A STABLE PARAREAL-LIKE METHOD FOR THE SECOND ORDER WAVE EQUATION

HIEU NGUYEN AND RICHARD TSAI

ABSTRACT. A new parallel-in-time iterative method is proposed for solving the homogeneous second-order wave equation. The new method involves a coarse scale propagator, allowing for larger time steps, and a fine scale propagator which fully resolves the medium using finer spatial grid and shorter time steps. The fine scale propagator is run in parallel for short time intervals. The two propagators are coupled in an iterative way that resembles the standard parareal method [22]. We present a data-driven strategy in which the computed data gathered from each iteration are re-used to stabilize the coupling by minimizing the energy residual of the fine and coarse propagated solutions. An example of Marmousi model is provided to demonstrate the performance of the proposed method.

Keywords: parallel-in-time, wave equation, Procrustes problem

1. INTRODUCTION

In this paper, we will focus on the initial value problem of the standard second order wave equation:

\[
\frac{\partial^2 u}{\partial t^2} = c^2(x) \Delta u, \quad x \in [0, 1)^d, \quad 0 \leq t < T,
\]

with initial conditions \( u(x, 0) = u_0(x), \quad u_t(x, 0) = p_0(x) \) and either periodic boundary conditions or absorbing boundary conditions from placing a perfectly matched layer around \([0, 1)^d\). The wave equation is a physical model for seismic wave and electromagnetic wave in certain simplified setups. It is also used as a test case for developing algorithms that are later generalized to more complicated elastic and electromagnetic wave equations. Our objective is to develop a stable parallel-in-time algorithm for such problem.

Time domain decomposition methods for evolution problems has been of increasing interest in the partial differential equation community due to the increasing number of cores available in modern supercomputers.
Despite rapid advance in parallel computer architecture, parallelizing the time evolution of the second order wave equation efficiently is still a challenging problem. One of the time domain decomposition implementations is parallel-in-time method in which whole time domain is partitioned into subintervals for parallel processing. The parallel-in-time method opens up another dimension to utilize computational resource in a supercomputer. The most relevant algorithm to this paper is the parareal method introduced by Lions, Maday and Turinici [22]. The parareal method combines iteratively two propagators, denoted by $F v$ the fine propagator and by $C v$ the coarse propagator. They advance the solution $v(x, t)$ to $v(x, t+\Delta t)$. The iteration can be described by

$$
\begin{align*}
 v^{k+1}_{n+1} &= C v^{k+1}_n + F v^k_n - C v^k_n, \\
v^1_0 &= v_0, v^1_{n+1} &= C v^1_n, \quad k = 1, 2, \ldots, n = 0, 1, \ldots, N.
\end{align*}
$$

Note that for each $k$, $F v^k_n$ is computed in parallel. For the second order wave equation under consideration, $v(x, t)$ is a vector of pair $[u(x, t); u_t(x, t)]$.

Typically, the coarse propagator runs on coarse grid and is cheaper to compute, while the fine propagator runs on finer grid and is assumed to fully resolve the small scales in the problem. The finer propagator is thus more costly to compute.

In [4], it is shown that the parareal method is stable and converges linearly to the serial fine solution if the coarse propagator is smooth and has sufficient dissipation. When certain conditions are met, parareal method can achieve high fidelity solution within few iterations, much smaller than the number of time steps which enables speed up in real world applications. A number of applications have been benefited from the parareal method: plasma turbulence in Tokamak reactor [34, 33, 32], Navier-Stoke equations [14, 35, 11], acoustic wave [25], shallow water [20], chemical kinetic [6], molecular dynamics [9], reaction wave [13], neutron diffusion [5, 24], lattice Boltzmann equation for laminar flow [26, 21, 27].

To further gain speed up, the coarse propagator can be chosen as a spatial coarsening of the fine propagator [31, 29] which allows larger coarse time step. Indeed, this coarsening technique provides additional speed up in some applications [23, 3, 21] because the coarse propagator has less grid points to compute, provided an appropriate grid restriction and interpolation operator. However as shown in [29], considerable coarse grid resolution and accurate interpolation are required in order to make the parareal iteration (2) converged.

The parareal method tends to suffer from slow convergence or instability when applied to hyperbolic problems. Using an oscillatory dynamical system as an example, [1, 2] pointed out that the phase error between the coarse and fine propagators is the reason for the slow convergence. Analogously for advection problem, the authors in [30] observe that numerical dispersion between the solvers make the parareal method
converging from above and hence causes instability. Intuitively, spatial constructive or destructive interference of two overlapping waves depends on their relative phase which is in turn sensitive to the frequency, yet the parareal iterative coupling is point-wise in space and time.

There have been some attempts to modify the classical parareal method in order to address the slow convergence issue. In [12], the fact that solutions to wave equation live on a submanifold of constant energy is exploited. In that work, the solutions are projected onto the submanifold to stabilize the parareal iterations. More precisely the algorithm can be presented as

\[ u_n^k = P[Cu_n^{k-1} + (Fu_n^{k-1} - (Cu_n^{k-1}))], \]

where \( P \) denotes the projection onto a predetermined submanifold. However, the projection is obtained by solving nonlinear equations which can be strongly sensitive to initial guess.

In the Krylov-subspace enhanced parareal methods [15, 31], the computed data is used to project the input conditions before applying the coarse propagator. More precisely, the Krylov-subspace parareal has the following form:

\[ u_n^k = (C(I - P)u_n^{k-1} + Fu_n^{k-1}) + Fu_n^{k-1} - (C(I - P)u_n^{k-1} + Fu_n^{k-1}), \]

where \( P \) is a linear projection operator, and \( I \) is the identity. The enhanced coarse propagator corresponds to

\[ C(I - P) + FP, \]

and the fine propagation of projected data, \( FPu \), is constructed by pre-computing the fine propagation of the basis in \( P \). Similarly, the reduced basis parareal method [10] develops more efficient ways to decompose the input condition and extends to solve nonlinear equations. The convergence and stability of these methods are analyzed and demonstrated by numerical examples of periodic constant variable in one and two dimension. However, combination with spatial grid coarsening and examples of variable wave speed have not been considered in these work.

On the other hand, it is known that the slow convergence and instability of the parareal method for hyperbolic problems can be due to some notions of phase errors [2, 1] and numerical dispersion [30]. In [1], effective multiscale parareal schemes relying on elaborate phase correction are proposed for a class of highly oscillatory dynamical systems. For dynamical systems on the complex plane, correcting the phase of the coarse solution corresponds to multiplication of a complex number. In [2], we derived convergence theory for the modified parareal schemes applying to linear systems of ordinary differential equations (ODEs). The theory resembles the classical linear stability theory for numerical schemes for ODEs, and can be used in a
similar fashion to stabilize the parareal iterations. Additionally, in [2] we investigated a few simple phase correction strategies systematically, and showed that appropriate phase correction can enable the resulting scheme to have superior performance.

A new proposed method uses computed data to boost the coarse propagator, based on the idea of $\theta$-parareal method [2]. Instead of decomposing the input data as in [15, 31], we use the computed data to build an operator, formally denoted as $\theta$, that directly brings the coarse solutions, $C u$, closer to the fine solutions, $F u$. In this paper, the $\theta$ operators are constructed by minimizing the residual between the fine and coarse solutions in a discrete semi-norm related to the wave energy.

2. Preliminary background

We briefly review the original plain parareal method and its properties. In a context of linear evolutionary problem $\dot{u}(t) = Au(t)$ for $t \in \{0, \Delta t, \ldots, N \Delta t = T\}$ and $A : \mathbb{R} \rightarrow \mathbb{R}$ linear function, let us denote the fine propagator/solver $F u_n \mapsto u_{n+1}^f$ and the coarse propagator/solver $C u_n \mapsto u_{n+1}^c$. Then the plain parareal iteration $k + 1$ can be written as a recurrence relation

$$u_{n+1}^{k+1} = C u_{n+1}^k + F u_n^k - C u_n^k.$$  

Starting solution $k = 1$ is the serial coarse solution $u_{n+1}^{k=1} = C^n u_0$. In addition, by rewriting the recurrence relation [2] in matrix form and manipulating inverse of Toeplitz, an error estimate $|u_n^k - u(t_n)|$ is derived in [2]

$$e_{n+1}^{k+1} \leq \|F - C\|_{\infty} \sum_{i=1}^{n-k-1} \|C\|_{\infty}^i e_n^k.$$  

First term on right hand side is equivalent to local truncation error of the coarse propagator, assuming the fine solver is an exact one. The summation term is bounded above by $N$ for stable schemes, e.g. $\|C\|_{\infty} \leq 1$. Above error estimate is equivalent to linear convergence analysis of the parareal method derived in [4, 16].

Wall-clock complexity of the parareal algorithm is estimated by

$$C_{\text{parareal}} = K \left(\frac{T}{\Delta t} + \frac{T}{n_{CPU}\delta t}\right).$$  

Comparing to complexity of the serial fine solver $C_{\text{fine}} = T/\delta t$, the parareal algorithm is more effective (from the perspective of total wall-clock computing time) if (i) a large number of computing nodes, $n_{CPU}$, are used; (ii) the coarse/fine time stepping ratio is sufficiently large $\Delta t/\delta t \gg 1$; and (iii) the number of iterations, $K$, needed to for the desired accuracy is small, assuming that the parareal iterations are convergent.

A key objective of this paper is to introduce a data-driven stabilization strategy to decrease the total number of needed iterations.
3. The proposed method

We propose a scheme that takes a general form:

\[ u_{n+1}^{k+1} = \theta_{n+1}[\tilde{C}u_n^{k+1}] + \tilde{F}u_n^k - \theta_n^{k+1}[\tilde{C}u_n^k]. \]

Here, \( u_n^k \) denotes the solutions computed on the grid, and it has two component blocks, one corresponds to the wave solution \( u \) and the other the time derivative \( u_t \). In this paper, for readability we shall also use \( \dot{u} \) to denote the time derivative of \( u \). The coarse and fine propagators, \( \mathcal{C} \) and \( \mathcal{F} \) will operate on different grids, and additional interpolation and restriction operators are needed for coupling the two propagators. Here we use \( \tilde{C} \) and \( \tilde{F} \) to denote the appropriately defined operations to be described in detail in this section.

A family of operators \( \theta_n^k[\cdot] \) are constructed such that

\[ \theta_{n+1} \approx \tilde{F}\tilde{C}^{-1} : \tilde{C}u \mapsto \tilde{F}u. \]

Clearly, direct calculation of \( \tilde{F}\tilde{C}^{-1} \) is not practical because it undermines time parallelization of the \( \theta \)-parareal method. Instead, we seek for an effective mapping that has similar property as \( \tilde{F}\tilde{C}^{-1} \) and is constructed from data computed along the parareal iterations.

3.1. Discretizations and data preparation. In this paper, we use the uniform Cartesian grids for the spatial domain and uniform stepping in time. Both the coarse and the fine propagators are defined by the standard second order central difference scheme for the spatial derivatives and velocity Verlet for time marching. The coarse propagator will operate on the coarse grid: \( \Delta x \cdot \mathbb{Z}^d \times \Delta t \cdot \mathbb{Z}^+ \), and the fine propagator will operate on the fine grid: \( \delta x \cdot \mathbb{Z}^d \times \delta t \cdot \mathbb{Z}^+ \), for \( d = 1 \) or \( 2 \). Let \( u_n \in \mathbb{R}^{N\Delta x}, U_n \in \mathbb{R}^{N\Delta x} \) denote respectively the solutions computed at time \( t_n = n\Delta t_{com}, n = 1, 2, 3, \ldots N \) on the fine and coarse grids.

These fine and coarse grid functions are coupled by an interpolation \( \mathcal{I} : U \mapsto u \) and a restriction \( \mathcal{R} : u \mapsto U \). The input wave speed for coarse propagator is point wise evaluated from the given wave speed. The fine and coarse propagators communicate at \( n\Delta t_{com} \). The fine propagator uses the step size \( \delta t = \Delta t_{com}/m_{\mathcal{F}} \) and the coarse propagator uses \( \Delta t = \Delta t_{com}/m_{\mathcal{C}} \), with \( m_{\mathcal{F}}, m_{\mathcal{C}} \in \mathbb{N} \) selected according to \( \delta x \) and \( \Delta x \) for stability in the respective time stepping. See Figure[1].

Given \( [u_{n-1}^k; \dot{u}_{n-1}^k] \) at \( t_{n-1} \), the fine and coarse propagators are applied to obtain the solutions at define

\[ [u_n, \dot{u}_n] := \mathcal{F}[u_{n-1}^k; \dot{u}_{n-1}^k] \]

and

\[ [U_n, \dot{U}_n] := \mathcal{C}[\mathcal{R}u_{n-1}^k, \mathcal{R}\dot{u}_{n-1}^k]. \]
Figure 1. Discretization diagram. Coarse propagator, $C$ uses spatial grid size $\Delta x$ and temporal step size $\Delta t$. Fine propagator, $F$ uses spatial grid size $\delta x$ and temporal step size $\delta t$. These propagators communicate at $(j\Delta x, n\Delta t_{com})$ for $j \in \mathbb{Z}, n = 1, 2, 3, \ldots, N$.

These solutions are propagated over a coupling time interval $[t_{n-1}, t_n]$. These propagators are expected to approximately preserve the wave energy.

Finally, we will quickly describe the data matrices that will be used to construct the operators $\theta_k$. We are interested in using the computed solution data, particularly the gradient of the wave field $\nabla u$ and a weighted momentum of $\dot{u}$. Each column of data matrices is formed by block(s) of the gradients $\nabla U_n$ followed by a block of momentum $\dot{U}_n$ of coarse grid solution at $n$-th coupling time. In practice, the gradient operator, $\nabla$, will be replaced by some numerical approximation $\nabla_h$. Then define the data:

$$F := \begin{bmatrix} \nabla_h R u_1 & \nabla_h R u_2 & \cdots & \nabla_h R u_N \\ c^{-1} \dot{R} u_1 & c^{-1} \dot{R} u_2 & \cdots & c^{-1} \dot{R} u_N \end{bmatrix},$$

$$G := \begin{bmatrix} \nabla_h U_1 & \nabla_h U_2 & \cdots & \nabla_h U_N \\ c^{-1} \dot{U}_1 & c^{-1} \dot{U}_2 & \cdots & c^{-1} \dot{U}_N \end{bmatrix}.$$  

Here and for the rest of the paper, $c^{-1} \dot{U}_m$ denotes the component-by-component multiplications of $c^{-1}(x_j)$ and $\dot{U}(x_j)$. The same convention is used for $c^{-1} \dot{R} u_j$.

Now, define the discrete wave energy function as

$$E([U_n, \dot{U}_n]) := \frac{1}{2} \sum_j |\nabla_h U_n(x_j)|^2 \Delta x^d + \frac{1}{2} \sum_j c_j^{-2} |\dot{U}_n(x_j)|^2 \Delta x^d.$$
We see that it is equivalent, up to a constant, to the Frobenius norm of the $G$:

$$\|G\|^2_F = \sum_{n=1}^{N} \left[ \sum_j |\nabla h U_n(x_j)|^2 + \sum_j c_j^{-2} |\dot{U}_n(x_j)|^2 \right] = \frac{2}{\Delta x^d} \sum_{n=1}^{N} \mathcal{E}([U_n, \dot{U}_n]).$$

### 3.2. Minimization of coarse-fine solution gaps in the discrete wave energy norm.

For wave propagation problems, it is often more physical to consider the associated wave energy. Therefore, we shall measure the discrepancy between the coarse and fine solutions in the discrete wave energy semi-norm $\|\|$.

Denote the $j$-th column of $F$ and $G$ by $f_j$ and $g_j$ respectively. We consider the following optimization problem:

$$\min_{\Omega \in \mathbb{R}^{(d+1)N_{\Delta x} \times (d+1)N_{\Delta x}}} \sum_{j=1}^{N} \|f_j - \Omega g_j\|^2_F, \text{ s.t. } \Omega^T \Omega = I.$$

Recall that the elements in the columns of $F$ and $G$ consist of the spatial gradients and weighted time derivatives of the solutions on the respective fine and coarse grid, and that the $\ell^2$ norm corresponds to the discrete wave energy $\|\|$.

Therefore, we look for a unitary matrix so that the discrete wave energy of the coarse solutions is preserved at different times. Intuitively this operator aligns the phase (in the above sense) of the coarse solution to fine solution for each $t_n$. It is similar to the local phase-alignment procedure in [1] as depicted in Figure 2. Indeed,

$$\|f_j - \Omega g_j\|^2_F = \|f_j\|^2_F + \|g_j\|^2_F - 2(f_j, \Omega g_j)$$

the minimization can be interpreted as minimizing the sum of the angles between the columns on the data matrices. Thus, we shall refer to $\Omega$ as a phase corrector.

The minimization problem (11) is equivalent to the ”Procrustes Problem” [17]:

$$\min_{\Omega \in \mathbb{R}^{(d+1)N_{\Delta x} \times (d+1)N_{\Delta x}}} \|F - \Omega G\|^2_F, \text{ s.t. } \Omega^T \Omega = I = \Omega^T \Omega,$$

where $\| \cdot \|_F$ denotes the Frobenius norm of a matrix. An in-depth review of the Procrustes problem can be found in [18]. Its variants have been instrumental to multidimensional statistical analysis, rigid body motion simulation, satellite tracking and machine learning [36, 28, 19].

### 3.3. Solution to the optimization problem.

The optimization problem (11) can be solved in a couple of different ways. One of them is to use the singular value decomposition (SVD) of the correlation matrix

$$M := FG^T = \sum_{j=1}^{n} \nabla h R u_j \otimes \nabla h U_j + c^{-1} R \dot{u}_j \otimes c^{-1} \dot{U}_j.$$
If matrix $M$ has full rank, the minimizer of (11) is uniquely

$$\Omega^* = XY^T,$$

where $X, Y$ are the left and right singular vectors of $M = X\Sigma Y^T$. Correspondingly, the minimum residual is

$$r_{\min}^2 = \|F\|_F^2 + \|G\|_F^2 - 2 \text{trace}(\Sigma).$$

Figure 2 illustrates the Procrustes problem and its solution in a simple setup in $\mathbb{R}^2$.

3.3.1. Low rank approximation of $\Omega^*$. We now consider a low rank approximation of $\Omega^*$ for computational efficiency. Since the number of time frames is usually much smaller than the number of spatial grid nodes; i.e. $N \ll (d+1)N_{\Delta x}$, we can factorize the data matrices using a reduced QR factorization. Denote the factorizations by $F = Q_F R_F$ and $G = Q_G R_G$, where $Q_F, Q_G \in \mathbb{R}^{(d+1)N_{\Delta x} \times N}$, $R_F, R_G \in \mathbb{R}^{N \times N}$.

With the singular value decomposition of the smaller system $R_F R_F^T = X_F \Sigma Y_G^T$, the correlation matrix can be factor into

$$M = Q_F X_F \Sigma Y_G^T Q_G^T.$$
The last relation shows that
\[ \text{rank}(M) = \text{rank}(R_F R^T_G) = \min(\text{rank}(F), \text{rank}(G)). \]
Hence we can use the factorization of the smaller \( N \times N \) matrix \( R_F R^T_G \) to obtain
\[ \Omega_* = (Q_F X_F)(Q_G Y_G)^T. \]  
By setting a tolerance to singular values in \( \Sigma \), there are \( s \) singular values such that \( \sigma_s \geq \text{tol} \) remained. As the result, we only need to store \( s \) number of columns in \( Q_F, Q_G \), and the truncated phase corrector becomes
\[ \Omega_* := \left( Q_F(:,1:s) X_F(1:s,1:s) \right) \left( Q_G(:,1:s) Y_G(1:s,1:s) \right)^T. \]

3.3.2. Enriching the phase corrector \( \Omega_* \). After every parareal iteration, there is more data become available. We can use them to enrich the phase corrector. Define
\[ M^{k+1} = M^k + F^k(G^k)^T. \]
The singular value decomposition of \( M^{k+1} = \widetilde{U} \Sigma \widetilde{V}^T \) can be updated using that of \( M^k = U S V^T \), see [7]. We summarize the update procedure is Algorithm 1.

3.4. Reconstruction of wave field from the gradient. After correcting the energy components, i.e. the gradients and the weighted time derivatives, of the coarse solutions, it is necessary to reconstruct the wave field pair from the corrected energy components. In other words, from \([q,p] \) defined as the corrected energy components of a wave field pair \([w,\dot{w}]\)
\[
\begin{bmatrix}
q \\
p
\end{bmatrix}
\equiv \Omega_* \Lambda
\begin{bmatrix}
w \\
\dot{w}
\end{bmatrix}
:= \Omega_*
\begin{bmatrix}
\nabla_h w \\
c^{-1} \dot{w}
\end{bmatrix},
\]
where the mapping \( \Lambda : [w,\dot{w}] \mapsto [\nabla_h w, c^{-1} \dot{w}] \) takes function to wave energy components. We want to deduce a wave field pair \([v,\dot{v}]\) such that
\[
\begin{bmatrix}
\nabla v \\
c^{-1} \dot{v}
\end{bmatrix}
\simeq
\begin{bmatrix}
q \\
p
\end{bmatrix}.
\]
It is straightforward to find the latter component \( \dot{v} = cp \). For the displacement component \( v \), we use the spectral property of differentiation \( \text{fft}\{\nabla v\} = i\xi \text{fft}\{v\} \) to recover its the Fourier modes as follow
\[
\text{fft}\{v\} = \begin{cases}
-i(\xi \cdot \text{fft}\{q\})|\xi|^{-2} & \text{for } |\xi| \neq 0, \\
\sum_{j=1}^{N_{\Delta t}} w(x_j) & \text{for } |\xi| = 0.
\end{cases}
\]
We denote this mapping from energy component to wave field component as \( \Lambda^\dagger : [\nabla v, c^{-1} \dot{v}] \mapsto [v,\dot{v}] \). In particular, when the gradient is approximated by Fourier method, this reconstruction is an identity.
Algorithm 1: Update SVD of the current correlation matrix

\[ M^{k+1} \equiv \tilde{U}\tilde{S}\tilde{V}^T = USV^T + FG^T \]

where \( USV^T = M^k \).

\[
[\tilde{U}, \tilde{S}, \tilde{V}] \leftarrow \text{UpdateSVD}(U, S, V, F, G, tol) :
\]

\textbf{if} \( S \) \textbf{is empty} \textbf{then}

\[
\begin{align*}
Q_F R_F &= \text{truncatedqr}(F) \\
Q_G R_G &= \text{truncatedqr}(G) \\
X_r \Sigma Y_r^T &= \text{svd}(R_F R_G^T) \\
\text{rank} M &= \text{sum}(\text{diag}(\Sigma)/\max(\text{diag}(\Sigma)) > tol) \\
\tilde{U} &= Q_F X_r(:, 1: \text{rank} M) \\
\tilde{V} &= Q_G Y_r(:, 1: \text{rank} M) \\
\tilde{S} &= \Sigma(:, 1: \text{rank} M)(:, 1: \text{rank} M)
\end{align*}
\]

\textbf{else}

\[
\begin{align*}
U_F &= U^T F \\
V_G &= V^T G \\
Q_F R_F &= \text{truncatedqr}(F - UU_F) \\
Q_G R_G &= \text{truncatedqr}(G - VV_G) \\
H &= [U_F; R_F][V_G; R_G]^T + [S; 0; 0; 0] \\
X_h \Sigma Y_h &= \text{svd}(H) \\
\text{rank} M &= \text{sum}(\text{diag}(\Sigma)/\max(\text{diag}(\Sigma)) > tol) \\
\tilde{U} &= [U Q_F] X_h(:, 1: \text{rank} M) \\
\tilde{V} &= [V Q_G] Y_h(:, 1: \text{rank} M) \\
\tilde{S} &= \Sigma(:, 1: \text{rank} M)(:, 1: \text{rank} M)
\end{align*}
\]

Proposition 3.1. Suppose the gradient of function \( v(x) \) is estimated by spectral method \( \nabla_h v \equiv \text{ifft}\{i\xi \text{fft}\{v\}} \), then

\[
(18) \quad \Lambda^\dagger \Lambda \begin{bmatrix} v \\ \dot{v} \end{bmatrix} = \Lambda^\dagger \begin{bmatrix} \text{ifft}\{i\xi \text{fft}\{v\}} \end{bmatrix} = \begin{bmatrix} v \\ \dot{v} \end{bmatrix}.
\]

\textbf{Proof.} Let

\[
\begin{bmatrix} w \\ \dot{w} \end{bmatrix} = \Lambda^\dagger \Lambda \begin{bmatrix} v \\ \dot{v} \end{bmatrix}
\]

Since \( \Lambda \) maps function to energy components we have

\[
\Lambda^\dagger \Lambda \begin{bmatrix} v \\ \dot{v} \end{bmatrix} = \Lambda^\dagger \begin{bmatrix} \nabla_h v \end{bmatrix}
\]
By construction of $\Lambda^\dagger$, for nonzero wavenumber $|\xi| \neq 0$

$$\text{fft}\{w\} = -i\xi \cdot \text{fft}\{\nabla_h v\}|\xi|^{-2}. $$

Here the gradient is approximated using spectral method then

$$\text{fft}\{w\} = -i\xi \cdot \{i\xi \text{fft}\{v\}\}|\xi|^{-2} = \text{fft}\{v\}. $$

And for zero wavenumber $|\xi| = 0$, $\text{fft}\{w\} = \sum_j v(x_j) = \text{fft}\{v\}$. Thus, $w = v$ while the second energy component $\dot{w} = cc^{-1}\dot{v} = \dot{v}$. This concludes that the mapping $\Lambda^\dagger\Lambda$ is equal to identity. □

If the gradient is approximated by central finite difference of $2m$-order instead of spectral method, for one dimensional setting equation (19) in the proof above becomes

$$\text{fft}\{w\} = -i\xi [\Delta x^{-1} \sum_{j=1}^{m} (e^{j\xi \Delta x} - e^{-ij\xi \Delta x})\beta_j \text{fft}\{v\}]|\xi|^{-2} $$

$$= 2(\Delta x)^{-1} \sum_{j=1}^{m} \sin(j\xi \Delta x)\beta_j \text{fft}\{v\}, $$

where $\beta_j$ are appropriate coefficients of the stencil. When the spatial grid is small enough $\xi \Delta x \ll 1$, above expression is approximately

$$\text{fft}\{w\} \approx 2(\Delta x)^{-1} \sum_{j=1}^{m} (j\xi \Delta x - \frac{1}{3!}(j\xi \Delta x)^3 + \mathcal{O}(j\xi \Delta x)^5)\beta_j \text{fft}\{v\} $$

$$= 2 \sum_{j=1}^{m} (j - \frac{1}{3!}j^3(\xi \Delta x)^2 + \mathcal{O}j^5(\xi \Delta x)^4)\beta_j \text{fft}\{v\}. $$

Particularly for second order central difference $m = 1$, we would have $\beta_1 = 1/2$, then

$$\text{fft}\{w\} = \text{sinc}(\xi \Delta x)\text{fft}\{v\}, $$

which says that $|\text{fft}\{w\}| \leq |\text{fft}\{v\}|$ because $\text{sinc}(\xi \Delta x) \leq 1$. In practice, we observe that the algorithm does not require spectral approximation of the gradient, but instead $\|\Lambda^\dagger\Lambda\|_2 \leq 1$ is necessary for stability of the method. When central finite difference is utilized, it is well known that the modified wavenumber is less than $|\xi|$, hence central difference satisfies the requirement $\|\Lambda^\dagger\Lambda\|_2 \leq 1$. Algorithm 2 summarizes above procedure.
Algorithm 2: Reconstruct function from the gradient.

\[
\begin{align*}
    w &\leftarrow \text{grad2func}(\nabla_h w, \sum w): \\
    \hat{p} &\leftarrow \text{fft}(\nabla_h w) \\
    \hat{q}(|\xi| \neq 0) &\leftarrow -i\xi \cdot \hat{p}|\xi|^{-2} \\
    \hat{q}(|\xi| = 0) &\leftarrow \sum w \\
    w &\leftarrow \text{iff}(\hat{q})
\end{align*}
\]

Algorithm 3: The proposed algorithm.

Initialization: \[ u_{n=1}^{k=1}, \dot{u}_{n=1}^{k=1} = IC[Ru_{n=1}^{k=1}, R\dot{u}_{n=1}^{k=1}] \]
\[ \Sigma = [ ], \quad X = [ ], \quad Y = [ ] \]

while tolerance not meet and \( k \leq K \) do

parfor \( n = 2 \rightarrow N \) do

\[
\begin{align*}
    &\quad [v_n, \dot{v}_n] = \mathcal{F}[u_{n-1}^{k=1}, \dot{u}_{n-1}^{k=1}] \\
    &\quad [U_n, \dot{U}_n] = \mathcal{C}[Ru_{n-1}^{k=1}, R\dot{u}_{n-1}^{k=1}]
\end{align*}
\]

end

Solve the orthogonal Procrustes problem:
\[
\begin{align*}
    \mathbf{F} &= [\nabla_h Ru_n, Ru^{-1}\dot{u}_n] \\
    \mathbf{G} &= [\nabla_h U_n, c^{-1}\dot{U}_n] \\
    [X, \Sigma, Y] &= \text{UpdateSVD}(X, \Sigma, Y, F, G, tol)
\end{align*}
\]

for \( n = 2 \rightarrow N \) do

\[
\begin{align*}
    &\quad [w, \dot{w}] = \mathcal{C}[Ru_{n=1}^{k+1}, R\dot{u}_{n=1}^{k+1}] \\
    &\quad [q, \dot{p}] = XY^T[\nabla_h w, c^{-1}\dot{w}] \\
    &\quad [\hat{q}, \hat{p}] = XY^T[\nabla_h U_n, c^{-1}\dot{U}_n] \\
    &\quad \text{Reconstruct function from gradient:} \\
    &\quad q_1 = \text{grad2func}(q, \sum w) \\
    &\quad \hat{q}_1 = \text{grad2func}(\hat{q}, \sum U_n) \\
    &\quad \text{Update next time step:} \\
    &\quad [u_{n=1}^{k+1}, \dot{u}_{n=1}^{k+1}] = [v_n, \dot{v}_n] + I\left(\left[\text{real}(q_1 - \hat{q}_1), c(p - \hat{p})\right]\right)
\end{align*}
\]

end

end

3.5. The proposed algorithm. The proposed algorithm couples the fine and the coarse propagators at times \( n\Delta t_{com}, n = 1, 2, \ldots, N \) over the fine grid (the spatial grid that the fine solutions are defined). However, the phase correction are applied
on the coarse grid. If the two grids are not identical, an interpolation is needed. We denote the interpolation operator by \( I \). Furthermore, denote the mappings between the wavefield \([v, \dot{v}]\) and its energy components \([\nabla v, c^{-1}\dot{v}]\) by \( \Lambda : [v, \dot{v}] \mapsto [\nabla v, c^{-1}\dot{v}] \) and \( \Lambda^\dagger : [\nabla v, c^{-1}\dot{v}] \mapsto [v, \dot{v}] \). With these notations, the \( \theta \) operator after \( k \) iterations can be written as

\[
\theta^k[v, \dot{v}] = I\Lambda^\dagger \Omega^k \Lambda [v, \dot{v}].
\]

Here we use \( \Omega^k \) to denote the phase corrector derived from the data matrix \( M^k \).

Finally, our new algorithm can be written compactly as in \( \theta \)-parareal form

\[
\begin{bmatrix}
u_{n+1}^{k+1} \\
\dot{u}_{n+1}^{k+1}
\end{bmatrix} = \theta^k C \begin{bmatrix}
\mathcal{R}u_{n+1}^k \\
\mathcal{R}\dot{u}_{n+1}^k
\end{bmatrix} + 
\mathcal{F} \begin{bmatrix}
u_n^k \\
\dot{u}_n^k
\end{bmatrix} - \theta^k C \begin{bmatrix}
\mathcal{R}u_n^k \\
\mathcal{R}\dot{u}_n^k
\end{bmatrix}
\]

(20)

Algorithm 3 describes the new scheme in a pseudo-code form with more details.

4. Complexity Analysis

There are three parts to our implementation: parallel fine propagator computation, construction of \( \Omega^* \) and the serial coarse updates. We assume that (i) no spatial domain decomposition, i.e. whole domain on a single core, (ii) standard QR complexity, i.e. no parallelization, (iii) communication between nodes and other tasks negligible here. Further speed-up for (i) and (ii) via parallelization will only improve the algorithm’s efficiency.

In each iteration, the wall clock time complexity for the parallel fine and coarse computations together is in the order of

\[
\frac{1}{n_{CPU}} \left( \frac{T}{\Delta t} N_{dx} + \frac{T}{\Delta t} N_{dx} \right) (d + 1)
\]

where \( n_{CPU} \) is the number of cores, \( N_{dx}, N_{dx} \) are respectively the total number of fine and coarse grid points. The complexity of serial coarse update in a iteration is

\[
\frac{T}{\Delta t} (d + 1) N_{dx}.
\]

The complexity of standard QR factorization for constructing \( \Omega \) is

\[
(d + 1) N_{dx} N^2.
\]

Therefore, the total complexity is

\[
K(d + 1) \left( \frac{T}{n_{CPU} \Delta t} N_{dx} + \frac{T}{n_{CPU} \Delta t} N_{dx} + \frac{T}{\Delta t} N_{dx} + N_{dx} N^2 \right),
\]

(21)

where \( K \) is the number of iterations. In this set up, the speed up over a serial fine computation is

\[
\left[ K \left( \frac{1}{n_{CPU}} + \left( \frac{1}{n_{CPU}} + 1 \right) \frac{T}{n_{dx} \Delta t} \frac{N_{dx} N \Delta t}{N_{dx} N \Delta t + N_{dx} N^2} \right) \right]^{-1}.
\]

(22)
Additionally, we have coarse/fine time step ratio $\Delta t / \delta t = m$, which implies $\Delta t_{\text{com}} / \delta t \geq m$, and their corresponding the mesh ratio is $\Delta x / \delta x = m_s$. Hence the speed up is approximately

$$\frac{1}{K} \min \{ n_{CPU}, m_d m_t, \frac{m_d m_t}{N} \}.$$  

5. Stability and convergence

In this section, we will derive some estimates that show the stability and the convergence of algorithm 3 under certain assumptions. We measure the difference, in the discrete energy semi-norm on the coarse grid, between the serially computed fine solution and the iterated solution.

Consider energy components of parareal iterated solution restricted on the coarse grid

$$\begin{bmatrix} \nabla h U_{n+1}^k \\ \frac{1}{c} \dot{U}_{n+1}^k \end{bmatrix} \equiv \Lambda R \begin{bmatrix} u_{n+1}^k \\ \dot{u}_{n+1}^k \end{bmatrix}.$$  

Its parareal iterative coupling is expressed as equation (20)

$$\begin{bmatrix} \nabla h U_{n+1}^k \\ \frac{1}{c} \dot{U}_{n+1}^k \end{bmatrix} = \Lambda \mathcal{R} \theta^{k-1} C \begin{bmatrix} \mathcal{R} u_{n}^k \\ \mathcal{R} \dot{u}_{n}^k \end{bmatrix} + \mathcal{F} \begin{bmatrix} u_{n-1}^k \\ \dot{u}_{n-1}^k \end{bmatrix} - \theta^{k-1} C \begin{bmatrix} \mathcal{R} u_{n-1}^k \\ \mathcal{R} \dot{u}_{n-1}^k \end{bmatrix}.$$  

Recall that $\theta[v, \dot{v}] := I \Lambda^\dagger \Omega \Lambda [v, \dot{v}]$, so

$$\Lambda \mathcal{R} \theta^{k-1} = \Lambda \mathcal{R} I \Lambda^\dagger \Omega^k \Lambda.$$  

Since the restriction operator takes point wise values, it cancels action of the interpolation $\mathcal{R} I = 1$. So equation (24) becomes

$$\begin{bmatrix} \nabla h U_{n+1}^k \\ \frac{1}{c} \dot{U}_{n+1}^k \end{bmatrix} = \Lambda \Lambda^\dagger \Omega^k \Lambda C \begin{bmatrix} U_{n}^k \\ \dot{U}_{n}^k \end{bmatrix} + \Lambda \mathcal{R} \mathcal{F} \begin{bmatrix} u_{n-1}^k \\ \dot{u}_{n-1}^k \end{bmatrix} - \Lambda \Lambda^\dagger \Omega^k \Lambda C \begin{bmatrix} U_{n-1}^k \\ \dot{U}_{n-1}^k \end{bmatrix}.$$  

Let us denote the square root of wave energy as $\mathcal{E}([U, \dot{U}]) := \sqrt{E([U, \dot{U}])}$, where $E$ is defined in (9). Thus,

$$\mathcal{E}([U_{n}^k, \dot{U}_{n}^k]) = \| \begin{bmatrix} \nabla h U_{n}^k \\ \frac{1}{c} \dot{U}_{n}^k \end{bmatrix} \|_2.$$  

**Theorem 5.1.** Suppose that

1. the coarse propagator $\mathcal{C}$ satisfies, for some $\epsilon > 0$;

$$\mathcal{E}(\mathcal{C}[U, \dot{U}]) \leq \mathcal{E}([U, \dot{U}]) + \epsilon,$$
(2) the residual of the energy minimization problem is bounded uniformly for 
k = 1, 2, \ldots:
\[ \|F^k - \Omega^*_k G^k\|_F \leq \eta \]
where \(F^k, G^k\) are data matrices in \([7], [8]\) gathered in the first \(k\) iterations;
(3) \(\|\Lambda^1\Lambda\|_2 \leq 1\), and \(\|1 - \Lambda^1\Lambda\|_2 < \lambda < 1/N\).
Then
\[ \max_{n \leq N} E([U^k_n, \dot{U}^k_n]) \leq \frac{1}{1 - \lambda N} \left( E([U_0, \dot{U}_0]) + (N + 1)\epsilon + C\eta \right), \]
where \(C\) is a norm equivalence constant between \(\ell_2, 1\) and Frobenius.

Proof. First, we apply triangle inequality on (25) to obtain
\[
E([U^k_{n+1}, \dot{U}^k_{n+1}]) \leq \|\Lambda\Lambda^\dagger \Omega^*\Lambda [U^k_{n+1}, \dot{U}^k_{n+1}]\|_2 + \|\Lambda R F [\dot{U}^k_{n+1} - \dot{U}^k_{n}] - \Lambda\Lambda^\dagger \Omega^* \Lambda [U^k_{n+1}, \dot{U}^k_{n+1}]\|_2
\]
\[
\leq \|\Lambda\Lambda^\dagger\|_2 \|\Omega\|_2 \|\Lambda\|_2 + \|\Lambda R F [\dot{U}^k_{n+1} - \dot{U}^k_{n}] - \Lambda\Lambda^\dagger \Omega^* \Lambda [U^k_{n+1}, \dot{U}^k_{n+1}]\|_2.
\]
By construction, \(\|\Omega\|_2 = 1\), and by the hypotheses that \(\|\Lambda\Lambda^\dagger\|_2 \leq 1\) and energy bound of the coarse propagator,
\[ \|\Omega^* \Lambda [U^k_n, \dot{U}^k_n]\|_2 = E(\Omega^* \Lambda [U^k_n, \dot{U}^k_n]) \leq E([U^k_n, \dot{U}^k_n]) + \epsilon, \]
we have
\[
E([U^k_{n+1}, \dot{U}^k_{n+1}]) \leq E([U^k_n, \dot{U}^k_n]) + \epsilon + \|\Lambda R F [\dot{U}^k_{n+1} - \dot{U}^k_{n}] - \Lambda\Lambda^\dagger \Omega^* \Lambda [U^k_{n+1}, \dot{U}^k_{n+1}]\|_2.
\]
Seeing the third term as part of the energy minimization problem in \([10]\),
\[
E([U^k_{n+1}, \dot{U}^k_{n+1}]) \leq E([U^k_n, \dot{U}^k_n]) + \epsilon + \|f_{n+1} - \Omega^* g_{n+1}\|_2 + \|(1 - \Lambda\Lambda^\dagger) \Omega^* \Lambda [U^k_{n+1}, \dot{U}^k_{n+1}]\|_2
\]
\[
\leq E([U^k_n, \dot{U}^k_n]) + \epsilon + \|f_{n+1} - \Omega^* g_{n+1}\|_2 + \|1 - \Lambda\Lambda^\dagger\|_2 \left( E([U^k_{1, n}, \dot{U}^k_{1, n}]) + \epsilon \right)
\]
\[
\leq E([U^k_n, \dot{U}^k_n]) + \epsilon + \|f_{n+1} - \Omega^* g_{n+1}\|_2 + \lambda \left( E([U^k_{1, n}, \dot{U}^k_{1, n}]) + \epsilon \right)
\]
\[
\leq E([U_0, \dot{U}_0]) + (n + 1)\epsilon + \sum_{j=1}^n \|f_j - \Omega^* g_j\|_2 + \sum_{j=0}^n \lambda \left( E([U^k_{1, j}, \dot{U}^k_{1, j}]) + \epsilon \right)
\]
\[
\leq E([U_0, \dot{U}_0]) + (n + 1)\epsilon + C\eta + \lambda n \left( \max_{j \leq N} E([U^k_{1, j}, \dot{U}^k_{1, j}]) + \epsilon \right).
\]
As the above relation also holds for $\max_{j \leq N} \mathcal{E}([U_j^k, \dot{U}_j^k])$, therefore,

$$\max_{j \leq N} \mathcal{E}([U_j^k, \dot{U}_j^k]) \leq \mathcal{E}([U_0, \dot{U}_0]) + (N + 1)\epsilon + C\eta \left( \max_{j \leq N} \mathcal{E}([U_j^{k-1}, \dot{U}_j^{k-1}]) + \epsilon \right)$$

$$= \lambda N \max_{j \leq N} \mathcal{E}([U_j^{k-1}, \dot{U}_j^{k-1}]) + \left( \lambda N \epsilon + \mathcal{E}([U_0, \dot{U}_0]) + (N + 1)\epsilon + C\ell_2, \eta \right).$$

Applying the discrete Grönwall inequality and geometric series we get

$$\max_{j \leq N} \mathcal{E}([U_j^k, \dot{U}_j^k]) \leq (\lambda N)^{k-1} \max_{j \leq N} \mathcal{E}([U_j^1, \dot{U}_j^1]) + \left( \lambda N \epsilon + \mathcal{E}([U_0, \dot{U}_0]) + (N + 1)\epsilon + C\ell_2, \eta \right) \sum_{l=0}^{k-1} (\lambda N)^l.$$

Thus,

$$\max_{j \leq N} \mathcal{E}([U_j^k, \dot{U}_j^k]) \leq \left( \lambda N \epsilon + \mathcal{E}([U_0, \dot{U}_0]) + (N + 1)\epsilon + C\eta \right) \frac{1}{1 - \lambda N}.$$ 

□

Next, we will show that, under some hypotheses, the proposed method converges to the solutions computed by applying the fine propagator serially. We shall use the following notation for those reference solutions:

$$(27) \quad [u(t_n), \dot{u}(t_n)] := \mathcal{F}^n[u_0, \dot{u}_0], \quad n = 1, 2, \ldots, N.$$ 

We measure the overall error on the fine grid as the square root of the difference in the discrete wave energy:

$$(28) \quad \mathcal{E}_n^k := \|\Lambda[u_n^k - u(t_n), \dot{u}_n^k - \dot{u}(t_n)]\|_2.$$ 

**Hypothesis 5.1.** (i) The phase corrected coarse solution is Lipschitz continuous in energy

$$\|\Lambda \theta \mathcal{CR} \begin{bmatrix} v \\ \dot{v} \end{bmatrix} - \Lambda \theta \mathcal{CR} \begin{bmatrix} w \\ \dot{w} \end{bmatrix}\|_2 = \|\Lambda \mathcal{I} \Lambda^\dagger \mathcal{AC} \mathcal{R} \begin{bmatrix} v \\ \dot{v} \end{bmatrix} - \Lambda \mathcal{I} \Lambda^\dagger \mathcal{AC} \mathcal{R} \begin{bmatrix} w \\ \dot{w} \end{bmatrix}\|_2$$

$$\leq (1 + \epsilon_{\mathcal{TR}}) (1 + \epsilon_{\lambda^\dagger \mathcal{AC}}) \|\Lambda \begin{bmatrix} v - w \\ \dot{v} - \dot{w} \end{bmatrix}\|_2.$$ 

Let $\epsilon_\theta$ denote the overall perturbation

$$(29) \quad \|\Lambda \theta \mathcal{CR} \begin{bmatrix} v \\ \dot{v} \end{bmatrix} - \Lambda \theta \mathcal{CR} \begin{bmatrix} w \\ \dot{w} \end{bmatrix}\|_2 \leq (1 + \epsilon_\theta) \|\Lambda \begin{bmatrix} v - w \\ \dot{v} - \dot{w} \end{bmatrix}\|_2.$$
(ii) The energy error between fine and corrected coarse operators is Lipschitz continuous

\[(30) \quad \| (\Lambda F - \Lambda \theta CR) \begin{bmatrix} v \\ \dot{v} \end{bmatrix} - (\Lambda F - \Lambda \theta CR) \begin{bmatrix} w \\ \dot{w} \end{bmatrix} \|_2 \leq \kappa \| \Lambda \begin{bmatrix} v - w \\ \dot{v} - \dot{w} \end{bmatrix} \|_2.\]

**Theorem 5.2.** Suppose that the fine and corrected coarse operators satisfy Hypothesis (29) and (30). Then,

\[(31) \quad \max_{j \leq N} \mathcal{E}_j^k \leq \kappa \frac{(1 + \epsilon_\theta)^N - 1}{\epsilon_\theta} \max_{j \leq N} \mathcal{E}_j^{k-1}.\]

**Proof.** In the following expansion of the parareal iteration, the superscript \(k\) in \(\theta^k\) are dropped for brevity

\[
\begin{bmatrix} \hat{u}_{n+1}^k \\ \hat{u}_n^k \end{bmatrix} = \theta CR \begin{bmatrix} \hat{u}_n^k \\ \hat{u}_n^k \end{bmatrix} + F \begin{bmatrix} u_{n-1}^k \\ \dot{u}_{n-1}^k \end{bmatrix} - \theta CR \begin{bmatrix} u_{n-1}^k \\ \dot{u}_{n-1}^k \end{bmatrix} \\
= \theta CR \left( \theta CR \begin{bmatrix} u_{n-1}^k \\ \dot{u}_{n-1}^k \end{bmatrix} + F \begin{bmatrix} u_{n-1}^k \\ \dot{u}_{n-1}^k \end{bmatrix} - \theta CR \begin{bmatrix} u_{n-1}^k \\ \dot{u}_{n-1}^k \end{bmatrix} \right) \\
+ F \begin{bmatrix} u_{n-1}^k \\ \dot{u}_{n-1}^k \end{bmatrix} - \theta CR \begin{bmatrix} u_{n-1}^k \\ \dot{u}_{n-1}^k \end{bmatrix} \\
= (\theta CR)^n \begin{bmatrix} u_0 \\ \dot{u}_0 \end{bmatrix} + (\theta CR)^n \left( F \begin{bmatrix} u_0 \\ \dot{u}_0 \end{bmatrix} - \theta CR \begin{bmatrix} u_0 \\ \dot{u}_0 \end{bmatrix} \right) \\
+ (\theta CR)^{n-1} \left( F \begin{bmatrix} \hat{u}_1^{k-1} \\ \dot{\hat{u}}_1^{k-1} \end{bmatrix} - \theta CR \begin{bmatrix} \hat{u}_1^{k-1} \\ \dot{\hat{u}}_1^{k-1} \end{bmatrix} \right) \ldots \\
+ (\theta CR) \left( F \begin{bmatrix} u_{n-1}^k \\ \dot{u}_{n-1}^k \end{bmatrix} - \theta CR \begin{bmatrix} u_{n-1}^k \\ \dot{u}_{n-1}^k \end{bmatrix} + (F - \theta CR) \begin{bmatrix} u_{n-1}^k \\ \dot{u}_{n-1}^k \end{bmatrix} \right). \]

It can be verified that the serial fine solution \([u(t_{n+1}), \dot{u}(t_{n+1})]\) also satisfies above expression when superscript \(k, k - 1\) are dropped in solution vector \([u^k, \dot{u}^k]\). Then we have an expression for the difference of the solutions

\[(32) \quad \begin{bmatrix} u_{n+1}^k - u(t_{n+1}) \\ \dot{u}_{n+1}^k - \dot{u}(t_{n+1}) \end{bmatrix} = (\theta CR)^{n-1} (F - \theta CR) \begin{bmatrix} u_1^k - u(t_1) \\ \dot{u}_1^k - \dot{u}(t_1) \end{bmatrix} + \ldots \]

\[(\theta CR)(F - \theta CR) \begin{bmatrix} u_{n-1}^k - u(t_{n-1}) \\ \dot{u}_{n-1}^k - \dot{u}(t_{n-1}) \end{bmatrix} + (F - \theta CR) \begin{bmatrix} u_{n-1}^k - u(t_{n-1}) \\ \dot{u}_{n-1}^k - \dot{u}(t_{n-1}) \end{bmatrix}. \]
Recall the square root of energy error is defined as

$$E_{n+1}^k = \| \Lambda \left[ u_{n+1}^k - u(t_{n+1}) \atop \dot{u}_{n+1}^k - \dot{u}(t_{n+1}) \right] \|_2.$$  

Using triangle inequality on $E_{n+1}^k$ with equation (32) we obtain

$$E_{n+1}^k \leq \| \Lambda(\theta CR)^N \cdot (F - \theta CR) \left[ u_{k-1}^1 - u(t_1) \atop \dot{u}_{k-1}^1 - \dot{u}(t_1) \right] \|_2 \ldots$$

$$+ \| \Lambda(\theta CR)(F - \theta CR) \left[ u_{k-1}^1 - u(t_{n-1}) \atop \dot{u}_{k-1}^1 - \dot{u}(t_{n-1}) \right] \|_2$$

$$+ \| \Lambda(F - \theta CR) \left[ u_{k-1}^1 - u(t_n) \atop \dot{u}_{k-1}^1 - \dot{u}(t_n) \right] \|_2.$$  

Apply equation (29) in Hypothesis 5.1 (i) to bound each term

$$E_{n+1}^k \leq (1 + \epsilon_\theta)^{n-1} \| \Lambda(F - \theta CR) \left[ u_{k-1}^1 - u(t_1) \atop \dot{u}_{k-1}^1 - \dot{u}(t_1) \right] \|_2 \ldots$$

$$+ (1 + \epsilon_\theta) \| \Lambda(F - \theta CR) \left[ u_{k-1}^1 - u(t_{n-1}) \atop \dot{u}_{k-1}^1 - \dot{u}(t_{n-1}) \right] \|_2$$

$$+ \| \Lambda(F - \theta CR) \left[ u_{k-1}^1 - u(t_n) \atop \dot{u}_{k-1}^1 - \dot{u}(t_n) \right] \|_2.$$  

Finally we use equation (30) in Hypothesis 5.1 (ii) to obtain

$$E_{n+1}^k \leq (1 + \epsilon_\theta)^{N-1} \| \Lambda \left[ u_{n-1}^k - u(t_{n-1}) \atop \dot{u}_{n-1}^k - \dot{u}(t_{n-1}) \right] \|_2 \ldots$$

$$+ (1 + \epsilon_\theta) \| \Lambda \left[ u_{n-1}^k - u(t_n) \atop \dot{u}_{n-1}^k - \dot{u}(t_n) \right] \|_2$$

$$+ \| \Lambda \left[ u_{n-1}^k - u(t_n) \atop \dot{u}_{n-1}^k - \dot{u}(t_n) \right] \|_2$$

$$= (1 + \epsilon_\theta)^{n-1} \epsilon_{1}^{k-1} \ldots + (1 + \epsilon_\theta) \epsilon_{n-1}^{k-1} + \epsilon^{k-1}$$

$$\leq \kappa \left( (1 + \epsilon_\theta)^{n-1} \ldots + (1 + \epsilon_\theta) + 1 \right) \max_{j \leq n} E_{j}^{k-1}$$

$$= \kappa \frac{(1 + \epsilon_\theta)^{n} - 1}{\epsilon_\theta} \max_{j \leq n} E_{j}^{k-1}.$$  

Thus

$$\max_{j \leq N} E_{j}^{k} \leq \kappa \frac{(1 + \epsilon_\theta)^{N} - 1}{\epsilon_\theta} \max_{j \leq N} E_{j}^{k-1}.$$
By assumption $\kappa N < 1$ and $\epsilon_\theta N \ll 1$, the error goes to zero as $k$ approaches infinity.

We see that the convergence depends on the Lipschitz constant $\kappa$ in Hypothesis 5.1(ii), which reflects the gap between the corrected coarse propagator to the fine propagator. This gap between propagators is quantified by the energy residual of the minimization (12).

6. Numerical Study of the New Algorithm

In this section, we study the influence of different components of the proposed algorithm to the overall stability and accuracy. To be precise, we consider the influence of (i) varying the low-rank approximation of the optimal phase correctors $\Omega_*$, (ii) different orders of approximation for the gradient, (iii) different interpolation schemes used as the interpolation operator $I$. Regarding to the last item, we will use the following interpolation methods, written as MATLAB functions, in this section:

- interpft: Fourier interpolation
- akima: cubic Hermite interpolation
- pchip: cubic interpolation
- linear: linear interpolation

We shall consider the simplest one dimensional setting with $c \equiv 1$ for both coarse and fine propagator, and the initial data:

$$u(x, 0) = \cos(10\pi x) \exp(-100x^2), \quad x \in [-0.5, 0.5]$$
$$u_t(x, 0) = 0.$$

We will assume that the coarse grid nodes overlap with the fine grid nodes, and that the restriction operator $R$ is just a point-wise evaluation on the coarse grid nodes.

The errors at final time $T_N = T$ are defined as square root of energy of difference on the fine grid

$$\sqrt{\frac{E([u_N^k - u(t_N), \dot{u}_N^k - \dot{u}(t_N)])}{E([u(t_N), \dot{u}(t_N)])}}.$$ 

And similarly the error can also be defined in $\ell^2$ of difference in displacement component

$$\frac{\|u_N^k - u(t_N)\|_2}{\|u(t_N)\|_2}.$$ 

The reference solution $[u(t_N), \dot{u}(t_N)]$ are serially computed using the fine propagators.
6.1. Rank tolerance. In this example, we study the sensitivity of the algorithm to rank-truncation of the optimal phase corrector $\Omega_*$. We use the same spatial grid for both the coarse and the fine propagators in order to avoid error coming from interpolation/restriction. The fine propagator has an CFL number that is 20 times smaller than the coarse, and the coupling take place every 10 coarse steps. We sample several values for tolerance in Algorithm 1 at $10^{-15}, 10^{-12}, 10^{-9}, 10^{-6}, 10^{-3}$. The parameters are tabulated below:

| $T$ | $\Delta t_{com}$ | $\Delta x$ | $\Delta t/\Delta x$ | $\Delta x/\delta x$ | $\Delta t/\delta t$ | $\nabla_h$ | $\text{interpft}$ | 2 order | $\text{tol}$ | $10\{-15,-12,-9,-6,-3\}$ |
|-----|-----------------|-----------|-------------------------|-----------------------|---------------------|---------|----------------|---------|----------|
| 5   | 0.05            | 0.01     | 0.5                     | 1                     | 20                  | interpft | 2 order       |         | $10^{-3}$ |

Figure 3 shows the relative errors along the iterations as the tolerance in the truncation of $\Omega_*$ is varied. The errors decrease in the first few. The rate of decrease seem independent of the set tolerance values. As more iterations progress, the errors convergence eventually stagnate at certain values that strongly correlate to the set tolerance values. Particularly, the stagnated error values scale as the square root of the tolerance as shown on the right plot of Figure 3. This scaling can be explained by the fact that the tolerance corresponds to the truncation of $\Omega_*$, which modifies the wave energy components, and we measure the square root of wave energy difference. Hence in general, the convergence rate of our method is expected to slow down after the error has passed $10^{-8}$ because the tolerance can only be as small as machine epsilon $10^{-16}$. Figure 4 shows the number of retained singular values for different values of tolerance. For this simple example, it is reasonable for the number of singular values to stay the same at every iteration because the iteration is convergent and there is no new mode emerged in the spatially constant medium.
6.2. No phase correction ($\Omega \equiv 1$). Assuming again that the coarse and fine propagators are on the same grid. Without the phase correction, i.e. $\Omega \equiv 1$, the proposed iteration takes the form
\[
\begin{bmatrix}
  u^k_h \\
  \dot{u}^k_h
\end{bmatrix}
= \Lambda^\dagger \Lambda C
\begin{bmatrix}
  u^{k-1}_{n-1} \\
  \dot{u}^{k-1}_{n-1}
\end{bmatrix}
+ \mathcal{F}
\begin{bmatrix}
  u^{k-1}_{n-1} \\
  \dot{u}^{k-1}_{n-1}
\end{bmatrix}
- \Lambda^\dagger \Lambda C
\begin{bmatrix}
  u^{k-1}_{n-1} \\
  \dot{u}^{k-1}_{n-1}
\end{bmatrix}.
\]

The above expression becomes the plain parareal method if the term $\Lambda^\dagger \Lambda = 1$. But when the first wave component $\nabla_h u$ is approximated by some finite differencing, the term $\Lambda^\dagger \Lambda \neq 1$ in general. In particular when $\nabla_h$ is approximated by the standard second order central differencing, i.e. $\nabla_h = D^0_{\Delta x}$, $\Lambda^\dagger \Lambda$ corresponds to multiplication of
\[
\frac{\sin \xi \Delta x}{\xi \Delta x} = \text{sinc}(\xi \Delta x)
\]
to the Fourier mode of the solutions. Since $|\text{sinc}(\xi \Delta x)| \leq 1$, $\Lambda^\dagger \Lambda$ damps high frequency modes, and thus stabilizes parareal-like iterations.

Nevertheless, for long time simulations, such high frequency damping may be not sufficient to stabilize the parareal-like iterations. To illustrate this, we take the same discretization as above but now consider four terminal times $T = 2.5, 5, 10, 50$: 

| $T$ | $\Delta t_{com}$ | $\Delta x$ | $\Delta t / \Delta x$ | $\Delta x / \delta x$ | $\Delta t / \delta t$ | $\mathcal{T}$ | $\nabla_h$ | tol |
|-----|----------------|-----------|---------------------|---------------------|---------------------|-----------|---------|-----|
| *   | 0.05           | 0.01      | 0.5                 | 0.5                 | 1                   | 20 interpft| 2 order FD | $10^{-14}$ |

*: \{2.5, 5, 10, 50\}. Figure 5 presents a comparison of the errors computed with $\Omega \equiv 1$ and with $\Omega = \Omega^k$, for different terminal times. For shorter time intervals, such as $T = 2.5$, the two choices of $\Omega$ yield similar convergence rates until after some iterations when the errors computed with $\Omega^k$ plateau around a much larger value.
Figure 5. Comparison of error convergence at different terminal time. Top row, left: $T = 2.5$, right: $T = 5$. Bottom row, left: $T = 10$, right: $T = 50$. Applying optimized $\Omega^k$ to solution is shown in cross solid curve. No optimization $\Omega = 1$, but fine and coarse solutions are coupled in wave energy components, is shown in diamond dashes.

For larger terminal times, $T = 5, 10, 50$, the instability that comes with using $\Omega \equiv 1$ becomes more and more apparent, while the computations with $\Omega = \Omega^k$ remain stable.

6.3. Influence of gradient approximation. Here we study the influence of the accuracy in approximating the gradient of the wave field in forming the data matrices. We observe from the following examples that higher order approximations of gradient estimation accelerates convergence rate of the proposed method. The parameters used in the simulations are tabulated below:

| $T$ | $\Delta t_{com}$ | $\Delta x$ | $\Delta t/\Delta x$ | $\Delta x/\delta x$ | $\Delta t/\delta t$ | $I$ | $\nabla h$ | tol |
|-----|------------------|-----------|---------------------|---------------------|---------------------|-----|-----------|------|
| 10  | 0.05             | 0.01      | 0.5                 | 1                   | 20                  | interpft | * order  | $10^{-14}$ |
Figure 6. Relative energy errors at $T = 10$, computed with different order of approximations to $\nabla_h U$.

$\ast$: $\{2, 4, 6, 8, \text{spectral}\}$. To isolate other factors that can also influence the convergence rate, the table below shows the relative residual in Sec 3.3 averaged over all iterations, denoted as $\left\langle \frac{\|F - \Omega^k G\|_F}{\|F\|_F} \right\rangle_k$. We see that the residual does not change while we increase the order of finite difference. In the last column, the errors in reconstruction of $U$ from its approximated gradient is provided. To be specific, we denote operation in equation (17) as $Y : \nabla_h v \mapsto v$.

$$\begin{array}{|c|c|c|} \hline \text{approx. order of } & \left\langle \frac{\|F - \Omega^k G\|_F}{\|F\|_F} \right\rangle_k & \max_{i,k} \frac{\|Y \nabla_h U^k - U^k_i\|_2}{\|U^k_i\|_2} \\ \hline 2 & 3.8634 \cdot 10^{-3} & 2.9435 \cdot 10^{-2} \\ 4 & 3.8101 \cdot 10^{-3} & 1.4023 \cdot 10^{-3} \\ 6 & 3.8079 \cdot 10^{-3} & 8.2100 \cdot 10^{-5} \\ 8 & 3.8077 \cdot 10^{-3} & 5.8569 \cdot 10^{-6} \\ \text{spectral} & 3.8077 \cdot 10^{-3} & 1.7706 \cdot 10^{-12} \\ \hline \end{array}$$

Figure 6 shows the convergence of errors for different central differencing and Fourier approximations for $\nabla_h$. The ones with second order approximation has the slowest convergence rate, while those using sixth order or higher converge faster.

6.4. The residual of the minimization problem. Here we study the influence of the residual of the minimization problem. Let us fix the setup as before

| $T$ | $\Delta t_{com}$ | $\Delta x$ | $\Delta t/\Delta x$ | $\Delta x/\delta x$ | $\Delta t/\delta t$ | $\mathcal{I}$ | $\nabla_h$ | $\text{tol}$ |
|-----|------------------|------------|----------------------|----------------------|----------------------|--------|-------|--------|
| 10  | 0.05             | 0.01       | 0.5                  | 1                    | 20                   | interpft | 2 order | $10^{-14}$ |
Figure 7. Error convergence as number of singular values removed in $\Omega_*$. Full singular value is shown in cross solid line while removing first 1, 3, 5 singular value are shown in circle, square, diamond respectively.

To increase the residual, we remove few columns in the left and right singular matrices of $M$. The table below shows the averaged relative residual increases as more number of first few columns are removed. Additionally to verify gradient estimation error does not change, the table also includes the relative error of reconstructing estimated gradient

| singular values removed | $\frac{\langle \| F - \Omega G \|_F \| F \|_F \rangle}{k}$ | $\max_{i,k} \frac{\| Y \nabla h U^k_i - U^k_i \|_2}{\| U^k_i \|_2}$ |
|------------------------|-------------------------------------------------|--------------------------------------------------|
| 0                      | $3.8634 \cdot 10^{-3}$                         | $2.9435 \cdot 10^{-2}$                         |
| 1                      | $3.2295 \cdot 10^{-1}$                         | $2.9435 \cdot 10^{-2}$                         |
| 3                      | $5.4013 \cdot 10^{-1}$                         | $2.9546 \cdot 10^{-2}$                         |
| 5                      | $7.8960 \cdot 10^{-1}$                         | $7.3908 \cdot 10^{-2}$                         |

Figure 7 shows the error convergence as number of singular values removed, which corresponding residual is presented in above table. We conclude that smaller residual leads to faster and more stable convergence.

6.5. The effect of parareal-like corrections. If the parareal-style additive correction is omitted, solution is propagated with just the phase corrected coarse propagator:

$$\begin{bmatrix}
    u^k_n \\
    \dot{u}^k_n
\end{bmatrix} = \theta^{k-1} C \begin{bmatrix}
    \mathcal{R} u^k_n \\
    \mathcal{R} \dot{u}^k_{n-1}
\end{bmatrix}.$$

The simulation parameters are given as follow
We first point out that if $\mathcal{C}$ preserves the discrete wave energy, then the above scheme will also preserve it by construction of $\theta^k$. Figure 8 shows the errors comparing to the serial fine solution. At iteration $k = 1$, the solution is serially computed with the coarse propagator $\mathcal{C}$. At iteration $k = 1$, a phase corrector $\theta^2$ is constructed based on the data computed in $k = 1$. The solution at $k = 2$ is serially computed with $\theta \mathcal{C}$. On the right subplot of Figure 8, we see that the coarse solution now has the same phase as the fine solution, but has a slightly different amplitude. For iteration after $k = 3$, however, the error does not decrease further since the parareal-style additive correction has been omitted. Comparing to the examples with similar simulation parameters presented in the previous subsection, we see that the parareal-style correction

$$\mathcal{F}[\dot{u}_{n-1}, \dot{u}_{n-1}] - \theta^k \mathcal{C}[\mathcal{R}u_{n-1}^k, \mathcal{R}\dot{u}_{n-1}^k]$$

is important, as it adds the missing amplitudes back to improve accuracy (when the solutions are properly aligned).

6.6. **Influence of interpolation.** So far in this section, we have only considered examples in which the coarse and fine propagators operate on the same spatial grid. When these propagators are on two different grids, interpolation is needed to couple the solutions. In this subsection, we study the effect of interpolation. To illustrate this point, take coarse/fine grid ratio to be 2 and keep the discretization as before.
Figure 9. Error convergence of the proposed method using different interpolation schemes.

\[
\begin{array}{cccccccc}
T & \Delta t_{com} & \Delta x & \Delta t/\Delta x & \Delta x/\delta x & \Delta t/\delta t & T & \nabla_h & \text{tol} \\
10 & 0.05 & 0.01 & 0.5 & 10 & 200 & * & 4 \text{ order} & 10^{-14}
\end{array}
\]

*: \{interpft, akima, pchip, linear\}. Input wave speed for coarse propagator is \( c = 1 \) as well. Figure 9 shows the error convergence with different methods for grid interpolation. We observe that spectral interpolation interpft performs the best comparing to the other methods because it resolves the initial wave form much better.

7. Numerical Examples

In this section, we shall consider one and two dimension examples, including an example that involve a large scale wave speed model commonly used in the seismic migration community. When the spatial grid of coarse and fine are different, wave speed field on coarse grid is point wise evaluation of the given wave speed field.

7.1. One dimensional examples. Consider a medium with the wave speed
\[
c(x) = 1 + 0.25 \cos(4\pi x),
\]
and the initial wave field in \([-0.5, 0.5]\)
\[
u(x, 0) = \cos(10\pi x) \exp(-100x^2),
\]
\[
u_t(x, 0) = 0.
\]

We present a numerical simulation using the parameters listed below:
Figure 10. Relative error of the solutions computed in numerical example Sec. 7.1. Left: the energy error, right: the $l^2$ error. Our method, shown in cross solid line, generalizes beyond constant wave speed while the plain parareal method, shown in circle dash, diverges right away.

| $T$ | $\Delta t_{com}$ | $\Delta x$ | $\Delta t/\Delta x$ | $\Delta x/\delta x$ | $\Delta t/\delta t$ | interpft | $\nabla_h$ | $\text{tol}$ |
|-----|------------------|------------|---------------------|---------------------|---------------------|-----------|---------|----------|
| 10  | 0.05             | 0.01       | 0.5                 | 10                  | 100                 | interpft  | 4 order | $10^{-14}$ |

The fine propagator operate on a spatial grid which is 10 times finer than the coarse grid, and uses a CFL which is 10 times smaller. Figure 10 shows convergence of the proposed method comparing to the plain parareal. Because fine and coarse solution in a variable medium may differ a lot, the plain parareal method becomes even more unstable.

7.2. Two dimensional cases. We apply the proposed method to three types of media: one with a smoothly varying wave speed (wave guide), one containing a piecewise constant wave speed (inclusion), and a more complicated wave speed profile which is often used in exploration seismology as a standard case study (Marmousi).

7.2.1. Waveguide. We consider a wave guide in $xy$-plane $[-1, 1] \times [-0.5, 0.5]$ with the wave speed

$$c(x, y) = 1 - 0.3 \cos(2\pi y).$$

The initial data is a plane wave traveling left to right along the $x$-axis:

$$u(x, y; 0) = \exp(-50(x + 0.5)^2),$$

$$u_t(x, y; 0) = 100 \exp(-50(x + 0.5)^2).$$

The parameters used in the simulation are set as follow
Figure 11. Relative error of parareal iterated solutions for the waveguide example in Sec 7.2.1. Left: the energy error, right: the $\ell^2$ error.

Figure 12. Relative error over time for the inclusion example described in Sec 7.2.2. The initial plane wave “hits” the small inclusion at around $T = n\Delta t_{\text{com}}$. Left: the energy error for $\Delta x/\delta x = 1$, right: the energy error for $\Delta x/\delta x = 5$.

| $T$ | $\Delta t_{\text{com}}$ | $\Delta x$ | $\Delta t/\Delta x$ | $\Delta x/\delta x$ | $\lambda_\Delta/\lambda_\delta$ | $T$ | $\nabla_h$ | $\text{interpft}$ | $\text{order}$ | tol |
|-----|---------------------|-----------|----------------------|----------------------|------------------------------|-----|---------|------------------|--------------|-----|
| 5   | 0.05                | 0.005     | 1/4                  | $\{1, 5, 10\}$       | 5                            | $\text{interpft}$ | 4 order | $10^{-13}$       |              |     |

Figure 11 shows error of the solution with different coarse fine grid ratio.

7.2.2. Inclusion. In this example, we consider the two dimensional domain in the $xy$-plane $[-1,1] \times [-0.5, 0.5]$ where a plane wave encounters an inclusion of radius $\sqrt{0.002}$ centered at $[0.5, 0.1]$, modeled by the wave speed

$$c(x, y) = 1 - 0.9 \chi((x-0.5)^2 + (y+0.1)^2 < 0.002).$$
Figure 13. Relative density error of solution of inclusion example for $\Delta x/\delta x = 5$ in Fourier modes $\frac{|\text{fft2}\{u_k^N(\xi_j) - u(\xi_j, t_N)\}|}{\sum_j |\text{fft2}\{u(\xi_j, t_N)\}|}$. The rectangular box indicates the Fourier domain of the coarse spatial grid. Top row, left: error at $n = 15$, right: error at $n = 40$. Bottom row, left: error at $n = 140$, right: error at $n = 200$.

We used the initial data traveling from left to right:

$u(x, y; 0) = \cos(4\pi(x + 0.5)) \exp(-50(x + 0.5)^2)$,

$u_t(x, y; 0) = \left(-4\pi \sin(4\pi(x + 0.5)) + 100(x + 0.5) \cos(4\pi(x + 0.5))\right) \exp(-50(x + 0.5)^2)$,

and discretization parameters

| $T$ | $\Delta t_{com}$ | $\Delta x$ | $\Delta t/\Delta x$ | $\Delta x/\delta x$ | $\lambda_\Delta/\lambda_\delta$ | $I$ | $\nabla_h$ | $\text{tol}$ |
|-----|------------------|-------------|----------------------|----------------------|------------------|-----|-------------|-----------|
| $4$ | $0.02$           | $0.005$     | $1/2$                | $\{1, 5\}$          | $5$              | interpft | $4$ order   | $10^{-13}$ |

When the coarse grid is the same as fine grid, the iterations converge to the serial fine solution for the whole time interval (shown in left subplot of Figure 12). On the
other hand, when coarse/fine grid ratio is 5, the right subplot of Figure 12 shows that the error escalates quickly at \( n = 50 \) (or \( t = 1 \)), when the initial plane wave hits the inclusion for the first time, and again at \( n = 150 \) (or \( t = 2 \)), as some parts of the initial plane wave wraps around the domain the interact with the inclusion again. The error does not decreasing for later iterations.

Figure 13 shows the relative density error in the Fourier modes of the computed solution at different times. For the short time range \( n = 15 \) (before the wave energy is scattered by the inclusion), most of the error concentrates at low frequencies which the coarse grid is able to resolve. Once the wave touches the inclusion at \( n = 40 \) and thereafter \( n = 140, n = 200 \), the errors in the higher frequencies becomes significant. These scattered higher frequency wave is not resolved by the coarse grid and cannot be corrected by the proposed method.

7.2.3. Marmousi experiment. We test our method with the Marmousi wave speed model [8], as shown in Figure 14. The fine scale domain has \( 2422 \times 7367 \) grid points while the coarse scale has \( 49 \times 147 \) grid points or 50 times smaller in each dimension. The initial data is a pulse waveform centered at \( x_0 = (400m, 3880m) \), where \( m \) denotes the length unit in meter,

\[
\begin{align*}
    u(x, y; 0) &= \cos(0.01(x - x_0)) \exp(-1.6 \cdot 10^{-5}((x - x_0)^2)), \\
    u_t(x, y; 0) &= 0.
\end{align*}
\]

The discretization parameters are in the following table where coarse and fine computation communicate every 500 coarse time steps.
Figure 15. Solution at $T = 2(s)$ of Marmousi example in Sec 7.2.3. Top row, left: serial fine solution, right: serial coarse solution. Bottom row, left: $k = 2$ iterated solution, right: $k = 4$ iterated solution.

Figure 16. Relative errors by the proposed method applied to simulate wave propagation in the Marmousi model. Left: the energy error, right: the $\ell^2$ error.
Figure 17. Absolute and energy of absolute error field at $T = 2(s)$ when fine and coarse propagators run on the same grid for Marmousi example in Sec 7.2.3. Top row, left: absolute error of wavefield for $k = 1$ solution, right: absolute error of wavefield for $k = 7$ solution. Bottom row, left: energy error for $k = 1$ solution, right: energy error for $k = 7$ solution.

The computation is executed on one node consisting of 20 cores on the Stampede2 system at Texas Advanced Computing Center (TACC) \footnote{www.tacc.utexas.edu}. With our non-optimized MATLAB code, it took 26 hours to run 6 parareal iterations and 12 hours to compute the serial fine solution. Hence each iteration takes about 4 hours, almost 3 times faster on the wall clock than the serial fine computation. And of course the ratio will increase if more cores are used in the computation.
Figure 18. Relative errors when coarse and fine propagators run on the same grid for the Marmousi model. Left: the energy error, right: the $\ell^2$ error.

Figure 15 shows the solutions computed by the proposed method. One observes that some finer details are added back to the computed solution along the iterations. However, Figure 16 reveals that the errors decrease rather slowly after the first few iterations. Indeed, the setup in this experiment is a challenging example of strong scattering due to discontinuities in the wave speed (compared to the previous Example).

It is natural to wonder if the proposed method computes solutions that would converge to the serially computed fine solutions, *when the coarse and fine propagators run on the same spatial grid*. For this purpose, it suffices to consider a smaller version of the Marmousi velocity model, which is defined on $485 \times 1474$ grid points. A different set of discretization parameters are described in the following table.

$$
\begin{array}{|c|c|c|c|c|c|c|c|c|}
\hline
T & \Delta t_{com} & \Delta x & \Delta t/\Delta x & \Delta x/\delta x & \Delta t/\delta t & I & \nabla h & \text{tol} \\
\hline
2(s) & 0.05(s) & 6.245(m) & 3.2026 \cdot 10^{-6} & 1 & 10 & \text{imresize} & 4 \text{ order} & 10^{-10} \\
\hline
\end{array}
$$

Figure 17 shows the absolute error $|u_n^k - u(t_n)|$ and energy error fields at iterations $k = 1$ and $k = 7$. On the left column, we see that the solution at $k = 1$ has larger point-wise absolute error and energy error in regions of high wave speed contrast (e.g. the lower left region in the image domain) than the regions of low wave speed contrast (e.g. upper left region in the image domain). On the right column, however, the solution at $k = 7$ has large patches of point-wise absolute error at regions of low wave speed contrast. These errors contribute to the increase of overall $\ell^2$ error in the initial few iterations shown in the right subplot in Figure 18.
We observe a discrepancy between the two errors curves. The energy error decreases while the $\ell^2$ error increases, particularly in the regions of low wave speed contrast. This discrepancy in regions of low wave speed contrast is likely due to the construction of the phase corrector. At regions of high contrast, when locally scattered wave emerged, the phase corrector is constructed to decrease the error there but because it is a global operator, it also perturbs solution everywhere else that in effect increases the overall error.

8. Summary and conclusion

We present here a new stable parareal-like method for the second order wave equation. The method uses the solutions computed along the iterations to construct linear operators which bridge the energy difference between the coarse and fine propagators. Such operators are referred to as the phase correctors in this paper. We presented an extensive set of numerical studies which aim at revealing the properties of the proposed method. From the experiments, we see that the proposed method works well for constant and smooth wave speeds. For piece-wise smooth wave speeds, the algorithm is stable, but does not seem to produce numerical solutions that converge to the solutions computed by the fine propagators (as the number of iterations increase), when the fine and coarse propagators run on different spatial resolution. This is expected because the higher Fourier modes of the solutions computed by the fine propagator on a finer spatial grid cannot be resolved by coarser grids. This is true even when the initial wave field is resolved by the grid coarse grid. As our simulations reveal, the stagnation of the errors may be caused additionally by a couple of different approximations used in the algorithm. This paper outline these factors for future improvement. In the last two examples involving piece-wise smooth wave speeds with high contrast, we observe that the relative errors are in general much larger than the previous cases. Most likely, this is due to strong local scattering of waves cause by the discontinuities in the wave speeds. Such scatterings cannot be corrected efficiently by the proposed Procrustean approach.

Acknowledgment

The authors are supported partially by NSF grants DMS-1620396 and DMS-1720171. Nguyen is supported by an ICES NIMS fellowship. Part of this research was performed while the second author was visiting the Institute for Pure and Applied Mathematics (IPAM), which is supported by the National Science Foundation (Grant No. DMS-1440415). The authors thanks TACC for providing computing resources. Tsai also thanks National Center for Theoretical Sciences Taiwan for hosting his visits where part of this research was conducted.
References

[1] G. Ariel, S. J. Kim, and R. Tsai. Parareal multiscale methods for highly oscillatory dynamical systems. *SIAM Journal on Scientific Computing*, 38(6):A3540–A3564, 2016.

[2] G. Ariel, H. Nguyen, and R. Tsai. theta-parareal scheme. 2017.

[3] A. Arteaga, D. Ruprecht, and R. Krause. A stencil-based implementation of parareal in the c++ domain specific embedded language stella. *Applied Mathematics and Computation*, 267:727–741, 2015.

[4] G. Bal. *On the Convergence and the Stability of the Parareal Algorithm to Solve Partial Differential Equations*, pages 425–432. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[5] A.-M. Baudron, J.-J. Lautard, Y. Maday, M. K. Riahi, and J. Salomon. Parareal in time 3d numerical solver for the lwr benchmark neutron diffusion transient model. *Journal of Computational Physics*, 279:67–79, 2014.

[6] A. Blouza, L. Boudin, and S. M. Kaber. Parallel in time algorithms with reduction methods for solving chemical kinetics. *Communications in Applied Mathematics and Computational Science*, 5(2):241–263, 2011.

[7] M. Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear algebra and its applications*, 415(1):20–30, 2006.

[8] A. Brougois, M. Bourget, P. Lailly, M. Poulet, P. Ricarte, and R. Versteeg. Marmousi, model and data. In *EAEG workshop-practical aspects of seismic data inversion*, 1990.

[9] E. J. Bylaska, J. Q. Weare, and J. H. Weare. Extending molecular simulation time scales: Parallel in time integrations for high-level quantum chemistry and complex force representations. *The Journal of chemical physics*, 139(7):074114, 2013.

[10] F. Chen, J. S. Hesthaven, and X. Zhu. On the use of reduced basis methods to accelerate and stabilize the parareal method. In *Reduced Order Methods for modeling and computational reduction*, pages 187–214. Springer, 2014.

[11] R. Croce, D. Ruprecht, and R. Krause. Parallel-in-space-and-time simulation of the three-dimensional, unsteady navier-stokes equations for incompressible flow. In *Modeling, Simulation and Optimization of Complex Processes-HPSC 2012*, pages 13–23. Springer, 2014.

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

[13] M. Duarte, M. Massot, and S. Descombes. Parareal operator splitting techniques for multiscale reaction waves: numerical analysis and strategies. *ESAIM: Mathematical Modelling and Numerical Analysis*, 45(5):825–852, 2011.

[14] P. F. Fischer, F. Hecht, and Y. Maday. A parareal in time semi-implicit approximation of the navier-stokes equations. In *Domain decomposition methods in science and engineering*, pages 433–440. Springer, 2005.

[15] M. J. Gander and M. Petcu. Analysis of a Krylov Subspace Enhanced Parareal Algorithm for Linear Problem. *ESAIM: Proc.*, 25:114–129, 2008.

[16] 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.

[17] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[18] J. C. Gower, G. B. Dijksterhuis, et al. *Procrustes problems*, volume 30. Oxford University Press on Demand, 2004.

[19] E. Grave, A. Joulin, and Q. Berthet. Unsupervised alignment of embeddings with wasserstein procrustes. *arXiv preprint arXiv:1805.11222*, 2018.
[20] T. Haut and B. Wingate. An asymptotic parallel-in-time method for highly oscillatory pdes. *SIAM Journal on Scientific Computing*, 36(2):A693–A713, 2014.

[21] A. Kreienbuehl, A. Naegel, D. Ruprecht, R. Speck, G. Wittum, and R. Krause. Numerical simulation of skin transport using parareal. *Computing and visualization in science*, 17(2):99–108, 2015.

[22] J.-L. Lions, Y. Maday, and G. Turinici. A ”parareal” in time discretization of pde’s. *Comptes Rendus de l’Academie des Sciences*, 332:661–668, 2001.

[23] T. Lunet, J. Bodart, S. Gratton, and X. Vasseur. Time-parallel simulation of the decay of homogeneous turbulence using parareal with spatial coarsening. *Computing and Visualization in Science*, 19(1-2):31–44, 2018.

[24] Y. Maday, O. Mula, and M.-K. Riahi. Towards a fully scalable balanced parareal method: Application to neutronics. 2015.

[25] D. Mercerat, L. Guillot, and J.-P. Vilotte. Application of the parareal algorithm for acoustic wave propagation. In *AIP Conference Proceedings*, volume 1168, pages 1521–1524. AIP, 2009.

[26] A. Randles and E. Kaxiras. Parallel in time approximation of the lattice boltzmann method for laminar flows. *Journal of Computational Physics*, 270:577–586, 2014.

[27] A. Randles and E. Kaxiras. A spatio-temporal coupling method to reduce the time-to-solution of cardiovascular simulations. In *Parallel and Distributed Processing Symposium, 2014 IEEE 28th International*, pages 593–602. IEEE, 2014.

[28] D. A. Ross, D. Tarlow, and R. S. Zemel. Unsupervised learning of skeletons from motion. In *European Conference on Computer Vision*, pages 560–573. Springer, 2008.

[29] D. Ruprecht. Convergence of parareal with spatial coarsening. *PAMM*, 14(1):1031–1034, 2014.

[30] D. Ruprecht. Wave propagation characteristics of parareal. *Comput. Vis. Sci.*, 19(1-2):1–17, June 2018.

[31] D. Ruprecht and R. Krause. Explicit parallel-in-time integration of a linear acoustic-advection system. *Computers & Fluids*, 59:72–83, 2012.

[32] D. Samaddar, D. Coster, X. Bonnin, C. Bergmeister, E. Havlíčková, L. A. Berry, W. R. Elwasif, and D. B. Batchelor. Temporal parallelization of edge plasma simulations using the parareal algorithm and the solps code. *Computer Physics Communications*, 221:19–27, 2017.

[33] D. Samaddar, D. Coster, X. Bonnin, L. Berry, W. Elwasif, and D. Batchelor. Application of the parareal algorithm to simulations of elms in itp plasma. *Computer Physics Communications*, 235:246–257, 2019.

[34] D. Samaddar, D. E. Newman, and R. Sánchez. Parallelization in time of numerical simulations of fully-developed plasma turbulence using the parareal algorithm. *Journal of Computational Physics*, 229(18):6558–6573, 2010.

[35] J. Trindade and J. Pereira. Parallel-in-time simulation of the unsteady navier–stokes equations for incompressible flow. *International journal for numerical methods in fluids*, 45(10):1123–1136, 2004.

[36] C. Wang and S. Mahadevan. Manifold alignment using procrustes analysis. In *Proceedings of the 25th international conference on Machine learning*, pages 1120–1127. ACM, 2008.