On the use of reduced basis methods to accelerate and stabilize the parareal method

Feng Chen, Jan S. Hesthaven, and Xueyu Zhu

Abstract We propose a modified parallel-in-time - parareal - multi-level time integration method which, in contrast to previously proposed methods, employs a coarse solver based on a reduced model, built from the information obtained from the fine solver at each iteration. This approach is demonstrated to offer two substantial advantages: it accelerates convergence of the original parareal method for similar problems and the reduced basis stabilizes the parareal method for purely advective problems where instabilities are known to arise. When combined with empirical interpolation methods (EIM), we develop this approach to solve both linear and nonlinear problems and highlight the minimal changes required to utilize this algorithm to accelerate existing implementations. We illustrate the advantages through algorithmic design, through analysis of stability, convergence, and computational complexity, and through several numerical examples.

1 Introduction

With the number of computational cores on large scale computing platforms increasing, the demands on scalability of computational methods likewise increase, due partly to an increasing imbalance between the cost of memory access, communication and arithmetic capabilities. Among other things, traditional domain decomposition methods tend to stagnate in scaling as the number of cores increases since the computational cost is overwhelmed by other tasks. This suggests a need to consider the development of computational techniques that better balance these constraints and allow for the acceleration of large scale computational challenges.

Feng Chen
Brown University, 182 George Street, Providence, RI 02912, e-mail: feng_chen_1@brown.edu

Jan S. Hesthaven
Brown University, 182 George Street, Providence, RI 02912 e-mail: Jan.Hesthaven@brown.edu

Xueyu Zhu
Brown University, 182 George Street, Providence, RI 02912 e-mail: xueyu_zhu@brown.edu

A recent development in this direction is the parareal method, introduced in [16], providing a strategy for 'parallel-in-time' computations that offers the potential for an increased level of parallelism. Relying on combining a computational inexpensive but inaccurate solver with an accurate and expensive but parallel solver, the parareal method utilizes an iterative, predictor-corrector procedure that allows the expensive solver to run across many processors in parallel. Under suitable conditions, the parareal iteration converges after a small number of iterations to the serial solution [3]. During the last decade, the parareal method has been applied successfully to a number of applications (cf. [18, 19]), demonstrating its potential, accuracy, and robustness.

As a central and serial component, the properties of the coarse solver can impact the efficiency and stability of the parareal algorithm, e.g., if an explicit scheme is used in both the coarse and the fine stage of the algorithm, the efficiency of the parareal algorithm is limited by the upper bound of the time step size [19]. One can naturally also consider a different temporal integration approach such as an implicit approach, although the cost of this can be considerable and often requires the development of a new solver. An attractive alternative is to use a simplified physics model as the coarse solver [2, 18, 17], thereby ignoring small scale phenomenon but potentially impacting the accuracy. The success of such an approach is problem specific.

While the choice of the coarse solver clearly impacts accuracy and overall efficiency, the stability of the parareal method is considerably more subtle. For parabolic and diffusion dominated problems, stability is well understood and observed in many applications [12]. However, for hyperbolic and convection dominated problems, the question of stability is considerably more complex and generally remains open [23, 8, 3]. In [8], the authors propose to regularly project the solution onto an energy manifold approximated by the fine solution. The performance of this projection method was demonstrated for the linear wave equation and the nonlinear Burgers' equation. Another attempt, the Krylov subspace parareal method, builds a new coarse solver by reusing all information from the corresponding fine solver at previous iterations. The stability of this approach was demonstrated for linear problems on linear structural dynamics [10] and a linear 2-D acoustic-advection system [21]. However, the Krylov subspace parareal method appears to be limited to linear problems.

The approach of combining the reduced basis method [20] with the parareal method for parabolic equations was initiated in [13] in which it is demonstrated that a coarse solver based on an existing reduced model offers better accuracy and reduces the number of iterations in the examples considered. However, in this work, there was no discussion on the construction of the reduced model, nor was there any attempt to analyze the stability and convergence of the method.

Inspired by [13, 21], we propose a modified parareal method, referred to as the reduced basis parareal method in which the Krylov subspace is replaced by a subspace spanned by a set of reduced bases, constructed on-the-fly from the fine solver. This method inherits most advantages of the Krylov subspace parareal method and is observed to retain stability and convergence for linear wave problems. We demonstrate that this approach accelerates the convergence in situations where the original parareal already converges. However, it also overcomes several known issues: (i) it deals with nonlinear problems by incorporating methodologies from the reduced basis methods; and (ii) the traditional coarse propagator is needed only once at the very beginning of the algorithm to generate an initial reduced basis. This allows for the time step restrictions to be relaxed as compared to the coarse solver of the original parareal method. The main difference between our method and [13] lies in the reduced approximation space and the construction of reduced bases.

The reduced model, playing the role of the coarse solver, is updated for each iteration while the reduced model in [13] is built only once during an initial offline process. Among other advantages, this allows the proposed method to adapt the dimension of the reduced approximation space based on the regularity of the solution, while in [13] the reduced model remains fixed and must be developed using some other approach.

What remains of this paper is organized as follows. We first review the original parareal method in Section 2.1 and the Krylov subspace parareal method in Section 2.2. This sets the stage for Section 2.3 where we introduce the reduced basis parareal method and discuss different strategies to develop reduced models for problems with nonlinear terms. Section 3 offers some analysis of the stability, convergence, and complexity of the reduced basis parareal method and Section 4 demonstrates the feasibility and performance of the reduced basis parareal method through various linear and nonlinear numerical examples. We conclude the paper in Section 5.

2 Parareal algorithms

To set the stage for the general discussion, let us first discuss the original and the Krylov subspace parareal methods in Section 2.1 and Section 2.2, respectively. We shall highlight issues related to stability and computational complexity to motivate the reduced basis parareal method, introduced in Section 2.3.

2.1 The original parareal method

Consider the following initial value problem:

$$\begin{aligned} &\mathbf{u}_t = \mathbf{L}(\mathbf{u}) := \mathbf{A}\mathbf{u}(t) + \mathbf{N}(\mathbf{u}(t)), \quad t \in (0, T], \\ &\mathbf{u}(0) = \mathbf{u}_0, \end{aligned} \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^N$ is the unknown solution, $\mathbf{L}$ is an operator, possibly arising from the spatial discretization of a PDE, with $\mathbf{A}$ being the linear part of $\mathbf{L}$, and $\mathbf{N}$ the nonlinear part.

In the following, we denote $F_{\delta t}$ as the accurate but expensive fine time integrator, using a constant time step size, δt. Furthermore, $G_{\Delta t}$ is the inaccurate but fast coarse time integrator using a larger time step size, Δt. Generally, it is assumed that $\Delta t \gg \delta t$.

The original parareal method is designed to solve (1) in a parallel-in-time fashion to accelerate the computation. First, $[0, T]$ is decomposed into N_c coarse time intervals or elements:

$$0 = t_0 < \cdots < t_i < \cdots < t_{N_c} = T, \quad t_i = i\Delta T, \quad \Delta T = \frac{T}{N_c}. \tag{2}$$

Assume that

$$\Delta T = N_f \delta t, \quad N_f \in \mathbb{N}, \tag{3}$$

which implies that $T = N_c N_f \delta t$. Denote $F_{\delta t}(\mathbf{u}, t_{i+1}, t_i)$ as the accurate numerical solution integrated from t_i to t_{i+1} by using $F_{\delta t}$ with the initial condition $\mathbf{u}$ and the constant time step size δt. Similarly for $G_{\Delta t}(\mathbf{u}, t_{i+1}, t_i)$. Denote also $\mathbf{u}_n = F_{\delta t}(\mathbf{u}_0, T, 0)$ as the numerical solution generated using only the fine integrator.

Now assume that the k-th iterated approximation $\mathbf{u}_n^k$ is known. The parareal approach proceeds to the $k+1$-th iteration as

$$\mathbf{u}_{n+1}^{k+1} = G_{\Delta t}(\mathbf{u}_n^{k+1}, t_{n+1}, t_n) + F_{\delta t}(\mathbf{u}_n^k, t_{n+1}, t_n) - G_{\Delta t}(\mathbf{u}_n^k, t_{n+1}, t_n), \quad 0 \le k \le N_c - 1. \tag{4}$$

It is easy to see that $F_{\delta t}(\mathbf{u}_n^{k+1}, t_{n+1}, t_n)$ can be done in parallel across all temporal elements. If we take the limit of $k \to \infty$ and assume that the limit of $\{\mathbf{u}_n^k\}$ exists, we obtain [16]:

$$\mathbf{u}_{n+1}^{k+1} \to \mathbf{u}_{n+1} = F_{\delta t}(\mathbf{u}_n, t_{n+1}, t_n). \tag{5}$$

In order to achieve a reasonable efficiency, the number of iterations, N_{it}, should be much smaller than N_c.

With the above notation, the original parareal method is

```
Initialization:
u_0^0 = u_0;
for i ← 0 to N_c do
    u_{i+1}^0 = G_Δt(u_i^0, t_{i+1}, t_i)
end
Iterations:
k = 0;
for k ← 0 to N_it do
    Parallel predictor step:
    for i ← 0 to N_c do
        u_{f i+1}^k = F_δt(u_i^k, t_{i+1}, t_i)
    end
    Sequential correction step:
    for i ← 0 to N_c do
        u_{i+1}^{k+1} = G_Δt(u_i^{k+1}, t_{i+1}, t_i) − u_{f i+1}^k + G_Δt(u_i^k, t_{i+1}, t_i)
    end
end
```

Algorithm 1: The original parareal method

To demonstrate the performance of the original parareal method, let us consider a few numerical examples. Consider first the viscous Burgers' equation:

$$
\begin{aligned}
&u_t + (\frac{u^2}{2})_x = \nu u_{xx}, \quad (x,t) \in (0, 2\pi) \times (0, T], \\
&u(x,0) = \sin(x),
\end{aligned}
\tag{6}
$$

where $T = 2$ and $\nu = 10^{-1}$. 2π-periodic boundary condition is used. The spatial discretization is a P_1 discontinuous Galerkin method (DG) with 100 elements [15]. The time integrator is a first-order forward Euler method and we use the following parameters in the parareal integration

$$
N_c = 100, \quad N_{it} = 5, \quad \Delta t = 10^{-3}, \quad \delta t = 10^{-4}. \tag{7}
$$

Figure 1 illustrates the L_∞-error of the parareal solution at $T = 2$ against the number of iterations. Notice that for this nonlinear problem the algorithm converges after only four iterations, resulting in an expected acceleration in a parallel environment.

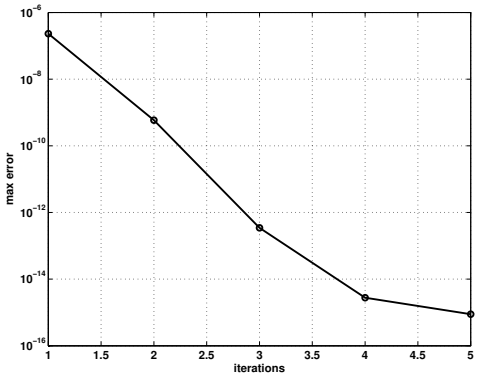

Fig. 1: The L_∞-error at $T = 2$ against the number of iterations of the 1-D Burgers' equation using the original parareal method

As a second example, we consider the Kuramoto-Sivashinsky equation [25]:

$$
\begin{aligned}
&\frac{\partial u}{\partial t} = (\frac{u^2}{2})_x - u_{xx} - u_{xxxx}, \quad (x,t) \in (-8, 8) \times (0, T], \\
&u(x,0) = \exp(-x^2)
\end{aligned}
\tag{8}
$$

with final time $T = 40$ and periodic boundary conditions.

As a spatial discretization we use a Fourier collocation method with 128 points [14] and an IMEX scheme [1] as a time integrator, treating the linear terms implicitly and the nonlinear term explicitly. The parameters in the parareal method are taken as

$$
N_c = 100, \quad N_{it} = 5, \quad \Delta t = 10^{-2}, \quad \delta t = 10^{-4}. \tag{9}
$$

Figure 2 (left) shows the time evolution of the chaotic solution to the Kuramoto-Sivashinsky equation with a Gaussian initial condition. In Figure 2 (right), we show the L_∞-error at $T = 40$

against the number of iterations. In this case, we take the solution computed by the fine solver as the exact solution. It is clear that the parareal solution converges, albeit slower. However, it should also be noted that $\Delta t/\delta t = 100$, indicating the potential for a substantial acceleration.

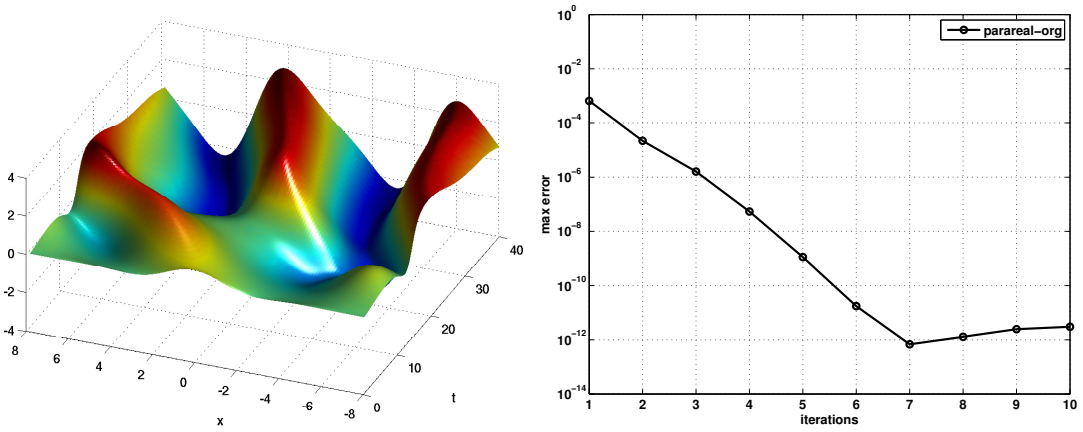

Fig. 2: The time evolution of the solution (left) and the L_∞-error at $T = 40$ against the number of iterations (right) of the 1-D Kuramoto-Sivashinsky equation using the original parareal method.

As a last and less encouraging example, we consider the 1-D advection equation

$$\begin{aligned} &u_t + a u_x = 0, \quad (x,t) \in (0, 2\pi) \times (0, T], \\ &u(x,0) = \exp(\sin(x - at)), \end{aligned} \tag{10}$$

with a final time $T = 10$, $a = 2\pi$ and a 2π-periodic boundary condition. We use a DG method of order 32 and 2 elements in space [15], a singly diagonal implicit fourth-order Runge-Kutta scheme in time (a five-stage fourth-order scheme, cf. S54b in [22]), and the parareal parameters:

$$N_c = 100, \quad N_{it} = 27, \quad \Delta t = 5 \times 10^{-2}, \quad \delta t = 10^{-4}. \tag{11}$$

Figure 3 shows the L_∞-error at $T = 10$ against the number of iterations. The instability of the original parareal method is apparent, as has also been observed by others [3, 23, 8].

2.2 The Krylov subspace parareal method

We notice in Algorithm 1 that only $\{\mathbf{u}_{\mathbf{f}}{}_{i+1}^{k}\}_{i=0}^{N_c-1}$ is used in the advancement of the solution to $k+1$. To fix the stability issue, [10] proposed to improve the coarse solver by reusing information computed at all previous iterations. They applied this idea to the linear hyperbolic problems in structural dynamics. Recently, a similar idea was successfully applied to linear hyperbolic problems [21].

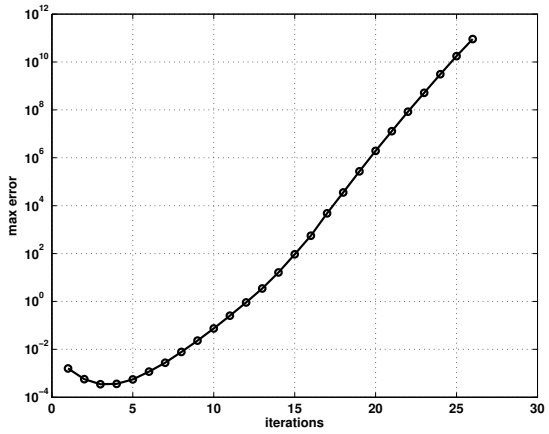

Fig. 3: The L_∞-error at $T = 10$ against the number of iterations of the 1-D linear advection equation using the original parareal method

The basic idea of the Krylov subspace parareal method is to project $\mathbf{u}_i^{k+1}$ onto a subspace spanned by all numerical solutions integrated by the fine solver at previous iterations. Denote the subspace as

$$\mathbf{S}^k := \text{span}\{\mathbf{u}_{\mathbf{f}_i}^j, 1 \le i \le N_c, 1 \le j \le k\}. \tag{12}$$

The corresponding orthogonal basis set $\{\mathbf{s}_1, ..., \mathbf{s}_r\}$ is constructed through a QR decomposition.

Denote $\mathbb{P}^k$ as the L_2-orthogonal projection onto $\mathbf{S}^k$. The previous coarse solver $G_{\Delta t}$ is replaced by $K_{\Delta t}$ as:

$$K_{\Delta t}(\mathbf{u}, t_{i+1}, t_i) = G_{\Delta t}((\mathbb{I} - \mathbb{P}^k)\mathbf{u}, t_{i+1}, t_i) + F_{\delta t}(\mathbb{P}^k\mathbf{u}, t_{i+1}, t_i). \tag{13}$$

For a linear problem, $F_{\delta t}(\mathbb{P}^k\mathbf{u}, t_{i+1}, t_i)$ can be computed efficiently as

$$F_{\delta t}(\mathbb{P}^k\mathbf{u}, t_{i+1}, t_i) = F_{\delta t}(\sum_{j=1}^{N_c k} C_j\mathbf{s}_j, t_{i+1}, t_i) = \sum_{j=1}^{N_c k} C_j F_{\delta t}(\mathbf{s}_j, t_{i+1}, t_i), \tag{14}$$

where $F_{\delta t}(\mathbf{s}_j, t_{i+1}, t_i)$ are computed and stored once the $\mathbf{s}_j$'s are available. Since this approach essentially produces an approximation to the fine solver, the new coarse solver is expected to be more accurate than the old coarse solver. It was shown in [11] that as the dimension of $\mathbf{S}^k$ increases, $\mathbb{P}^k \to \mathbb{I}$ and $K_{\Delta t} \to F_{\delta t}$, thus achieving convergence. The algorithm outline is presented in Algorithm 2.

To demonstrate the performance of the Krylov subspace parareal method, we use it to solve the linear advection equation, (10). In Figure 4 (left) we show the L_∞-error at $T = 10$ against the number of iterations. It is clear that the Krylov subspace parareal method stabilizes the parareal solver for this problem.

Two observations are worth making. First, the Krylov subspace parareal method needs to store all the values of $\mathbf{S}^k$ and $F(\mathbf{S}^k)$. As k increases, this induces a memory requirement scaling $O(kN_cN)$

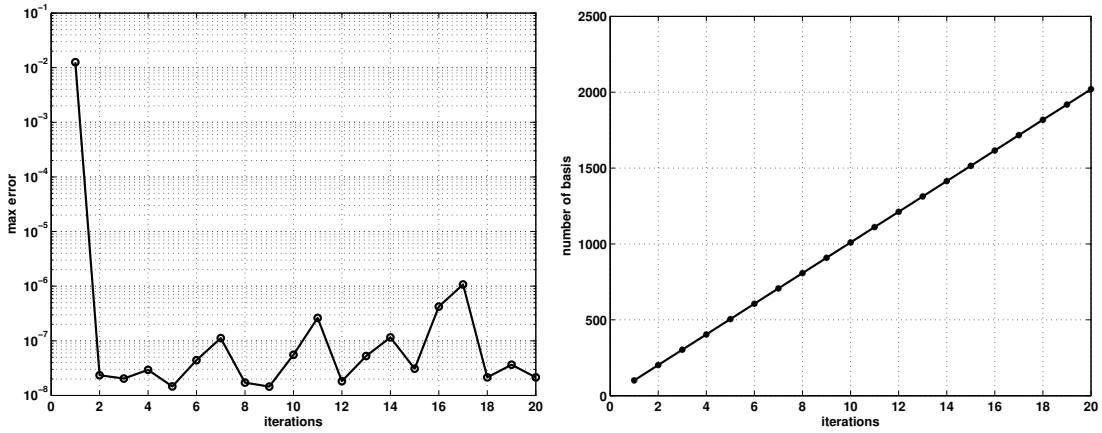

Fig. 4: The L_∞-error at $T = 10$ against the number of iterations (left), and the number of bases (right) for solving the 1-D linear advection equation using the Krylov subspace parareal method

and may be become a bottleneck demanding, as illustrated in Figure 4 (right). Furthermore, the efficiency of the coarse solver depends critically on the assumption of linearity of the operator and it is not clear how to extend this framework to nonlinear problems. These constraints appear to limit the practicality of the method.

2.3 The reduced basis parareal method

Let us first recall a few properties of reduced basis methods that will subsequently serve as key elements of the proposed reduced basis parareal method.

2.3.1 Reduced basis methods

We are generally interested in solving the nonlinear ODE (1). As a system, the dimensionality of the problem can be very large, e.g., if the problem originates from a method-of-lines discretization of a nonlinear PDE, so to achieve a high accuracy, requiring a high number of degrees of freedom, N, and it is tempting to seek to identify an approximate model to enhance the computational efficiency without significantly impacting the accuracy.

A general representation of a reduced model in matrix-form can be expressed as

$$\mathbf{u}(t) \approx \mathbf{V}_r \tilde{\mathbf{u}}(t), \tag{15}$$

Initialization:
$\mathbf{u}_0^0 = \mathbf{u}_0$;
for $i \leftarrow 0$ **to** N_c **do**
  $\mathbf{u}_{i+1}^0 = G_{\Delta t}(\mathbf{u}_i^0, t_{i+1}, t_i)$
end
Iterations:
$k = 0$;
for $k \leftarrow 0$ **to** N_{it} **do**
  Parallel predictor step:
  for $i \leftarrow 0$ **to** N_c **do**
    $\mathbf{u}_{f\,i+1}^k = F_{\delta t}(\mathbf{u}_i^k, t_{i+1}, t_i)$
  end
  Constructing reduced basis:
  Update $\mathbf{S}^{k-1}$ to $\mathbf{S}^k$ based on $\mathbf{u}_{i-1}^k, \mathbf{u}_{\mathbf{f}i}^k$
  Marching the basis:
  for $i \leftarrow 1$ **to** N_r **do**
    $\mathbf{S}_{f_i} = F_{\delta t}(\mathbf{s}_i, 0, \Delta t)$;
  end
  Sequential correction step:
  for $i \leftarrow 0$ **to** N_c **do**
    $\mathbf{u}_{i+1}^{k+1} = K_{\Delta t}(\mathbf{u}_i^{k+1}, t_{i+1}, t_i) - \mathbf{u}_{f\,i+1}^k + K_{\Delta t}(\mathbf{u}_i^k, t_{i+1}, t_i)$
  end
end

Algorithm 2: The Krylov subspace parareal method

where the r columns of the matrix $\mathbf{V}_r$ represent a linear space - the reduced basis - and $\tilde{\mathbf{u}}(t) \in \mathbb{R}^r$ are the coefficients of the reduced model. Projecting the ODE system (1) onto $\mathbf{V}_r$, we recover the reduced system:

$$\mathbf{V}_r^T \mathbf{V}_r \frac{d\tilde{\mathbf{u}}(t)}{dt} = \mathbf{V}_r^T \mathbf{A} \mathbf{V}_r \tilde{\mathbf{u}}(t) + \mathbf{V}_r^T \mathbf{N}(\mathbf{V}_r \tilde{\mathbf{u}}(t)). \tag{16}$$

Assuming that $\mathbf{V}_r$ is orthonormal, we recover

$$\frac{d\tilde{\mathbf{u}}(t)}{dt} = \mathbf{V}_r^T \mathbf{A} \mathbf{V}_r \tilde{\mathbf{u}}(t) + \mathbf{V}_r^T \mathbf{N}(\mathbf{V}_r \tilde{\mathbf{u}}(t)). \tag{17}$$

One is now left with specifying how to choose a good subspace, $\mathbf{V}_r$, to adequately represent the dynamic behavior of the solution and developing a strategy for how to recover the coefficients for the reduced model in an efficient manner. There are several ways to address this question, most often based on the construction of $\mathbf{V}_r$ through snapshots of the solution.

Proper orthogonal decomposition The proper orthogonal decomposition (POD) [6, 5] is perhaps the most widely used approach to generate a reduced basis from a collection of snapshots. In this case, we assume we have a collection of N_s snapshots

$$\mathbf{U} = [\mathbf{u}_1, ..., \mathbf{u}_{N_s}], \tag{18}$$

where each $\mathbf{u}_i$ is a vector of length N; this N can be large as it reflects the number of degrees of freedom in system. The POD basis, denoted by $\{\phi_i\}_1^r \in \mathbb{R}^N$, is chosen as the orthonormal vectors that solve the minimization problem:

$$\begin{aligned} &\min_{\phi_i \in \mathbb{R}^N} \sum_{j}^{N_s} \|\mathbf{u}_j - \sum_{i=1}^{r} (\mathbf{u}_j^T \phi_i)\phi_i\|_2^2, \\ &\text{subject to } \phi_i^T \phi_j = \delta_{ij} = \begin{cases} 1, \ i = j, \\ 0, \text{ otherwise.} \end{cases} \end{aligned} \tag{19}$$

The solution to this minimization problem is found through the singular value decomposition (SVD) of $\mathbf{U}$:

$$\mathbf{U} = \mathbf{V}\Sigma\mathbf{W}^T, \tag{20}$$

where $\mathbf{V} \in \mathbb{R}^{N \times r}$ and $\mathbf{W} \in \mathbb{R}^{N_s \times r}$ are the left and right singular vectors, respectively, and $\mathbf{V}$ is the sought after basis. The entries of the diagonal matrix Σ provides a measure of the relative energy of each of the orthogonal vectors in the basis.

Once the basis is available, we can increase the computational efficiency for solving (17) by pre-computing $\mathbf{V}_r^T \mathbf{A} \mathbf{V}_r$, which is of size $r \times r$. However, the computational complexity of the nonlinear term remains dependent on N and, hence, potentially costly.

Discrete Empirical Interpolation. To address this, [7] proposed an approach, originating in previous work on empirical interpolation methods [4] but limited the case of an existing discrete basis set, in which $\mathbf{N}(\mathbf{V_r}\tilde{\mathbf{u}}(t))$ is represented by $\tilde{\mathbf{N}}(t) \in \mathbb{R}^N$ which is subsequently approximated as

$$\mathbf{N}(\mathbf{V_r}\tilde{\mathbf{u}}(t)) \approx \tilde{\mathbf{N}}(t) \approx \mathbf{V_p}\mathbf{c}(t). \tag{21}$$

Here $\mathbf{V_p} = [\mathbf{v}_1, ..., \mathbf{v}_m]$ is an orthogonal POD basis set based on snapshots of $\mathbf{N}(t)$. To recover $\mathbf{c}(t)$, we seek a solution to an overdetermined system. However, rather than employing an expensive least square method, we extract m equations from the original set of snapshots. Denote

$$\mathbf{P} = [\mathbf{e}_{p_1}, ..., \mathbf{e}_{p_m}] \in \mathbb{R}^{N \times m}, \tag{22}$$

where $\mathbf{e}_{p_1} = [0, ..., 0, 1, 0, ..., 0]^T \in \mathbb{R}^N$ (1 only appears on the p_1-th position of the vector). If $\mathbf{P}^T\mathbf{V_p}$ is nonsingular, $\mathbf{c}(t)$ can be uniquely determined by

$$\mathbf{P}^T\tilde{\mathbf{N}}(t) = \mathbf{P}^T\mathbf{V_p}\mathbf{c}(t),$$

resulting in a final approximation of $\tilde{\mathbf{N}}(t)$ as

$$\tilde{\mathbf{N}}(t) \approx \mathbf{V_p}(\mathbf{P}^T\mathbf{V_p})^{-1}\mathbf{P}^T\tilde{\mathbf{N}}(t).$$

The interpolation index p_i is selected iteratively by minimizing the largest magnitude of the residual $\mathbf{r} = \mathbf{u}_k - \mathbf{V_{p,k}}\mathbf{c}$. The procedure, sometimes referred to as discrete empirical interpolation, is outlined in Algorithm 3. Notice that in contrast to the original parareal algorithm, in this approach, the coarse time integrator is used only once to initiate the scheme.

input : $\{\mathbf{v_k}\}_{k=1}^{m} \subset \mathbb{R}^N$ linearly independent POD bases of the nonlinear term
output: the interpolation operator $\mathbf{P}_m = [p_1, ..., p_m]$.

begin
  $\epsilon = \max|\mathbf{u}_1|, p_1 = argmax|\mathbf{u}_1|$;
  $\mathbf{P} \leftarrow \{p_1\}$; $\mathbf{V_{p,1}} \leftarrow \{\mathbf{v}_1\}$;
  for $k \leftarrow 2$ **to** M **do**
    Solve $\mathbf{P}^T\mathbf{v}_k = \mathbf{P}^T\mathbf{V_{p,k}}\mathbf{c}(t)$ to obtain $\mathbf{c}(t)$;
    Compute the residual; $\mathbf{r} = \mathbf{v}_k - \mathbf{V_{p,k}}\mathbf{c}$;
    $\epsilon = \max|\mathbf{r}|, p_k = argmax|\mathbf{r}|$;
    $\mathbf{V}_{p,k} \leftarrow \mathbf{V}_{p,k-1} \cup \{\mathbf{v}_k\}$;
    $\mathbf{P}_k \leftarrow \mathbf{P}_{k-1} \cup \{p_k\}$;
  end
end

Algorithm 3: Empirical interpolation with a given discrete basis set

With the above approximation, we can now express the reduced system as

$$\frac{d\tilde{\mathbf{u}}(t)}{dt} = \mathbf{V}_r^T\mathbf{A}\mathbf{V}_r\tilde{\mathbf{u}}(t) + \mathbf{V}_r^T\mathbf{V_p}(\mathbf{P}^T\mathbf{V_p})^{-1}\mathbf{N}(\mathbf{P}^T\mathbf{V}_r\tilde{\mathbf{u}}(t)). \tag{23}$$

Full Empirical Interpolation Pursuing the above approach further, one is left wondering if we can use a basis other than the computational expensive POD basis, and whether we can choose the interpolation position based on other guidelines. Addressing these questions leads us to propose a full empirical interpolation method.

It is well-known that the original empirical interpolation method is commonly used to separate the dependence of parameters and spatial variables [4], and that the method choosing 'optimal' interpolation points in a certain sense. Wee propose to consider time as a parameter, and use the empirical interpolation to construct the reduced bases $\mathbf{V_{E,k}}$ of $\mathbf{u}$ and the reduced bases $\mathbf{V_{pE,k}}$ of the nonlinear term, i.e.,

$$\mathbf{u}(t) \approx \mathbf{V_{E,k}}\mathbf{c}(t), \quad \tilde{\mathbf{N}}(t) \approx \mathbf{V_{pE,k}}\mathbf{c}(t). \tag{24}$$

The resulting reduced model can be written as

$$\frac{d\tilde{\mathbf{u}}(t)}{dt} = \mathbf{V_{E,k}}^T\mathbf{A}\mathbf{V_{E,k}}\tilde{\mathbf{u}}(t) + \mathbf{V_{E,k}}^T\mathbf{V_{pE,k}}(\mathbf{P}^T\mathbf{V_{pE,k}})^{-1}\mathbf{N}(\mathbf{P}^T\mathbf{V_{E,k}}\tilde{\mathbf{u}}(t)). \tag{25}$$

The essential difference between the models based on discrete empirical interpolation and the full empirical interpolation approach is found in the way in which one constructs the reduced basis set. In the former case, the importance of the basis elements is guided by the SVD and the relative size of the singular values, resulting in a potentially substantial cost. The latter case is based on the interpolation error and the basis in constructed in a full greedy fashion. A detailed comparative study of the performance between the two approaches is ongoing and will be presented in a forthcoming paper.

2.3.2 The reduced basis parareal method

Let us now introduce the new reduced basis parareal method. Our first observation is that the first term in (13) can be dropped under the assumption that the projection error vanishes asymptotically. Hence, for linear problems, we can replace $K_{\Delta t}$ by $\hat{K}_{\Delta t}$ as

$$\hat{K}_{\Delta t}(\mathbf{u}, t_{i+1}, t_i) = F_{\delta t}(\mathbb{P}^k \mathbf{u}, t_{i+1}, t_i) = \sum_{j=1}^{N_c k} C_j F_{\delta t}(\mathbf{s}_j, t_{i+1}, t_i). \tag{26}$$

This is essentially an approximation to the fine time integrator with an admissible truncation error. Keeping in mind that $F_{\delta t}$ is an expensive operation, we seek to reduce the dimension of $\mathbf{S}^k$ to achieve a better efficiency. If the solution to the ODE is sufficiently regular, it is reasonable to seek an r-dimensional subspace, $\mathbf{S}_r^k$ (the reduced basis space), of the original space $\mathbf{S}^k$. Now redefine $\mathbb{P}_r^k$ to be the orthogonal projection from $\mathbf{u}$ onto $\mathbf{S}_r^k$. Then (26) becomes (27), which is essentially an approximation to the fine time integrator using the reduced model, i.e.,

$$\hat{K}_{\Delta t}(\mathbf{u}, t_{i+1}, t_i) = F_{\delta t}(\mathbb{P}_r^k \mathbf{u}, t_{i+1}, t_i) = \sum_{j=1}^{r} C_j F_{\delta t}(\mathbf{s}_j, t_{i+1}, t_i). \tag{27}$$

Consequently, our reduced basis parareal method for linear problems is as follows:

$$\mathbf{u}_{n+1}^{k+1} = F_{\delta t}(\mathbb{P}_r^k \mathbf{u}_n^{k+1}, t_{n+1}, t_n) + F_{\delta t}(\mathbf{u}_n^k, t_{n+1}, t_n) - F_{\delta t}(\mathbb{P}_r^k \mathbf{u}_n^k, t_{n+1}, t_n), \quad 0 \le k \le N_c - 1. \tag{28}$$

Depending on the construction of the reduced model, we refer to it as the POD parareal method or the EIM parareal method.

Algorithm 4 describes the basic steps of the reduced basis parareal method for linear problems. It follows a procedure similar to Algorithm 2, but requires less memory for storing the bases. Notice that for linear problems, the coarse solver is needed only for initializing the algorithm. After this first step, the fine solver produces all the information needed for the reduced model, and the algorithm no longer depends on the coarse solver.

For nonlinear problems, the relationship

$$F_{\delta t}(\mathbb{P}_r^k \mathbf{u}, t_{i+1}, t_i) = \sum_{j=1}^{r} C_j F_{\delta t}(\mathbf{s}_j, t_{i+1}, t_i) \tag{29}$$

does not generally hold, even if $\mathbb{P}^k \mathbf{u} \to \mathbf{u}$. Therefore, the Krylov subspace parareal method is not applicable. Fortunately, the knowledge of the development of reduced models using empirical interpolation offers insight into dealing with nonlinear problems, as mentioned in Section 2.3.1. We construct the coarse time integrator as follows:

$$\hat{K}_{\Delta t}(\mathbf{u}, t_{i+1}, t_i) = F_{\delta t}^r(\mathbb{P}_r^k \mathbf{u}, t_{i+1}, t_i), \tag{30}$$

where $F_{\delta t}^r$ is the reduced model constructed by POD or EIM as we described in the previous section. Consequently, our reduced basis parareal method for nonlinear problems becomes

$$\mathbf{u}_{n+1}^{k+1} = F_{\delta t}^{r}(\mathbb{P}_r^k \mathbf{u}_n^{k+1}, t_{n+1}, t_n) + F_{\delta t}(\mathbf{u}_n^k, t_{n+1}, t_n) - F_{\delta t}^{r}(\mathbb{P}_r^k \mathbf{u}_n^k, t_{n+1}, t_n), \quad 0 \le k \le N_c - 1. \tag{31}$$

As long as there exists a suitable reduced model for the problem, we can evaluate $\hat{K}_{\Delta t}$ efficiently while maintaining an accuracy commensurate with the fine solver. The reduced basis parareal method for nonlinear problems is outlined in Algorithm 5.

Initialization:
$\mathbf{u}_0^0 = \mathbf{u}_0$;
for $i \leftarrow 0$ **to** N_c **do**
  $\mathbf{u}_{i+1}^0 = G_{\Delta t}(\mathbf{u}_i^0, t_{i+1}, t_i)$
end
Iterations:
$k = 0$;
for $k \leftarrow 0$ **to** N_{it} **do**
  Parallel predictor step:
  for $i \leftarrow 0$ **to** N_c **do**
    $\mathbf{u}_{f\,i+1}^k = F_{\delta t}(\mathbf{u}_i^k, t_{i+1}, t_i)$
  end
  Constructing reduced basis by POD or EIM:
  $U^k = \{\mathbf{u}_{f\,i+1}^k, i = 0, ..., N_c, j = 0, ..., k\}$
  $\mathbf{S} = \mathrm{POD}(U^k)$ or $\mathbf{S} = \mathrm{EIM}(U^k)$ where $\mathbf{S} = \{\mathbf{s}_i, i = 1, ..., r\}$
  Marching the basis:
  for $i \leftarrow 1$ **to** r **do**
    $\mathbf{S}_{f\,i} = F_{\delta t}(\mathbf{s}_i, 0, \Delta t)$;
  end
  Sequential correction step:
  for $i \leftarrow 0$ **to** N_c **do**
    $\mathbb{P}^k \mathbf{u}_i^k = \sum_{j=1}^{r} C_j \mathbf{s}_j \leftarrow C_j$
    $\hat{K}_{\Delta t}(\mathbf{u}_i^{k+1}, t_{i+1}, t_i) = \sum_{j=1}^{N_r} C_j \mathbf{S}_{f\,j}$
    $\mathbf{u}_{i+1}^{k+1} = \hat{K}_{\Delta t}(\mathbf{u}_i^{k+1}, t_{i+1}, t_i) - \mathbf{u}_{f\,i+1}^k + \hat{K}_{\Delta t}(\mathbf{u}_i^k, t_{i+1}, t_i)$
  end
end

Algorithm 4: The reduced parareal method for a linear problem

Initialization:
$\mathbf{u}_0^0 = \mathbf{u}_0$;
for $i \leftarrow 0$ **to** N_c **do**
  $\mathbf{u}_{i+1}^0 = G_{\Delta t}(\mathbf{u}_i^0, t_{i+1}, t_i)$
end
Iterations:
$k = 0$;
for $k \leftarrow 0$ **to** N_{it} **do**
  Parallel predictor step:
  for $i \leftarrow 0$ **to** N_c **do**
    $\mathbf{u}_{f_{i+1}^k} = F_{\delta t}(\mathbf{u}_i^k, t_{i+1}, t_i)$
  end
  Constructing reduced basis:
  $U^k = \{\mathbf{u}_{f_{i+1}^k}, i = 0,...,N_c, j = 0,...,k\}$
  $\mathbf{S}$ = POD-DEIM(U^k) or $\mathbf{S}$ = EIM(U^k) where $\mathbf{S} = \{\mathbf{s}_i, i = 1,...,r\}$
  Sequential correction step:
  for $i \leftarrow 0$ **to** N_c **do**
    $\hat{K}_{\Delta t}(\mathbf{u}_i^k, t_{i+1}, t_i) = F_{\delta t}^r(\mathbb{P}_r^k \mathbf{u}_i^k, t_{i+1}, t_i)$
    $\hat{K}_{\Delta t}(\mathbf{u}_i^{k+1}, t_{i+1}, t_i) = F_{\delta t}^r(\mathbb{P}_r^k \mathbf{u}_i^{k+1}, t_{i+1}, t_i)$
    $\mathbf{u}_{i+1}^{k+1} = \hat{K}_{\Delta t}(\mathbf{u}_i^{k+1}, t_{i+1}, t_i) - \mathbf{u}_{f_{i+1}^k} + \hat{K}_{\Delta t}(\mathbf{u}_i^k, t_{i+1}, t_i)$
  end
end

Algorithm 5: The reduced parareal method for a nonlinear problem

3 Analysis of the reduced basis parareal method

In the following we provide some analysis of the reduced basis parareal method to understand its stability, convergence and overall computational complexity. Throughout, we assume that there exists a reduced space for the continuous problem.

3.1 Stability analysis

We first consider the linear case. Define the projection error:

$$g_j^k = \|(\mathbb{I} - \mathbb{P}_r^k)\mathbf{u}_j^k\|_{L_2(0,T)}, \tag{32}$$

where r is the dimension of the reduced space. We assume a projection error

$$g_j^k \leq \varepsilon, \quad \forall j,k, \tag{33}$$

and define:

$$C_{p,r} = \frac{\varepsilon}{\Delta T}, \quad \forall j,k. \tag{34}$$

It is reasonable to assume that the fine propagator is L_2 stable, i.e., there exists a nonnegative constant C_F independent of the discretization parameters, such that,

$$\|F_{\delta t}(\mathbf{v}, t_{i+1}, t_i)\|_{L_2(0,T)} \leq (1 + C_F \Delta T)\|\mathbf{v}\|_{L_2(0,T)}, \quad \forall \mathbf{v} \in L_2(0,T). \tag{35}$$

Theorem 1 (Stability for the linear case) *Under the assumption of* (33) *and* (35)*, the reduced basis parareal method is stable for* (1) *with* $\mathbf{N} \equiv \mathbf{0}$*, i.e., for each* i *and* k*,*

$$\|\mathbf{u}_{i+1}^{k+1}\|_{L_2(0,T)} \leq C_L e^{C_F(i+1)\Delta T}, \tag{36}$$

where C_L *is a constant depending only on* $C_{p,r}$*,* C_F*, and* $\mathbf{u}_0$*.*

Proof. Using the triangle inequality, linearity of the operator, and assumption (35), we obtain

$$\|\mathbf{u}_{i+1}^{k+1}\|_{L_2(0,T)} \leq \|F_{\delta t}(\mathbb{P}_r^k \mathbf{u}_i^{k+1}, t_{i+1}, t_i)\|_{L_2(0,T)} + \|F_{\delta t}(\mathbf{u}_i^k, t_{i+1}, t_i) - F_{\delta t}(\mathbb{P}_r^k \mathbf{u}_i^k, t_{i+1}, t_i)\|_{L_2(0,T)} \tag{37}$$

$$\leq (1 + C_F \Delta T)\|\mathbf{u}_i^{k+1}\|_{L_2(0,T)} + (1 + C_F \Delta T)\|(\mathbb{I} - \mathbb{P}_r^k)\mathbf{u}_i^k\|_{L_2(0,T)}. \tag{38}$$

Then, by the discrete Gronwall's lemma [9] and (33), we recover

$$\|\mathbf{u}_{i+1}^{k+1}\|_{L_2(0,T)} \leq (1 + C_F \Delta T)^{i+1}(\|\mathbf{u}_0^{k+1}\|_{L_2(0,T)} + \Delta T \sum_{j=0}^{i} (1 + C_F \Delta T)^{-j} C_{p,r}) \tag{39}$$

$$= (1 + C_F \Delta T)^{i+1}\|\mathbf{u}_0^{k+1}\|_{L_2(0,T)} + \frac{1}{C_F}((1 + C_F \Delta T)^{i+1} - 1)C_{p,r} \tag{40}$$

$$\leq e^{C_F(i+1)\Delta T}\|\mathbf{u}_0\|_{L_2(0,T)} + \frac{1}{C_F}(e^{C_F(i+1)\Delta T} - 1)C_{p,r}. \tag{41}$$

This completes the proof.

Note that if there exists an small integer M (indicating a good reduced approximation space) such that,

$$\lim_{r \to M} C_{p,r} = 0, \tag{42}$$

then we recover the same stability property as that of the fine solver:

$$\|\mathbf{u}_{i+1}^{k+1}\|_{L_2(0,T)} \leq e^{C_F(i+1)\Delta T}\|\mathbf{u}_0\|_{L_2(0,T)}.$$

For the nonlinear case, we further assume that there exists a nonnegative constant C_r, independent of the discretization parameters, such that,

$$\|F_{\delta t}(\mathbf{v}, t_{i+1}, t_i) - F_{\delta t}^r(\mathbb{P}_r^k \mathbf{v}, t_{i+1}, t_i)\|_{L_2(0,T)} \leq (1 + C_r \Delta T) q_i^k, \quad \forall \mathbf{v} \in L_2(0,T), \tag{43}$$

where q_i^k is the L_2-difference between the fine propagator and the reduced model using the same initial condition $\mathbf{v}$ at t_i. We assume that

$$q_j^k \leq \varepsilon, \quad \forall j,k. \tag{44}$$

Theorem 2 (Stability for the nonlinear case) *Under assumptions* (35), (43) *and* (44), *the reduced basis parareal method is stable for* (1) *in the sense that for each* i *and* k

$$\|\mathbf{u}_{i+1}^{k+1}\|_{L_2(0,T)} \leq C_N e^{C_\star (i+1)\Delta T}, \tag{45}$$

where $C_\star = \max\{C_F, C_r\}$ *and* C_N *is a constant depending only on* $C_{p,r}$, C_F, C_r, *and* $\mathbf{u}_0$.

Proof. Using the triangle inequality and assumptions (35) and (43), we have

$$\|\mathbf{u}_{i+1}^{k+1}\|_{L_2(0,T)} \leq \|F_{\delta t}^r(\mathbb{P}_r^k \mathbf{u}_i^{k+1}, t_{i+1}, t_i)\|_{L_2(0,T)} + \|F_{\delta t}(\mathbf{u}_i^k, t_{i+1}, t_i) - F_{\delta t}^r(\mathbb{P}_r^k \mathbf{u}_i^k, t_{i+1}, t_i)\|_{L_2(0,T)} \tag{46}$$

$$\leq (1 + C_F \Delta T)\|\mathbf{u}_i^{k+1}\|_{L_2(0,T)} + (1 + C_r \Delta T) q_i^k. \tag{47}$$

Next, by the discrete Gronwall's lemma and (44), we derive

$$\|\mathbf{u}_{i+1}^{k+1}\|_{L_2(0,T)} \leq (1 + C_F \Delta T)^{i+1} (\|\mathbf{u}_0^{k+1}\|_{L_2(0,T)} + \Delta T \sum_{j=0}^{i} (1 + C_r \Delta T)^{-j} C_{p,r}) \tag{48}$$

$$= (1 + C_F \Delta T)^{i+1} \|\mathbf{u}_0^{k+1}\|_{L_2(0,T)} + \frac{1}{C_r}((1 + C_r \Delta T)^{i+1} - 1) C_{p,r} \tag{49}$$

$$\leq e^{C_F (i+1)\Delta T} \|\mathbf{u}_0\|_{L_2(0,T)} + \frac{1}{C_r}(e^{C_r (i+1)\Delta T} - 1) C_{p,r}. \tag{50}$$

This completes the proof.

3.2 Convergence analysis

To show the convergence for the linear case, we first assume that there exists a nonnegative constant C_F, such that,

$$\|F_{\delta t}(\mathbf{x}, t_{i+1}, t_i) - F_{\delta t}(\mathbf{y}, t_{i+1}, t_i)\|_{L_2(0,T)} \leq (1 + C_F \Delta T)\|\mathbf{x} - \mathbf{y}\|_{L_2(0,T)}, \quad \forall t_i > 0. \tag{51}$$

We define

$$w_j^k = \|(\mathbb{I} - \mathbb{P}_r^k)\mathbf{u}_j\|_{L_2(0,T)}, \tag{52}$$

and assume that

$$w_j^k \leq \varepsilon, \quad \forall j,k. \tag{53}$$

Theorem 3 (Convergence for the linear case) *Under assumption* (33), (42), (51), (53) *and* $\mathbf{N} \equiv \mathbf{0}$ *in* (1), *the reduced basis parareal solution converges to* $\mathbf{u}_{i+1}$ *for each* i.

Proof. Using the reduced basis parareal formula and the linearity of the operator, we obtain

$$\mathbf{u}_{i+1}^{k+1} - \mathbf{u}_{i+1} = F_{\delta t}(\mathbb{P}_r^k \mathbf{u}_i^{k+1}, t_{i+1}, t_i) + F_{\delta t}(\mathbf{u}_i^k, t_{i+1}, t_i) - F_{\delta t}(\mathbb{P}_r^k \mathbf{u}_i^k, t_{i+1}, t_i) - F_{\delta t}(\mathbf{u}_i, t_{i+1}, t_i) \tag{54}$$

$$= F_{\delta t}(\mathbb{P}_r^k \mathbf{u}_i^{k+1}, t_{i+1}, t_i) - F_{\delta t}(\mathbb{P}_r^k \mathbf{u}_i, t_{i+1}, t_i) \tag{55}$$

$$+ F_{\delta t}(\mathbf{u}_i^k, t_{i+1}, t_i) - F_{\delta t}(\mathbb{P}_r^k \mathbf{u}_i^k, t_{i+1}, t_i) \tag{56}$$

$$+ F_{\delta t}(\mathbb{P}_r^k \mathbf{u}_i, t_{i+1}, t_i) - F_{\delta t}(\mathbf{u}_i, t_{i+1}, t_i). \tag{57}$$

By the triangular inequality and assumption (51), we recover

$$\|\mathbf{u}_{i+1}^{k+1} - \mathbf{u}_{i+1}\|_{L_2(0,T)} \leq (1 + C_F \Delta T) \|\mathbf{u}_i^{k+1} - \mathbf{u}_i\|_{L_2(0,T)} \tag{58}$$

$$+ (1 + C_F \Delta T) \|(\mathbb{I} - \mathbb{P}_r^k) \mathbf{u}_i^k\|_{L_2(0,T)} \tag{59}$$

$$+ (1 + C_F \Delta T) \|(\mathbb{I} - \mathbb{P}_r^k) \mathbf{u}_i\|_{L_2(0,T)}. \tag{60}$$

Finally by the discrete Gronwall's lemma, (33) and (53), we obtain

$$\|\mathbf{u}_{i+1}^{k+1} - \mathbf{u}_{i+1}\|_{L_2(0,T)} \leq (1 + C_F \Delta T)^{i+1} (\|\mathbf{u}_0^{k+1} - \mathbf{u}_0\|_{L_2(0,T)} \tag{61}$$

$$+ \Delta T \sum_{j=0}^{i} (1 + C_F \Delta T)^{-j} C_{p,r} + \Delta T \sum_{j=0}^{i} (1 + C_F \Delta T)^{-j} C_{p,r}) \tag{62}$$

$$\leq 2 \Delta T \sum_{j=0}^{i} (1 + C_F \Delta T)^{-j} C_{p,r} \tag{63}$$

$$\leq \frac{2}{C_F} ((1 + C_F \Delta T)^{i+1} - 1) C_{p,r} \tag{64}$$

$$\leq \frac{2}{C_F} (e^{C_F (i+1) \Delta T} - 1) C_{p,r}, \tag{65}$$

which approaches zero as r increases. This completes the proof.

For the nonlinear case, we must also assume that there exists a nonnegative constant C_r, such that,

$$\begin{aligned} \|F_{\delta t}(\mathbf{u}_i^k, t_{i+1}, t_i) - F_{\delta t}^r(\mathbb{P}_r^k \mathbf{u}_i^k, t_{i+1}, t_i)\|_{L_2(0,T)} &\leq (1 + C_r \Delta T) q_i^k, \\ \|F_{\delta t}(\mathbf{u}_i, t_{i+1}, t_i) - F_{\delta t}^r(\mathbb{P}_r^k \mathbf{u}_i, t_{i+1}, t_i)\|_{L_2(0,T)} &\leq (1 + C_r \Delta T) p_i^k, \end{aligned} \tag{66}$$

where q_i^k and p_i^k are the L_2-difference between the fine operator and the reduced solver using the same initial condition $\mathbf{u}_i^k$ and $\mathbf{u}_i$. We assume that

$$p_j^k \leq \varepsilon, \quad \forall j, k. \tag{67}$$

Theorem 4 (Convergence of the nonlinear case) *Under assumptions* (42), (43), (44), (66) *and* (67), *the reduced basis parareal solution of* (1) *converges to* $\mathbf{u}_{i+1}$ *for each* i.

Proof. Using the reduced basis parareal formula, we obtain

$$\mathbf{u}_{i+1}^{k+1} - \mathbf{u}_{i+1} = F_{\delta t}^r(\mathbb{P}_r^k \mathbf{u}_i^{k+1}, t_{i+1}, t_i) + F_{\delta t}(\mathbf{u}_i^k, t_{i+1}, t_i) - F_{\delta t}^r(\mathbb{P}_r^k \mathbf{u}_i^k, t_{i+1}, t_i) - F_{\delta t}(\mathbf{u}_i, t_{i+1}, t_i) \tag{68}$$

$$= F_{\delta t}^r(\mathbb{P}_r^k \mathbf{u}_i^{k+1}, t_{i+1}, t_i) - F_{\delta t}^r(\mathbb{P}_r^k \mathbf{u}_i, t_{i+1}, t_i) \tag{69}$$

$$+ F_{\delta t}(\mathbf{u}_i^k, t_{i+1}, t_i) - F_{\delta t}^r(\mathbb{P}_r^k \mathbf{u}_i^k, t_{i+1}, t_i) \tag{70}$$

$$+ F_{\delta t}^r(\mathbb{P}_r^k \mathbf{u}_i, t_{i+1}, t_i) - F_{\delta t}(\mathbf{u}_i, t_{i+1}, t_i). \tag{71}$$

By the triangular inequality and assumptions (66) and (43), we have

$$\|\mathbf{u}_{i+1}^{k+1} - \mathbf{u}_{i+1}\|_{L_2(0,T)} \leq (1 + C_F \Delta T)\|\mathbf{u}_i^{k+1} - \mathbf{u}_i\|_{L_2(0,T)} + (1 + C_r \Delta T) q_i^k + (1 + C_r \Delta T) p_i^k. \tag{72}$$

Then, by the discrete Gronwall's lemma, (44) and (67) we recover

$$\|\mathbf{u}_{i+1}^{k+1} - \mathbf{u}_{i+1}\|_{L_2(0,T)} \leq \frac{2}{C_r}((1 + C_r \Delta T)^{i+1} - 1) C_{p,r} \tag{73}$$

$$\leq \frac{2}{C_r}(e^{C_r (i+1)\Delta T} - 1) C_{p,r}, \tag{74}$$

which approaches zero as r increases under assumption (42).

Remark 3.1 *For the above analysis it is worth emphasizing two points:*

- *The accuracy of the new parareal algorithm is $O(\varepsilon)$, since $C_{p,r}$ depends on ϵ as a measure of the quality of the reduced model. We shall confirm this point by the numerical tests in Section 4.*
- *Theorem 3 and 4 indicate that if there exists a good reduced approximation space for the problem, the new parareal algorithm converges in one iteration.*

3.3 Complexity analysis

Let us finally discuss the computational complexity of the reduced basis parareal method. Recall that the dimension of the reduced space is r and that of the fine solution is N. This is assumed to be the same for the coarse and fine solvers although this may not be a requirement in general. The compression ratio is $R = r/N$. Following the notation of [21]: $\tau_{QR}(k), \tau_{RB}(k)$ (representing $\tau_{SVD}(k)$, $\tau_{EIM}(k)$, and $\tau_{DEIM}(k)$ in different scenarios) reflect computing times required by the corresponding operations at the k-th iteration. τ_c and τ_f is the time required by the coarse and fine solvers, respectively. $N_t = N_c N_f$ is the total number of time steps in one iteration with N_c being the number of the coarse time intervals and N_f the number of fine time steps on each coarse time interval. N_p is the number of processors.

In [21], the speedup is estimated as

$$S(N_p) \approx \frac{N_t \tau_f}{N_c \tau_c + N_{it}(N_c \tau_c + N_t/N_p \tau_f) + N_{it} \tau_{QR}(it)} \tag{75}$$

$$= \frac{1}{(1+N_{it})(\frac{N_c}{N_t}\frac{\tau_c}{\tau_f}) + \frac{N_{it}\tau_{QR}(N_{it})}{N_t \tau_f} + \frac{N_{it}}{N_p}}. \tag{76}$$

In the reduced basis parareal method, $\tau_c = R^2 \tau_f$, since the complexity of the computation of the right hand side of system is $\mathbf{O}(r^2)$. In addition, τ_{QR} becomes τ_{SVD} or τ_{EIM}. With this in mind, the speedup can be estimated as

$$S(N_p) = \frac{1}{(1+N_{it})(\frac{N_c}{N_t}R^2) + \frac{N_{it}\tau_{RB}(N_{it})}{N_t \tau_f} + \frac{N_{it}}{N_p}}. \tag{77}$$

Next, we examine the first two terms in the denominators of (76) and (77).

- In the first term, τ_c/τ_f takes the role of R^2. Hence, we can achieve a comparable performance, if $R \approx \sqrt{\tau_c/\tau_f}$, i.e, if the underlying PDE solution can be represented by a reduced basis set of size $\mathbf{O}(\sqrt{\tau_c/\tau_f}N)$. Suppose that $\sqrt{\tau_c/\tau_f} = \sqrt{1/20} \approx 0.23$. This requires that $R < 1/4$, which is a reasonable compression ratio for many problems. In addition, it is possible to use a reduced basis approximation to achieve a better performance for cases where CFL conditions lead to restrictions for the coarse solver.
- For the second term, $\tau_{SVD} \approx \tau_{QR} \approx \mathbf{O}(NN_{it}^2 N_c^2)$, while $\tau_{EIM} \approx \mathbf{O}(r^3/2N_{it}N_c + rNN_{it}N_c)$. Therefore, $\tau_{SVD}/\tau_{EIM} \approx \mathbf{O}(2N_{it}N_c/Rr^2)$. As N_c increases, τ_{EIM} becomes smaller. In addition, EIM has a very good parallel efficiency and requires less memory during the computation.

Also note that N_{it} would typically be different for the reduced basis parareal method and the original parareal method. If a reduced space exists, the modified algorithm usually converges within a few iterations, hence accelerating the overall convergence significantly.

4 Numerical results

In the following, we demonstrate the feasibility and efficiency of the reduced basis parareal method for both linear and nonlinear problems. We use the solution obtained from the fine time integrator as the exact solution.

4.1 The linear advection equation

We begin by considering the performance of the reduced basis parareal method and illustrate that it is stable for the 1-D linear advection equation (10). The spatial and temporal discretizations are the same as in Section 2 and the parameters as in (11) are used.

In Figure 5 (left), we show the L_∞-error at $T = 10$ against the number of iterations for the original parareal method, the POD parareal method, and the EIM parareal method. The accuracy of the fine time integrator at $T = 10$ is 4×10^{-13}. The original parareal method is clearly unstable, while the other two remain stable. In Figure 5 (right), we show the number of bases used to satisfy the tolerance ϵ in the POD parareal method and the EIM parareal method. Here ϵ in the POD context is defined as the relative energy in the truncated mode and in the EIM context it is the interpolation error. In both cases, the tolerance in the basis selection using POD or EIM is set to 10^{-13}. We note that the EIM parareal method achieves higher accuracy but requires more memory to store the bases. This suggests that one can explore a tradeoff between accuracy and efficiency in a particular application.

Remark 4.1 *It should be noted that if only snapshots from the previous iteration is used in the EIM basis construction, the scheme becomes unstable. However, when including all snapshots collected up to the previous iteration level, the stability is restored.*

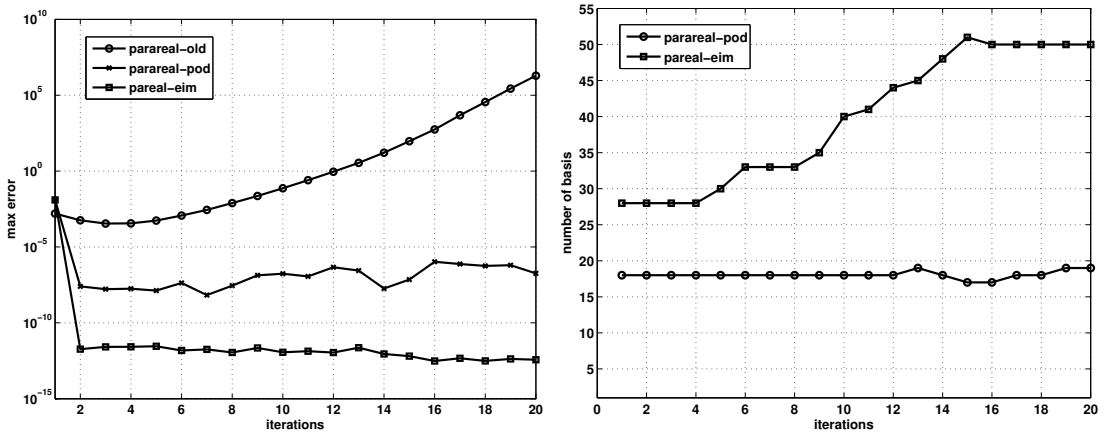

Fig. 5: Results obtained using the original parareal method, the POD parareal method, and the EIM parareal method for the 1-D advection equation. On the left we show the L_∞-error at $T = 10$ against the number of iterations while the right shows the number of bases used for satisfying the tolerance of $\epsilon(10^{-13})$ in the POD and EIM parareal methods across the iterations.

Figure 6 (upper left) shows the convergence behavior of the EIM parareal algorithm with different tolerances ($\epsilon = 10^{-k}, k = 1, 3, 5, 7, 9, 11$). The convergence stagnates at a certain level and instability may set in after further iterations. There are two reasons for this: 1) as ϵ becomes small, the reduced bases may become linear dependent, leading to a bad condition number of the related matrices that may impact stability; 2) the newly evolved reduced bases $\mathbf{S}_{f_i}$ for the fine solution may not be within $\mathbf{S}$ anymore. To resolve this problem, we first perform the reorthogonalization of the reduced bases to obtain a new space $\tilde{\mathbf{S}}$ and then project the newly evolved solution $\hat{K}_{\Delta t}(\mathbf{u}_i^{k+1}, t_{i+1}, t_i)$ back to $\tilde{\mathbf{S}}$. In Figure 6 (lower left) we show the convergence results following this approach. Most importantly, it is clear that stability is restored. Furthermore, the dependence of the final accuracy on ϵ is clear. These results are consistent with Theorem 3, stating that the parareal solution converges to the serial solution integrated by the fine solver as long as the subspace $\mathbf{S}$ saturates in terms of accuracy.

In practice, we can choose an ϵ such that the accuracy of the parareal solution and the serial fine solution are comparable.

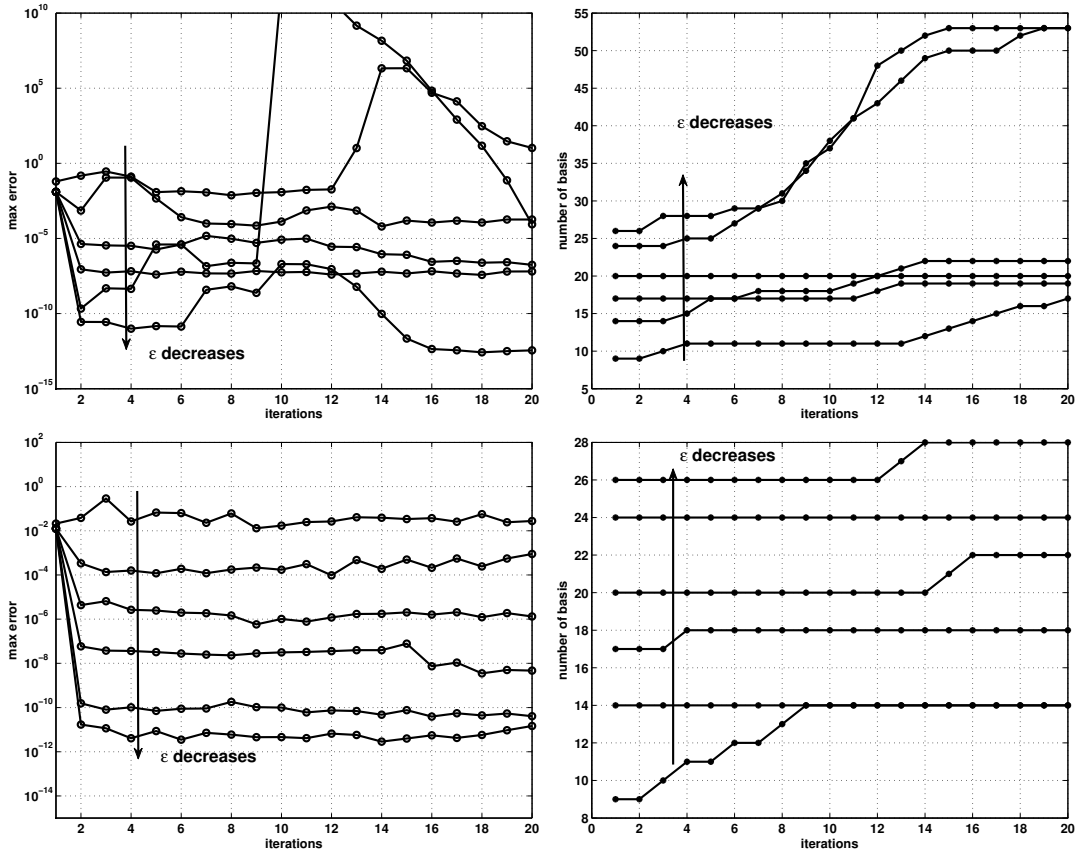

Fig. 6: The performance of the EIM parareal method for the 1-D advection equation against the tolerance used in the design of the reduced basis. On the upper left we show the L_∞-error at $T = 10$ against the number of iterations as the tolerance ϵ decreases and on the upper right the number of bases used for satisfying the tolerance as ϵ decreases, where $\epsilon = 10^{-k}, k = 1, 3, 5, 7, 9, 11$; On the lower left and right, we show the corresponding convergence results and the number bases with the reorthogonalization procedure of the evolved basis.

4.2 The second order wave equation

To further evaluate the stability of the new parareal algorithm, we consider the second-order wave equation from [8]:

$$\begin{aligned} u_{tt} &= c^2 u_{xx}, \quad (x,t) \in (0, 2\pi) \times (0, T], \\ u(x,0) &= f(x), \quad u_t(x,0) = g(x), \end{aligned} \tag{78}$$

where $T = 10$ and $c = 5$ and a 2π-periodic boundary condition is used. With

$$f(x) = \sum_{l=-N}^{N} \hat{u}_l e^{ilx}, \quad g(x) = 0 \tag{79}$$

and

$$\hat{u}_l = \begin{cases} \frac{1}{|l|^p}, & l \neq 0, \\ 0 & l = 0. \end{cases}$$

and set $p = 4$. In the following we use a Fourier spectral discretization with 33 modes in space [14] and the velocity Verlet algorithm in time [24]. The following parameters are used in the parareal algorithm:

$$N_c = 100, \quad N_{it} = 10, \quad \Delta t = 10^{-3}, \quad \delta t = 10^{-4}. \tag{80}$$

We set the tolerance for POD to 10^{-11}.

In Figure 7 (left), we show the L_∞-error at $T = 10$ against the number of iterations for the original parareal method and the POD parareal method. The original parareal method is clearly unstable, while the POD parareal remains stable and converges in one iteration. This confirms our analysis: if the reduced model is accurate enough, the reduced basis parareal should converge in one iteration. In Figure 5 (right), we show the number of bases needed to satisfy the tolerance ϵ in the POD parareal method.

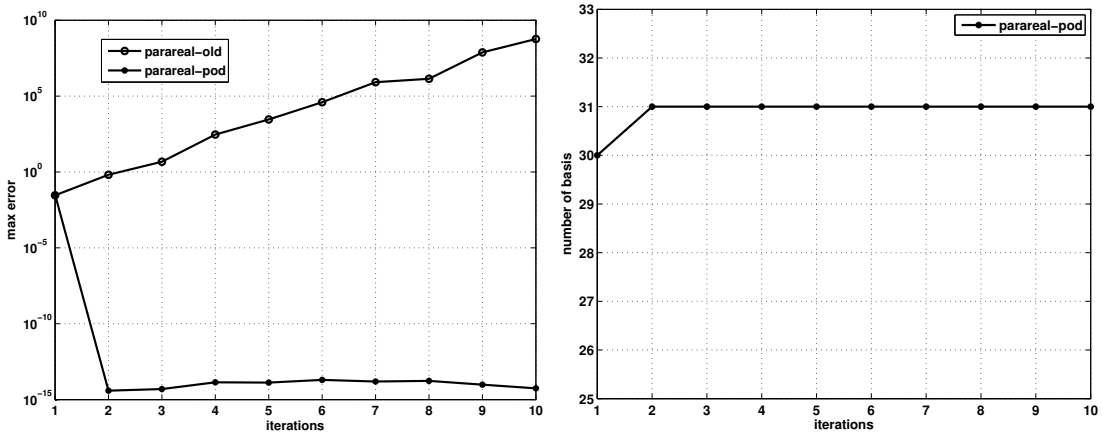

Fig. 7: Results obtained using the original parareal method, the POD parareal method for the 1-D second order wave equation. On the left we show the L_∞-error at $T = 10$ against the number of iterations while the right shows the number of bases used for satisfying the tolerance of $\epsilon(10^{-11})$ in the POD parareal method across the iterations.

4.3 Nonlinear equations

Let us also apply the reduced basis parareal method to examples of nonlinear PDEs. We recall that the Krylov based approach is not applicable in this case.

4.3.1 Viscous Burgers' equation

We first consider the viscous Burgers' equation (6), with the same spatial and temporal discretization and the same parameters as in (7). To build the reduced basis, we set the tolerance for POD and EIM to be 10^{-15} and 10^{-10}.

In Figure 8 (left), we show the L_∞-error at $T = 2$ against the number of iterations for the original parareal method, the POD parareal method, and the EIM parareal method. Note that in this case, the RB parareal performs worse than the original parareal does. It is a result of the reduced model not adequately capturing the information of the fine solver. Recall that in the nonlinear case, we have to deal with two approximations: one for the state variables and one for the nonlinear term. For the POD parareal algorithm, we choose the number of reduced bases based on the tolerance for the state variable u; alternatively, we can choose the dimension of the reduced approximation space based on the tolerance for the nonlinear term. The latter approach shows better convergence behavior in Figure 8 (left, parareal-pod-modified). It is apparent that the quality of the reduced model directly impacts the convergence.

We emphasize that although the reduced basis parareal method converges slower than the original parareal, it is less expensive, as discussed in Section 2.3.1.

4.3.2 Kuramoto-Sivashinsky equation

Next we consider the Kuramoto-Sivashinsky equation (8). The same spatial and temporal discretization and the same parameters as in (9) are used. To build the reduced basis, we set the tolerance for POD and EIM to be 10^{-13} and 10^{-8}, respectively.

In Figure 9 we show the L_∞-error at $T = 40$ against the number of iterations for the original parareal method, the POD parareal method, the modified POD parareal, and the EIM parareal method. It is clear that the reduced basis parareal method converges faster than the original parareal method. This is likely caused by the solution of the problem being smooth enough to ensure that there exists a compact reduced model. Moreover, to keep the corresponding tolerance, the number of degrees of freedom in the reduced basis parareal methods is roughly one-third that of the original parareal method.

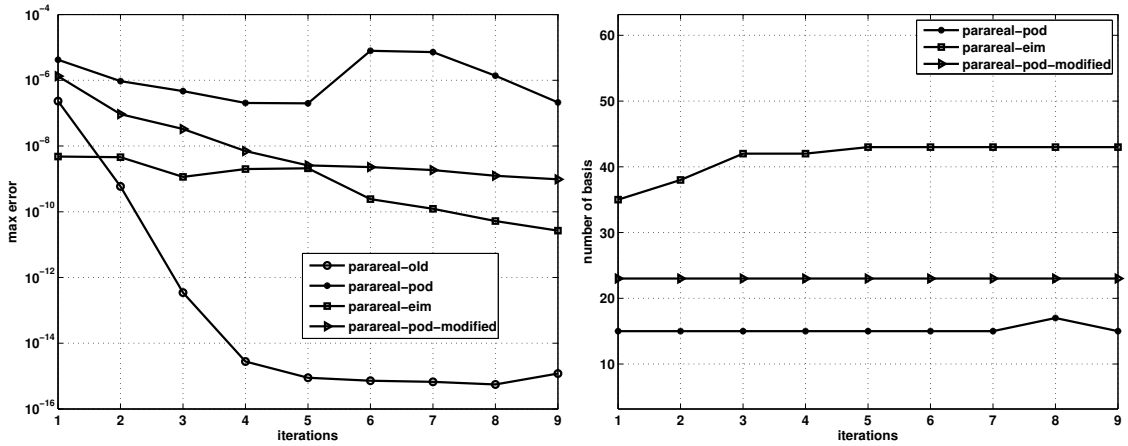

Fig. 8: We compare the performance of the original parareal method, the POD parareal method, the modified POD parareal and the EIM parareal method for the 1-D Burgers' equation. On the left we show the L_∞-error at $T = 2$ against the number of iterations, while the right illustrates the number of bases.

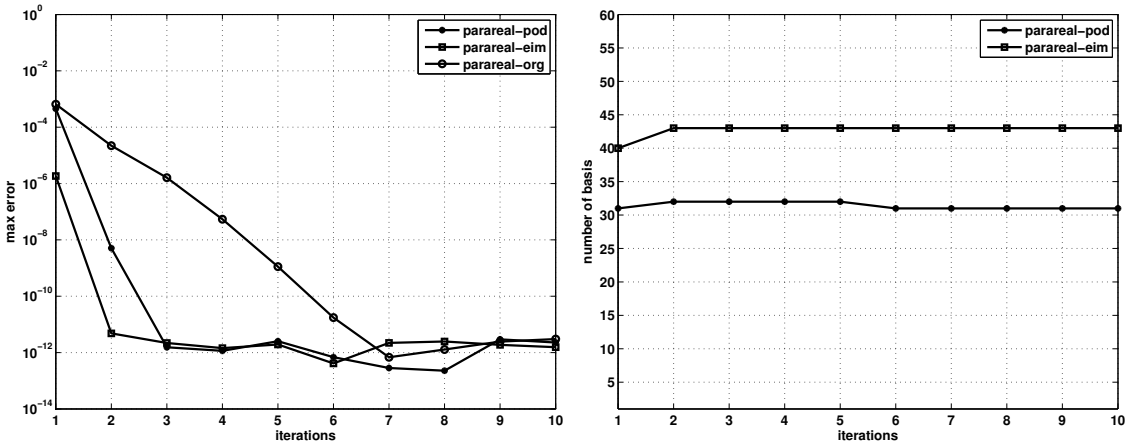

Fig. 9: We compare the performance of the original parareal method, the POD parareal method, and the EIM parareal method for the 1-D Kuramoto-Sivashinsky equation. On the left we show the L_∞-error at $T = 40$ against the number of iterations, while the right shows the number of bases used against the number of iterations.

4.3.3 Allan-Cahn equation: nonlinear source

As a third nonlinear example we consider the 1-D Allan-Cahn equation:

$$\begin{aligned} &\frac{\partial u}{\partial t} = \nu u_{xx} + u - u^3, \quad (x,t) \in (0, 2\pi) \times (0, T], \\ &u(x,0) = 0.25 \sin(x), \end{aligned} \tag{81}$$

where $T = 2$ and $\nu = 2, 1, 10^{-1}, 10^{-2}$. A periodic boundary condition is assumed. We use a P_1 DG method with 100 elements in space [15] and a forward Euler scheme in time. The following parameters are used in the parareal algorithm

$$N_c = 200, \quad N_{it} = 5, \quad \Delta t = 1 \times 10^{-4}, \quad \delta t = 5 \times 10^{-6}. \tag{82}$$

We set the tolerance for POD and EIM to be 10^{-12} and 10^{-8}.

In Figure 10 (left), we show the L_∞-error at $T = 2$ against the number of iterations for the POD parareal method with different values of ν's. It is clear that for larger values of ν, the solution converges faster and less elements in the reduced basis is needed. This is expected since a larger ν indicates a smoother and more localized solution which allows an efficient representation in a lower dimensional space. Similar results are obtained by an EIM based parareal approach and are

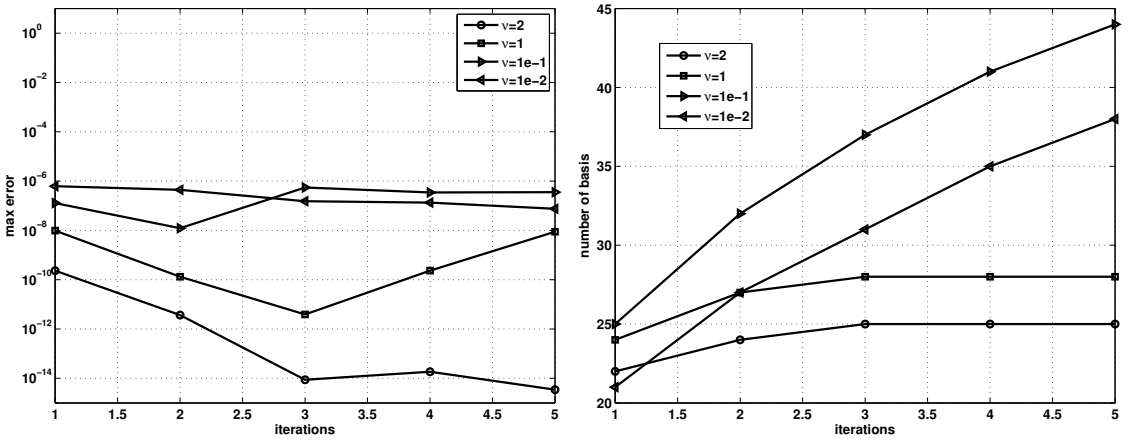

Fig. 10: The POD parareal method for the 1-D Allan-Cahn equation. On the left we show the L_∞-error at $T = 2$ against the number of iterations for different values of ν and on the right we show the number of bases.

not reproduced here.

4.3.4 KdV equation: nonlinear flux

As a last example we consider the KdV equation (taken from [26]):

$$\begin{aligned} &\frac{\partial u}{\partial t} = -(\frac{u^2}{2})_x - \nu u_{xxx}, \quad (x,t) \in (-1,1) \times (0,T], \\ &u(x,0) = 1.5 + 0.5\sin(2\pi x), \end{aligned} \tag{83}$$

where $T = 2$ and $\nu = 10^{-3}$ and we assume a periodic boundary condition. The equation conserves energy, much like the linear wave equation, but the nonlinearity induces a more complex behavior with the generation of propagating waves. In the parareal algorithm we use

$$N_c = 100, \quad N_{it} = 10, \quad \Delta t = 10^{-4}, \quad \delta t = 10^{-5}. \tag{84}$$

We use a first order local discontinuous Galerkin method (LDG) with 100 elements in space [26, 15] and an IMEX scheme in time [1], with the linear terms treated implicitly and the nonlinear term explicitly. We set the tolerance for POD and EIM to be 10^{-13} and 10^{-8} respectively.

In Figure 11 (left) we show the L_∞-error at $T = 2$ against the number of iterations for the original parareal method, the POD parareal method, and the EIM parareal method. While the POD parareal method does not work well in this case, the EIM parareal method shows remarkable performance, i.e., it converges much faster than the original parareal method. Note that even if the tolerance for the POD is smaller than that of the EIM, it does not guarantee that the reduced model error based on the POD approach is smaller. There are two reasons: 1) the meaning of the tolerance in the context of the POD and the EIM are different. 2) in the convergence proof of (73), the constants $C_r, C_{p,r}$ depend on the details of the reduced approximation and the dimension of reduced approximation space, which impact the final approximation error.

5 Conclusion

In this paper, we propose an approach to produce and use a reduced basis method to replace the coarse solver in the parareal algorithm. We demonstrate that, as compared with the original parareal method, this new reduced basis parareal method has improved stability characteristics and efficiency, provided that the solution can be represented well by a reduced model. The analysis of the method is confirmed by the computation results, e.g., the accuracy of the parareal method is determined by the accuracy of the fine solver and the reduced model, used to replace the coarse solver. Unlike the Krylov subspace parareal method, this approach can be extended to include both linear problems and nonlinear problems, while requiring less storage and computing resources. The robustness and versatility of the method has been demonstrated through a number of different problems and sets the stage for the evaluation on more complex problems.

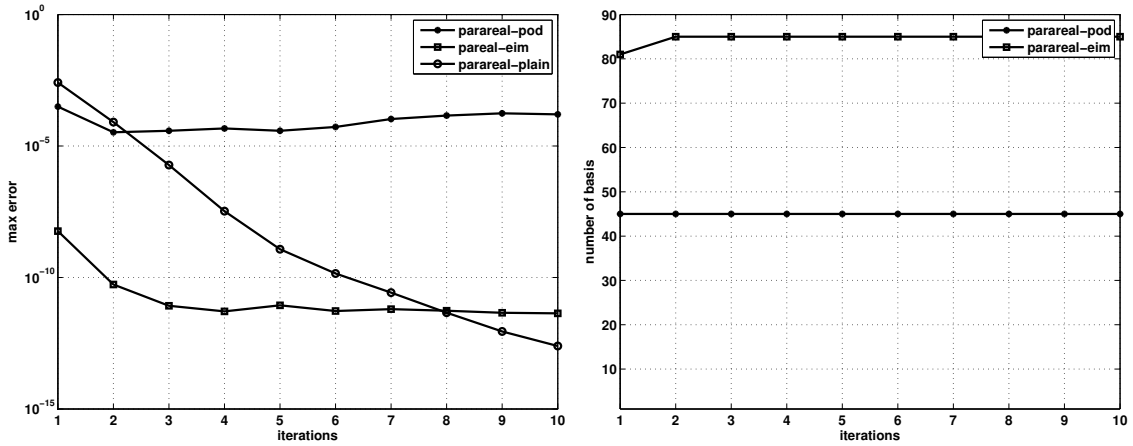

Fig. 11: We compare the performance of the original parareal method, the POD parareal method, and the EIM parareal method for the 1-D KdV equation. On the left we show the L_∞-error at $T = 2$ against the number of iterations, while the right shows the number of bases used against the number of iterations.

6 Acknowledgement

The authors acknowledge partial support by OSD/AFOSR FA9550-09-1-0613 and AFOSR FA9550-12-1-0463.

References

1. U.M. Ascher, S.J. Ruuth, and B.T.R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32(3):797–823, 1995.
2. L. Baffico, S. Bernard, Y. Maday, G. Turinici, and G. Zérah. Parallel-in-time molecular-dynamics simulations. *Physical Review E*, 66(5):057701, 2002.
3. G. Bal. On the convergence and the stability of the parareal algorithm to solve partial differential equations. *Domain decomposition methods in science and engineering*, pages 425–432, 2005.
4. M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera. An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.
5. D. Cavar and K.E. Meyer. Les of turbulent jet in cross flow: Part 2–POD analysis and identification of coherent structures. *International Journal of Heat and Fluid Flow*, 2012.
6. A. Chatterjee. An introduction to the proper orthogonal decomposition. *CURRENT SCIENCE-BANGALORE-*, 78(7):808–817, 2000.
7. S. Chaturantabut and D.C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.
8. X. Dai and Y. Maday. Stable parareal in time method for first and second order hyperbolic system. *arXiv preprint arXiv:1201.1064*, 2012.
9. E. Emmrich. *Discrete versions of Gronwall's lemma and their application to the numerical analysis of parabolic problems*. TU, Fachbereich 3, 1999.
10. C. Farhat, J. Cortial, C. Dastillung, and H. Bavestrello. Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses. *International journal for numerical methods in engineering*, 67(5):697–724, 2006.
11. M. Gander and M. Petcu. Analysis of a Krylov subspace enhanced parareal algorithm for linear problems. In *ESAIM: Proceedings*, volume 25, pages 114–129, 2008.
12. M.J. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM Journal on Scientific Computing*, 29(2):556–578, 2007.
13. Liping He. The reduced basis technique as a coarse solver for parareal in time simulations. *J. Comput. Math*, 28:676–692, 2010.
14. Gottlieb S. Hesthaven, J.S and G. Gottlieb. *Spectral Methods for Time-Dependent Problems*. Cambridge University Press, 2007.
15. J.S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*, volume 54. Springer, 2007.
16. J.L. Lions, Y. Maday, and G. Turinici. A"parareal"in time discretization of pde's. *Comptes Rendus de l'Academie des Sciences Series I Mathematics*, 332(7):661–668, 2001.
17. Y Maday. Parareal in time algorithm for kinetic systems based on model reduction. *High-dimensional partial differential equations in science and engineering*, 41:183–194.
18. Y. Maday and G. Turinici. Parallel in time algorithms for quantum control: Parareal time discretization scheme. *International journal of quantum chemistry*, 93(3):223–228, 2003.
19. A.S. Nielsen. Feasibility study of the parareal algorithm. 2012.
20. Gianluigi Rozza, DBP Huynh, and Anthony T Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):229–275, 2008.
21. D. Ruprecht and R. Krause. Explicit parallel-in-time integration of a linear acoustic-advection system. *Computers & Fluids*, 59:72–83, 2012.
22. L. M. Skvortsov. Diagonally implicit Runge-Kutta methods for stiff problems. *Computational Mathematics and Mathematical Physics*, 46(12):2110–2123, December 2006.
23. G. Staff and E. Rønquist. Stability of the parareal algorithm. *Domain decomposition methods in science and engineering*, pages 449–456, 2005.
24. Loup Verlet. Computer" experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical review*, 159(1):98, 1967.
25. Y. Xu and C.W. Shu. Local discontinuous galerkin methods for the Kuramoto–Sivashinsky equations and the Ito-type coupled KdV equations. *Computer methods in applied mechanics and engineering*, 195(25):3430–3447, 2006.

26. J. Yan and C.W. Shu. A local discontinuous galerkin method for KdV type equations. *SIAM Journal on Numerical Analysis*, 40(2):769–791, 2002.