COMMUNICATIONS IN COMPUTATIONAL PHYSICS
Vol. **6**, No. 4, pp. 883-902

Commun. Comput. Phys.
October 2009

Parareal-Richardson Algorithm for Solving Nonlinear ODEs and PDEs

Shulin Wu, Baochang Shi and Chengming Huang*

School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan 430074, China.

Received 28 September 2008; Accepted (in revised version) 11 March 2009

Available online 6 April 2009

Abstract. The parareal algorithm, proposed firstly by Lions *et al.* [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], is an effective algorithm to solve the time-dependent problems parallel in time. This algorithm has received much interest from many researchers in the past years. We present in this paper a new variant of the parareal algorithm, which is derived by combining the original parareal algorithm and the Richardson extrapolation, for the numerical solution of the nonlinear ODEs and PDEs. Several nonlinear problems are tested to show the advantage of the new algorithm. The accuracy of the obtained numerical solution is compared with that of its original version (i.e., the parareal algorithm based on the same numerical method).

AMS subject classifications: 65Y05, 65Y10, 65Y20, 37M05, 65-05

Key words: Parallel computation, parareal algorithm, Richardson extrapolation, accuracy, nonlinear problems.

1 Introduction

In the seminal paper [24] the concept of a new domain decomposition for the numerical solution of time-dependent problems, the parareal algorithm, was proposed by Lions, Maday and Turinici. The name, *parareal*, was chosen to indicate that the algorithm is constructed to compute simultaneously in time the solution of evolution problems whose solution cannot be obtained in real time using one processor only. The method has received much interest from many researchers in the past years, especially in the area of domain decompositions, see, e.g., [21]. Many excellent results about this algorithm have been obtained and below we will make a brief retrospection.

*Corresponding author. *Email addresses:* wushulin_ylp@163.com (S. L. Wu), sbchust@yahoo.com (B. C. Shi), chengming_huang@hotmail.com (C. M. Huang)

The idea targeted a large scale time parallelism of simulating the evolution problems was proposed already in 1964 by Nievergelt [29]. The idea of Nievergelt eventually developed into the well known multiple shooting method for boundary value problems [20]. For much more investigation in this direction we refer the interested reader to [4, 5, 22]. Later in 1967, Miranker and Liniger [26] introduced a family of naturally Runge-Kutta methods for small scale parallelism coupled with predictor-corrector strategies. The parallelization of these methods lays on that, the prediction and correction steps can be performed simultaneously over several time steps.

The parareal algorithm was first introduced in [24] and an improved version was given by Bal and Maday in [2]. Some further improvements and understanding, as well as new applications of this algorithm, were investigated by Baffico *et al.* in [3] and Maday and Turinici in [27,28]. Its stability was investigated in [1,30]. Recently, several variants of this algorithm have been proposed in [6, 10, 14] and extensive experiments can be found for fluid and structure problems in [6, 10], for the Navier-Stokes equations in [7, 8], for reservoir simulation in [11], and for various nonlinear problems, such as Brusselator, Arenstorf orbit and viscous Burgers' equation etc. in [13].

We pay special attention to the recent results presented by Gander and Vandewalle in [12]. In that paper, the relation of the parareal algorithm to the space-time multigrid methods [16,18,19,25,31–33] and multiple shooting methods was first investigated. It has been shown that the parareal algorithm can be regarded as the practical implementation of the multiple shooting and time-multigrid methods. The new convergence results that show superlinear convergence of the algorithm on bounded time intervals and linear convergence on unbounded intervals were also presented in that paper. It also provides a up-to-date historical review and references in this field.

Along the lines of [12], we investigate in this paper a new variant of the parareal algorithm, namely Parareal-Richardson algorithm, for the time dependent problems. The new algorithm is derived by combining the original parareal algorithm and the Richardson extrapolation, and hence the accuracy of the numerical solution obtained by the proposed algorithm is higher than that of the original parareal algorithm. The aim of this paper is to show the advantages of this new algorithm in terms of the accuracy when applied to nonlinear ODEs and PDEs. Moreover, the advantages with respect to the stability and convergence rate for the proposed algorithm are presented.

The remainder of this paper is organized as follows. The Parareal-Richardson algorithm is described in detail in Section 2. We also discuss the stability property and the accuracy of the Parareal-Richardson scheme. It is demonstrated that for some one-step numerical methods, the stability region of the Parareal-Richardson algorithm is larger than that of the parareal algorithm. In Section 3, we apply the parareal and Parareal-Richardson algorithms to several classical nonlinear ODEs and PDEs, and to demonstrate that the Parareal-Richardson algorithm is more flexible and outperforms the parareal algorithm. In Section 4, we discuss the effects of the parameters used in the proposed algorithm to the convergence speed. In Section 5, we give some conclusions of this paper and discuss future directions of the research for the Parareal-Richardson algorithm.

2 Derivation of the Parareal-Richardson algorithm

In this section, we first review the parareal algorithm and the idea of the Richardson extrapolation for systems of ordinary differential equations of the form

$$\begin{cases} y'(t)=f(t,y(t)), & t\in[0,T], \\ y(0)=y_0, \end{cases} \tag{2.1}$$

where $f:\mathbb{R}\times\mathbb{R}^m\rightarrow\mathbb{R}^m$ and $y:\mathbb{R}\rightarrow\mathbb{R}^m$. The Parareal-Richardson algorithm is then derived naturally.

2.1 The parareal algorithm

The parareal algorithm is defined using two propagation operators G and F. The operator $G(t_n,y_n)$ provides a coarse approximation to $y(t_{n+1})$ and operator $F(t_n,y_n)$ provides a more accurate approximation of $y(t_{n+1})$. To start the algorithm, we need initial approximations Y_n^0 of $y(t)$ at time t_n ($n=0,1,\cdots,N$) which are given, e.g., by the sequential computation of $Y_{n+1}^0=G(t_n,Y_n^0)$ with $Y_0^0=y_0$. Then in each subinterval $[t_n,t_{n+1}]$ we perform the parallel computation $F(t_n,Y_n^0)$ to obtain a more accurate approximations of $y(t)$ at time t_n. At last, we perform sequentially, for $k=0,1,2,\cdots$, the correction iteration

$$Y_{n+1}^{k+1}=G(t_n,Y_n^{k+1})+F(t_n,Y_n^k)-G(t_n,Y_n^k). \tag{2.2}$$

It is obvious that for $k\rightarrow+\infty$ the parareal algorithm (2.2) generates a series of values Y_n that satisfy $Y_{n+1}=F(t_n,Y_n)$ provided the iterative process is convergent. This indicates that the approximations at the time t_n will have achieved the accuracy of the $F-$propagator.

2.2 The Richardson extrapolation

For an one-step method which we write, in Henrici's notation [15], as

$$y_{n+1}=y_n+h\Phi(t_n,y_n,h), \quad n=0,1,\cdots, \tag{2.3}$$

we assume that the method is of order p. Then with $(n+1)h$ being a constant number (maybe h is varying, but $(n+1)h$ is constant), suppose that starting from y_n the error between y_{n+1} and the exact solution $y(t)$ of (2.1) at time $t_{n+1}=t_0+(n+1)h$ takes an asymptotic expansion of the form

$$y(t_{n+1})-y_{n+1}=e_p(t_{n+1})h^p+e_q(t_{n+1})h^q+\mathcal{O}(h^{q+1}), \tag{2.4}$$

where $q>p$.

Let

$$y_{n+1}=\Psi(t_n,y_n,h)=y_n+h\Phi(t_n,y_n,h) \tag{2.5}$$

and $y^*_{n+1}=\Psi_M(t_n,y_n,h/M)$ be the approximation of $y(t)$ at t_{n+1} obtained by the method Φ with step size h/M after M steps. We then have the following global asymptotic expansion at time t_{n+1} :

$$y(t_{n+1})=y_{n+1}+e_p(t_{n+1})h^p+e_q(t_{n+1})h^q+\mathcal{O}(h^{q+1}), \tag{2.6a}$$

and

$$y(t_{n+1})=y^*_{n+1}+e_p(t_{n+1})(h/M)^p+e_q(t_{n+1})(h/M)^q+\mathcal{O}\Big((h/M)^{q+1}\Big). \tag{2.6b}$$

Define

$$\alpha=\frac{1}{1-M^p},\quad \beta=\frac{M^p}{M^p-1}. \tag{2.7}$$

It then follows by multiplying (2.6a) and (2.6b) with α and β respectively that

$$y(t_{n+1})=\alpha y_{n+1}+\beta y^*_{n+1}+e_q(t_{n+1})h^q\left(\alpha+\beta M^{-q}\right)+\mathcal{O}(h^{q+1}). \tag{2.8}$$

Define

$$Y_{n+1}=\alpha\Psi(t_n,Y_n,h)+\beta\Psi_M(t_n,Y_n,h/M),\quad Y_0=y_0. \tag{2.9}$$

Then we have

$$y(t_{n+1})=Y_{n+1}+e_q(t_{n+1})h^q\left(\alpha+\beta M^{-q}\right)+\mathcal{O}(h^{q+1}). \tag{2.10}$$

From (2.10), we know that the numerical solution Y_n may obtain higher order of accuracy at cost of much more computation time. To overcome this disadvantage, we consider parallel computation of $\Psi_M(t_n,y_n,h/M)$ in each subinterval $[t_n,t_{n+1}]$. To this end, we adopt the idea of the parareal algorithm and therefore introduce the following algorithm.

Algorithm 2.1: Parareal-Richardson Algorithm

```
• Initialization: Perform sequential computation $Y^0_{n+1}=\Psi(t_n,Y^0_n,h)$ with $Y^0_0=y_0$, $n=0,\cdots,N-1$;
• For $k=0,1,\cdots,$
    1. Perform simultaneously in each subinterval $[t_n,t_{n+1}]$ the computation
           $\hat{Y}_{n+\frac{m+1}{M}}=\Psi\left(t_{n+m/M},\hat{Y}_{n+m/M},h/M\right)$
       with initial value $Y^k_n$, where $t_{n+m/M}=t_n+mh/M$ and $m=0,1,\cdots,M-1$;
    2. Perform sequential corrections
           $Y^{k+1}_{n+1}=(\alpha+\gamma)\Psi(t_n,Y^{k+1}_n,h)+\beta\hat{Y}_{n+1}-\gamma\Psi(t_n,Y^k_n,h)$
       with $Y^{k+1}_0=y_0$, $n=0,1,\cdots,N-1$;
    3. If for $n=0,1,\cdots,N-1$, $Y^{k+1}_n$ satisfy some termination criteria, stop the iteration; else go
       to step 1.
```

The above algorithm can be written compactly as

$$Y_{n+1}^{k+1}=(\alpha+\gamma)\Psi(t_n,Y_n^{k+1},h)+\beta\Psi_M\left(t_n,Y_n^k,h/M\right)-\gamma\Psi(t_n,Y_n^k,h), \tag{2.11}$$

where k is the iteration index and the parameter γ is the relaxation factor which is chosen to speedup the convergence. Note that for $k\to+\infty$ method (2.11) will upon convergence generates a series of values Y_n which satisfy

$$Y_{n+1}=\alpha\Psi(t_n,Y_n,h)+\beta\Psi_M(t_n,Y_n,h/M).$$

This implies that the converged solution $Y_n^k(k\to+\infty)$ obtained by the Parareal-Richardson algorithm at time t_n will achieve the accuracy of the one defined by (2.9).

We note that both the parareal and Parareal-Richardson algorithms can be illustrated in Fig. 1: the symbols $'\mathbf{F}'$ and $'\mathbf{G}'$ stand for the finer propagation operator F and the coarse propagation operator G, respectively; the symbol $'\mathbf{C}'$ denotes some combination of the computed values $G(t_n,y_{n-1}^{k+1}),F(t_n,y_{n-1}^k)$ and $G(t_n,y_{n-1}^k)$ at iteration k — for the parareal algorithm the combination is (2.2) and for the Parareal-Richardson algorithm the combination is (2.11). In Fig. 1, at iteration k the finer approximations z_n^k are computed simultaneously.

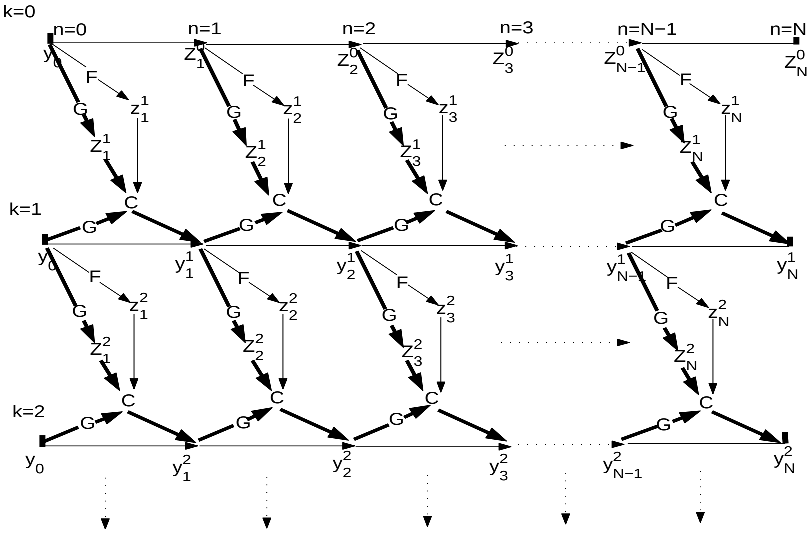

Figure 1: Illustration of the parareal and Parareal-Richardson algorithms.

Remark 2.1. If we set $\alpha=0,\beta=1,\gamma=1$ in (2.11), the Parareal-Richardson algorithm reduces to the parareal algorithm

$$Y_{n+1}^{k+1}=\Psi(t_n,Y_n^{k+1},h)+\Psi_M\left(t_n,Y_n^k,h/M\right)-\Psi(t_n,Y_n^k,h), \tag{2.12}$$

which is a special case of the parareal algorithm (for the general parareal algorithm, the propagators F and G can be based on different numerical method; see [12] for more details). Therefore, we can regard the Parareal-Richardson algorithm as the "weighted" version of the parareal algorithm. Also, it can be regarded as the parallel implementation of the Richardson extrapolation (2.9). Moreover, we note that the Parareal-Richardson algorithm can be regarded as a special case of the original parareal algorithm if the propagation operators F and G are replaced by

$$\hat{F}(t,y)=\alpha\Psi(t,y,h)+\beta\Psi_M(t,y,h/M),\hat{G}(t,y)=(\alpha+\gamma)\Psi(t,y,h) \tag{2.13}$$

in (2.2), respectively. We have not found this kind of modification in the literature of this field and our aim in this paper is to demonstrate some advantages of the new algorithm compared with its original version (2.12).

Remark 2.2. It is clear that at every iteration the storage and the computation time costed by the Parareal-Richardson algorithm equals to the parareal algorithm, while the Parareal-Richardson algorithm takes the advantage of higher accuracy. Moreover, for $M\gg 1$ we find $\alpha\approx 0$ and $\beta\approx 1$, and thus if $\gamma=1$ we may expect that the convergence speed of the Parareal-Richardson algorithm approaches to that of the parareal algorithm. We will validate this in Section 3 by several classical nonlinear ODEs and PDEs.

Remark 2.3. The convergence condition and the convergence speed of the parareal algorithm were investigated in [12] by the model problem

$$\begin{cases} y'(t)=\lambda y, \ \lambda\in\mathbb{C}, \\ y(0)=y_0. \end{cases} \tag{2.14}$$

Following along the lines of [12], we have studied the convergence condition and the convergence speed of the Parareal-Richardson algorithm in [34] by using the model problem (2.14), where we have proved that the algorithm converges superlinearly on any bounded time interval and only linearly on unbounded time intervals. Moreover, under some suitable condition there exists an optimal relaxation factor γ by which the Parareal-Richardson algorithm converges to the converged solution with only one iteration.

Remark 2.4. From [12], we know that the general parareal algorithm (2.2) is stable if it holds that

$$\mathcal{K}(z)=\frac{|\mathcal{R}_f(z)-\mathcal{R}(z)|}{1-|\mathcal{R}(z)|}<1 \text{ and } |\mathcal{R}(z)|<1, \ z\in\mathbb{C}, \tag{2.15}$$

where $\mathcal{R}(z)$ and $\mathcal{R}_f(z)$ are the stability functions of the G-propagator and F-propagator, respectively. Therefore, from Remark 2.1 the stability region of the Parareal-Richardson algorithm and its original version (2.12) can be written as

$$\mathcal{D}_{PR}=\left\{z\Big|\frac{|\beta\mathcal{R}^M(\frac{z}{M})-\gamma\mathcal{R}(z)|}{1-|(\alpha+\gamma)\mathcal{R}(z)|}<1 \text{ and } |\mathcal{R}(z)|<\frac{1}{|\alpha+\gamma|}, z\in\mathbb{C}\right\}, \tag{2.16}$$

and

$$\mathcal{D}=\left\{z\Big|\frac{|\mathcal{R}^M(\frac{z}{M})-\mathcal{R}(z)|}{1-|\mathcal{R}(z)|}<1 \text{ and } |\mathcal{R}(z)|<1,\, z\in\mathbb{C}\right\}, \tag{2.17}$$

respectively. For some one step numerical methods, the stability region of the Parareal-Richardson algorithm is larger than the one of the parareal algorithm. To illustrate this, for $\gamma=1$ and $M=2$, we plot the stability region of these two algorithms coupled with the forward Euler method, the 2 stage-order 2 method used in our paper and the trapezoidal rule, in the left, middle and right panels of Fig. 2, respectively.

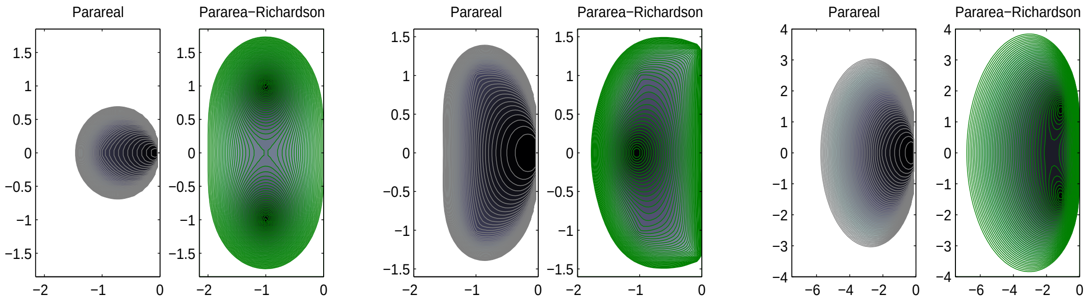

Figure 2: Stability region of the two algorithms coupled with the forward Euler (left), the 2 stage-order 2 method (right) and the trapezoidal rule (middle).

We close this section by analyzing the accuracy of the Parareal-Richardson algorithm. From (2.10) it is easy to know that

$$y(t_{n+1})-Y_{n+1}=\mathcal{O}\left(\left(\frac{1}{1-M^p}+\frac{M^p}{M^q(M^p-1)}\right)h^q\right). \tag{2.18}$$

Therefore, the Parareal-Richardson algorithm will upon convergence generate a series of values Y_n which achieve the accuracy of the right-hand side of (2.18), while the converged solution obtained by the parareal algorithm will have only achieved accuracy $\mathcal{O}((h/M)^p)$. Consequently, by using the Parareal-Richardson algorithm, the improvement of accuracy of the numerical solution is

$$\mathcal{O}\left(\left(\frac{M^p}{1-M^p}+\frac{M^{2p}}{M^q(M^p-1)}\right)h^{q-p}\right)=\mathcal{O}\left(\frac{M^p}{1-M^p}\left(1-\frac{1}{M^{q-p}}\right)h^{q-p}\right). \tag{2.19}$$

Obviously, for $M\gg1$ the improvement of accuracy approaches to $\mathcal{O}(h^{q-p})$.

3 Numerical results

To perform the parareal and Parareal-Richardson algorithms, we use the following Runge-Kutta methods.

- $\boxed{\textbf{2 stage-order 2 RK}}$:

$$A=\begin{bmatrix}0 & 0\\ \frac{1}{2} & 0\end{bmatrix},\quad b=(0,1); \tag{3.1}$$

- $\boxed{\textbf{3 stage-order 2 RK}}$:

$$A=\begin{bmatrix}0 & 0 & 0\\ \frac{1}{2} & 0 & 0\\ 0 & 1 & 0\end{bmatrix},\quad b=\left(\frac{1}{4},\frac{1}{2},\frac{1}{4}\right); \tag{3.2}$$

- $\boxed{\textbf{3 stage-order 3 RK}}$:

$$A=\begin{bmatrix}0 & 0 & 0\\ \frac{2}{3} & 0 & 0\\ \frac{1}{6} & \frac{1}{2} & 0\end{bmatrix},\quad b=\left(\frac{1}{4},\frac{1}{4},\frac{1}{2}\right). \tag{3.3}$$

For nonlinear ODEs (2.1), the step sizes for the coarse and the fine approximation are ΔT and $\Delta T/M$, respectively. Moreover, we use a very accurate numerical solution obtained using MATLAB's `ode45` solver with minimal tolerances (`AbsTol`$=1\times10^{-14}$ and `RelTol`$=1\times10^{-15}$), and we denote the solution at time t_n by $\bar{y}_n$ $(n=0,1,\cdots,T/\Delta T)$. We denote the converged numerical solution obtained by the Parareal-Richardson algorithm and its original version by $\tilde{Y}_n$ and Y_n, respectively. The numerical solution at time t_n and iteration k obtained by the Parareal-Richardson algorithm and its original version is denoted by $\tilde{Y}_n^k$ and Y_n^k, respectively. We therefore introduce the algebraic error with iteration index k as

$$\mathrm{Err}_k=\max_{n=0,1,\cdots,\frac{T}{\Delta T}}\|Y_n-Y_n^k\|_\infty \tag{3.4}$$

and

$$\widetilde{\mathrm{Err}}_k=\max_{n=0,1,\cdots,\frac{T}{\Delta T}}\|\tilde{Y}_n-\tilde{Y}_n^k\|_\infty, \tag{3.5}$$

for the parareal and Parareal-Richardson algorithms, respectively. The discretization error at time t_n of the parareal and Parareal-Richardson algorithms is defined as

$$\mathrm{DErr}_n=\|\bar{y}_n-Y_n\|_\infty \tag{3.6}$$

and

$$\widetilde{\mathrm{DErr}}_n=\|\bar{y}_n-\tilde{Y}_n\|_\infty, \tag{3.7}$$

respectively. Both algorithms terminate when the algebraic error is less than 10^{-12}.

Example 3.1. (***Lotka-Volterra system***) Consider the following system

$$\begin{cases}x'(t)=x(1-y),\\ y'(t)=-y(1-x).\end{cases} \tag{3.8}$$

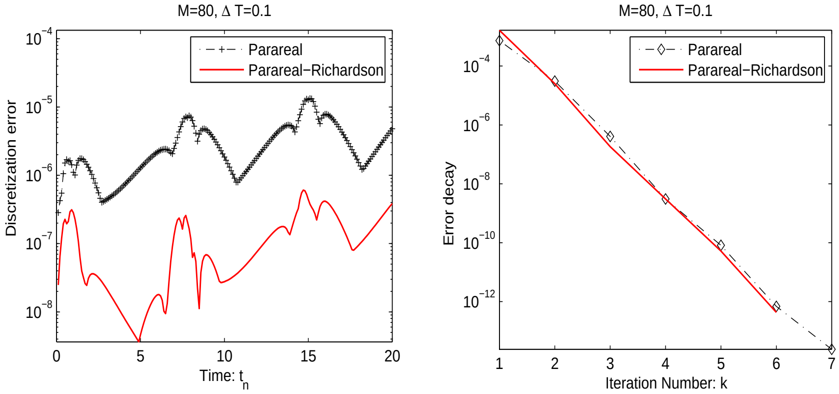

Figure 3: 2 stage-order 2 RK method (3.1): Left: discretization errors DErr_n and $\widetilde{\text{DErr}}_n$; Right: algebraic error decay Err_k and $\widetilde{\text{Err}}_k$.

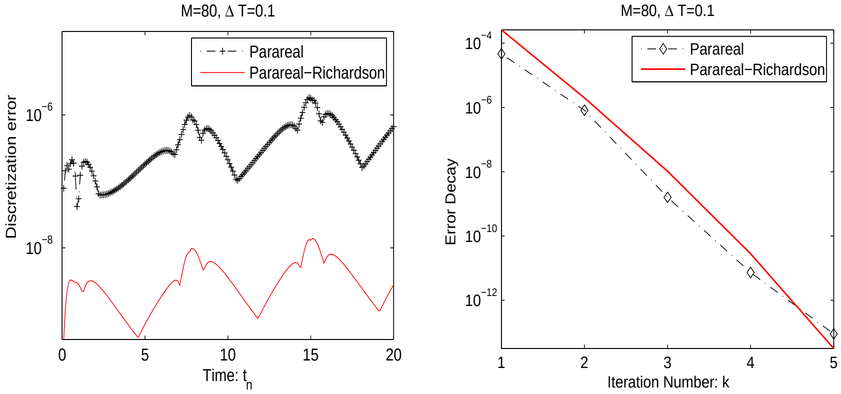

Figure 4: 3 stage-order 2 RK method (3.2):Left: discretization errors DErr_n and $\widetilde{\text{DErr}}_n$; Right: algebraic error decay Err_k and $\widetilde{\text{Err}}_k$.

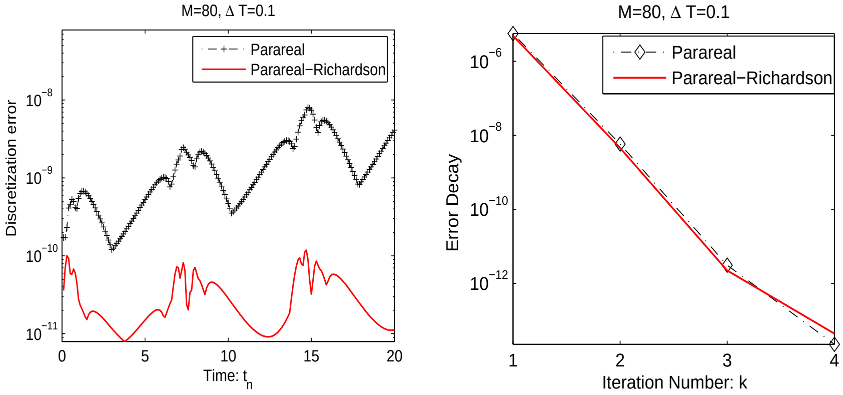

Figure 5: 3 stage-order 3 RK method (3.3): Left: discretization errors DErr_n and $\widetilde{\text{DErr}}_n$; Right: algebraic error decay Err_k and $\widetilde{\text{Err}}_k$.

To perform the parareal and Parareal-Richardson algorithms, we choose $T=20,\Delta T=0.1,M=80,\Delta t=\Delta T/M$. The discretization error at time t_n, the actual error decay for the above three RK methods (3.1)-(3.3) are plotted in Figs. 3-5, respectively.

We see clearly in these figures that the Parareal-Richardson algorithm performs much better than the original parareal algorithm in the sense of the accuracy of the converged solution. Moreover, from the right panels of these figures we see that the Parareal-Richardson algorithm converges with almost equal convergence speed of the parareal algorithm.

Example 3.2. (***HIRES Problem***) The HIRES model was proposed by Schafer *et al.* in 1975 [9]. The problem originates from plant physiology and describes how light is involved in morphogenesis. The equations are given by

$$\begin{cases} y_1'(t)=-1.71y_1+0.43y_2+8.32y_3+0.0007, \\ y_2'(t)=-1.71y_1-8.75y_2, \\ y_3'(t)=-10.03y_3+0.43y_4+0.035y_5, \\ y_4'(t)=8.32y_2+1.71y_3-1.12y_4, \\ y_5'(t)=-1.745y_5+0.43y_6+0.43y_7, \\ y_6'(t)=-280y_6y_8+0.69y_4+1.71y_5-0.43y_6+0.69y_7, \\ y_7'(t)=280y_6y_8-1.81y_7, \\ y_8'(t)=-280y_6y_8+1.81y_7, \end{cases} \tag{3.9}$$

with initial values $y_{1,\cdots,8}(0)=(1,0,0,0,0,0,0,0.0057)^T$.

For the HIRES problem (3.9), we choose $\Delta T=0.05,M=50,\Delta t=\Delta T/M$ to perform the parareal and Parareal-Richardson algorithms on time interval $[0,40]$. For this problem, we note that both the parareal and Parareal-Richardson algorithms coupled with the 2 stage-order 2 RK method (3.1) and the 3 stage-order 3 RK method (3.3) are not convergent, while both algorithms are convergent when coupled with the 3 stage-order 2 RK method (3.2). The computational results are plotted in Fig. 6, where one can see that the converged solution of the Parareal-Richardson algorithm is much more accurate than that of the parareal algorithm, while the convergence speed of this algorithm strictly equals to the parareal algorithm (see the right panel of Fig. 6).

Example 3.3. (***Burgers' Equation***) We finally show numerical experiments for the well known viscous Burger's equation:

$$u_t+f(u)_x=0, \quad (x,t)\in[0,1]\times[0,4], \tag{3.10}$$

where $f(u)=u^2/2$. Problem (3.10) is completed with initial-boundary conditions

$$u(x,0)=\begin{cases} 1, & \text{if } x\le\frac{1}{2}, \\ 0, & \text{if } \frac{1}{2}<x\le 1, \end{cases}$$

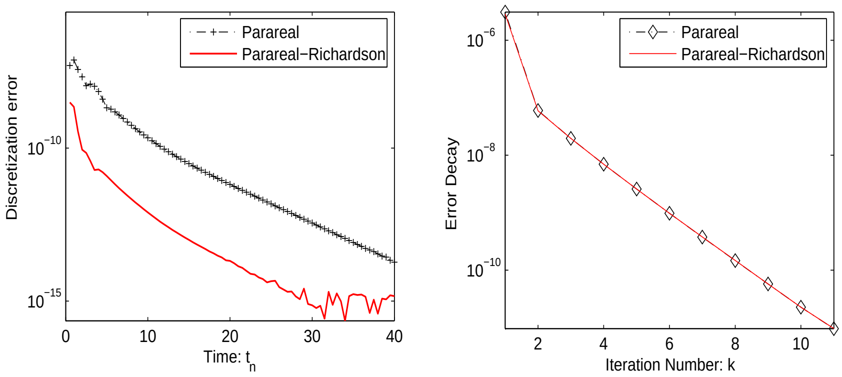

Figure 6: 3 stage-order 2 RK method (3.2): Left: discretization errors DErr_n and $\widetilde{\text{DErr}}_n$; Right: algebraic error decay Err_k and $\widetilde{\text{Err}}_k$.

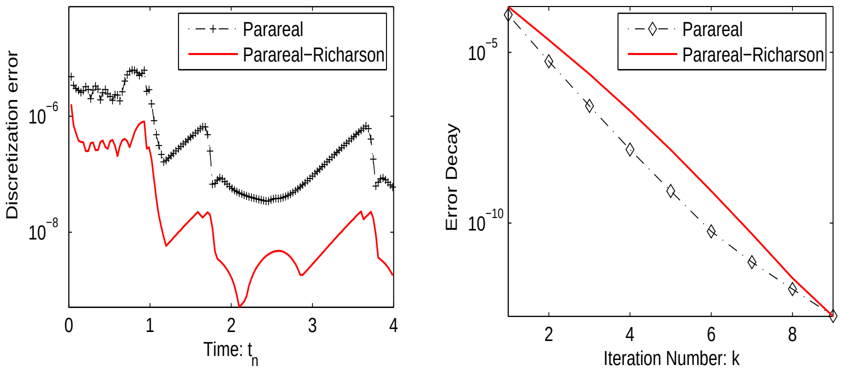

Figure 7: 2 stage-order 2 RK method (3.1): Left: discretization errors DErr_n and $\widetilde{\text{DErr}}_n$; Right: algebraic error decay Err_k and $\widetilde{\text{Err}}_k$.

and $u(0,t)=1-\frac{1}{2}\sin(\pi t)$.

The term $f(u)_x$ is discretized as $f(u)_x \approx \frac{f(u_i)-f(u_{i-1})}{\Delta x}$ at $x_i = i\Delta x$ with $\Delta x = 0.02$, $i = 1,2,\cdots,1/\Delta x$, and then we do the same experiment as in the previous examples with parameters $\Delta T = 0.008, M = 40, \Delta t = \Delta T/M$. The computational results are plotted in Figs. 7-9.

It is interesting that, as in the previous examples, the 3 stage-order 2 RK method (3.2) is still the best one, since in this case the advantage of the Parareal-Richardson algorithm in the sense of accuracy is much more significant; moreover, both algorithms converge with least iterations, as one can see in the right panels of the Figs. 7-9.

Remark 3.1. From the above numerical results, one can see clearly that the 3 stage-order 2 RK method (3.2) (see [15] on page 223) has significant advantage in the sense of achieving much better accuracy of the numerical solution when applied to the Parareal-Richardson algorithm. An exercise in this monograph asks to show that for this method "one step

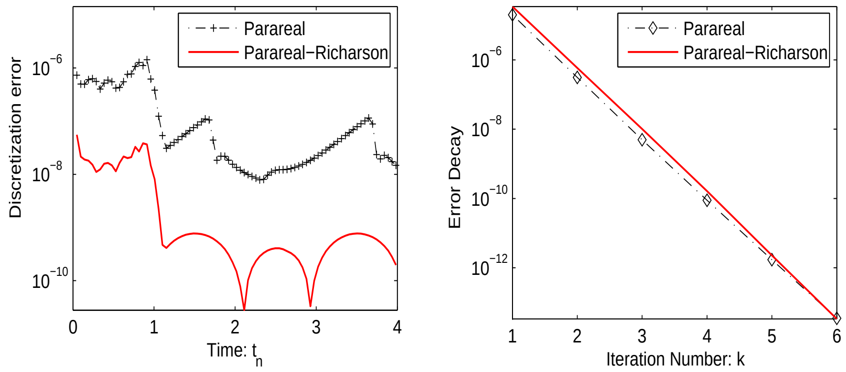

Figure 8: 3 stage-order 2 RK method (3.2): Left: discretization errors DErr_n and $\widetilde{\mathrm{DErr}}_n$; Right: algebraic error decay Err_k and $\widetilde{\mathrm{Err}}_k$.

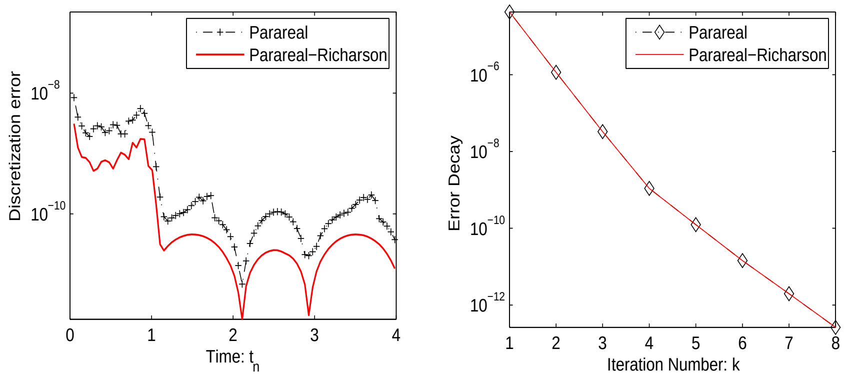

Figure 9: 3 stage-order 3 RK method (3.3): Left: discretization errors DErr_n and $\widetilde{\mathrm{DErr}}_n$; Right: algebraic error decay Err_k and $\widetilde{\mathrm{Err}}_k$.

of Richardson extrapolation increases the order of the method by two", and this explains why for this integrator the Parareal-Richardson algorithm gains a factor $(\Delta T)^2$ compared to the original parareal algorithm.

Remark 3.2. For $M=50$ and $\gamma=1$, the stability region of the parareal and Parareal-Richardson algorithms (defined by (2.16), (2.17)), coupled with the three RK methods (3.1)-(3.3) is shown in Figs. 10 and 11, respectively.

One can see clearly that the stability region of the two algorithms coupled with the 3 stage-order 2 RK method is larger than the case when coupled with the other two RK methods. This may interpret why in Example 3.2 both algorithms are not convergent when coupled with the 2 stage-order 2 RK method (3.1) and the 3 stage-order 3 RK method (3.3). Moreover, for $M=50$, the quantities α and β defined by (2.7) are very close to 0 and 1, respectively; hence, the stability region of these two algorithms coupled with the same numerical method is very close. This can be also seen in Figs. 10 and 11.

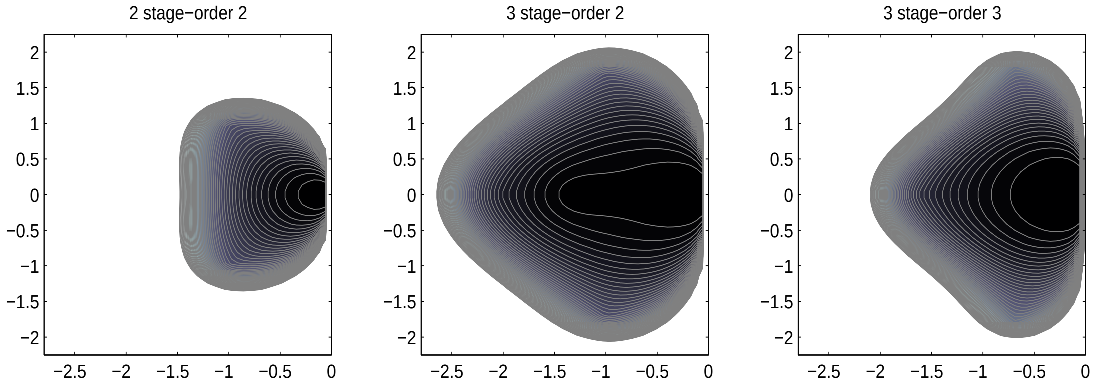

Figure 10: Stability region of the parareal algorithms coupled with the 2 stage-order 2 RK method (left), 3 stage-order 2 (middle) and 3 stage-order 3 (right).

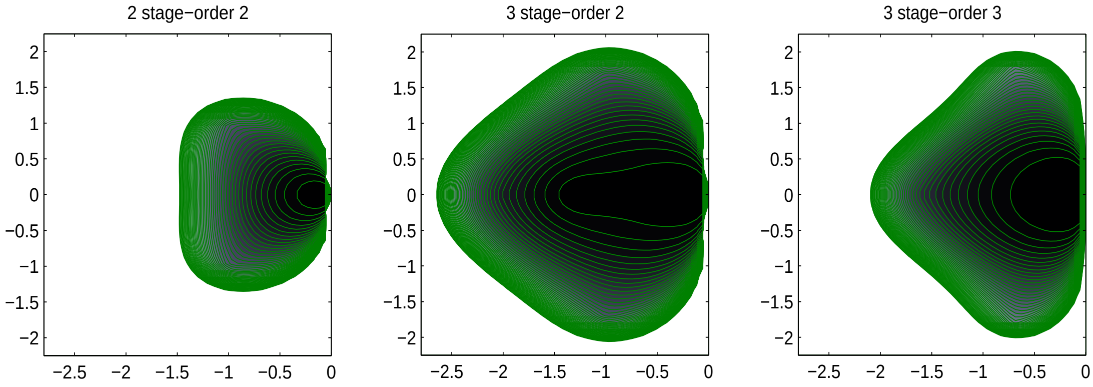

Figure 11: Stability region of the Parareal-Richardson algorithms coupled with the 2 stage-order 2 RK method (left), 3 stage-order 2 (middle) and 3 stage-order 3 (right).

4 Discussion of the parameters γ and M

In this section, we give some discussion of the effect of the parameters γ and M to the convergence speed of the Parareal-Richardson algorithm with the 3 stage-order 2 RK method (3.2) and the Lotka-Volterra system (3.8). We choose $T=20,\Delta T=0.05$ and $M=10,50$ and for each M, we perform the Parareal-Richardson algorithm with $\gamma=0.95,1,(2-\alpha)/2,1-\alpha$ and $(1-\alpha)+0.005$, where α is defined by (2.7). For each M, the relation of the line types to the parareal and Parareal-Richardson algorithms with different γ are specified in Figs. 12 and 13.

We first test $M=10$. In such case, the parareal and Parareal-Richardson algorithms with the above five parameters γ are convergent, and the discretization error DErr_n and $\widetilde{\text{DErr}}_n$ at every time point t_n is plotted in the left panel of Fig. 12. In the middle panel, we show the measured convergence speed of the parareal and Parareal-Richardson al-

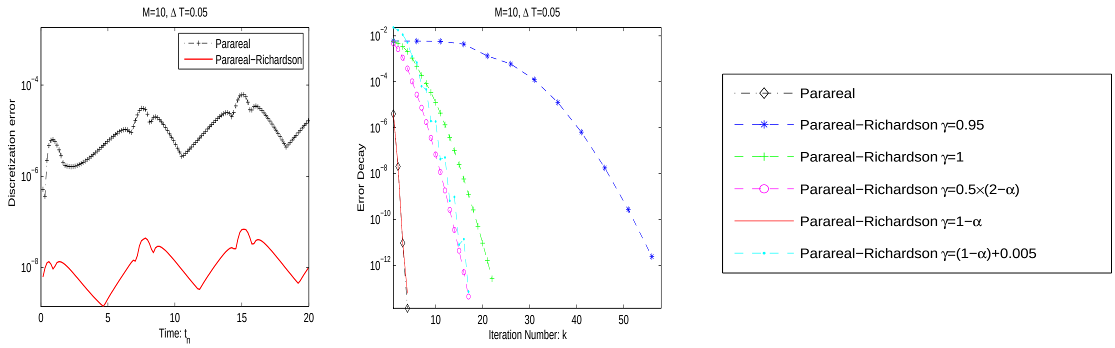

Figure 12: Computational results with $M=10$: Left: discretization errors DErr_n and $\widetilde{\text{DErr}}_n$; Middle: algebraic error decay Err_k and $\widetilde{\text{Err}}_k$ with different γ; Right: relation of the line types to the parareal and Parareal-Richardson algorithms with different γ.

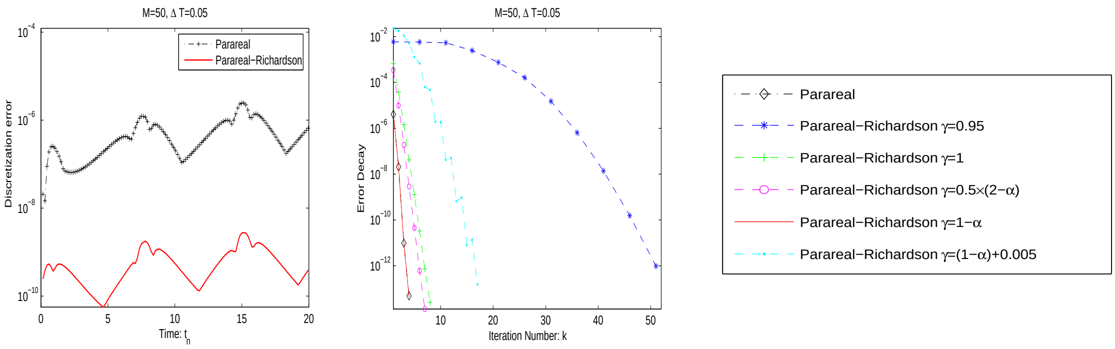

Figure 13: Same as Fig. 12, except with $M=50$.

gorithms with different γ (the line types shown in this panel are specified in the right panel).

We see from Fig. 12 that the convergence speed of the Parareal-Richardson algorithm coupled with $\gamma=1-\alpha$ outperforms the other cases, and in such case the convergence speed of the Parareal-Richardson algorithm strictly equals to the parareal algorithm.

We next test $M=50$. The discretization error DErr_n, $\widetilde{\text{DErr}}_n$ at every time point t_n and the measured convergence speed of the parareal and Parareal-Richardson algorithms with different γ are plotted in the left and middle panels of Fig. 13, respectively. In the middle panel of Fig. 13, one can see that the convergence speed of the Parareal-Richardson algorithm with $\gamma=1,(2-\alpha)/2$ and $1-\alpha$ closes to the parareal algorithm. This can be interpreted as: for $M=50$, the quantities α, β and γ approach to 0,1 and 1, respectively, and therefore from Remark 2.2 we know that the Parareal-Richardson algorithm almost equals to the parareal algorithm. However, the accuracy of the converged solution computed by the Parareal-Richardson algorithm is still significantly better than that of the parareal algorithm as one can see in the left panel of Fig. 13.

By the above experiments, one can see that the convergence speed of the Parareal-

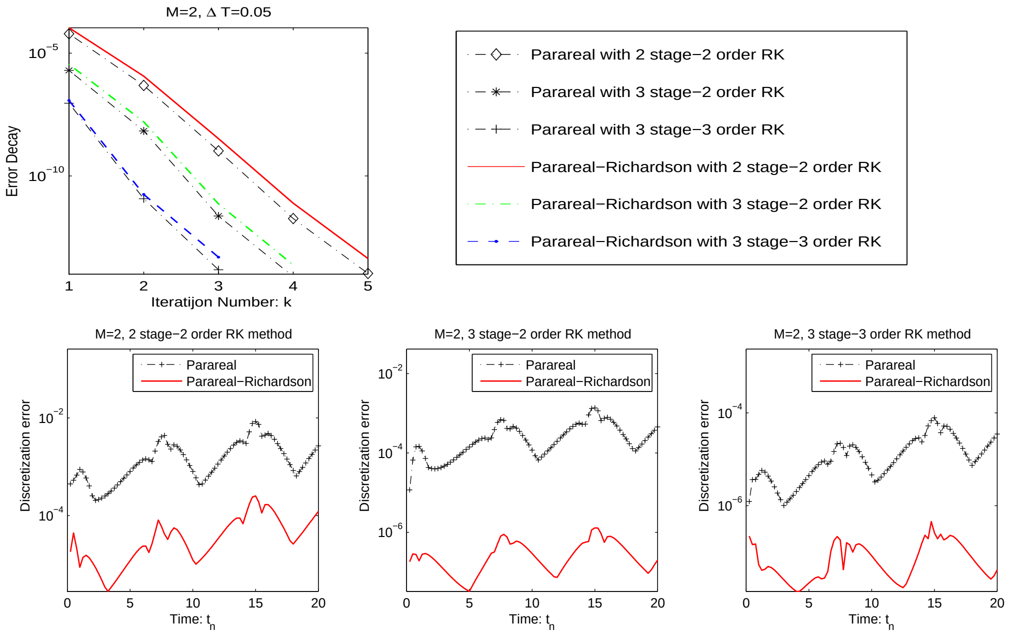

Figure 14: Computational results with $M=2,\gamma=1-\alpha$ for the Lotka-Volterra system (3.8): Top left: algebraic error decay Err_k and $\widetilde{\mathrm{Err}}_k$ with different RK methods; Top right: the relation of the line types to the parareal and Parareal-Richardson algorithms coupled with three RK methods (3.1)-(3.3); Bottom, from left to right: discretization errors DErr_n and $\widetilde{\mathrm{DErr}}_n$ with RK methods (3.1), (3.2) and (3.3).

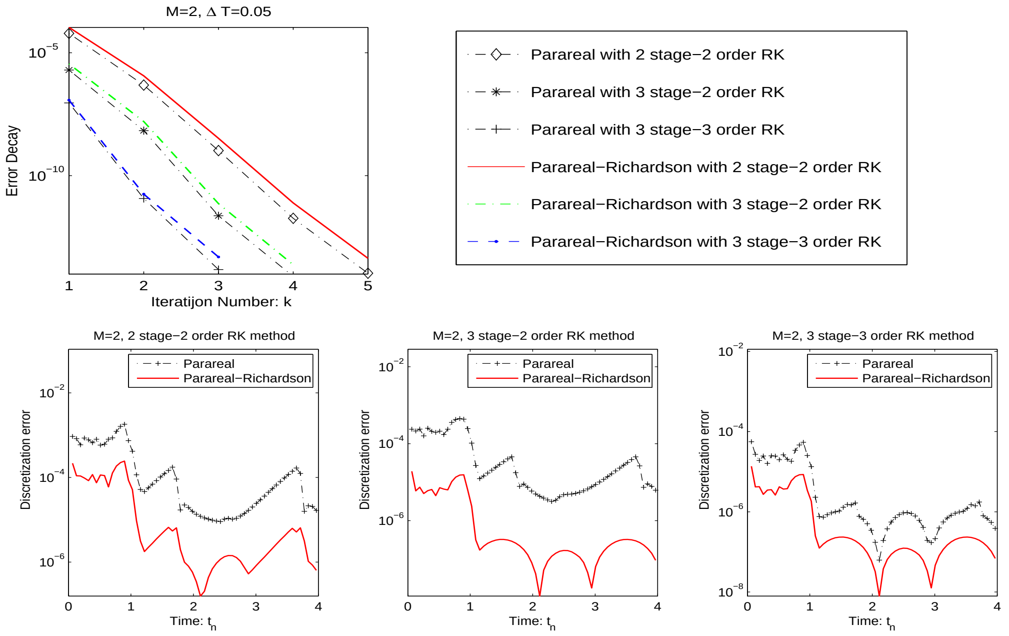

Figure 15: Same as Fig. 14, except for Burgers' equation (3.10).

Richardson algorithm is very sensitive to the parameter γ. Moreover, at the moment $\gamma = 1-\alpha$ may be the best choice for the Parareal-Richardson algorithm, by which the Parareal-Richardson algorithm converges with almost equal speed of the original parareal algorithm, even if the quantity M is small. To further validate our conjecture, we will do more experiments with much smaller M in the remainder of this section.

Obviously, the meaningful minimal M is 2, and in such case our computation results show that in most cases the Parareal-Richardson algorithm with $\gamma = 1$ converges very slow (at the moment, we just report this result but not intend to show it numerically). Fortunately, we have the choice $\gamma = 1-\alpha$! In the following of this section, for $M = 2$, we perform the parareal and Parareal-Richardson algorithms with $\gamma = 1-\alpha$ and the RK methods (3.1)-(3.3) for the Lotka-Volterra system (3.8) and the Burger's equation (3.10).

The computational results for these two nonlinear equations are plotted in Figs. 14 and 15, respectively. In the top left panel of these two figures, we show the measured convergence speed of the parareal and Parareal-Richardson algorithms, where one can see clearly that both algorithms converge with almost equal speed. On the bottom panels, one can see that the advantage of the Parareal-Richardson algorithm in the sense of accuracy is very significant. The relation of the line types to the parareal and Parareal-Richardson algorithms coupled with the three RK methods is specified in Figs. 14 and 15.

5 Conclusions and further research

We propose a new algorithm, namely *Parareal-Richardson*, to solve the time-dependent problems simultaneously in time. The new algorithm can be regarded as a weighted version of the parareal algorithm and also can be regarded as the parallel implementation of the Richardson extrapolation. By testing several classical nonlinear ODEs and PDEs, one can see that the Parareal-Richardson algorithm takes the advantage of much better accuracy of the converged numerical solution. We have shown that, for $\gamma = 1$ and larger M, the Parareal-Richardson algorithm converges with almost equal speed of the parareal algorithm, while for small M the choice $\gamma = 1$ is not advisable. We have also presented an experiential optimal choice $\gamma = 1-\alpha$, by which the Parareal-Richardson algorithm converges with almost equal speed of the original parareal algorithm in most cases, even if the quantity M is 2, the meaningful minimal value of M.

However, several basic problems are still open. One is that how the underlaying numerical method affects the convergence and the convergence speed of the Parareal-Richardson algorithm when applied to nonlinear problems. The other problem is that how the parameter γ affects the convergence speed. Even though the choice $\gamma = 1-\alpha$ can guarantee the almost equal convergence speed of the Parareal-Richardson algorithm with the original parareal algorithm as shown in Section 4, can we choose some better γ which significantly speedups the convergence of the Parareal-Richardson algorithm?

To validate that there really exists some optimal parameter γ by which the Parareal-Richardson algorithm converges faster than the parareal algorithm, we consider the fol-

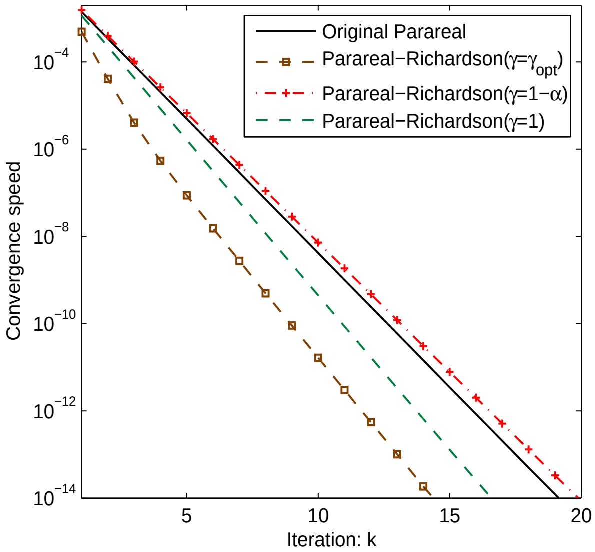

Figure 16: Convergence speed of the parareal and Parareal-Richardson algorithms with $\gamma = \gamma_{opt}, 1-\alpha$ and 1.

lowing reaction-diffusion equation

$$\begin{cases} u_t = u_{xx} + \cos(t+x) + \sin(t+x), & (t,x) \in [0,10] \times [0,1], \\ u(0,x) = \sin(x), & x \in [0,1], \\ u(t,0) = \sin(t), u(t,1) = \sin(1+t), & t \in [0,10]. \end{cases} \tag{5.1}$$

By applying the central difference formula to discretize the diffusion term u_{xx} with $\Delta x = 0.025$, we obtain

$$U'(t) = AU(t) + g(t), \quad t \in [0,10], \tag{5.2}$$

with

$$A = \frac{1}{\Delta x^2} \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix}_{39 \times 39} \tag{5.3}$$

and some function $g(t)$. We will show in our forthcoming paper that, for the linear system (5.2) (even without the special structure of the matrix A as (5.3)), there exists a special parameter γ such that the convergence speed of the Parareal-Richardson algorithm is sharper than that of the parareal algorithm. To illustrate this, we test these two algorithms coupled with the backward Euler method with $\Delta T = 0.1$ and $M = 20$. The parameter γ is chosen as $\gamma = \gamma_{opt} = 0.89347368421053$, $\gamma = 1-\alpha$ and $\gamma = 1$. The computational results are plotted in Fig. 16. In Fig. 16, it is clearly shown that the Parareal-Richardson algorithm with $\gamma=\gamma_{opt}$ converges faster than the parareal algorithm. Moreover, for these three parameters γ, the choice of $\gamma=\gamma_{opt}$ is the best one. Particularly, the iteration number for the parareal and Parareal-Richardson algorithms with $\gamma=\gamma_{opt}$, $\gamma=1-\alpha$ and $\gamma=1$ is 20, 15, 20 and 17, respectively.

Acknowledgments

This work was supported by NSF of China (Nos. 10671078, 60773195) and by Program for NCET, the State Education Ministry of China.

The authors are grateful to the anonymous referees for the careful reading of a preliminary version of the manuscript and their valuable suggestions and comments, which really improve the quality of this paper.

References

[1] 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, Lect. Notes Comput. Sci. Eng. 40, R. Kornhuber, R. H. W. Hoppe, J. Péeriaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., Springer, Berlin, 2003, 426–432.

[2] 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 Comput. Sci. Eng. 23, Springer, Berlin, 2002, 189–202.

[3] L. Baffico, S. Bernard, Y. Maday, G. Turinici and G. Zérah. Parallel in time molecular–dynamics simulations. Physical Review. E, 66 (2002), pp. 057706–1–4.

[4] A. Bellen and M. Zennaro. Parallel algorithms for initial value problems for difference and differential equations. J. Comput. Appl. Math., 25 (1989), 341–350.

[5] P. Chartier and B. Philippe. A parallel shooting technique for solving dissipative ODEs. Computing, 51 (1993), 209–236.

[6] C. Farhat and M. Chandesris. Time–decomposed parallel time–integrators: Theory and feasibility studies for fluid, structure, and fluid–structure applications. Internat. J. Numer. Methods Engrg., 58 (2003), 1397–1434.

[7] 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, Lect. Notes Comput. Sci. Eng. 40, R. Kornhuber, R. H. W. Hoppe, J. Péeriaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., Springer, Berlin, 2003, 433–440.

[8] P. F. Fischer. Investigation of the parareal algorithm for semi–implicit incompressible navier-stokes simulations. Talk at 15th domain decomposition conferance in Berlin, 2003.

[9] E. Schäfer. A new approach to explain the 'high irraiance responses' of photomorphogenesis on the basis of phytochrome. J. Math. Biology, 2 (1975), 41–56.

[10] I. Garrido, B. Lee, G. E. Fladmark and M. E. Espedal. Convergent iterative schemes for time parallelization. Math. Comp., 26 (2006), 1403–1428.

[11] I. Garrido, M. S. Espedal and G. E. Fladmark. A convergence algorithm for time parallelization applied to reservoir simulation. In Proceedings of the 15th International Domain Decomposition Conference, Lect. Notes Comput. Sci. Eng. 40, R. Kornhuber, R. H. W. Hoppe, J. Péeriaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., Springer, Berlin, 2003, 469–476.

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

[13] M. J. Gander and E. Hairer. Nonlinear convergence analysis for the parareal algorithm. Proceedings of the 17th International Conference on Domain Decomposition Methods, 2007.

[14] D. Guibert and D. Tromeur–Dervout. Parallel adaptive time domain decomposition for stiff systems of ODEs/DAEs. Computers and Structures, 85 (2007), 553–562.

[15] E. Hairer and G. Wanner. Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. Springer–Verlag Berlin Heidelberg, 1996.

[16] W. Hackbusch. Parabolic multigrid methods. Computing Methods in Applied Sciences and Engineering, VI, R. Glowinski and J.-L.Lions, eds., North–Holland, Amsterdam 1984, 189–197.

[17] E. Hairer, S. P. Nørsett and G. Wanner. Solving Ordinary Dierential Equations I: Nonstiff Problems. Springer Series in Computational Mathematics, Springer, Berlin, 1991, 216–217.

[18] G. Horton. The time–parallel multigrid method. Comm. Appl. Numer. Methods, 8 (1992), 585–595.

[19] G. Horton and S. Vandewalle. A space–time multigrid method for parabolic partial differential equations. SIAM J. Sci. Comput., 16 (1995), 848–864.

[20] H. B. Keller. Numerical Methods for Two–Point Boundary Value Problems. Blaisdell, Waltham, MA, 1968.

[21] R. Kornhuber, R. H. W. Hoppe, J. Péeriaux, O. Pironneau, O. B. Widlund and J. Xu, eds. In Proceedings of the 15th International Domain Decomposition Conference. Lect. Notes Comput. Sci. Eng. 40, Springer, Berlin, 2003.

[22] M. Kiehl. Parallel multiple shooting for the solution of initial value problems. Parallel Comput., 20 (1994), 275–295.

[23] R. J. LeVeque. Finite Volume Methods for Hyperbolic Problems. Combridge University Press, 2002

[24] 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), 661–668.

[25] C. Lubich and A. Ostermann. Multi–grid dynamic iteration for parabolic equations. BIT, 27 (1987), 216–234.

[26] W. L. Miranker and W. Liniger. Parallel methods for the numerical integrating of ordinary differential equations. Math. Comp., 91 (1967), 303–320.

[27] Y. Maday and G. Turinici. A parareal in time procedure for the control of partial differential equations. C. R. Acad Sci. Paris Sér. I Math, 335 (2002), 387–392.

[28] Y. Maday and G. Turinici. A parallel in time approach for quantum control: the parareal algorithm. In Proceedings of the 41th IEEE Conference on Decision and Control, Las Vegas, Nevada, USA, 1 (2002), 62–66.

[29] J. Nievergelt. Parallel methods for integrating ordinary differential equations. Comm. ACM, 7 (1964), 731–733.

[30] G. A. Staff and E. M. Rønquist. Stability of the parareal algorithm. In Proceedings of the 15th International Domain Decomposition Conference, Lect. Notes Comput. Sci. Eng. 40, R. Kornhuber, R. H. W. Hoppe, J. Péeriaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., Springer, Berlin, 2003, 449–56.

[31] S. Vandewalle and R. Piessens. Efficient parallel algorithms for solving initial–boundary value and time–periodic parabolic partial differential equations. SIAM J. Sci. Statist. Comput., 13 (1992), 1330–1346.

[32] S. Vandewalle. Parallel Multigrid Waveform Relaxation for Parabolic Problems. B. G. Teub-

ner, Stuttgart, 1993.

[33] S. Vandewalle and E. V. de Velde. Space–time concurrent multigrid waveform relaxation. Ann. Numer. Math., 1 (1994), 347–360.

[34] S. L. Wu, C. M. Huang and B. C. Shi. Convergence analysis of the Parareal–Richardson algorithm. Manuscript.