ANALYSIS OF A KRYLOV SUBSPACE ENHANCED PARAREAL ALGORITHM FOR LINEAR PROBLEMS

M. GANDER* AND M. PETCU[‡][♭]

*Section de Mathématiques, University of Geneva, Switzerland
‡ Laboratoire de Mathématiques, Université de Poitiers, France
♭ The Institute of Mathematics of the Romanian Academy, Bucharest, Romania

ABSTRACT. The parareal algorithm is a numerical method to integrate evolution problems on parallel computers. The performance of the algorithm is well understood for diffusive problems, and it can have spectacular performance when applied to certain non-linear problems. Its convergence properties are however less favorable for hyperbolic problems. We present and analyze in this paper a variant of the parareal algorithm, recently proposed in the PITA framework for systems of second order ordinary differential equations.

1. INTRODUCTION

The parareal algorithm is a time parallel time integration method for evolution problems introduced by Lions, Maday and Turinici in [6]. It can be used to compute in parallel an approximate solution of systems of ordinary differential equations (ODEs). The method was introduced in order to obtain a speed-up on parallel computers when solving evolution problems, which need to be solved in real time, which explains its name. The advantage of this method, as shown in [6], is that it approximates the solution later in time, before having fully accurate approximations at earlier times, while the global accuracy of the iterative process after few iterations is comparable to that given by a sequential method used on a fine discretization in time.

The method has received great attention over the last years, and it was used for different applications in financial mathematics (see e.g. [1]) in robotics, biology and engineering (see [2]), as well as in quantum chemistry (see [7]). The parareal algorithm was also written in different forms: by Farhat and Chandesris (see [2]) as an algorithm called PITA (Parallel Implicit Time Integrator), and by Gander and Vanderwalle (see [5]) as a multiple shooting method. Gander and Vandewalle showed in [5] that the parareal algorithm is a multiple shooting method for initial value problems with a coarse grid approximation for the Jacobian in Newtons method.

1

They also proved that when the method is applied to linear, diffusive problems, it converges linearly on long time intervals and super-linearly on short time intervals. In [4], [1], [2], the authors showed that the parareal algorithm produces a speed-up for first order ODEs, but the method does not have the same potential for second order ODEs, see also [5]. In [3], the authors proposed a method in order to adapt the existing PITA algorithm to second order differential equations.

This paper is structured in two parts: in the first part we introduce the parareal algorithm as a multiple shooting method and recall the existing convergence analysis for this method. In the second part, we introduce the modified algorithm, explain the reason for which this method gives better results for systems of linear ODEs, and prove convergence results similar to the ones for the original method.

2. Parareal algorithm as a multiple shooting method

The parareal algorithm is a time parallel method to compute in parallel the solution of the system of ODEs

$$(2.1) \qquad u'(t) = f(u(t)), \ t \in (0, T), \ u(0) = u_0,$$

where $f : \mathbb{R}^d \to \mathbb{R}^d$ and $u : \mathbb{R} \to \mathbb{R}^d$.

The parareal algorithm is described by Gander and Vandewalle [5] as a multiple shooting method with the use of two propagation operators. The inexpensive coarse propagator $G(T_n, T_{n-1}, x)$ gives a rough approximation to $u(T_n)$, where u is the solution of equation (2.1) having $u(T_{n-1}) = x$ as initial condition, while the expensive fine propagator $F(T_n, T_{n-1}, x)$ gives a more accurate approximation to the same $u(T_n)$. Partitioning the time domain $(0, T)$ into N-time subdomains $\Omega_n = (T_n, T_{n+1})$, the algorithm works as follows:

- Step 0: The algorithm starts with an initial approximation U_n^0, $n = 0, \ldots, N$, which can be found for example using the coarse propagator sequentially,

$$(2.2) \qquad U_{n+1}^0 = G(T_{n+1}, T_n, U_n^0), \qquad U_0^0 = u_0.$$

- Step $k + 1$: We perform a correction step, using both the coarse and fine propagator,

$$(2.3) \qquad U_{n+1}^{k+1} = G(T_{n+1}, T_n, U_n^{k+1}) + F(T_{n+1}, T_n, U_n^k) - G(T_{n+1}, T_n, U_n^k).$$

We note that the initial step (2.2) of the algorithm is sequential, but not expensive, since we use the coarse and cheap G-propagator. In the iteration (2.3), the approximations can potentially become better since we improve the accuracy with the aid of the F-propagator. A significant advantage of the method is that since we already know the values U_n^k, for all n, the expensive computations for $F(T_{n+1}, T_n, U_n^k)$ can be performed in parallel and (2.3) is reduced to a sequential computation with cost comparable to $G(T_{n+1}, T_n, U_n^{k+1})$.

We now recall a convergence result proved in [4], where it was assumed for simplicity that the fine propagator F is the exact solution of the problem, and equal time subdomains are used, $T_{n+1} - T_n = \triangle T$. Furthermore, it was assumed that the difference between the approximate solution given by the coarse propagator G and the exact solution satisfies the relation

$$(2.4) \qquad F(T_n, T_{n-1}, x) - G(T_n, T_{n-1}, x) = c_{p+1}(x)\Delta T^{p+1} + c_{p+2}(x)\Delta T^{p+2} + \dots,$$

where the coefficients c_j with $j = p+1, p+2, \dots$ are continuously differentiable, and that G satisfies the Lipschitz condition

$$(2.5) \qquad \|G(t + \Delta T, t, x) - G(t + \Delta T, t, y)\| \leq (1 + C_2\Delta T)\|x - y\|.$$

Theorem 2.1. *Under the conditions above, the error* $u(T_n) - U_n^k$ *at step* k *of the parareal algorithm satisfies the convergence estimate*

$$(2.6) \quad \begin{aligned} |u(T_n) - U_n^k| &\leq C_1 \frac{(\Delta T^{p+1})^{k+1}}{(k+1)!}(1 + C_2\Delta T)^{n-k-1}\prod_{j=0}^{k}(n - j) \\ &\leq C_1 \frac{T_n^{k+1}}{(k+1)!}\exp C_2(T_n - T_{k+1})\Delta T^{p(k+1)}, \end{aligned}$$

where C_1 *is a constant depending on the coefficients in* (2.4), *and* C_2 *is the constant in* (2.5).

3. PITA ALGORITHM AND EQUIVALENCE TO THE PARAREAL ALGORITHM

We now present the PITA algorithm introduced by Farhat et al. in [2], and prove that for linear problems the algorithm is equivalent to the multiple shooting method. We consider the linear system of first order ODEs

$$(3.1) \qquad\qquad u'(t) = Au(t) + b(t),$$

with $A \in \mathcal{M}_d(\mathbb{R})$ and $b(\cdot) \in \mathbb{R}^d$.

The time interval $[0, T]$ is partitioned into N time subdomains Ω_n of constant size ΔT, obtaining this way a coarse grid. Each time subdomain is then partitioned in turn into J subintervals of size Δt, obtaining a fine grid. Then the PITA algorithm is given by

- Step 0: Provide an initial approximate solution U_n^0, $n = 0, \dots, N$ by applying a sequential numerical method to problem (3.1) on the coarse grid.
- Step $k + 1$:
  - Using the available approximate solution U_n^k as initial condition for problem (3.1) considered on each time slice, apply a sequential numerical

method on the decomposed fine grid,

(3.2)
$$(u_n^k)'(t) = A u_n^k(t) + b(t), \quad \text{on } \Omega_n,$$
$$u_n^k(T_n) = U_n^k.$$

Note that the computations (3.2) can be done in parallel for each time slice.

– Evaluate the jumps

(3.3)
$$S_n^k = u_{n-1}^k(T_n) - U_n^k, \ 1 \le n \le N,$$

on the coarse grid. If the jumps are small enough, the algorithm terminates.

– Solve, by the numerical method applied on the coarse grid, sequentially the correction problem

(3.4)
$$(c_n^k)'(t) = A c_n^k(t), \quad c_0^k(T_0) = 0, \ c_n^k(T_n) = c_{n-1}^k(T_n) + S_n^k,$$

and compute the correction $C_n^k = c_{n-1}^k(T_n)$.

– Update the approximate solution using the formula

(3.5)
$$U_n^{k+1} = u_{n-1}^k(T_n) + C_n^k = U_n^k + S_n^k + C_n^k, \quad \text{for all } 1 \le n \le N.$$

Proposition 3.1. *For the linear problem (3.1), the PITA algorithm described above is equivalent to the multiple shooting algorithm (2.2), (2.3).*

Proof. We first note that the fine propagator of the multiple shooting method corresponds to applying a sequential numerical method on the fine grid, while the coarse propagator corresponds to applying a sequential numerical method on the coarse grid. Clearly the initial step is then the same for both methods. Rewriting the PITA method using the propagators, we find that the jumps are

$$S_n^k = F(T_n, T_{n-1}, U_{n-1}^k) - U_n^k.$$

Since we propagate the jumps on the coarse grid, the corrections are

(3.6)
$$C_{n+1}^k = G(T_{n+1}, T_n, C_n^k + S_n^k) - G(T_{n+1}, T_n, 0).$$

Using the updating formula (3.5), we obtain

$$U_n^{k+1} = u_{n-1}^k(T_n) + C_n^k = U_n^k + S_n^k + C_n^k,$$

which implies that

(3.7)
$$S_n^k + C_n^k = U_n^{k+1} - U_n^k.$$

Rewriting the updating formula (3.5) using the fine propagator, we find with the expression for the correction (3.6), relation (3.7), and using linearity

$$
\begin{aligned}
U_{n+1}^{k+1} &= F(T_{n+1}, T_n, U_n^k) + C_{n+1}^k \\
&= F(T_{n+1}, T_n, U_n^k) + G(T_{n+1}, T_n, C_n^k + S_n^k) - G(T_{n+1}, T_n, 0) \\
&= F(T_{n+1}, T_n, U_n^k) + G(T_{n+1}, T_n, U_n^{k+1} - U_n^k) - G(T_{n+1}, T_n, 0) \\
&= F(T_{n+1}, T_n, U_n^k) + G(T_{n+1}, T_n, U_n^{k+1}) - G(T_{n+1}, T_n, U_n^k),
\end{aligned}
$$

which coincides with the updating formula (2.3) of the multiple shooting algorithm.

$\square$

When the parareal algorithm is applied to the system of second order ODEs

$$
(3.8) \qquad Mq'' + Dq' + Kq = f(t), \ q(T_0) = q_0, \ q'(T_0) = q_0',
$$

where $M, D, K \in \mathcal{M}_{\tilde{d},\tilde{d}}(\mathbb{R})$ represent respectively the mass, the damping and the stiffness matrices, the algorithm does not perform well, because of a beating phenomenon, see [3]. In order to understand this, we rewrite equation (3.8) in the homogeneous case as a system of first order ODEs, assuming that the mass matrix is invertible, and obtain

$$
(3.9) \qquad u' = Au, \ u(T_0) = u_0,
$$

where the matrix $A \in \mathcal{M}_{d,d}(\mathbb{R})$ is given by

$$
\begin{pmatrix} -M^{-1}D & -M^{-1}K \\ Id & O \end{pmatrix},
$$

and $u_0 = (q_0', q_0)' \in \mathbb{R}^d$, with $d = 2\tilde{d}$. Now the exact solution of the problem can be expressed as a superposition of harmonic functions, which are related to the natural frequencies of the system, i.e. the eigenvalues of matrix A, which are purely imaginary numbers. When we compute the approximate solution using formula (2.3), we can have a beating phenomenon caused by the interference of two modes not having the same frequencies,

$$
F(T_{n+1}, T_n, \phi_i) = e^{i\alpha_i}\phi_i,
$$

with α_i a constant depending on the eigenvalues of the matrix A and on the time-step ΔT, and ϕ_i a natural mode of the problem. Suppose that α and β are two natural frequencies that appear in U_n^{k+1} and U_n^k. Then, when we compute $U_n^{k+1} - U_n^k$ we find that this difference is characterized by the frequencies $(\alpha + \beta)/2$ and $(\alpha - \beta)/2$. If α and β are frequencies close to each other, the order of the new frequency $(\alpha - \beta)/2$ becomes very small and the frequency $(\alpha + \beta)/2$ becomes very close to the natural frequencies α and β and while computing $G(T_{n+1}, T_n, U_n^{k+1}) - G(T_{n+1}, T_n, U_n^k)$ the result might not be accurate unless ΔT is very small.

4. Krylov subspace enhanced parareal algorithm

In [1], [2] and [3], the authors showed that the parareal algorithm produces a speed-up for certain first order ODEs, but the method does not have the same potential when applied to systems of second order ODEs. It was also shown in [1], as well as in [8] that the parareal algorithm is not stable for certain second order ODEs (in fact the method is unstable when the system presents purely imaginary eigenvalues) or for most hyperbolic problems (since the situation of purely complex eigenvalues or complex eigenvalues with large imaginary part appears). We propose and analyze here a modified parareal algorithm, based on an idea presented in the PITA framework in [3] as a remedy for a beating phenomenon when the algorithm is applied to a system of second order ODEs. We consider the inhomogeneous linear system of first order ODEs

$$(4.1) \qquad u'(t) = Au(t) + f(t), \ u(T_0) = u_0,$$

where $A \in \mathcal{M}_{d,d}(\mathbb{R})$, $f : \mathbb{R} \to \mathbb{R}^d$ and $u : \mathbb{R} \to \mathbb{R}^d$.

4.1. **Homogeneous case.** We start by considering the homogeneous case; we will see later that the inhomogeneous case can also be treated in a similar manner. In the correction step (2.3) of the original parareal algorithm, the fine propagator is used for every time subdomain, and if the equation is linear and homogeneous, one knows by linearity rapidly the evolution of the solution for a larger and larger subspace of initial conditions. This information can be used to obtain a more accurate approximation for the coarse propagator G when we compute $G(T_{n+1}, T_n, U_n^{k+1} - U_n^k)$ in (2.3) (we used linearity), at no extra cost. The idea is to use the fine propagator F for the part of $U_n^{k+1} - U_n^k$ in the subspace for which the evolution is already known from previous evaluations of the fine propagator, and to only propagate the rest with the coarse propagator. More formally, we define the space

$$(4.2) \qquad \mathcal{S}^k = \text{span}\{U_n^l; \ 0 \le l \le k, \ 0 \le n \le N\},$$

and replace in (2.3) the coarse propagation $G(T_{n+1}, T_n, Y)$ by $K(T_{n+1}, T_n, Y)$, where the new propagator K is defined by

$$(4.3) \qquad K(T_{n+1}, T_n, Y) = F(T_{n+1}, T_n, P^k Y) + G(T_{n+1}, T_n, (I - P^k)Y),$$

and P^k is the orthogonal projection onto the space $\mathcal{S}^k$,

$$(4.4) \qquad P^k = S^k (S^k)^T.$$

Introducing this into (2.3), we obtain by linearity after simplification the modified update

$$(4.5) \qquad U_{n+1}^{k+1} = F(T_{n+1}, T_n, P^k U_n^{k+1}) + G(T_{n+1}, T_n, (I - P^k)U_n^{k+1}).$$

Note that if we formally take $k \to \infty$ in (4.5), we have that the projection P_k tends to the identity and so U_n^{k+1} tends to the approximate solution given by the

fine propagator. A rigorous explanation for the fact that P_k tends to the identity when the algorithm converges can be found in Remark 4.2 where we show that we converge when the spaces $\{\mathcal{S}_k\}_k$ do not increase anymore.

The modified parareal algorithm computes this update performing the following steps:

- Step 0: As for the original parareal algorithm, we start with the initial approximations U_n^0 from (2.2), and the initial subspace $\mathcal{S}^{-1}$ is empty, since we have done no evaluation of the fine propagator F.
- Step $k + 1$:
  - Compute in parallel $F(T_{n+1}, T_n, U_n^k)$.
  - We enhance the subspace of known fine evolution by

$$(4.6) \qquad \mathcal{S}^k = \operatorname{span}(\mathcal{S}^{k-1} \cup \{U_n^k, n = 0, \ldots, N\}),$$

   since now the fine evolution for all initial conditions U_n^k is known. We also construct an orthogonal basis for $\mathcal{S}^k$, whose columns we store in the matrix S^k, and the associated orthogonal projector P^k using (4.4).
  - We perform the sequential updating step (4.5) for $n = 0, \ldots, N$, at no extra evaluation of the fine propagator, since the fine evolution on the subspace $\mathcal{S}^k$ is known.

4.2. **Properties of the subspaces $\mathcal{S}^k$.** Initially, the matrix S^0 contains as columns the linearly independent vectors that generate

$$\mathcal{S}^0 = \operatorname{span}(U_n^0; \ 0 \le n \le N).$$

In the algorithm, the vectors U_n^0 are found using the coarse propagator G,

$$U_n^0 = G(T_n, T_{n-1}, U_{n-1}^0) = G U_{n-1}^0, \quad U_0^0 = U_0,$$

where we use the same symbol G for the matrix that defines the coarse propagator G in the homogeneous linear case under consideration. We therefore obtain that

$$(4.7) \qquad \mathcal{S}^0 = \mathcal{K}_{N+1}(U_0, G),$$

where $\mathcal{K}_{N+1}(U_0, G)$ is a Krylov space. Therefore, the dimension of $\mathcal{S}^0$ is given by

$$(4.8) \qquad p := \dim \mathcal{K}_{N+1}(U_0, G) = \min(N + 1, \operatorname{grade}(U_0)),$$

where $\operatorname{grade}(U_0)$ is the degree of the minimal polynomial of G with respect to U_0. It is then well known that

$$\mathcal{S}^0 = \mathcal{K}_{N+1}(U_0, G) = \mathcal{K}_p(U_0, G),$$

and hence the matrix S^0 contains as columns the orthonormalized vectors $U_0, U_1^0, \ldots, U_{p-1}^0$.

Proposition 4.1. *If the matrix G has only simple eigenvalues, then the degree p of the matrix G with respect to the vector U_0 coincides with the number of natural modes excited by the initial condition.*

Proof. Let ϕ_i, $i = 1, \ldots, N$ be the natural modes of the system (3.9), i.e. the eigenvectors of the matrix A. One can easily see that the matrices A and G have the same eigenvectors. Let λ_i be the eigenvalues of G corresponding to the eigenvectors ϕ_i, and suppose that the initial condition can be written in the form

$$U_0 = \beta_1 \phi_1 + \ldots + \beta_l \phi_l, \quad \text{with } \beta_i \neq 0 \text{ for } i = 1, 2, \ldots, l.$$

We know that the grade of G with respect to U_0 is the smallest degree of a polynomial p such that $p(G)U_0 = 0$. Since

$$(4.9) \qquad p(G)U_0 = \beta_1 p(G)\phi_1 + \ldots + \beta_l p(G)\phi_l = \beta_1 p(\lambda_1)\phi_1 + \ldots + \beta_l p(\lambda_l)\phi_l = 0,$$

and taking into account the linear independence of the eigenvectors and the fact that all the coefficients β_i are nonzero, we must have $p(\lambda_i) = 0$ for all $i = 1, \ldots, l$. Thus the smallest degree of the polynomial is l, the number of natural modes excited by the initial condition U_0. $\qquad\qquad\square$

After this initial step, the algorithm enhances at each step the subspace $\mathcal{S}^{k-1}$ with the vectors obtained from the current step, $U_0^k, \ldots, U_N^k$. If the matrix S^{k-1} is known from the previous iteration, we can construct the new matrix S^k for example by using modified Gram-Schmidt to form the extended orthogonal basis, or QR factorization steps.

Remark 4.2. The structure of the space $\mathcal{S}^0$ is quite simple, as we have seen before. For $k \geq 1$ the spaces $\mathcal{S}^k$ are much more complicated. To see why, we just give a simple example: consider in what follows that both the coarse and the fine propagators are obtained using the Euler method respectively on a coarse and fine grid. Then, $F = (I + \Delta t A)^J$ and $G = I + \Delta T A$. The approximations at Step 0 are $U_k^0 = (1 + \Delta T A)^k$ and the space $\mathcal{S}^0$ is:

$$\mathcal{S}^0 = \text{span}\{U_0^0, \ldots, U_N^0\} = \text{span}\{U_0, AU_0, \ldots, A^N U_0\}.$$

In order to find $\mathcal{S}^1$, we need to first compute the approximations at Step 1. It can be easily seen that $U_0^1 = U_0$ and, since $U_0 \in \mathcal{S}^0$, we find $U_1^1 = F(T_1, T_0, P^0 U_0^1) = (I + \Delta t A)^J U_0$ and $U_2^1 = F(T_2, T_1, P^0 U_1^1) + G(T_2, T_1, (I - P^0)U_1^1)$. To further compute U_2^1 we need to know if U_1^1 is an element of $\mathcal{S}^0$. If so (case possible if for example $J \leq dim(\mathcal{S}^0)$) then $U_2^1 = (I + \Delta t A)^{2J} U_0$; if not, we need to project U_1^1 into the space $\mathcal{S}^0$, use the fine projector for the part $P^0 U_1^1$ and the coarse projector for the remaining part. So, the approximations at Step k are linear combinations of vectors of the form $A^l U_0$, but l will in general be from a non-contiguous subset of $\mathbb{N}$, and thus form a "broken" Krylov space, in contrast to the case at Step 0.

4.3. **Inhomogeneous case.** We now show how the inhomogeneous case can be treated with the enhanced subspace in the same manner as the homogeneous one.

To this end, we first note that the action of the fine propagator for a linear inhomogeneous problem can be decomposed by linearity into

$$\begin{aligned}
F(T_{n+1}, T_n, \alpha X + \beta Y) &= F^J(\alpha X + \beta Y) + F(T_{n+1}, T_n, 0) \\
&= \alpha F^J X + \beta F^J Y + F(T_{n+1}, T_n, 0) \\
&= \alpha(F(T_{n+1}, T_n, X) - F(T_{n+1}, T_n, 0)) \\
&\quad + \beta(F(T_{n+1}, T_n, Y) - F(T_{n+1}, T_n, 0)) + F(T_{n+1}, T_n, 0),
\end{aligned}$$

where F is the matrix that defines the fine propagator. The affine computation for $F(T_{n+1}, T_n, P_k U_n^{k+1})$, and similarly for $G(T_{n+1}, T_n, P_k U_n^{k+1})$, requires thus an additional preprocessing step, which leads to the algorithm:

- Step -1: We start the algorithm by computing the quantities $F(T_{n+1}, T_n, 0)$ and $G(T_{n+1}, T_n, 0)$. The utility of these terms is seen at Step $k+1$ when we need to update the approximation.
- Step 0: At this step we are doing the same computations as for the homogeneous case, in order to obtain initial approximations U_n^0. Also the initial subspace $\mathcal{S}^{-1}$ is empty.
- Step $(k+1)$:
    - Compute in parallel $F(T_{n+1}, T_n, U_n^k)$.
    - Enhance the subspace of known fine evolution by
      $$\mathcal{S}^k = \text{span}(\mathcal{S}^{k-1} \cup \{U_n^k, n = 0, \ldots, N\}),$$
      as in the homogeneous case, and compute the matrix S^k and the projector P^k.
    - Update the approximate solution using the formula

$$(4.10) \quad U_{n+1}^{k+1} = F(T_{n+1}, T_n, P_k U_n^{k+1}) + G(T_{n+1}, T_n, (I - P_k)U_n^{k+1}) - G(T_{n+1}, T_n, 0).$$

Note here that if we formally take $k \to \infty$ in (4.10), we find that P_k tends to I and so U_{n+1}^{k+1} tends to the fine solution. The last term in (4.10) was zero for the homogeneous case, here it is not zero but it is a term that was computed at Step -1.

We also note that at Step $k+1$ the term $F(T_{n+1}, T_n, P_k U_n^{k+1})$ can be computed at minimal cost, since $P_k U_n^{k+1}$ is a linear combination of approximations for which the evolution is known. In fact, $F(T_{n+1}, T_n, P_k U_n^{k+1})$ is computed as a linear combination of $F(T_{n+1}, T_n, 0)$ computed at Step -1 and of $F(T_{n+1}, T_n, S^k)$,

$$F(T_{n+1}, T_n, P_k U_n^{k+1}) = [F(T_{n+1}, T_n, S^k) - F(T_{n+1}, T_n, \mathbf{0})](S^k)^T U_n^{k+1} + F(T_{n+1}, T_n, 0),$$

where by $F(T_{n+1}, T_n, S^k)$ we understand the matrix having as columns the evolution by the fine propagator F of the column vectors of the matrix S^k, and $F(T_{n+1}, T_n, \mathbf{0})$ is the matrix of the same dimension as S^k, having as columns the vector $F(T_{n+1}, T_n, 0)$. The matrix $F(T_{n+1}, T_n, S^k)$ is known once we computed in parallel the evolution of

U_n^k using the fine propagator: if U is the matrix of all the generators of $\mathcal{S}^k$, we compute (or update) the QR factorization of U ($U = QR$), and we simply compute

$$F(T_{n+1}, T_n, S^k) = [F(T_{n+1}, T_n, U) - F(T_{n+1}, T_n, \mathbf{0})] \cdot R^{-1} + F(T_{n+1}, T_n, 0).$$

4.4. Modified PITA algorithm and equivalence with the Krylov enhanced parareal algorithm.
The idea of modifying the multiple shooting method in order to be effective for second order ODEs came from [3], where the authors proposed a modified PITA algorithm. In what follows, we describe the modified PITA algorithm and we show that the two methods are equivalent for linear problems. We denote for the new PITA method the amplification matrices of a given sequential numerical method associated with the coarse and fine time-grids by G and F respectively. Then, the new PITA method works as follows:

- Step 0: Provide an initial approximate solution U_n^0 with $0 \leq n \leq N$ by applying a numerical method to problem (3.1) on the coarse grid.
- Step $k + 1$:
  - Using the available approximate solution U_n^k, $n = 0, \ldots, N$ as initial condition for problem (3.1) considered on each time slice, apply a sequential numerical method on the decomposed fine grid:

(4.11)
$$\begin{aligned}(u_n^k)'(t) &= A u_n^k(t) + b(t), \quad \text{on } \Omega_n, \\ u_n^k(T_n) &= U_n^k.\end{aligned}$$

    We note that the computations (4.11) can be done in parallel for each time slice.
  - Evaluate the jumps:

(4.12)
$$S_n^k = u_{n-1}^k(T_n) - U_n^k, \ 1 \leq n \leq N,$$

    on the coarse grid. If the jumps are small enough, we terminate the algorithm.
  - Construct the space $\mathcal{S}^k$ and the corresponding projection P^k (same definition as in (4.2) and (4.4)).
  - Compute the correction propagating the jump on both the coarse and fine time-grids:

(4.13)
$$\begin{aligned}C_0^k &= 0, \\ C_{n+1}^k &= (F^J P^k + G(I - P^k))(C_n^k + S_n^k),\end{aligned}$$

    with $0 \leq n \leq N - 1$.
  - Update the approximate solution:

(4.14)
$$U_n^{k+1} = u_{n-1}^k(T_n) + C_n^k = U_n^k + S_n^k + C_n^k, \quad \text{for all } 1 \leq n \leq N.$$

Proposition 4.3. *For linear problems, the modified PITA algorithm described above is equivalent to the modified multiple shooting algorithm proposed.*

Proof. We first note that for the linear case, the fine propagator of the multiple shooting method can be written in terms of the amplification matrix F,

$$F(T_n, T_{n-1}, Y) = F^J Y + F(T_n, T_{n-1}, 0),$$

and the coarse propagator can be written in terms of the amplification matrix G,

$$G(T_n, T_{n-1}, Y) = GY + G(T_n, T_{n-1}, 0).$$

The corrections can therefore be written in the form

$$\begin{aligned}
C_0^k &= 0, \\
C_{n+1}^k &= (F^J P^k + G(I - P^k))(C_n^k + S_n^k) \\
&= F(T_{n+1}, T_n, P^k(C_n^k + S_n^k)) - F(T_n, T_{n-1}, 0) \\
&\quad + G(T_{n+1}, T_n, (I - P^k)(C_n^k + S_n^k)) - G(T_n, T_{n-1}, 0).
\end{aligned}$$

Since we know that $S_n^k + C_n^k = U_n^{k+1} - U_n^k$, we find that the updated solution is

$$\begin{aligned}
U_{n+1}^{k+1} &= F(T_{n+1}, T_n, U_n^k) + F(T_{n+1}, T_n, P^k(U_n^{k+1} - U_n^k)) - F(T_n, T_{n-1}, 0) \\
&\quad + G(T_{n+1}, T_n, (I - P^k)(U_n^{k+1} - U_n^k)) - G(T_n, T_{n-1}, 0) \\
&= F(T_{n+1}, T_n, P^k U_n^{k+1}) + G(T_{n+1}, T_n, (I - P^k)U_n^{k+1}),
\end{aligned}$$

where we used the fact that $P^k U_n^k = U_n^k$. We thus obtained the same formula for updating the approximate solution in the PITA algorithm as for the modified multiple shooting method. $\square$

5. MATHEMATICAL ANALYSIS

In [3], the authors have already shown for the modified PITA algorithm that when choosing two p-th order sequential coarse and fine numerical methods, then the algorithm converges in a finite number of steps and the algorithm preserves the accuracy of the fine numerical method. We present now a refined convergence analysis for the new parareal algorithm. For simplicity, we consider the homogeneous case only, the proofs for the inhomogeneous case follow the same ideas. In what follows, we assume that the initial condition excites l-natural modes, i.e. $U_0 = \beta_1 \phi_1 + \ldots + \beta_l \phi_l$, where all the coefficients β_i are non-zero. We can then prove the following convergence result:

Theorem 5.1. *The new parareal algorithm converges as soon as $\mathcal{S}^{k-1} = \mathcal{S}^k$. Furthermore, we have that $\mathcal{S}^k \subset span(\phi_1, \ldots, \phi_l)$ for all k.*

Proof. The relation $\mathcal{S}^{k-1} = \mathcal{S}^k$ means in fact that $\{U_n^k\}_n \subset \mathcal{S}^{k-1}$. Since U_n^k belongs to $\mathcal{S}^{k-1}$, the projection of U_n^k on $\mathcal{S}^{k-1}$ is just the identity. Now, recalling the way

we update the approximate solution,

$$U_{n+1}^k = F(T_{n+1}, T_n, P^{k-1}U_n^k) + G(T_{n+1}, T_n, (I - P^{k-1})U_n^k)$$
$$= F(T_{n+1}, T_n, U_n^k),$$

we can see that $\{U_n^k\}_n$ is exactly the solution given by a sequential computation using the fine propagator. In other words, the moment $\mathcal{S}^{k-1} = \mathcal{S}^k$, we converged to the approximate solution given by $F(T_{n+1}, T_n, U_n^k)$.

It remains to prove that $\mathcal{S}^k \subset \text{span}(\phi_1, \ldots, \phi_l)$. The argument is inductive: For $k = 0$ we know that the space $\mathcal{S}^0$ coincides with $\mathcal{K}_N(U_0, G)$, where $\mathcal{K}_N(U_0, G)$ is the Krylov space associated with U_0 and the matrix G of the coarse propagator. Since the matrices A, G and F have the same eigenvectors, we obtain

$$\mathcal{S}^0 \subset \text{span}(\phi_1, \ldots, \phi_l).$$

We now suppose that $\mathcal{S}^k \subset \text{span}(\phi_1, \ldots, \phi_l)$ and we want to show that $\mathcal{S}^{k+1} \subset \text{span}(\phi_1, \ldots, \phi_l)$, which is equivalent to proving that U_n^{k+1} belongs to $\text{span}(\phi_1, \ldots, \phi_l)$, for all n.

The proof is by induction, this time on the index n. For $n = 0$ the assumption holds since $U_0^{k+1} = U_0$ and the initial condition is in $\text{span}(\phi_1, \ldots, \phi_l)$ by definition. If $U_n^{k+1} \in \text{span}(\phi_1, \ldots, \phi_l)$, then

$$P^k U_n^{k+1} \in \mathcal{S}^k \subset \text{span}(\phi_1, \ldots, \phi_l).$$

Taking into account that the matrices A, G and F have the same eigenvectors, we find

$$U_{n+1}^{k+1} = F(T_{n+1}, T_n, P^k U_n^{k+1}) + G(T_{n+1}, T_n, (I - P^k)U_n^{k+1}) \in \text{span}(\phi_1, \ldots, \phi_l),$$

which concludes the proof.   □

We now present a convergence estimate of the new method. In what follows, we denote by $\tilde{F}$ the exact solution of the equation and we assume that the coarse propagator G satisfies the property

$$(5.1) \qquad \tilde{F}(T_n, T_{n-1}, x) - G(T_n, T_{n-1}, x) = c_{p+1}(x)\Delta T^{p+1} + c_{p+2}(x)\Delta T^{p+2} + \ldots$$

The fine propagator acts on a subdivision of each subdomain Ω_n, the size of the subdivision being $\Delta t = t_n - t_{n-1}$. We then assume that F satisfies

$$(5.2) \qquad \tilde{F}(t_n, t_{n-1}, x) - F(t_n, t_{n-1}, x) = c'_{p+1}(x)\Delta t^{p+1} + c'_{p+2}(x)\Delta t^{p+2} + \ldots$$

We also assume that both propagators satisfy the Lipschitz condition (2.5).

Theorem 5.2. *Let $||\cdot||$ be the norm on $\mathbb{R}^N$. If the propagators F and G considered to solve the linear homogeneous system of ODEs are of order p, then the hybrid propagator K is also of order p.*

Proof. We recall that the case considered here is the linear one. We evaluate the difference between the exact solution and the approximation given by K. For an arbitrary vector Y from $\mathbb{R}^N$ and a time step ΔT small enough, we obtain

(5.3)
$$
\begin{aligned}
&||\tilde{F}(T_{n+1}, T_n, Y) - K(T_{n+1}, T_n, Y)|| \\
&= ||\tilde{F}(T_{n+1}, T_n, Y) - F(T_{n+1}, T_n, P^k Y) - G(T_{n+1}, T_n, (I - P^k)Y)|| \\
&\leq C||P^k Y||\Delta t^{p+1} + C||(I - P^k)Y||\Delta T^{p+1} \\
&\leq C(\Delta t^{p+1} + \Delta T^{p+1})||Y||,
\end{aligned}
$$

where C is a constant independent of the time-steps, related to the constants from (5.1) and (5.2). $\square$

Theorem 5.3. *If F and G are two propagators of order p, then the error of the new parareal algorithm satisfies at iteration k the estimate*

(5.4) $\quad ||u(T_n) - U_n^k|| \leq C_1 \dfrac{T_n^{k+1}}{(k+1)!} \exp C_2(T_n - T_{k+1})\triangle T^{p(k+1)} + C_3 T_n e^{C_2 \Delta T}\Delta t^p,$

where C_1, C_2, C_3 are constants independent to the time-steps Δt and ΔT, but related to the constants from (2.5), (5.1), (5.2).

Proof. Let $u(T_n) = \tilde{F}(T_n, T_{n-1}, u(T_{n-1}))$ be the exact solution of the problem. We need to estimate the error of the method at iteration $k + 1$,

(5.5)
$$
\begin{aligned}
E_n^{k+1} &= u(T_n) - U_n^{k+1} \\
&= \tilde{F}(T_n, T_{n-1}, u(T_{n-1})) - F(T_n, T_{n-1}, U_{n-1}^k) - K(T_n, T_{n-1}, U_{n-1}^{k+1} - U_{n-1}^k) \\
&= \tilde{F}(T_n, T_{n-1}, u(T_{n-1}) - U_{n-1}^k) - K(T_n, T_{n-1}, u(T_{n-1}) - U_{n-1}^k) \\
&\quad + \tilde{F}(T_n, T_{n-1}, U_{n-1}^k) - F(T_n, T_{n-1}, U_{n-1}^k) + K(T_n, T_{n-1}, u(T_{n-1}) - U_{n-1}^{k+1}),
\end{aligned}
$$

where in the last relation of (5.5) we added and subtracted $K(T_n, T_{n-1}, u(T_{n-1})) + \tilde{F}(T_n, T_{n-1}, U_{n-1}^k)$. Using (5.1), (5.2) and the fact that the propagators F and G satisfy the Lipschitz condition, we find

(5.6) $\quad ||E_n^{k+1}|| \leq \alpha||u(T_{n-1}) - U_{n-1}^k|| + (1 + C_2\Delta T)||u(T_{n-1}) - U_{n-1}^{k+1}|| + C\Delta t^{p+1},$

with $\alpha = C(\Delta t^{p+1} + \Delta T^{p+1})$ given by relation (5.3). Initially, we have

(5.7)
$$
\begin{aligned}
||E_n^0|| &= ||u(T_n) - U_n^0|| = ||\tilde{F}(T_n, T_{n-1}, u(T_{n-1})) - G(T_n, T_{n-1}, U_{n-1}^0)|| \\
&\leq ||\tilde{F}(T_n, T_{n-1}, u(T_{n-1})) - G(T_n, T_{n-1}, u(T_{n-1}))|| \\
&\quad + ||G(T_n, T_{n-1}, u(T_{n-1}) - U_{n-1}^0)|| \\
&\leq C\Delta T^{p+1} + (1 + C_2\Delta T)||u(T_{n-1}) - U_{n-1}^0||.
\end{aligned}
$$

Comparing (5.6) and (5.7) with the results obtained by Gander and Hairer in [4] for the original parareal algorithm, we see that the error of the new method satisfies a similar recurrence relation. Following the approach in [4], we study the recurrence relation

$$(5.8) \qquad e_n^{k+1} = \alpha e_{n-1}^k + \beta e_{n-1}^{k+1} + \delta, \quad e_n^0 = \gamma + \beta e_{n-1}^0,$$

where $\alpha = C(\Delta t^{p+1} + \Delta T^{p+1})$, $\beta = 1 + C_2\Delta T$, $\delta = C\Delta t^{p+1}$ and $\gamma = C\Delta T^{p+1}$. We multiply relation (5.8) by x^n, and summing over n, we find that the generating function $\rho_k(x) = \sum_{n \geq 1} e_n^k x^n$ satisfies the recurrence relation

$$(5.9) \qquad \begin{aligned} \rho_{k+1}(x) &= \alpha x \rho_k(x) + \beta x \rho_{k+1}(x) + \delta \frac{x}{1-x}; \\ \rho_0(x) &= \beta x \rho_0(x) + \gamma \frac{x}{1-x}. \end{aligned}$$

Solving (5.9) by induction, we obtain

$$(5.10) \quad \rho_{k+1}(x) = \gamma \alpha^k \frac{x^{k+1}}{1-x} \frac{1}{(1-\beta x)^{k+1}} + \delta \frac{x}{1-x} \frac{1}{1-(\alpha+\beta)x} \left(1 - \alpha^k \frac{x^k}{(1-\beta x)^k}\right),$$

which we bound from above by

$$(5.11) \qquad \rho_{k+1}(x) \leq \gamma \alpha^k x^{k+1} \frac{1}{(1-\beta x)^{k+2}} + \delta x \frac{1}{(1-(\alpha+\beta)x)^2}.$$

Using the polynomial series expansion

$$\frac{1}{(1-\beta x)^{k+2}} = \sum_{j \geq 0} \binom{k+1+j}{j} \beta^j x^j,$$

we find for the n-th coefficient e_n^k of the function ρ_k the bound

$$e_n^k \leq \gamma \alpha^k \beta^{n-k-1} \binom{n}{k+1} + \delta \binom{n}{n-1}(\alpha+\beta)^{n-1}.$$

We thus obtain

$$(5.12) \qquad \begin{aligned} \|E_n^k\| &\leq C'^{k+2}(\Delta t^{p+1} + \Delta T^{p+1})^{k+1}(1 + C_2\Delta T)^{n-k-1}\binom{n}{k+1} \\ &\quad + C\Delta t^{p+1} n (1 + C_2\Delta T)^{n-1} \\ &\leq C_1 \frac{T_n^{k+1}}{(k+1)!} e^{C_2(T_n - T_{k+1})} \Delta T^{p(k+1)} + C_3 T_n e^{C_2 T_n} \Delta t^p, \end{aligned}$$

which proves the theorem.   □

Remark 5.4. From Theorem 5.3 we see that at each iteration the accuracy of the method improves, but it is limited by the accuracy of the method given by the fine propagator F. More precisely, the error is bounded by

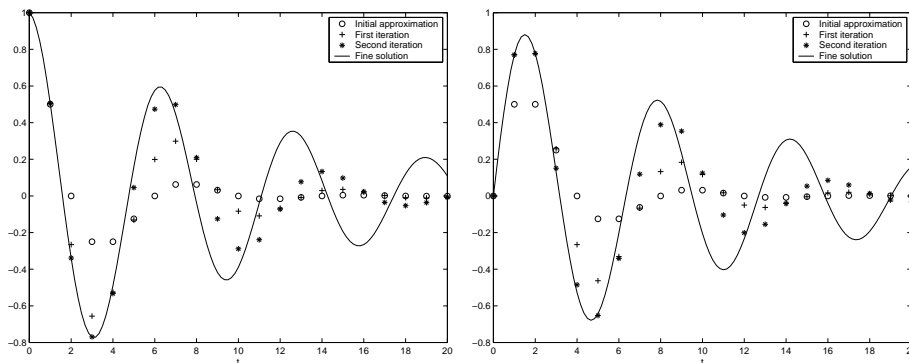$$\|E_n^k\| \leq C_k \max(\Delta T^{(k+1)p}, \Delta t^p).$$

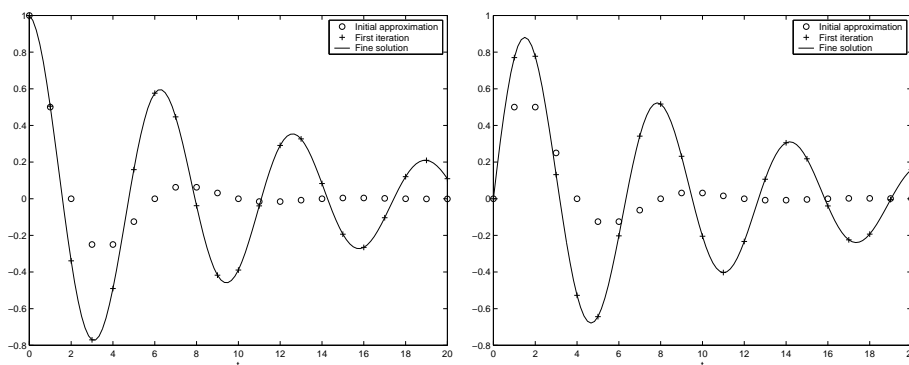FIGURE 1. First few iterations of the original parareal algorithm for the model problem $u'' = -u$.

FIGURE 2. Initial approximation and first iteration of the modified parareal algorithm for the model problem $u'' = -u$.

6. NUMERICAL EXPERIMENTS

We first consider the simple model problem $u'' = -u$ with initial conditions $u(0) = 1$ and $u'(0) = 0$. We transform the equation into a first order system with two components, and perform the simulations on the time interval $[0, 20]$. We choose for the coarse time step $\Delta T = 1$, and for the fine time step $\Delta t = 1/6$. In Figure 1, we show the first few iterations of the original parareal algorithm, on the left for the first component, and on the right for the second component, together with the fine grid solution. One can see that the algorithm is converging slowly, and that the convergence early in the time interval is significantly better than later in the time interval. In Figure 2, we show the initial guess and the first iteration of the modified parareal algorithm, which converges with the first iteration. This illustrates our first convergence result, see Theorem 5.1, since in this low dimensional problem, the fine solution is already contained in the subspace after one iteration.
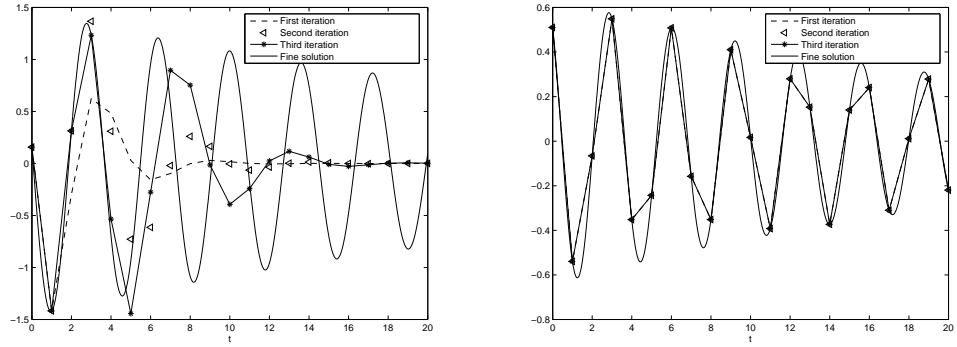
FIGURE 3. The first component of the solution for $u'' + Ku = 0$ after the first few iterations of the original parareal algorithm and of the modified algorithm.
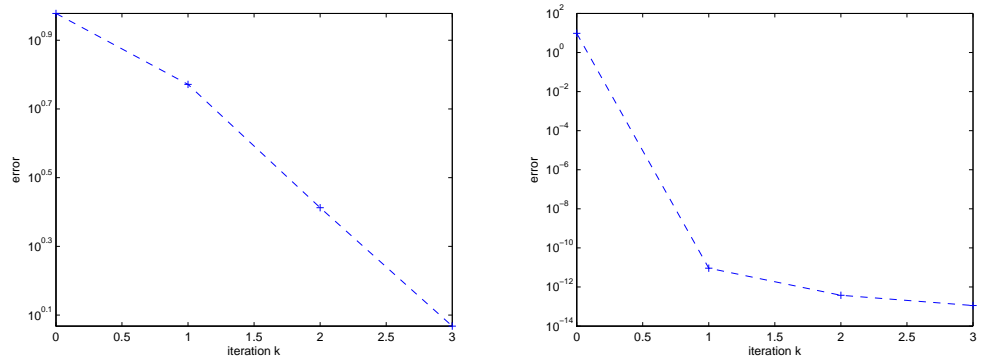

FIGURE 4. Error for the classical and the modified parareal algorithm for $u'' + Ku = 0$.

In Figure 3, we consider the model problem $u'' + Ku = 0$ with K a random matrix of dimension 100, for the time interval $[0, 20]$. We compare the first component of the solution obtained with the classical parareal algorithm and with the modified algorithm. We notice again that the classical parareal algorithm converges slowly, and the convergence is better for short time than for long time. In contrast, the modified parareal algorithm converges very rapidly.

In Figure 4, we show the error between the solutions of this system on the time interval $[0, 5]$, obtained with the fine propagator and the parareal method, using first the original parareal algorithm and then the modified one, for the first three iterations. We can see that the convergence is significantly improved using the modified parareal algorithm.
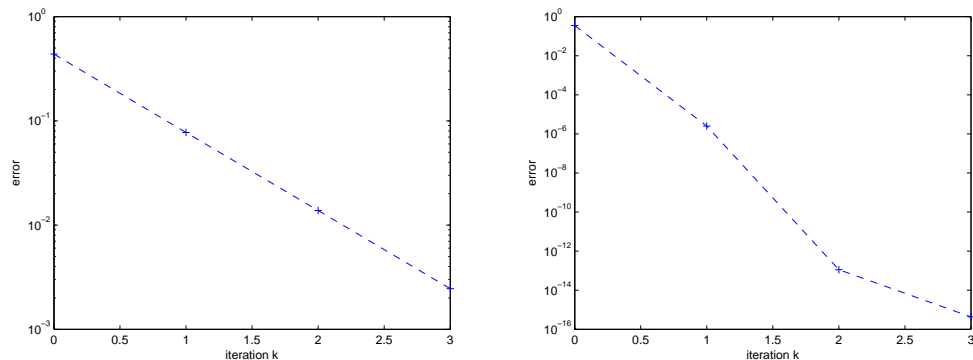
FIGURE 5. Error for the classical and the modified parareal algorithm, applied to the heat equation.

We finally show a comparison for the case of the heat equation $u_t - u_{xx} = 0$ with homogeneous Dirichlet conditions on the spatial domain $[0, 1]$, and with a random initial condition. The time-interval is $[0, 1]$. The modified parareal algorithm also converges faster, as one can see in Figure 5, where we plotted the error measured in the $L^\infty(0, T, L^2)$-norm between the fine solution and the approximation produced by the two algorithms at each iteration.

REFERENCES

[1] G. BAL, *On the convergence and the stability of the parareal algorithm to solve partial differential equations*, Domain decomposition methods in science and engineering, Lect. Notes Comput. Sci. Eng., 40, Springer-Verlag (2004), pp. 425–432.

[2] 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), pp. 1397–1434.

[3] C. FARHAT, J. CORTIAL, C. DASTILLUNG, AND H. BAVESTRELLO, *Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses*, Internat. J. Numer. Methods Engrg., 67 (2006), pp. 697–724.

[4] M. J. GANDER AND E. HAIRER, *Nonlinear convergence analysis for the parareal algorithm*. Lect. Notes Comput. Sci. Eng., 60, Springer-Verlag (2007), pp. 45–56.

[5] M. J. GANDER AND S. VANDERWALLE, *Analysis of the parareal time-parallel time-integration method*, SIAM J. Sci. Comput. 29(2) (2007), pp. 556–578.

[6] J.-L. LIONS, Y. MADAY, AND G. TURINICI, *Résolution d'EDP par un schéma en temps "pararéel"*, C. R. Acad. Sci. Paris Sér. I Math., 332 (2001), pp. 661–668.

[7] Y. MADAY, AND G. TURINICI, *Parallel in time algorithms for quantum control: the parareal time discretization scheme*, Int. J. Quant. Chem., 93 (2003), pp. 223–228.

[8] G. A. STAFF, AND E. M. RØNQUIST, *Stability of the parareal algorithm*, Domain decomposition methods in science and engineering, Lect. Notes Comput. Sci. Eng., 40, Springer-Verlag (2004), pp. 449–456.