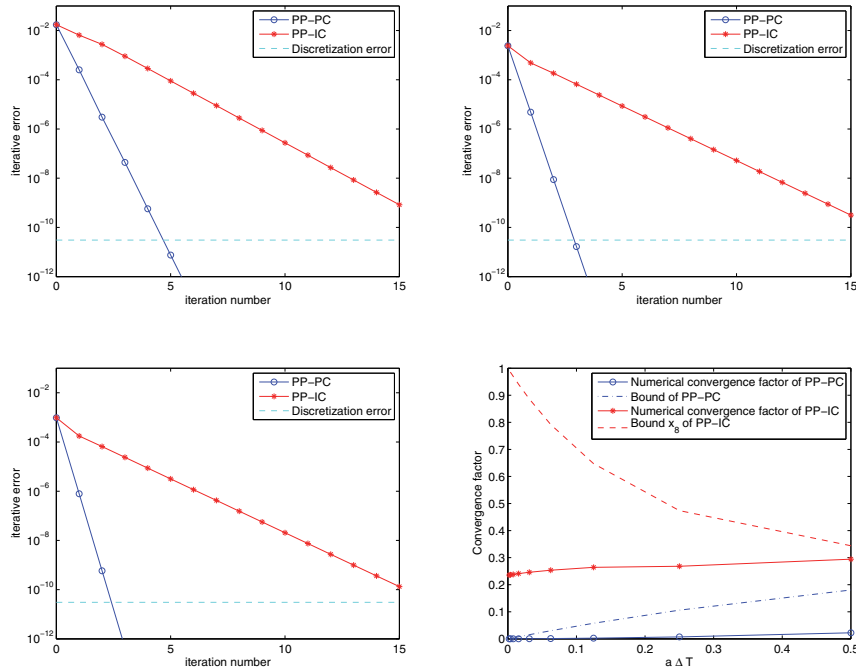


FIG. 6.2. Parareal convergence for the linear scalar case (3.1) with  $T = 1$ : comparison of the convergence behavior for  $\Delta T = 1/5$  (top left),  $\Delta T = 1/40$  (top right),  $\Delta T = 1/100$  (bottom left), with  $a = 1$  and  $\Delta t = 1/200$  in each case, and dependence of the convergence factor on  $a\Delta T$  (bottom right), as described in subsection 6.1.

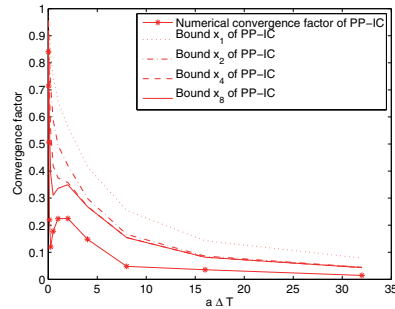


FIG. 6.3. Comparison of the numerical and theoretical convergence factors of the PP-IC algorithm for the linear scalar case (3.1) with  $\Delta T = 1/10$  and  $\Delta t = 1/200$ , as described in subsection 6.1.

that when  $a\Delta T$  goes to 0, the numerical convergence factor of the PP-IC algorithm does not go to 1, unlike the bottom right picture of Figure 6.1.

Next, we compare the numerical convergence factor of the PP-IC algorithm with the theoretical bounds  $x_l$  from Theorem 3.7 in Figure 6.3. One can see how  $x_l$  converges: when  $l$  is more than 4,  $x_l$  changes only little.

**6.2. Diffusion problem.** We solve the problem (4.2) on the time interval  $[0, T]$  and the spatial domain  $\Omega = [0, 1]$ . We take an exact solution  $u_{exact} = x(x-1) \sin(2\pi t)$ .

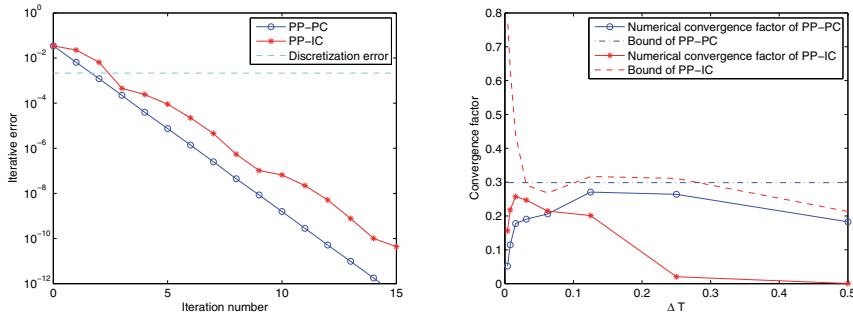


FIG. 6.4. Convergence of the PP-PC algorithm versus the PP-IC algorithm for the diffusion problem (4.2) with  $T = 1$ ,  $\Delta T = 1/10$ , and  $\Delta t = 1/200$  on the left and dependence of the convergence factor on  $\Delta T$  with  $T = 2$  and  $\Delta t = 1/512$  on the right with spatial mesh size  $h = 1/128$  in each case.

The problem (4.2) is discretized in space by a second-order centered finite difference scheme with mesh size  $h = 1/128$ . Then we get the linear system of ODEs

$$\frac{d\mathbf{u}(t)}{dt} = A_h \mathbf{u} + \mathbf{f}(t), \quad t \in [0, T],$$

where  $A_h$  is the discrete Laplacian. Here we use for the  $F$ -propagator backward Euler with time step  $\Delta t$  on each subinterval. The convergence results are shown in Figure 6.4.

**6.3. Nonlinear scalar problem.** We next consider the nonlinear scalar problem (5.1) on the time interval  $[0, 1]$  with the nonlinear function  $f(u, t) = -au^2 + \varphi(t)$ , where  $\varphi(t) = 2t(1 - t)(1 - 2t) + a(t^2(1 - t)^2 + 1)^2$  such that the exact solution is  $u_{exact} = t^2(1 - t)^2 + 1$ . In this example, the  $F$ -propagator and  $G$ -propagator are used to solve a nonlinear scalar initial value problem. We use Newton’s method to solve the nonlinear problems arising in the implicit time discretization. In addition, for the PP-PC algorithm, a nonlinear time-periodic problem has to be solved in each iteration, i.e., in the  $k + 1$ st iteration, we need to solve

$$(6.1) \quad \begin{bmatrix} 1 & & & & & & -G(T_N, T_{N-1}, \cdot) \\ -G(T_1, T_0, \cdot) & 1 & & & & & \\ & -G(T_2, T_1, \cdot) & 1 & & & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & & & -G(T_{N-1}, T_{N-2}, \cdot) & \\ & & & & & & 1 \end{bmatrix} \begin{bmatrix} U_0^{k+1} \\ U_1^{k+1} \\ U_2^{k+1} \\ \vdots \\ U_{N-1}^{k+1} \end{bmatrix} = \begin{bmatrix} F(T_N, T_{N-1}, U_{N-1}^k) - G(T_N, T_{N-1}, U_{N-1}^k) \\ F(T_1, T_0, U_0^k) - G(T_1, T_0, U_0^k) \\ F(T_2, T_1, U_1^k) - G(T_2, T_1, U_1^k) \\ \vdots \\ F(T_{N-1}, T_{N-2}, U_{N-2}^k) - G(T_{N-1}, T_{N-2}, U_{N-2}^k) \end{bmatrix}.$$

The vector on the right-hand side is known, and we denote it by  $\mathbf{b}^{k+1}$ , and let  $\mathbf{U}^{k+1} = [U_0^{k+1}, U_1^{k+1}, \dots, U_{N-1}^{k+1}]^T$ . To solve (6.1), we apply the fixed point iteration

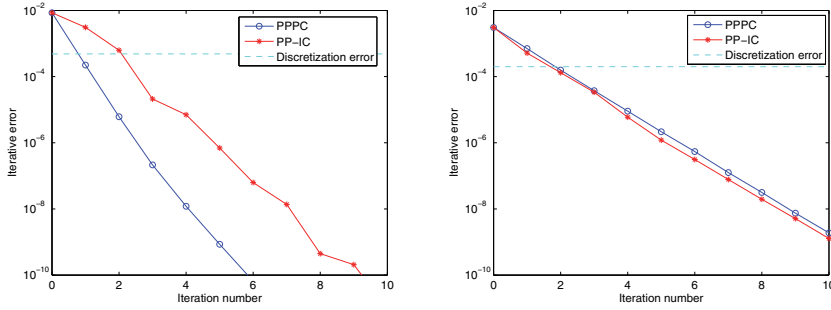


FIG. 6.5. Convergence of the PP-PC algorithm versus the PP-IC algorithm for the nonlinear model problem (5.1) and  $a = 1$  on the left and  $a = 10$  on the right with  $T = 1$ ,  $\Delta T = 1/10$ , and  $\Delta t = 1/200$  in each case.

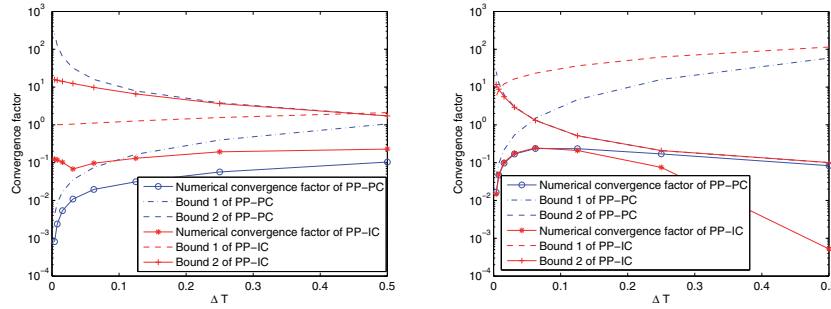


FIG. 6.6. Dependence of the convergence factor on  $\Delta T$  for the nonlinear model problem (5.1) and  $a = 1$  on the left and  $a = 10$  on the right with  $T = 1$  and  $\Delta t = 1/512$  in each case.

$$\mathbf{U}^{k+1,s+1} = \begin{bmatrix} 0 & & & & & & G(T_N, T_{N-1}, \cdot) \\ G(T_1, T_0, \cdot) & 0 & & & & & \\ & G(T_2, T_1, \cdot) & 0 & & & & \\ & & & \ddots & & & \\ & & & & \ddots & & \\ & & & & & G(T_{N-1}, T_{N-2}, \cdot) & \\ & & & & & & 0 \end{bmatrix} \times \mathbf{U}^{k+1,s} + \mathbf{b}^{k+1},$$

where  $\mathbf{U}^{k+1,0}$  is the initial guess. Because of Assumption 5.1 on the function  $f(u, t)$  in the nonlinear problem (5.1), the above fixed point iteration is convergent.

We show in Figure 6.5 the numerical experiments with the PP-PC and PP-IC algorithms. Here we use for the  $F$ -propagator backward Euler with time step  $\Delta t$  on each subinterval. We see again that the PP-PC algorithm converges faster than the PP-IC algorithm.

Next, we show in Figure 6.6 the numerical experiments for comparing the estimates obtained in section 5 with the numerical convergence factors. For the PP-PC algorithm, “Bound 1 of PP-PC” is the bound from Theorem 5.2, and “Bound 2 of PP-PC” is the bound from Theorem 5.3, while for the PP-IC algorithm, “Bound 1 of PP-IC” is the bound from Theorem 5.4, and “Bound 2 of PP-IC” is the bound from Theorem 5.5. These numerical experiments illustrate well our analysis in section 5.

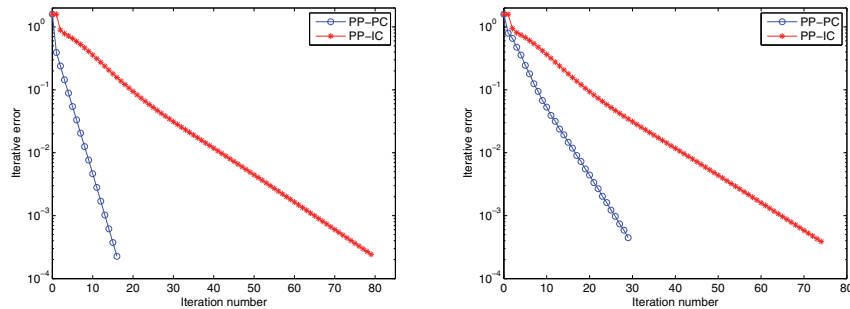


FIG. 6.7. Convergence of the PP-PC algorithm versus the PP-IC algorithm for the RFR model and 20 subintervals on the left and 40 subintervals on the right with  $T = 2$  and  $\Delta t = 1/200$  in each case.

**6.4. Nonlinear cooled reverse-flow reactor model.** A reverse-flow reactor (RFR) is a packed-bed reactor in which the flow direction is periodically reversed to trap a hot zone within the reactor. The model is taken from [28] and is given by the coupled system of nonlinear PDEs

$$\begin{aligned}\theta_t &= K_1\theta_{xx} - K_2\theta_x + K_3g(\theta)(1 - \chi) + K_4(1 - \theta), \\ \chi_t &= K_5\chi_{xx} - K_6\chi_x + K_7g(\theta)(1 - \chi)\end{aligned}$$

with boundary conditions

$$\begin{aligned}K_1\theta_x(t, 0) &= K_2(\theta(t, 0) - 1), K_5\chi_x(t, 0) = K_6\chi(t, 0), \\ \theta_x(t, 1) &= 0, \chi_x(t, 1) = 0,\end{aligned}$$

where  $\theta(t, x) : [0, \infty) \times [0, 1] \rightarrow [0, \infty)$  is the temperature and  $\chi(t, x) : [0, \infty) \times [0, 1] \rightarrow [0, 1]$  is the conversion. The constants  $K_j$ ,  $j = 1, 2, \dots, 7$ , are the physical parameters that appear in the model, and the function  $g(\theta(t, x))$  is nonlinear. The specific values from [28] are

- $K_1 = 6.9393 \times 10^{-4}$ ,  $K_2 = 0.1749$ ,  $K_3 = 1.5577 \times 10^{-6}$ ,  $K_4 = 0.0175$ ,  
 $K_5 = 2.4038 \times 10^{-3}$ ,  $K_6 = 174.06$ ,  $K_7 = 0.01$ ,
- $g(\theta) = \frac{1.6656 \times 10^{-5} e^{25.785(\theta-1)/\theta}}{1.6656 \times 10^{-5} + e^{-25.785/\theta}}$ .

A periodic state is a solution that satisfies

$$\begin{aligned}\theta(0, x) &= \theta(1, 1 - x), \\ \chi(0, x) &= \chi(1, 1 - x).\end{aligned}$$

Note that one cycle has a duration of two time units, and after each integer value of  $t$ , the flow direction reverses and the evolution equations and boundary conditions change accordingly.

We discretize in space using a centered finite difference scheme with mesh size  $\Delta x = 1/100$  for the diffusion terms and an upwind scheme for the advection terms. This results in a system of 202 time-periodic ODEs, which is then integrated in time by the fully implicit Newton's method with time step  $\Delta t = 1/200$ .

To apply the PP-PC algorithm and the PP-IC algorithm over one cycle, the time interval  $[0, 2]$  is first decomposed into  $N$  subintervals  $[T_n, T_{n+1}]$ ,  $n = 0, 1, \dots, N - 1$ , with  $\Delta T = 2/N$ , and then we take one step of backward Euler solved by Newton

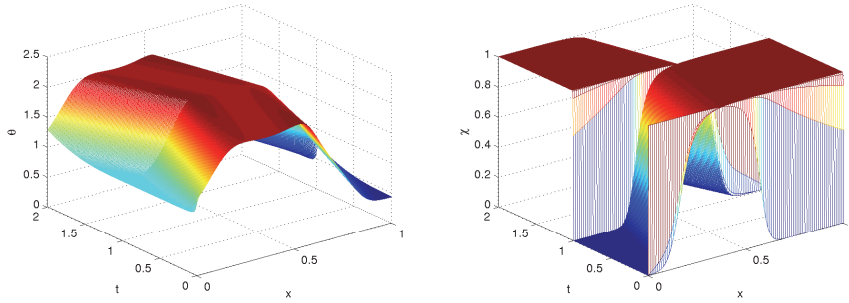


FIG. 6.8. The temperature  $\theta$  on the left and the conversion  $\chi$  on the right in a cycle for the RFR model by the PP-IC algorithm with 40 subintervals,  $T = 2$  and  $\Delta t = 1/200$ .

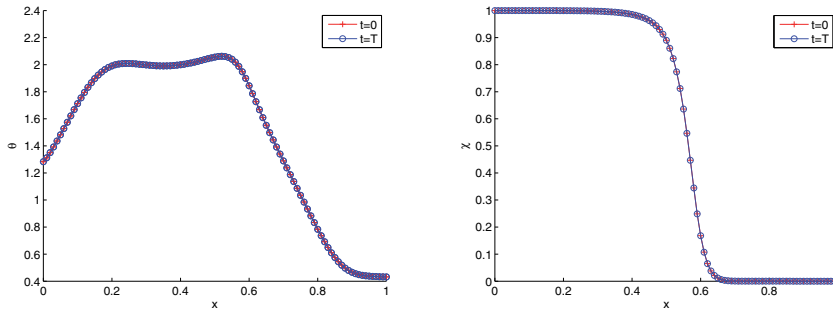


FIG. 6.9. The cross-section profile of the temperature  $\theta$  on the left and the conversion  $\chi$  on the right at  $t = 0$  and  $t = T$  for the RFR model by the PP-IC algorithm with 40 subintervals,  $T = 2$  and  $\Delta t = 1/200$ .

iterations for the  $G$ -propagator and  $M$  ( $M > 1$ , and  $NM = 400$ ) steps for the  $F$ -propagator. The initial guess for both parareal algorithms is chosen to be constant with  $\theta^0(x, T_n) = 2.02$  and  $\chi^0(x, T_n) = 0.7$  for  $n = 0, 1, \dots, N$ .

Figure 6.7 shows convergence results of the PP-PC and PP-IC algorithms for the RFR; we see that the PP-PC algorithm converges much more quickly than the PP-IC algorithm. The temperature  $\theta$  and the conversion  $\chi$  over one cycle are shown in Figure 6.8. Furthermore, from Figure 6.9, we can observe that the differences for both the temperature  $\theta$  and the conversion  $\chi$  between  $t = 0$  and  $t = 2$  are very small (in fact less than  $10^{-8}$ ), which indicates that the periodicity is satisfied for both the temperature and the conversion over one cycle.

**7. Conclusion.** We designed and analyzed two new parareal algorithms for time-periodic problems, the PP-PC and PP-IC algorithms. We considered linear and nonlinear scalar problems and also a diffusion problem, and we proved sharp bounds on convergence factors of both algorithms. The bounds of the convergence factors of the PP-PC algorithm were obtained in closed form, while the bounds for the PP-IC algorithm were obtained by some monotone iteration. In particular, we found that for the scalar problem, when  $a\Delta T$  is small, the PP-PC algorithm is better than the PP-IC algorithm, while when  $a\Delta T$  is large, the PP-IC algorithm is better. It is interesting to note that the new parareal algorithms for time-periodic problems do not have any superlinear convergence regime, in sharp contrast to the classical

parareal algorithm for initial value problems. It would be interesting to investigate if then acceleration methods could be used, such as Krylov subspace methods, for which interior variants were investigated in [5, 12]. It would also be interesting to develop the more recent PARAEXP [7, 8] algorithm for the time parallel solution of linear time-periodic problems.

**Acknowledgments.** The third author gratefully thanks the University of Geneva for hosting his visit.

## REFERENCES

- [1] F. BACHINGER, U. LANGER, AND J. SCHÖBERL, *Efficient solvers for nonlinear time-periodic eddy current problems*, *Comput. Visual. Sci.*, 4 (2006), pp. 197–207.
- [2] G. BAL, *On the convergence and the stability of the parareal algorithm to solve partial differential equations*, in *Proceedings of the 15th International Domain Decomposition Conference*, R. Kornhuber, R. H. W. Hoppe, J. Périaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., *Lecture Notes in Comput. Sci. Engrg.* 40, Springer, Berlin, 2003, pp. 426–432.
- [3] G. BAL AND Y. MADAY, *A “parareal” time discretization for non-linear PDE’s with application to the pricing of an American put*, in *Recent Developments in Domain Decomposition Methods*, *Lect. Notes in Comput. Sci. Engrg.* 23, Springer, Berlin, 2002, pp. 189–202.
- [4] C. BERNARDI, *Numerical approximation of a periodic linear parabolic problem*, *SIAM J. Numer. Anal.*, 19 (1982), pp. 1196–1207.
- [5] C. FARHAT AND M. CHANDESRI, *Time-decomposed parallel time-integrators: Theory and feasibility studies for fluid, structure, and fluid-structure applications*, *Internat. J. Numer. Methods Engrg.*, 58 (2003), pp. 1397–1434.
- [6] P. F. FISCHER, F. HECHT, AND Y. MADAY, *A parareal in time semi-implicit approximation of the Navier-Stokes equations*, in *Proceedings of the 15th International Domain Decomposition Conference*, R. Kornhuber, R. H. W. Hoppe, J. Périaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., *Lecture Notes in Comput. Sci. Engrg.* 40, Springer, Berlin, 2003, pp. 433–440.
- [7] S. GÜTTEL, *A parallel overlapping time-domain decomposition for ODEs*, *Proceedings of the 20th International Conference on Domain Decomposition Methods*, 2012.
- [8] M. J. GANDER AND S. GÜTTEL, *PARAEXP: A parallel integrator for linear initial-value problems*, *SIAM J. Sci. Comput.*, 35 (2013), pp. C123–C142.
- [9] M. J. GANDER AND S. VANDEWALLE, *Analysis of the parareal time-parallel time-integration method*, *SIAM J. Sci. Comput.*, 29 (2007), pp. 226–578.
- [10] M. J. GANDER AND E. HAIRER, *Nonlinear convergence analysis for the parareal algorithm*, in *Domain Decomposition Methods in Science and Engineering XVII*, *Lecture Notes in Comput. Sci. Engrg.* 60, Springer-Verlag, Berlin, 2007, pp. 45–56.
- [11] M. J. GANDER, *Analysis of the parareal algorithm to hyperbolic problems using characteristics*, in *Boletín de la Sociedad Española de Matemática Aplicada*, 2008, pp. 5–19.
- [12] M. J. GANDER AND M. PETCU, *Analysis of a Krylov subspace enhanced parareal algorithm for linear problems*, *ESAIM Proc.*, 25 (2008), pp. 114–129.
- [13] M. J. GANDER AND E. HAIRER, *Analysis for parareal algorithms applied to Hamiltonian differential equations*, *J. Comput. Appl. Math.*, (2013), DOI:10.1016/j.cam.2013.01.011.
- [14] R. M. GRAY, *Toeplitz and Circulant Matrices: A Review*, Now Publishers, Boston, 2005.
- [15] W. HACKBUSCH, *Fast numerical solution of time-periodic parabolic problems by a multigrid method*, *SIAM J. Sci. Statist. Comput.*, 2 (1981), pp. 198–206.
- [16] Y. L. JIANG, *Waveform Relaxation Methods*, Science Press, Beijing, 2009.
- [17] Y. L. JIANG, *Model Order Reduction Methods*, Science Press, Beijing, 2010.
- [18] Y. L. JIANG, R. M. M. CHEN, AND O. WING, *Periodic waveform relaxation of nonlinear dynamic systems by quasi-linearization*, *IEEE Trans. Circuits Syst. I*, 50 (2003), pp. 589–593.
- [19] Y. L. JIANG, *Periodic waveform relaxation of nonlinear dynamic equations*, *Appl. Math. Comput.*, 135 (2003), pp. 219–226.
- [20] Y. L. JIANG AND R. M. CHEN, *Computing periodic solutions of linear differential-algebraic equations by waveform relaxation*, *Math. Comput.*, 74 (2005), pp. 781–804.
- [21] 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.

- [22] Y. MADAY AND G. TURINICI, *A parareal in time procedure for the control of partial differential equations*, C. R. Math. Acad. Sci. Paris. Sér. I Math., 335 (2002), pp. 387–391.
- [23] Y. MADAY AND G. TURINICI, *Monotonic parareal control for quantum systems*, SIAM J. Numer. Anal., 45 (2007), pp. 2468–2482.
- [24] Y. MADAY AND G. TURINICI, *The parareal in time iterative solver: A further direction to parallel implementation*, in Proceedings of the 15th International Domain Decomposition Conference, R. Kornhuber, R. H. W. Hoppe, J. Périaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., Lecture Notes in Comput. Sci. Engrg. 40, Springer, Berlin, 2003, pp. 441–448.
- [25] C. V. PAO, *Numerical methods for time-periodic solutions of nonlinear parabolic boundary value problems*, SIAM J. Numer. Anal., 39 (2001), pp. 647–667.
- [26] G. A. STAFF AND E. M. RØNQUIST, *Stability of the parareal algorithm*, in Proceedings of the 15th International Domain Decomposition Conference, R. Kornhuber, R. H. W. Hoppe, J. Périaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., Lecture Notes in Comput. Sci. Engrg. 40, Springer, Berlin, 2003, pp. 449–456.
- [27] M. STEUERWALT, *The existence, computation, and number of solutions of periodic parabolic problems*, SIAM J. Numer. Anal., 16 (1979), pp. 402–420.
- [28] T. L. VAN NOORDEN, S. M. VERDUYN LUNEL, AND A. BLIEK, *The efficient computation of periodic states of cyclically operated chemical processes*, IMA J. Appl. Math., 68 (2003), pp. 149–166.
- [29] S. VANDEWALLE AND R. PIESENS, *On dynamic iteration methods for solving time-periodic differential equations*, SIAM J. Numer. Anal., 30 (1993), pp. 286–303.
- [30] B. A. VAN DE ROTTEN, S. M. VERDUYN LUNEL, AND A. BLIEK, *Efficient simulation of periodically forced reactors with radial gradients*, Chem. Engrg. Sci., 61 (2006), pp. 6981–6994.
- [31] R. S. VARGA, *Matrix Iterative Analysis*, 2nd ed., Springer-Verlag, Berlin, 2000.
- [32] 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.