An Adjoint Approach for Stabilizing the Parareal Method

Feng Chen[a], Jan S Hesthaven[b], Yvon Maday[c], Allan Nielsen[b]

[a]*Department of Mathematics, Baruch College, New York, NY 10010, USA*

[b]*Mathematics Institute of Computational Science and Engineering, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland*

[c]*Sorbonne Universités, UPMC Univ Paris 06, CNRS, UMR 7598, Laboratoire Jacques-Louis Lions, 4, place Jussieu 75005, Paris, France, Institut Universitaire de France and D.A.M., Brown University, Providence (RI) USA*

Received *****; accepted after revision +++++

Presented by ?????

Abstract

The parareal algorithm seeks to extract parallelism in the time-integration direction of time-dependent differential equations. While it has been applied with success to a wide range of problems, it suffers from some stability issues when applied to non-dissipative problems. We express the method through an iteration matrix and show that the problematic behavior is related to the non-normal structure of the iteration matrix. To enforce monotone convergence we propose an adjoint parareal algorithm, accelerated by the Conjugate Gradient Method. Numerical experiments confirm the stability and suggest directions for further improving the performance. *To cite this article: F. Chen, J.S. Hesthaven, Y. Maday, A. Nielsen, C. R. Acad. Sci. Paris, Ser. I ??? (2015).*

Résumé

L'algorithme parareel en temps vise à proposer la parallélisation dans la direction temporelle de l'intégration approchée déquations différentielles instationnaires. Cet algorithme a été appliqué avec succès sur une large gamme de problèmes mais souffre néanmoins de certains problèmes de stabilité lorsqu'il est appliqué à des problèmes non dissipatifs. Nous exprimons l'algorithme comme une méthode d'itération matricielle et nous montrons que le comportement problématique est liée à la structure non-normale de la matrice d'itération. Pour retrouver une convergence monotone nous proposons une version adjointe de l' algorithme pararéel, accélérée par la méthode du gradient conjugué. Des expériences numériques confirment la stabilité et proposent des pistes pour améliorer les performances. **Méthode adjointe pour la stabilisation de l'algorithme pararéel.** *Pour citer cet article : F. Chen, J.S. Hesthaven, Y. Maday, A. Nielsen, C. R. Acad. Sci. Paris, Ser. I ? ? ? (2015).*

Email addresses: feng.chen@baruch.cuny.edu (Feng Chen), jan.hesthaven@epfl.ch (Jan S Hesthaven), maday@ann.jussieu.fr (Yvon Maday), allan.nielsen@epfl.ch (Allan Nielsen).

1. Parareal in time algorithm

The increasing number of processors available at large computers presents a substantial challenge for the development of applications that scale well. One potential path to increased parallelism is the development of parallel time-integration methods. Once some standard methods-of-lines approach have been applied, the problem is traditionally viewed as a strongly sequential process. Attempts to extract parallelism have nevertheless been explored : a simple example is parallel Runge-Kutta methods where independent stages allow for the introduction of parallelism [8]. The parareal method, first proposed in [4], is another and more elaborate approach to achieve a similar goal. This algorithm borrows ideas from spatial domain decomposition to construct an iterative approach for solving the temporal problem in a parallel global-in-time approach. To present the method, consider the problem

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + \mathcal{A}\left(t, \mathbf{u}\right) = 0 \\ \mathbf{u}\left(T_0\right) = \mathbf{u}_0 \quad t \in [T_0, T] \end{cases} \tag{1}$$

where $\mathcal{A} : \mathbb{R} \times V \to V'$ is a general operator depending on $\mathbf{u} : \Omega \times \mathbb{R}^+ \to V$ with V being a Hilbert space and V' its dual. Assume the existence of a unique solution $\mathbf{u}\left(t\right)$ to (1) and decompose the time domain of interest into N individual time slices $T_0 < T_1 < \cdots < T_{N-1} < T_N = T$, where $T_n = n\Delta T$. Now, for any $n \in \mathbb{N}$, we define the numerical solution operator $\mathcal{F}_{\Delta T}^{T_n}$ that advances the solution from T_n to T_{n+1} as

$$\mathcal{F}_{\Delta T}^{T_n}\left(\mathbf{u}\left(T_n\right)\right) \approx \mathbf{u}\left(T_{n+1}\right) \tag{2}$$

This allows, starting from $\mathbf{U}_0 = \mathbf{u}_0$ to define recursively approximations $\mathbf{U}_1, \cdots, \mathbf{U}_N$ to $\mathbf{u}(T_1), \cdots, \mathbf{u}(T_N)$ by setting, for $n = 0, \ldots, N-1$: $\mathbf{U}_{n+1} = \mathcal{F}_{\Delta T}^{T_n}\left(\mathbf{U}_n\right)$. It is pertinent, for what follows, to realize that this numerical solution corresponds to the forward substitution solution method applied to the matricial system $M_{\mathcal{F}}\bar{\mathbf{U}} = \bar{\mathbf{U}}_0$ where $M_{\mathcal{F}}$, $\bar{\mathbf{U}}$ and $\bar{\mathbf{U}}_0$ are defined as follows

$$M_{\mathcal{F}} = \begin{bmatrix} 1 & & & \\ -\mathcal{F}_{\Delta T}^{T_0} & \ddots & & \\ & \ddots & \ddots & \\ & & -\mathcal{F}_{\Delta T}^{T_{N-1}} & 1 \end{bmatrix}, \quad \bar{\mathbf{U}} = \begin{bmatrix} \mathbf{U}_0 \\ \vdots \\ \vdots \\ \mathbf{U}_N \end{bmatrix}, \quad \bar{\mathbf{U}}_0 = \begin{bmatrix} \mathbf{u}_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{3}$$

If we instead solve the system using a point-iterative approach, i.e., seeking the solution on form $\bar{\mathbf{U}}^{k+1} = \bar{\mathbf{U}}^k + \left(\bar{\mathbf{U}}_0 - M_{\mathcal{F}}\bar{\mathbf{U}}^k\right)$, we may observe that $\bar{\mathbf{U}}^k$ is known at each iteration, allowing a complete decoupled computation of $\mathcal{F}_{\Delta T}^{T_1}, \cdots, \mathcal{F}_{\Delta T}^{T_N}$ on all intervals. However, for any k, we clearly have to wait k iterations in order to get $\mathbf{U}_k^k = \mathbf{U}_k$ hence we have to wait N iterations of the above iterative approach to solve the problem. Consequently, the above approach does not provide any reduction in the time solution. In order to reduce the number of iterations K needed to reach convergence, we need to find an appropriate preconditioner to accelerate the iteration, a typical approach is to utilize an approximation $M_{\mathcal{G}} \approx M_{\mathcal{F}}$, where $M_{\mathcal{G}}$ is cheap to apply. We can readily create such an $M_{\mathcal{G}}$ by defining another operator $\mathcal{G}_{\Delta T}$ that proposes a coarser approximation

$$\mathcal{G}_{\Delta T}^{T_n}\left(\mathbf{u}\left(T_n\right)\right) \approx \mathbf{u}\left(T_{n+1}\right) \tag{4}$$

by reducing accuracy and choosing a coarse grained model or an entirely different numerical model. Solving the system using standard preconditioned Richardson iterations we may write

$$\bar{\mathbf{U}}^{k+1} = \bar{\mathbf{U}}^k + \left(M_{\mathcal{G}}\right)^{-1}\left(\bar{\mathbf{U}}_0 - M_{\mathcal{F}}\bar{\mathbf{U}}^k\right). \tag{5}$$

This recovers the parareal algorithm on its standard form

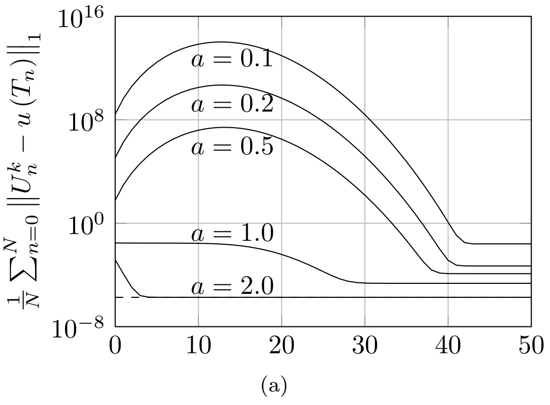
$$\mathbf{U}_{n+1}^{k+1} = \mathcal{G}_{\Delta T}^{T_n}\mathbf{U}_n^{k+1} + \mathcal{F}_{\Delta T}^{T_n}\mathbf{U}_n^k - \mathcal{G}_{\Delta T}^{T_n}\mathbf{U}_n^k \quad \text{with} \quad \mathbf{U}_{n+1}^0 = \mathcal{G}_{\Delta T}^{T_n}\mathbf{U}_n^0 \quad \text{and} \quad \mathbf{U}_0^k = \mathbf{u}(T_0). \tag{6}$$

This approach has no inherent limits to the amount of parallelism that can be extracted. Important contributions to the analysis of the method can be found in [6,1,3].

2. Instabilities

As the parareal algorithm is essentially a preconditioned point iterative method, (5) can be written as $\bar{\mathbf{U}}^{k+1} = \left[\left(I - (M_{\mathcal{G}})^{-1} M_{\mathcal{F}}\right)\right] \bar{\mathbf{U}}^k + M_{\mathcal{G}}^{-1}\bar{\mathbf{U}}_0$. It is clear that if $\mathcal{G}_{\Delta T}^{T_0}$ is sufficiently accurate, then one essentially recovers the serial solution process as expected. However, of $\mathcal{G}_{\Delta T}^{T_n}$ is far from accurate, the iteration matrice, $(M_{\mathcal{G}})^{-1} M_{\mathcal{F}}$ develops a degree of non-normality that increases as the quality of $\mathcal{G}_{\Delta T}^{T_n}$ decreases. When solving problems that have a natural dissipation, e.g., a parabolic problem, this does not pose a problem. However, when considering problems without dissipation, e.g., a wave dominated problem, the impact of this non-normality becomes very apparent.

To demonstrate this problematic behavior, Figs. 1a and 1b illustrate convergence when the parareal algorithm is used to solve the test equation $\frac{d}{dt}u(t) = \lambda u(t)$ over $[0, 100]$, using a forward Euler method, with a small time step $dt = 10^{-3}$, for the fine solver $\mathcal{F}$ and a varying time step $dT = R10^{-3}$ for the coarse solver $\mathcal{G}$ (hence solved R-time faster than $\mathcal{F}$). One clearly observes the problematic transient behavior for problems with small dissipation and recognize the intermittent behavior as characteristic of iteration with non-normal matrices [7]. It is also clear that strong intermittent behavior can be controlled by improving the accuracy of $\mathcal{G}_{\Delta T}^{T_n}$ as one would expect. We refer to [2] for some ways to cure these problems based on some conservation properties for hyperbolic equations.



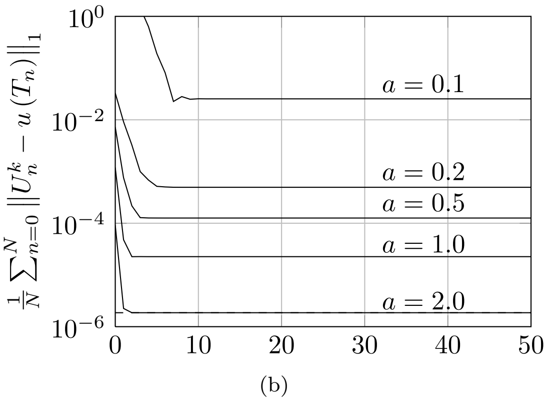


Figure 1. Error as a function of iterations when solving $\frac{d}{dt}u(t) = \lambda u(t)$ with lambda $\lambda = -a - i$ and $dt = 0.001$, $\Delta T = 1$. Ratio between timestep dT of $\mathcal{G}_{\Delta T}$ and dt of $\mathcal{F}_{\Delta T}$ (a) $R = 500$ (b) $R = 50$

3. An Adjoint Approach

As observed above, the parareal method exhibits problematic behavior for non-dissipative problems due to the inherent non-normal nature of the iteration matrices, leading to increasing number interactions k for the algorithm to converge. This adversely impacts the parallel efficiency which scales as $1/k$. To overcome this, we consider a symmetrized version, as was done in [5] for the application to control problem, by constructing the adjoint operators $\mathcal{F}_{\Delta T}$ and $\mathcal{G}_{\Delta T}$ to (1). Consider the adjoint operator $\mathcal{F}_{\Delta T}^{*}$ and

$$M_{\mathcal{F}}^{*} M_{\mathcal{F}} \bar{\mathbf{U}} = M_{\mathcal{F}}^{*} \bar{\mathbf{U}}_0 = \bar{\mathbf{U}}_0. \tag{7}$$

For simplicity, we assume that the adjoint operator is the conjugate transposed of $M_{\mathcal{F}}$, but this not an essential assumption. Clearly, $M_{\mathcal{F}}^{*} M_{\mathcal{F}}$ is symmetric positive definite, and we can use the preconditioned conjugate gradient (PCG) method to solve this problem. Inspired by the parareal method, it is natural to consider $\left(M_{\mathcal{G}}^{*} M_{\mathcal{G}}\right)^{-1}$ as a preconditioner. It is noteworthy that $M_{\mathcal{G}}^{*} M_{\mathcal{G}}$ is an approximation to the Cholesky factorization of $M_{\mathcal{F}}^{*} M_{\mathcal{F}}$ and it can be applied directly in the PCG method. The overall algorithm is outlined in algorithm 1 below. We present a method developed for linear systems, but the algorithm itself can be applied to nonlinear systems without modifications.

Algorithm 1 Pseudo code for PCG based Adjoint Parareal

```
$\bar{U}^0 = (M_{\mathcal{G}})^{-1} \bar{U}_0$ \\Creation of initial iterate
$r_{(0)} = \bar{U}_0 - M_{\mathcal{F}}^{*} M_{\mathcal{F}} \bar{U}^0$ \\Parallel application of $\mathcal{F}_{\Delta T}^{T_n}$ and $\left(\mathcal{F}_{\Delta T}^{T_n}\right)^*$ to $U_n^0$
$d_{(0)} = \left(M_{\mathcal{G}}^{*} M_{\mathcal{G}}\right)^{-1} r_{(0)}$ \\Sequential application of $\mathcal{G}_{\Delta T}^{T_n}$ and $\left(\mathcal{G}_{\Delta T}^{T_n}\right)^*$ on residual
$G_{(0)} = d_{(0)}$
\\Iterate
for $k = 0$ to $K_{max} \leq N - 1$ do
  $F_{(k)} = M_{\mathcal{F}}^{*} M_{\mathcal{F}} d_{(k)}$ \\Parallel application of $\mathcal{F}_{\Delta T}^{T_n}$ and $\left(\mathcal{F}_{\Delta T}^{T_n}\right)^*$ to $d_{(k)}$
  $\alpha_{(k)} = \frac{r_{(k)}^T G_{(k)}}{d_{(k)}^T F_{(k)}}$
  $\bar{U}^{k+1} = \bar{U}^k + \alpha_{(k)} d_{(k)}$
  $r_{(k+1)} = r_{(k)} - \alpha_{(k)} F_{(k)}$
  if $|U_n^{k+1} - U_n^k| < \epsilon \, \forall \, n$ then
    BREAK \\Terminate loop if converged
  end if
  $G_{(k+1)} = \left(M_{\mathcal{G}}^{*} M_{\mathcal{G}}\right)^{-1} r_{(k+1)}$ \\Sequential application of $\mathcal{G}_{\Delta T}^{T_n}$ and $\left(\mathcal{G}_{\Delta T}^{T_n}\right)^*$ on residual
  $\beta_{(k+1)} = \frac{r_{(k+1)}^T G_{(k+1)}}{r_{(k)}^T G_{(k)}}$
  $d_{(k+1)} = G_{(k+1)} + \beta_{(k+1)} d_{(k)}$
end for
```

4. Numerical Experiments

We again consider the test equation

$$\frac{d}{dt} u(t) = \lambda u(t), \quad u(0) = 1, \quad t \in [0, 100] \tag{8}$$

with a forward Euler discretization and $\lambda = i$, i.e., it is strictly non-dissipative. In Fig. 2a we show the error between the exact solution $u(t) = \exp(\lambda t)$ and the parareal iterative solution U_n^k as a function of the iteration count for different ratios between the coarse and the fine time-step. As the ratio R between timestep dT in $\mathcal{G}_{\Delta T}$ and dt in $\mathcal{F}_{\Delta T}$ decreases, the coarse solver improves in accuracy and the level of non-normality decreases.

In Fig. 2b we illustrate convergence of the PCG accelerated adjoint parareal algorithm for the same test case and the expected rapid convergence is clear. The convergence rate of the PCG algorithm depends on the condition number and the clustering of eigenvalues of the preconditioned system. In Fig. 4a the eigenvalues of $\left(M_{\mathcal{G}}^{*}M_{\mathcal{G}}\right)^{-1}M_{\mathcal{F}}^{*}M_{\mathcal{F}}$ for (8) is given for different values of R. It is clear that as R increases, small eigenvalues are introduced, contributing to the reduced convergence rate.

To improve the convergence rate, we define the projection $P=\left(I+\varepsilon V_1 V_1^T\right)$ where V_1 is the eigenvector of $\left(M_{\mathcal{G}}^{*}M_{\mathcal{G}}\right)^{-1}M_{\mathcal{F}}^{*}M_{\mathcal{F}}$ associated with the smallest eigenvalue μ_1 that we attempt to shift. V_1 is an eigenvector of $P\left(M_{\mathcal{G}}^{*}M_{\mathcal{G}}\right)^{-1}M_{\mathcal{F}}^{*}M_{\mathcal{F}}$, but now associated a new eigenvalue $\lambda_1'=(1+\varepsilon)\,\mu_1$. In Fig. 4a the modified preconditioned system is solved with $P=\left(I+\frac{1-\mu_1}{\mu_1}V_1V_1^T\right)$ to shift the smallest eigenvalue of the preconditioned system to 1, illustrating the potential for a substantially improved convergence rate.

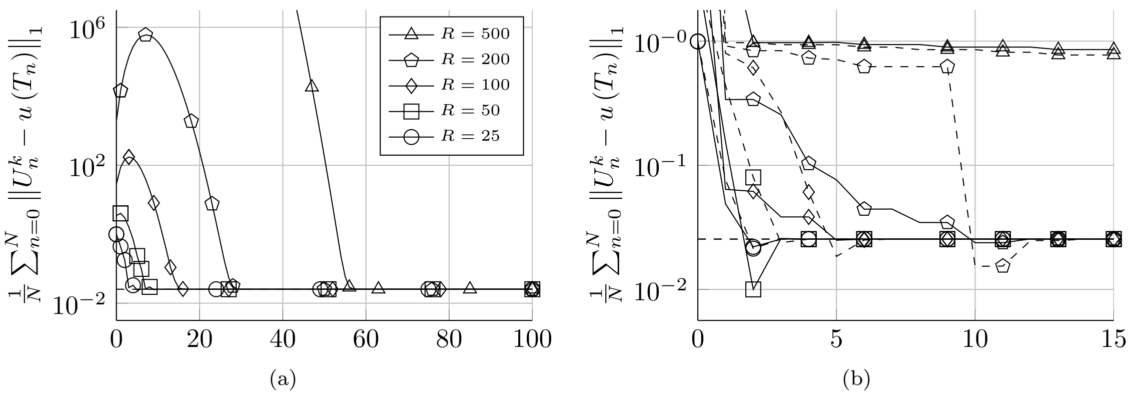


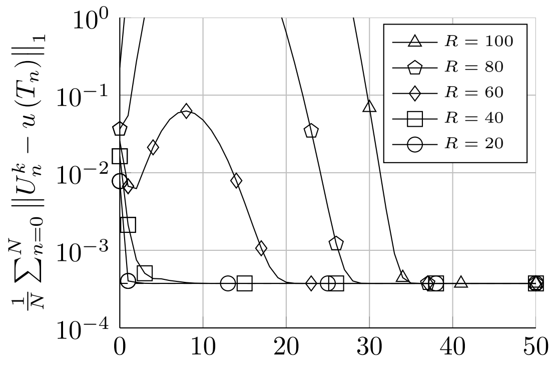
Figure 2. Error as a function of parareal iterations k for solving (8) with $dt=10^{-3}$. R is the ratio between timestep dT in $\mathcal{G}_{\Delta T}$ and dt in $\mathcal{F}_{\Delta T}$. (a) Original parareal. (b) Adjoint parareal (dashed), modified adjoint, (solid).

As a second test we consider

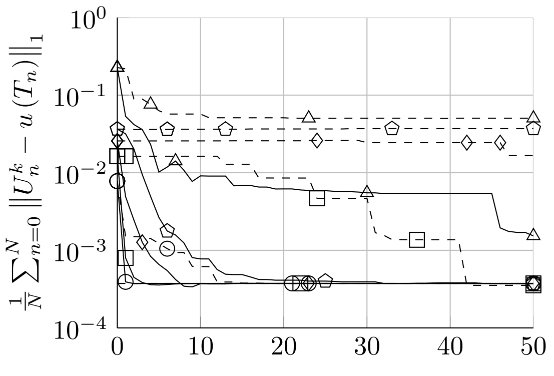
$$\begin{cases}\dfrac{\partial u\left(x,t\right)}{\partial t}+\dfrac{\partial u\left(x,t\right)}{\partial x}=0 & ,\ (t,x)\in[0,100]\times[0,2]\\ u\left(x,0\right)=\exp\left(\sin\left(2\pi x\right)\right) & ,\ u\left(0,t\right)=u\left(2,t\right)\forall t\in[0,100]\end{cases}\tag{9}$$

Using a spectral discretization in space and a forward Euler in time, we show in Fig. 3a the convergence rate of the original parareal algorithm as a function of the ratio between the coarse and the fine time-step, clearly exposing the same identified transient effect of the non-normal iteration matrix. In Fig. 3b the convergence of the PCG based adjoint parareal algorithm is presented. While the convergence is monotone, the rate of convergence is prohibitively slow. Figure 4b shows the eigenvalues of the preconditioned system for (9), spanning from 10^{-4} to 10^{8} in the case of the coarse discretization. While convergence can be improved by shifting the small eigenvalues as discussed above, this quickly becomes prohibitive due to the large number that needs to be shifted.

During the numerical tests we observed that by constructing a new preconditioner LL^* as a modified incomplete Cholesky factorization of $M_{\mathcal{G}}^{*}M_{\mathcal{G}}$, this improves the convergence rate dramatically. This is illustrated in Fig. 3b. The eigenvalues of the preconditioned system $\left(LL^*\right)^{-1}M_{\mathcal{F}}^{*}M_{\mathcal{F}}$ are also better clustered, Fig. 4b. It is surpricing that using a less accurate approximation of $M_{\mathcal{G}}^{*}M_{\mathcal{G}}$ leads to a substantially better preconditioner, we do not have a complete understanding of this observation.

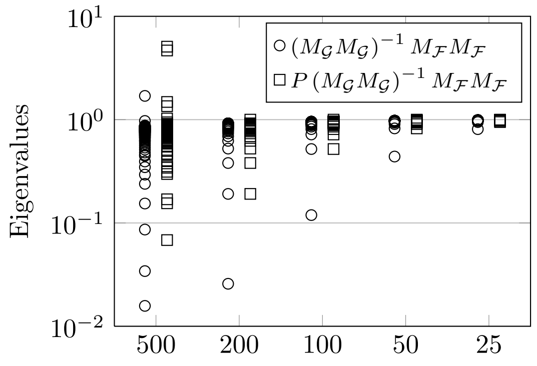


(a)

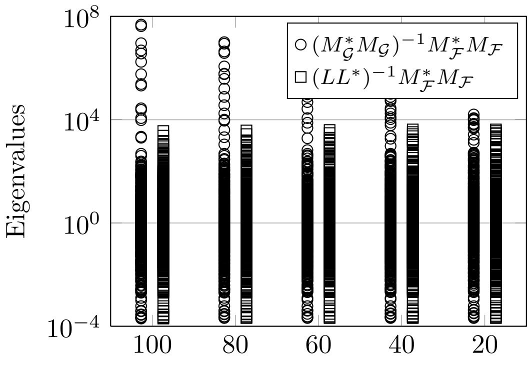


(b)

Figure 3. Error as a function of parareal iterations k solving (9) with $dx = \frac{2}{49}$ and $dt = 10^{-6}$. R is the ratio between timestep dT in $\mathcal{G}_{\Delta T}$ and dt in $\mathcal{F}_{\Delta T}$. (a) Original parareal. (b) Adjoint parareal (dashed), LL^* precondtioner (solid).



(a)



(b)

Figure 4. Eigenvalues of the preconditioned systems as a function of R. (a) Equation (8) and (b) Equation (9).

References

[1] G. Bal, 2005, *On the convergence and the stability of the parareal algorithm to solve partial differential equations*, Domain decomposition methods in science and engineering, Springer Berlin Heidelberg, 425-432.

[2] X. Dai, and Y. Maday, 2013, *Stable parareal in time method for first-and second-order hyperbolic systems.* SIAM Journal on Scientific Computing, **35**(1), A52-A78.

[3] M.J. Gander and S. Vandewalle, 2007, *Analysis of the parareal time-parallel time-integration method*, SIAM Journal on Scientific Computing **29**(2), 556-578.

[4] J.L. Lions, Y. Maday and G. Turinici. *Rśolution d'EDP par un schéma en temps pararéel*, 2001, Comptes Rendus de l'Académie des Sciences - Mathematics, **332**(7), 661-668.

[5] Y. Maday and G. Turinici, 2002, *A parareal in time procedure for the control of partial differential equations.* Comptes Rendus Mathematique, **335**(4), 387-392.

[6] G.A. Staff and E.M. Rønquist, 2005, *Stability of the parareal algorithm*, Domain decomposition methods in science and engineering. Springer Berlin Heidelberg, 449-456.

[7] L.N. Trefethen., M. Embree, 2005, *Spectra and pseudospectra: the behavior of nonnormal matrices and operators*, Princeton University Press.

[8] P. J. Van Der Houwen, B. P. Sommeijer and P. A. Van Mourik, 1989, *Note on explicit parallel multistep RungeKutta methods*, Journal of computational and applied mathematics **27**(3), 411-420.