Computing and Visualization in Science
https://doi.org/10.1007/s00791-018-0296-z

SPECIAL ISSUE PARALLEL-IN-TIME METHODS

# Wave propagation characteristics of Parareal

**Daniel Ruprecht**[1]

Received: 10 April 2017 / Accepted: 10 November 2017


**Abstract**
The paper derives and analyses the (semi-)discrete dispersion relation of the Parareal parallel-in-time integration method. It investigates Parareal's wave propagation characteristics with the aim to better understand what causes the well documented stability problems for hyperbolic equations. The analysis shows that the instability is caused by convergence of the amplification factor to the exact value from above for medium to high wave numbers. Phase errors in the coarse propagator are identified as the culprit, which suggests that specifically tailored coarse level methods could provide a remedy.

**Keywords** Parareal · Convergence · Dispersion relation · Hyperbolic problem · Advection-dominated problem

## 1 Introduction

Parallel computing has become ubiquitous in science and engineering, but requires suitable numerical algorithms to be efficient. Parallel-in-time integration methods have been identified as a promising direction to increase the level of concurrency in computer simulations that involve the numerical solution of time dependent partial differential equations [5]. A variety of methods has been proposed [8,10,12,22,25], the earliest going back to 1964 [27]. While even complex diffusive problems can be tackled successfully [11,14,24,33]—although parallel efficiencies remain low—hyperbolic or advection-dominated problems have proved to be much harder to parallelise in time. This currently prevents the use of parallel-in-time integration for most problems in computational fluid dynamics, even though many applications struggle with excessive solution times and could benefit greatly from new parallelisation strategies.

For the Parareal parallel-in-time algorithm there is some theory available illustrating its limitations in this respect. Bal shows that Parareal with a sufficiently damping coarse method is unconditionally stable for parabolic problems but not for hyperbolic equations [2]. An early investigation of Parareal's stability properties showed instabilities for imaginary eigenvalues [31]. Gander and Vandewalle [17] give a detailed analysis of Parareal's convergence and show that even for the simple advection equation $u_t + u_x = 0$, Parareal is either unstable or inefficient. Numerical experiments reveal that the instability emerges in the nonlinear case as a degradation of convergence with increasing Reynolds number [32]. Approaches exist to stabilise Parareal for hyperbolic equations [3,4,7,13,15,23,29], but typically with significant overhead, leading to further degradation of efficiency, or limited applicability.

Since a key characteristic of hyperbolic problems is the existence of waves propagating with finite speeds, understanding Parareal's wave propagation characteristics is important to understand and, hopefully, resolve these problems. However, no such analysis exists that gives insight into *how* the instability emerges. A better understanding of the instability could show the way to novel methods that allow the efficient and robust parallel-in-time solution of flows governed by advection. Additionally, just like for "classical" time stepping methods, detailed knowledge of Parareal's theoretical properties for test cases will help understanding its performance for complex test problems where mathematical theory is not available.

To this end, the paper derives a discrete dispersion relation for Parareal to study how plane wave solutions $u(x,t) = \exp(-i\omega t)\exp(i\kappa x)$ are propagated in time. It studies the discrete phase speed and amplification factor and how they depend on e.g. the number of processors, choice of coarse propagator and other parameters. The analysis reveals that the source of the instability is a convergence from above in the amplification factor in higher wave number modes. In diffusive problems, where high wave numbers are naturally

✉ Daniel Ruprecht
d.ruprecht@leeds.ac.uk

[1] School of Mechanical Engineering, University of Leeds, Leeds LS2 9JT, UK

Published online: 29 May 2018

strongly damped, this does not cause any problems, but in hyperbolic problems with little or no diffusion it causes the amplification factors to exceed a value of one and thus triggers instability. Furthermore, the paper identifies phase errors in the coarse propagator as the source of these issues. This suggests that controlling coarse level phase errors could be key to devising efficient parallel-in-time methods for hyperbolic equations.

All results presented in this paper have been produced using `pyParareal`, a simple open-source Python code. It is freely available [28] to maximise the usefulness of the here presented analysis, allowing other researchers to test different equations or to explore sets of parameters which are not analysed in this paper.

## 2 Parareal for linear problems

Parareal [25] is a parallel-in-time integration method for an initial value problem

$$\dot{u}(t) = Au(t), \quad u(0) = u_0 \in \mathbb{C}^n, \quad 0 \leq t \leq T. \tag{1}$$

For the sake of simplicity we consider only the linear case where the right hand side is given by a matrix $A \in \mathbb{C}^{n,n}$. To parallelise integration of (1) in time, Parareal decomposes the time interval $[0, T]$ into $P$ time slices

$$[0, T] = [0, T_1) \cup [T_1, T_2) \cup \cdots \cup [T_{P-1}, T], \tag{2}$$

with $P$ indicating the number of processors. Denote as $\mathcal{F}_{\delta t}$ and $\mathcal{G}_{\Delta t}$ two "classical" time integration methods with time steps of length $\delta t$ and $\Delta t$ (e.g. Runge–Kutta methods). For the sake of simplicity, assume that all slice $[T_{j-1}, T_j)$ have the same length $\Delta T$ and that this length is an integer multiple of both $\delta t$ and $\Delta t$ so that $\Delta T = N_c \Delta t$ and $\Delta T = N_f \delta t$. Below, $\delta t$ will always denote the time step size of the fine method and $\Delta t$ the time step size of the coarse method, so that we omit the indices and just write $\mathcal{G}$ and $\mathcal{F}$ to avoid clutter. Standard serial time marching using the method denoted as $\mathcal{F}$ would correspond to evaluating

$$u_p = \mathcal{F}(u_{p-1}), \quad p = 1, \ldots, P, \tag{3}$$

where $u_p \approx u(T_p)$. Instead, after an initialisation procedure to provide values $u_p^0$—typically running the coarse method once—Parareal computes the iteration

$$u_p^k = \mathcal{G}\left(u_{p-1}^k\right) + \mathcal{F}\left(u_{p-1}^{k-1}\right) - \mathcal{G}\left(u_{p-1}^{k-1}\right), \quad p = 1, \ldots, P \tag{4}$$

for $k = 1, \ldots, K$ where the computationally expensive evaluation of the fine method can be parallelised over $P$ processors. If the number of iterations $K$ is small enough and the coarse method much cheaper than the fine, iteration (4) can run in less wall clock time than serially computing (3).

### 2.1 The Parareal iteration in matrix form

As a first step toward deriving Parareal's dispersion relation we will need to derive its stability function which will require writing it in matrix form. Consider now the case where both the coarse and the fine integrator are one-step methods with stability functions $R_f$ and $R_c$. Then, $\mathcal{G}$ and $\mathcal{F}$ can be expressed as matrices

$$u_p = \mathcal{F}(u_{p-1}) = Fu_{p-1}, \quad u_p = \mathcal{G}(u_{p-1}) = Gu_{p-1} \tag{5}$$

with $F := \left(R_f(A\delta t)\right)^{N_f}$ and $G := \left(R_c(A\Delta t)\right)^{N_c}$. Denote as $\mathbf{u}^k = (u_0, \ldots, u_P) \in \mathbb{R}^{(P+1)n}$ a vector that contains the approximate solutions at all time points $T_j$, $j = 1, \ldots, P$ and the initial value $u_0$. Simple algebra shows that one step of iteration (4) is equivalent to the block matrix formulation

$$\mathbf{M}_g \mathbf{u}^k = \left(\mathbf{M}_g - \mathbf{M}_f\right) \mathbf{u}^{k-1} + \mathbf{b} \tag{6}$$

with matrices

$$\mathbf{M}_f := \begin{bmatrix} I & & & \\ -F & I & & \\ & \ddots & \ddots & \\ & & -F & I \end{bmatrix} \in \mathbb{R}^{(P+1)n,(P+1)n} \tag{7}$$

and

$$\mathbf{M}_g := \begin{bmatrix} I & & & \\ -G & I & & \\ & \ddots & \ddots & \\ & & -G & I \end{bmatrix} \in \mathbb{R}^{(P+1)n,(P+1)n} \tag{8}$$

and a vector $\mathbf{b} = (u_0, 0, \ldots, 0) \in \mathbb{R}^{(P+1)n}$. Formulation (6) interpretes Parareal as a preconditioned linear iteration [1].

### 2.2 Stability function of Parareal

From the matrix formulation of a single Parareal iteration (6), we can now derive its stability function, that is we can express the update from the initial value $u_0$ to an approximation $u_P$ at time $T = T_P$ using Parareal with $K$ iterations as multiplication by a single matrix. The fine propagator solution satisfies

$$\mathbf{M}_f \mathbf{u}_f = \mathbf{b} \tag{9}$$

and is a fixed point of iteration (6). Therefore, propagation of the error

$$\mathbf{e}^k := \mathbf{u}^k - \mathbf{u}_f \tag{10}$$

is governed by the matrix

$$\mathbf{E} := \left(\mathbf{I} - \mathbf{M}_g^{-1}\mathbf{M}_f\right) \tag{11}$$

in the sense that

$$\mathbf{e}^k = \mathbf{E}\mathbf{e}^{k-1}. \tag{12}$$

Using this notation and applying (6) recursively, it is now easy to show that

$$\mathbf{u}^k = \mathbf{E}\mathbf{u}^0 + \sum_{j=0}^{k-1} \mathbf{E}^j \mathbf{M}_g^{-1}\mathbf{b}. \tag{13}$$

If, as is typically done, the start value $\mathbf{u}^0$ for the iteration is produced by a single run of the coarse method, that is if

$$\mathbf{M}_g\mathbf{u}^0 = \mathbf{b}, \tag{14}$$

Equation (13) further simplifies to

$$\mathbf{u}^k = \left(\sum_{j=0}^{k} \mathbf{E}^j\right) \mathbf{M}_g^{-1}\mathbf{b}. \tag{15}$$

The right hand side vector can be generated from the initial value $u_0$ via

$$\mathbf{b} = \mathbf{C}_1 u_0 \tag{16}$$

by defining

$$\mathbf{C}_1 = [\mathbf{I}; \mathbf{0}] \in \mathbb{R}^{(P+1)n,n}, \quad \mathbf{I} \in \mathbb{R}^{n,n}, \mathbf{0} \in \mathbb{R}^{Pn,n}. \tag{17}$$

Finally, denote as

$$\mathbf{C}_2 = [\mathbf{0}, \mathbf{I}], \quad \mathbf{0} \in \mathbb{R}^{n,Pn}, \ \mathbf{I} \in \mathbb{R}^{n,n}, \tag{18}$$

the matrix that selects the last $n$ entries out of $\mathbf{u}^k$. Now, a full Parareal update from some initial value $u_0$ to an approximation $u_P$ using $K$ iterations can be written compactly as

$$u_P = \mathbf{C}_2 \left(\sum_{j=0}^{K} \mathbf{E}^j\right) \mathbf{M}_g^{-1}\mathbf{C}_1 u_0 =: M_{\text{Parareal}} u_0. \tag{19}$$

The stability matrix $M_{\text{Parareal}} \in \mathbb{R}^{n,n}$ depends on $K$, $T$, $\Delta T$, $P$, $\Delta t$, $\delta t$, $\mathcal{F}$, $\mathcal{G}$ and $A$. Note that Staff and Rønquist derived the stability function for the scalar case using Pascal's tree [31].

### 2.3 Weak scaling versus longer simulation times

There are two different application scenarios for Parareal that we can study when increasing the number of processors $P$. If we fix the final time $T$, increasing $P$ will lead to better resolution since the coarse time step $\Delta t$ cannot be larger than the length of a time slice $\Delta T$—the coarse method has to perform at least one step per time slice. In this scenario, more processors are used to absorb the cost of higher temporal resolution ("weak scaling").

Alternatively, we can use additional processors to compute until a later final time $T$ and this is the scenario investigated here. Consequently, we study here the case where $T$ and $P$ increase together and always assume that $T = P$, that is each time slice has length one and increasing $P$ means parallelising over more time slices covering a longer time interval. Since dispersion properties of numerical methods are typically analysed for a unit interval, this causes some issues that we resolve by "normalising" the Parareal stability function, see Sect. 3.1.

### 2.4 Maximum singular value

The matrix $\mathbf{E}$ defined in (11) determines how quickly Parareal converges. Note that $\mathbf{E}$ is nil-potent with $\mathbf{E}^P = 0$, owing to the fact that after $P$ iterations Parareal will always reproduce the fine solution exactly. Therefore, all eigenvalues of $\mathbf{E}$ are zero and the spectral radius is not useful to analyse convergence. Below, to investigate convergence, we will therefore compute the maximum singular value $\sigma$ of $\mathbf{E}$ instead. Since

$$\sigma = \|\mathbf{E}\|_2,$$

it follows from (12) that

$$\left\|\mathbf{e}^k\right\|_2 \leq \|\mathbf{E}\|_2 \left\|\mathbf{e}^{k-1}\right\|_2 = \sigma \left\|\mathbf{e}^{k-1}\right\|_2 \leq \sigma^k \left\|\mathbf{e}^0\right\|_2 \tag{20}$$

so that if $\sigma < 1$ Parareal will converge monotonically without stalling. In particular, this rules out behaviour as found by Gander and Vandewalle for hyperbolic problems, where the error would first increase substantially over the first $P/2$ iterations before beginning to decrease [16]. However, achieving fast convergence and good efficiency will typically require $\sigma \ll 1$. Note that if the coarse method is used to generate $\mathbf{u}_0$, it follows from (9) and (14) that the initial error is

$$\mathbf{e}^0 = \mathbf{u}^0 - \mathbf{u}_f = \left(\mathbf{M}_g^{-1} - \mathbf{M}_f^{-1}\right)\mathbf{b}. \tag{21}$$

The size of $\sigma$ depends on the accuracy "gap" between the coarse and fine integrator and the wave number. Figure 1 shows $\sigma$ for varying values of $\Delta t$ when backward Euler is used for both coarse and fine method. Clearly, as the coarse

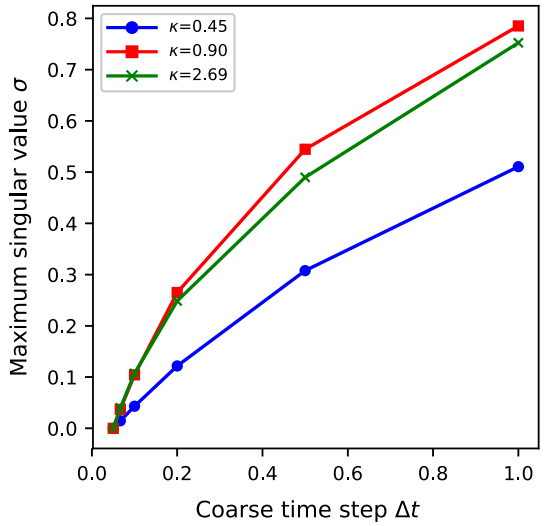


**Fig. 1** Maximum singular value $\sigma$ for decreasing coarse time step $\Delta t$ for $\nu = 0$. Backward Euler is used for both $\mathcal{F}$ and $\mathcal{G}$ and for $\Delta t = \delta t = 0.05$, both methods coincide so that $\sigma = 0$

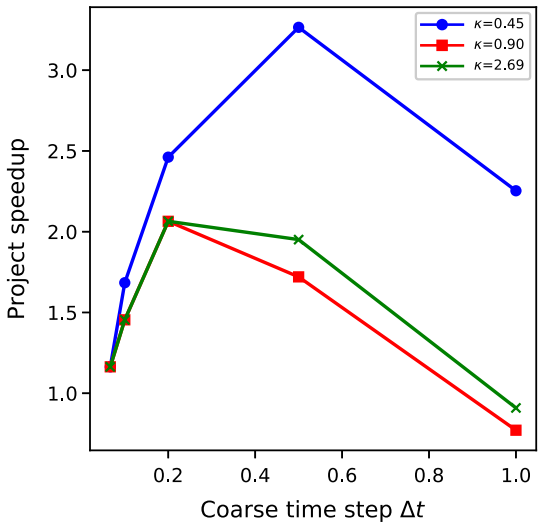


**Fig. 2** Projected speedup for pipelined Parareal [26] with $P = 16$ processors for the same parameters as in Fig. 1 and the number of iterations $k$ fixed such that $\sigma^k \leq$ tol with tol $= 0.01$

time step approaches to fine time step of $\delta t = 0.05$, the maximum singular value approaches zero. However, in this limit the coarse and fine propagator are identical and no speedup is possible. The larger $\Delta t$ compared to $\delta t$, the cheaper the coarse method becomes but since $\sigma$ also grows, more iterations are likely required. Note that higher wave numbers lead to higher values of $\sigma$ while lower wave numbers tend to have values of $\sigma \ll 1$ even for large coarse-to-fine time step ratios.

Looking at $\sigma$ also provides a way to refine performance models for Parareal. Typically, in models projecting speedup, the number of iterations has to be fixed in addition to $\Delta t$, $\delta t$ and $P$. Instead, at least for linear problems, we can fix $k$ such that

$$\sigma^k \leq \text{tol} \tag{22}$$

for some fixed tolerance tol. The resulting projected speedup for $P = 16$ processors and a tolerance of tol $=$ 1e$-$2 is shown in Fig. 2. First, as the coarse time step increases, the reduced cost of the coarse propagator improves achievable speedup. Simultaneously, the decreasing accuracy of $\mathcal{G}$ manifests itself in an increasing number of iterations required to match the selected tolerance. These two counteracting effects create a "sweet spot" where $\mathcal{G}$ is accurate enough to still enable relatively fast convergence but cheap enough to allow for speedup. It is noteworthy, however, that this sweet spot is different for lower and higher wave numbers. Therefore, the potential for speedup from Parareal does not solely depend on the solved equations and discretization parameters but also on the solution—the more prominent high wave number modes are, the more restricted achievable speedup.

## 2.5 Convergence and (in)stability of Parareal

Two different but connected issues with Parareal are discussed throughout the paper, convergence and (in)stability. Here, convergence refers to how fast Parareal approaches the fine solution within a single instance of Parareal, that is

$$M_{\text{Parareal}} \to F \quad \text{as} \quad k \to P. \tag{23}$$

As discussed above, after $k = P$ iterations we always have $M_{\text{Parareal}} = F$, but particularly for hyperbolic problems this can happen only at the final iteration $k = P$ while at $k = P - 1$ there is still a substantial difference [16]. Clearly, speedup is not obtainable in such a situation. The maximum singular value $\sigma$ of $\mathbf{E}$ gives an upper bound or worst-case scenario of how fast Parareal converges to the fine solution, cf. Equation (20). While $\sigma < 1$ does not necessarily guarantee converge quick enough to generate meaningful speedup, it guarantees monotonous convergence and rules out an error that increases first before decreasing only in later iterations.

The other issue investigated in the paper is that of stability of repeated application of Parareal ("restarting"). Below, stability is normally assessed for Parareal with a fixed number of iterations $k$. A configuration of Parareal is referred to as *unstable* if it leads to an amplification factor of more than unity. This corresponds to an artificial increase in wave amplitudes and, just as for classical time stepping methods, would result in a diverging numerical approximation if the method is used recursively

$$(M_{\text{Parareal}})^n \to \infty \quad \text{as} \quad n \to \infty. \tag{24}$$

While for classical methods this recursive application simply means stepping through time steps, for Parareal with $P$ processors it would mean computing one window $[0, T_P]$, then restarting it with the final approximation as initial value for the next window $[T_P, 2T_P]$ and so on.

## 3 Discrete dispersion relation for the advection–diffusion equation

Starting from (19) we now derive the (semi-)discrete dispersion relation of Parareal for the one dimensional linear advection diffusion problem

$$u_t + Uu_x = \nu u_{xx}. \tag{25}$$

First, assume a plane wave solution in space

$$u(x,t) = \hat{u}(t)e^{i\kappa x} \tag{26}$$

with wave number $\kappa$ so that (25) reduces to the initial value problem

$$\hat{u}_t(t) = -\left(Ui\kappa + \nu\kappa^2\right)\hat{u}(t) =: \delta(\kappa, U, \nu)\hat{u}(t) \tag{27}$$

with initial value $\hat{u}(0) = 1$. Integrating (27) from $t = 0$ to $t = T$ in one step gives

$$\hat{u}_T = R(\delta, T)\hat{u}_0 \tag{28}$$

where $R$ is the stability function of the employed method. Now assume that the approximation of $\hat{u}$ is a discrete plane wave so that the solution at the end of the $n$th time slice is given by

$$\hat{u}_n = e^{-i\omega n\Delta T}. \tag{29}$$

Inserting this in (28) gives

$$e^{-i\omega T}\hat{u}_0 = R\hat{u}_0 \Rightarrow \omega = i\frac{\log(R)}{T}. \tag{30}$$

For $R$ in polar coordinates, that is $R = |R|\exp(i\theta)$ with $\theta = \texttt{angle}(R)$, we get

$$\omega = i\left(\log(|R|) + i\theta\right)T^{-1}. \tag{31}$$

The exact integrator, for example, would read

$$R_{\text{exact}} = e^{\delta(\kappa,U,\nu)T}. \tag{32}$$

Using (30) to compute the resulting frequency yields $\omega = i\delta(k, U, \nu)$ and retrieves the continuous plane wave solution

$$u(x,T) = \hat{u}_T e^{i\kappa x} = e^{-\nu\kappa^2 T}e^{i\kappa(x-UT)} \tag{33}$$

of (25). It also reproduces the dispersion relation of the continuous system

$$\omega = i\frac{\log(R)}{T} = i\delta(\kappa, U, \nu) = U\kappa - i\nu\kappa^2. \tag{34}$$

However, if we use an approximate stability function $R$ instead, we get some approximate $\omega = \omega_r + i\omega_i$ with $\omega_r, \omega_i \in \mathbb{R}$. The resulting semi-discrete solution becomes

$$u_j^n = e^{-i\omega t_n}e^{i\kappa x} = e^{\omega_i t_n}e^{i\kappa\left(x-\frac{\omega_r}{\kappa}t_n\right)}. \tag{35}$$

Therefore, $\omega_r/\kappa$ governs the propagation speed of the solution while $\omega_i$ governs the growth or decay in amplitude. Consequently, $\omega_r/\kappa$ is referred to as *phase velocity* while $\exp(\omega_i)$ is called *amplification factor*. In the continuous case, the phase speed is equal to $U$ and the amplification factor equal to $e^{-\nu\kappa^2}$. Note that for (25) the exact phase speed should be independent of the wave number $\kappa$. However, the discrete phase speed of a numerical method often will change with $\kappa$, thus introducing numerical dispersion. Also note that for $\nu > 0$ higher wave numbers decay faster because the amplification factor decreases rapidly as $\kappa$ increases.

### 3.1 Normalisation

The update function R for Parareal in Eq. (19) denotes not an update over [0, 1] but over $[0, T_P]$ where $T_P = P$ is the number of processors. A phase speed of $\omega_r/\kappa = 1.0$, for example, indicates a wave that travels across a whole interval [0, 1] during the step. If scaled up to an interval $[0, P]$, the corresponding phase speed would become $\omega_r/\kappa = P$ instead.

This enlarged range of values causes problems with the complex logarithm in (30). As an example, take the stability function of the exact propagator (32). Analytically, the identity

$$\omega = i\frac{R}{T} = i\delta(\kappa, U, \nu)\frac{T}{T} = i\delta(\kappa, U, \nu) \tag{36}$$

holds, resulting in the correct dispersion relation (34) of the continuous system. However, depending on the values of $\kappa$, $U$, $T$ and $\nu$, this identity is not necessarily satisfied when computing the complex logarithm using `np.log`. For example, for $\kappa = 2$, $U = 1$, $\nu = 0$ and $T > 0$, the exact stability function is $R = e^{-2iT}$. In Python, we obtain for $T = 1$

$$\texttt{1j} * \texttt{np.log}(e^{-2i})/1 = 2 = \kappa \tag{37}$$

but for $T = 2$ we obtain

$$\texttt{1j} * \texttt{np.log}(e^{-4i})/2 \approx -1.1416 \neq \kappa \tag{38}$$

and so identity (36) is not fulfilled. The reason is that the logarithm of a complex number is not unique and so `np.log` returns only *a* complex logarithm but not necessarily the right one in terms of the dispersion relation.

To circumvent this problem, we "normalise" the update $R$ for Parareal to [0, 1]. To this end, decompose

$$R = \sqrt[P]{R} \cdot \ldots \cdot \sqrt[P]{R} \tag{39}$$

where $\sqrt[P]{R}$ corresponds to the propagation over [0, 1] instead of $[0, P]$. Since there are $P$ many roots $\sqrt[P]{R}$, we have to select the right one. First, we use the `zeros` function of `numPy` to find all $P$ complex roots $z_i$ of

$$z^n - R = 0. \tag{40}$$

Then, we select as root $\sqrt[P]{R}$ the value $z_i$ that satisfies

$$\left|\theta(\sqrt[P]{R}) - \theta_{\text{targ}}\right| = \min_{p=1,\ldots,P} \left|\theta(z_p) - \theta_{\text{targ}}\right| \tag{41}$$

where $\theta$ is the `angle` function and $\theta_{\text{targ}}$ some target angle, which we still need to define.

We compute $\omega$ and the resulting phase speed and amplification factor for a range of wave numbers $0 \leq \kappa_1 \leq \kappa_2 \leq \cdots \leq \kappa_N \leq \pi$. For $\kappa_1, \theta_{\text{targ}}$ is set to the angle of the frequency $\omega$ computed from the analytic dispersion relation. After that, $\theta_{\text{targ}}$ is set to the angle of the root selected for the previous value of $\kappa$. The rationale is that small changes in $\kappa$ should only result in small changes of frequency and phase so that $\theta(\omega_{i-1}) \approx \theta(\omega_i)$ if the increment between wave numbers is small enough. From the selected root $\sqrt[P]{R}$ we then compute $\omega$ using (30), the resulting discrete phase speed and amplification factor and the target angle $\theta_{\text{targ}}$ for the next wave number.

## 4 Analysis of the dispersion relation

After showing how to derive Parareal's dispersion relation and normalising it to the unit interval, this section now provides a detailed analysis of different factors influencing its discrete phase speed and amplification factor.

### 4.1 Influence of diffusion

Figure 3 shows the discrete phase speed and amplification factor of Parareal for $P = 16$, backward Euler with $\Delta t = 1.0$ as coarse and the exact integrator as fine propagator. Both levels use $\delta = -\left(Ui\kappa + \nu\kappa^2\right)$, that is the symbol of the continuous spatial operator. The two upper figures are for $U = 1.0$ and $\nu = 0.0$ (no diffusion) while the two lower figures are for $U = 1.0$ and $\nu = 0.1$ (diffusive).

In both cases, the discrete phase speed of Parareal converges almost monotonically toward the continuous phase speed. Even for ten iterations, Parareal still causes significant slowing of medium to large wave number modes. Parareal requires almost the full number of iterations, $k = 15$, before it faithfully reproduces the correct phase speed across most of the spectrum. However, for any number of iterations where speedup might still be possible, Parareal will introduce significant numerical dispersion. Slight artificial acceleration is also observed for high wave number modes for $k = 15$ in the non-diffusive and $k = 10$ in the diffusive case, but generally phase speeds are quite similar in the diffusive and non-diffusive case.

The amplification factor in the non-diffusive case (upper right figure) illustrates Parareal's instability for hyperbolic equations: for $k = 10$ and $k = 15$ it is larger than one for a significant part of the spectrum, indicating that these modes are unstable and will be artificially amplified. For $k = 5$, the iteration has not yet corrected for the strong diffusivity of the coarse propagator and remains stable for all modes but with significant numerical damping of medium to high wave numbers. The reason for the stability problems is discernible from the amplification factor for the diffusive case (lower-right): from $k = 0$ (blue circles) to $k = 5$, Parareal reproduces the correct amplification factor for small wave number modes but significantly overestimates the amplitude of medium to large wave numbers. It then continues to converge to the correct value *from above*. For the diffusive case where the exact values are smaller than one this does not cause instabilities. In the non-diffusive case, however, any overestimation of the analytical amplification factor immediately causes instability. There is, in a sense, "no room" for the amplification factor to converge to the correct value from above. This also means that using a non-diffusive method as coarse propagator, for example trapezoidal rule, leads to disastrous consequences (not shown) where most parts of the spectrum are unstable for almost any value of $k$.

Figure 4 illustrate how the phase speed and amplitude errors discussed above manifest themselves. It shows a single Gauss peak advected with a velocity of $U = 1.0$ with $\nu = 0.0$ on a spatial domain [0, 4] over a time interval [0, 16] distributed over $P = 16$ processors and $N_c = 2$ coarse time steps per slice. A spectral discretisation is used in space, allowing to represent the derivative exactly. For $k = 5$ iterations, most higher wave numbers are damped out and the result looks essentially like a low wave number sine function. The artificially amplified medium to high wave number modes create a "bulge" for $k = 10$ while dispersion leads to a significant trough at the sides of the domain. After fifteen iterations, the solution approximates the main part of

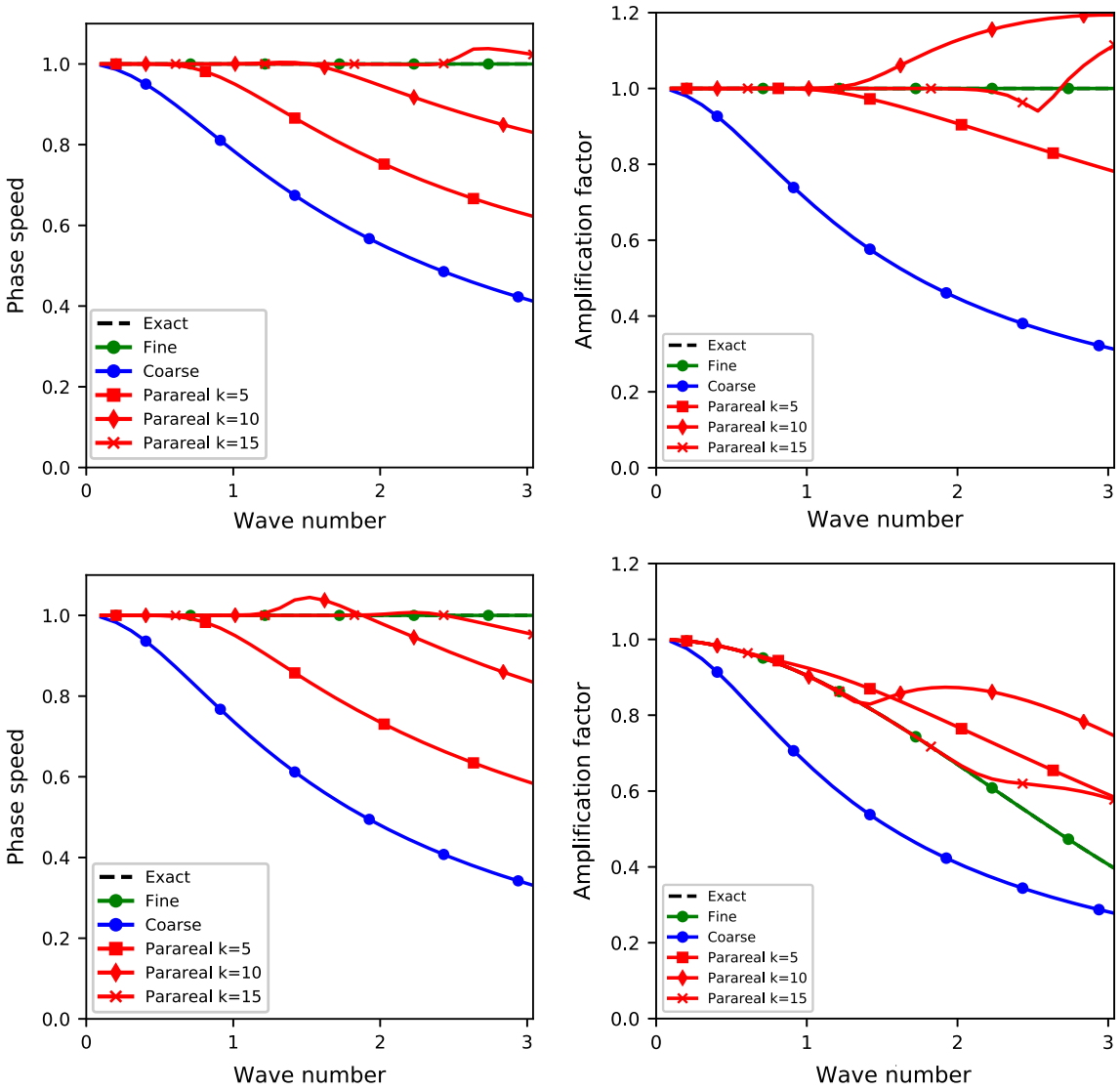


**Fig. 3** Discrete phase speed and amplification factor for Parareal with backward Euler as $\mathcal{G}$ and the exact integrator for $\mathcal{F}$. The symbol for the spatial discretisation is $\delta = -\left(i\kappa + \nu\kappa^2\right)$. The diffusion coefficient is $\nu = 0.0$ (upper) and $\nu = 0.1$ (lower)

the Gauss peak reasonably well, but dispersion still leads to visible errors along the flanks. The right figure shows a part of the resulting spectrum. For $k = 5$, only the lowest wave number modes are present, leading to the sine shaped solution. After $k = 10$ iterations, most of the spectrum is still being truncated but a small range of wave numbers around $\kappa = 0.05$ is being artificially amplified which creates the "bulge" seen in the left figure. Finally, for $k = 15$ iterations, Parareal starts to correctly capture the spectrum but the still significant overestimation of low wave number amplitudes and underestimation of higher modes causes visible errors.

**Observation 1** *The amplification factor in Parareal for higher wave numbers converges "from above". In diffusive problems these wave numbers are damped, so the exact amplification factor is significantly smaller than one, leaving room for Parareal to overestimate it without crossing the threshold to instability. For non-diffusive problems where the exact amplification factor is one, every overestimation causes the mode to become unstable.*

## 4.2 Low order finite difference in coarse method

In a realistic scenario, some approximation of the spatial derivatives would have to be used instead of the exact symbol $\delta$. For simple finite differences, we can study the effect this has on the dispersion relation. Consider the first order upwind finite difference

$$u_x(x_j) \approx \frac{u_j - u_{j-1}}{\Delta x}$$

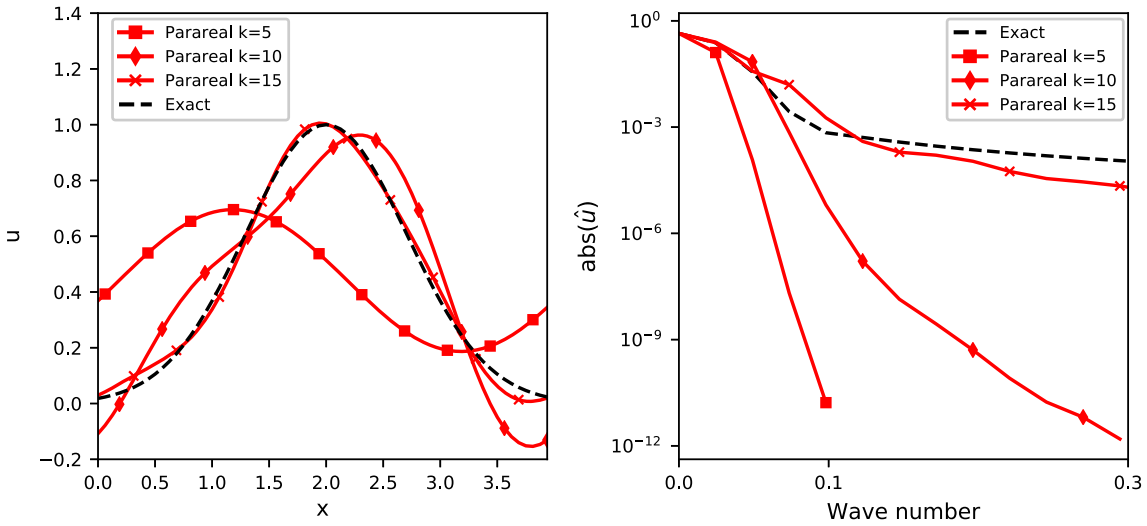


**Fig. 4** Gauss peak in physical space (left) and corresponding spectrum (right) for $U = 1.0$ and $\nu = 0.0$ integrated using a pseudo-spectral method with 64 modes in space and Parareal with $P = 16$ processors in time

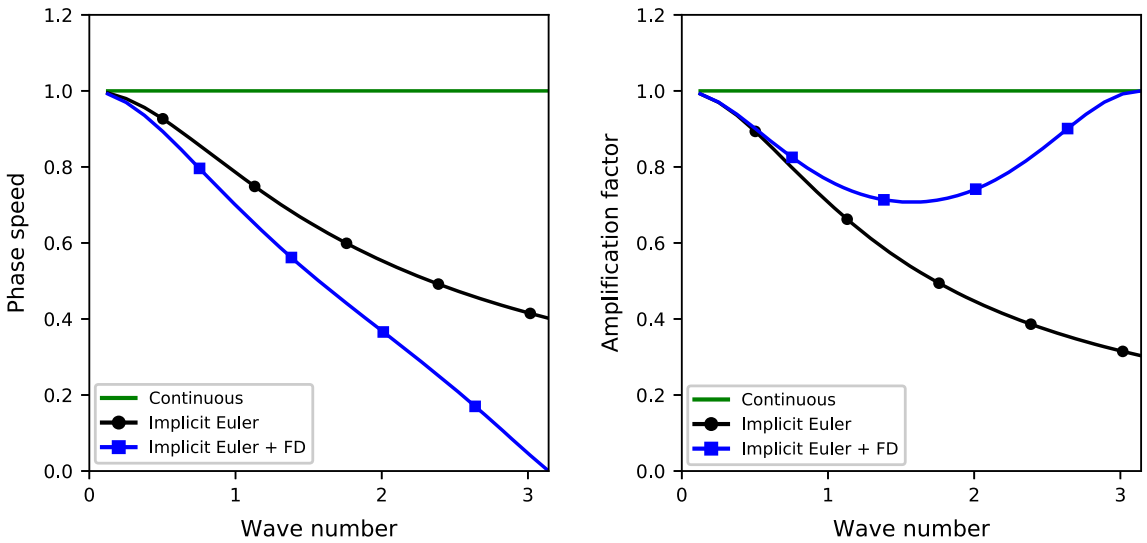


**Fig. 5** Phase speed (left) and amplification factor (right) of the implicit Euler method using the exact symbol $\delta$ (black circles) or the approximate symbol $\tilde{\delta}$ of the second order centred finite difference (blue squares)

as approximation for $u_x$ in (25). Assuming a discrete plane wave

$$u_j = \hat{u}(t) e^{i\kappa j \Delta x}$$

on a uniform spatial mesh $x_j = j\Delta x$ instead of the continuous plane wave (26), this leads to

$$u_x(x_j) \approx \frac{e^{i\kappa x_j} - e^{i\kappa(x_j - \Delta x)}}{\Delta x} = e^{i\kappa x_j} \frac{1 - e^{-i\kappa \Delta x}}{\Delta x}.$$

For $\nu = 0$ this results in the initial value problem

$$\hat{u}_t(t) = -U \frac{1 - e^{-i\kappa \Delta x}}{\Delta x} \hat{u}(t) =: \tilde{\delta}(k, U, \delta x) \hat{u}(t)$$

with initial value $\hat{u}(0) = 1$ and a discrete symbol $\tilde{\delta}$ instead of $\delta$ as in (27). Note that

$$\lim_{\Delta x \to 0} \frac{1 - e^{-i\kappa \Delta x}}{\Delta x} = i\kappa$$

so that $\tilde{\delta} \to \delta$ as $\Delta x \to 0$. Durran gives details for different stencils [6].

The dispersion properties of the implicit Euler method together with the first order upwind finite difference are qualitatively similar to the ones for implicit Euler with the

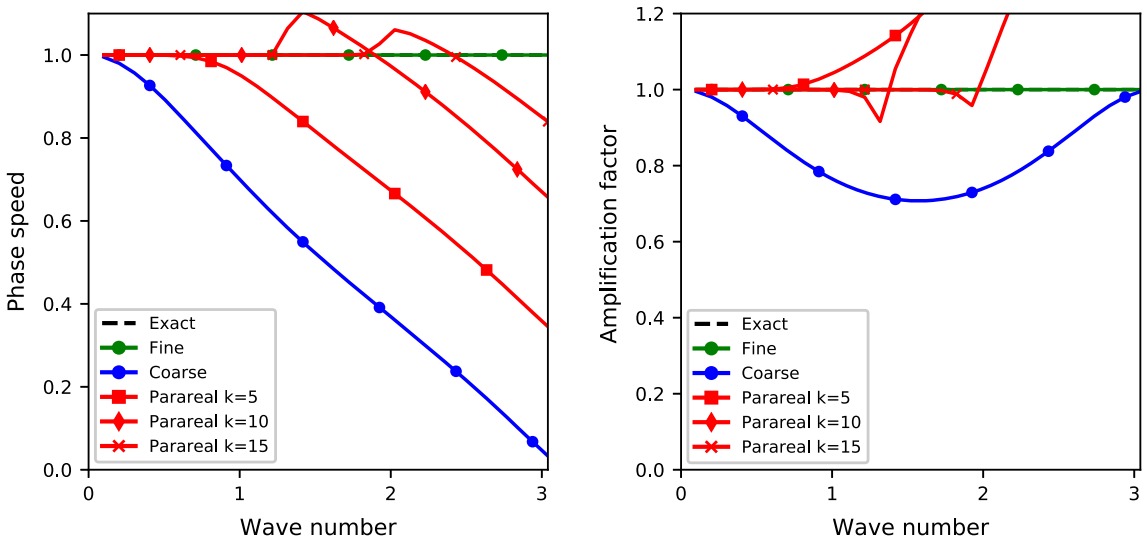


**Fig. 6** Phase speed (left) and amplification factor (right) for same configuration as in Fig. 3 but with a second order centred finite difference in the coarse propagator instead of the exact symbol

exact symbol (not shown).[1] Using the upwind finite difference instead of the exact symbol gives qualitatively similar wave propagation characteristics for the coarse propagator. Numerical slowdown increases (up to the point where modes at the higher end of the spectrum almost do not propagate at all) and numerical diffusion becomes somewhat stronger. As a result, Parareal's dispersion properties (also not shown) are also relatively similar, except for too small phase speeds even for $k = 15$.

However, if we use the centred finite difference

$$u_x(x_j) \approx \frac{u_{j+1} - u_{j-1}}{2\Delta x}$$

instead, this leads to an approximate symbol

$$\tilde{\delta} = -U\frac{e^{i\kappa\Delta x} - e^{-i\kappa\Delta x}}{2\Delta x} = -Ui\frac{\sin(\kappa\Delta x)}{\Delta x}. \tag{42}$$

In this case, it turns out that the dispersion properties of implicit Euler with $\delta$ and $\tilde{\delta}$ are quite different. Figure 5 shows the discrete phase speed (left) and amplification factor (right) for both configurations. For the phase speed, both version agree qualitatively, even though using $\tilde{\delta}$ leads to noticeable stronger slow down, particularly of higher wave numbers. For the amplification factor, however, there is a significant difference between the semi-discrete and fully discrete method. While the former damps high wave numbers strongly, the combination of implicit Euler and centred finite differences strongly damps medium wave numbers while damping of high wave numbers is weak.

In Parareal, this causes a situation similar to what happens when using the trapezoidal rule as coarse propagator, albeit less drastic. Figure 6 shows again the phase speed (left) and amplification factor (right) for the same configuration as before but implicit Euler with centred finite difference for $\mathcal{G}$. The failure of the coarse method to remove high wave number modes again leads to an earlier triggering of the instability. Whereas for Parareal using $\delta$ on the coarse level the iteration $k = 5$ was will stable (see Fig. 3), it is now unstable. For iterations $k = 10$ and $k = 15$, large parts of the spectrum remain unstable. Also, the stronger numerical slow down of the coarse method makes it harder for Parareal to correct for phase speed errors. Where before Parareal with $k = 15$ iteration captured the exact phase speed reasonably well, in Fig. 6 we still see significant numerical slow down of the higher wave number modes.

**Observation 2** *The choice of finite difference stencil used in the coarse propagator can have a significant effect on Parareal. It seems that centred stencils that fail to remove high wave number modes cause similar problems as non-diffusive time stepping methods, suggesting that stencils with upwind-bias are a much better choice.*

### 4.3 Influence of phase and amplitude errors

To investigate whether *phase errors* or *amplitude errors* in the coarse method trigger the instability, we construct coarse propagators where either the phase or the amplitude is exact. Denote as $R_{\text{euler}}$ the stability function of backward Euler and as $R_{\text{exact}}$ the stability function of the exact integrator. Then, a

[1] The script `plot_ieuler_dispersion.py` supplied together with the Python code can be used to visualize the dispersion properties of the coarse propagator alone.

method with the amplification factor of backward Euler and no phase speed error can be constructed as

$$R_1 := |R_{\text{euler}}|\, e^{i\theta(R_{\text{exact}})} \tag{43}$$

while a method with no amplification error and the phase speed of backward Euler can be constructed as

$$R_2 := |R_{\text{exact}}|\, e^{i\theta(R_{\text{euler}})}. \tag{44}$$

These artificially constructed propagators are now used within Parareal.

Figure 7 shows the resulting amplification factor when using $R_1$ (upper) or $R_2$ (lower) as coarse propagator. For $R_1$, where there is no phase speed error in the coarse propagator, there is no instability. Already for $k = 10$ it produces a good approximation of the exact amplification factor across the whole spectrum. In contrast, for $R_2$ where there is no amplification error produced by $\mathcal{G}$, the instability is clearly present for $k = 5$, $k = 10$ and $k = 15$.

Figure 8 shows the solution for the same setup that was used for Fig. 4, except using the $R_1$ artificial coarse propagator without phase errors instead of the backward Euler. For $k = 5$ iterations, the peak is strongly damped but, because $\mathcal{G}$ has no phase errors, in the correct place. After $k = 10$ iterations, Parareal has corrected for most the numerical damping and already provides a reasonable approximation, even though the amplitude of most wave numbers in the spectrum is still severely underestimated. However, the lack of phase errors and resulting numerical dispersion avoids the "bulge" and distortions that were present in Fig. 4. Finally, for $k = 15$ iterations, the solution provided by Parareal is indistinguishable from the exact solution. Small underestimation of the amplitudes of larger wave numbers can still be seen in the spectrum, but the effect is minimal. Note that this does not mean that Parareal will provide speedup—in a realistic scenario, where $\mathcal{F}$ is not exact but a time stepping method, too, it would depend on how many iterations are required for Parareal to be as accurate as the fine method run serially and the actual runtimes of both propagators. All that can be said so far is that avoiding coarse propagator phase errors avoids the instability and leads to faster convergence.

The effect of eliminating phase errors in the coarse method can also be illustrated by analysing the maximum singular value $\sigma$ of the error propagation matrix. Figure 9 shows $\sigma$ depending on the wave number $\kappa$ for three different coarse propagators: the backward Euler, the artificially constructed propagator $R_1$ with no phase error and the artificially constructed propagator $R_2$ with no amplitude error. For the backward Euler method, $\sigma$ is larger than one for significant parts of the spectrum, indicating possible non-monotonous convergence for these modes. The situation is even worse for $R_2$, mirroring the problems with a non-diffusive coarse

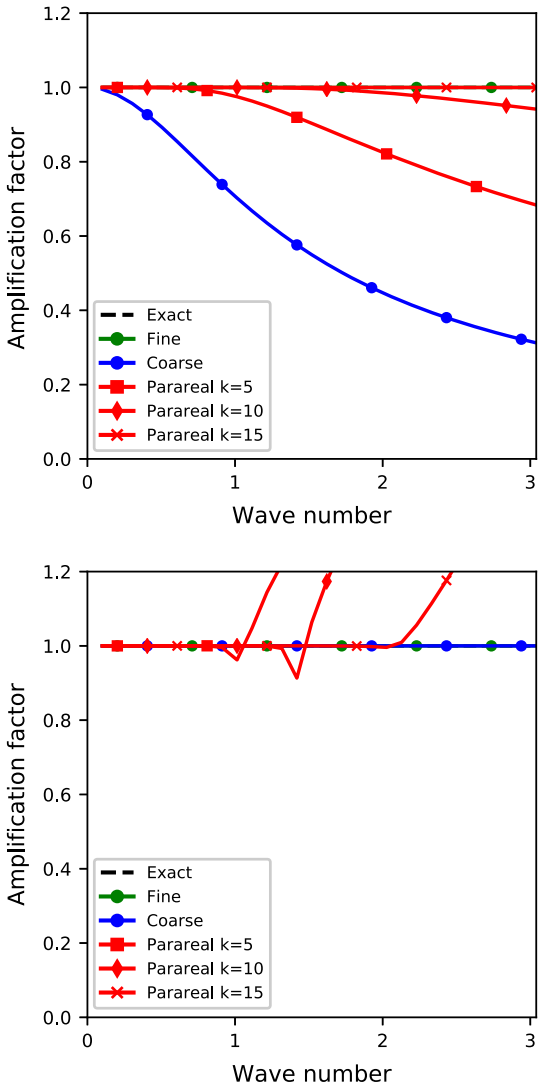


**Fig. 7** Amplification factor of Parareal for the advection equation for an artificially constructed coarse method with exact phase speed (upper) or exact amplification factor (lower)

method like the trapezoidal rule mentioned above. For $R_1$, however, $\sigma$ remains below one across the whole spectrum, so that Parareal will converge monotonically for every mode. Since $\sigma$ approaches one for medium to high wave numbers, convergence there is potentially very slow, in line with the errors seen in the upper part of the spectrum of the Gauss peak. However, in contrast to the other two cases, these wave numbers will not trigger instabilities.

In summary, these results strongly suggest that *phase errors* in the coarse method are responsible for the instability, which is in line with previous findings that Parareal can quickly correct even for very strong numerical diffusion as long as a wave is placed at roughly the correct position by the coarse predictor [29].

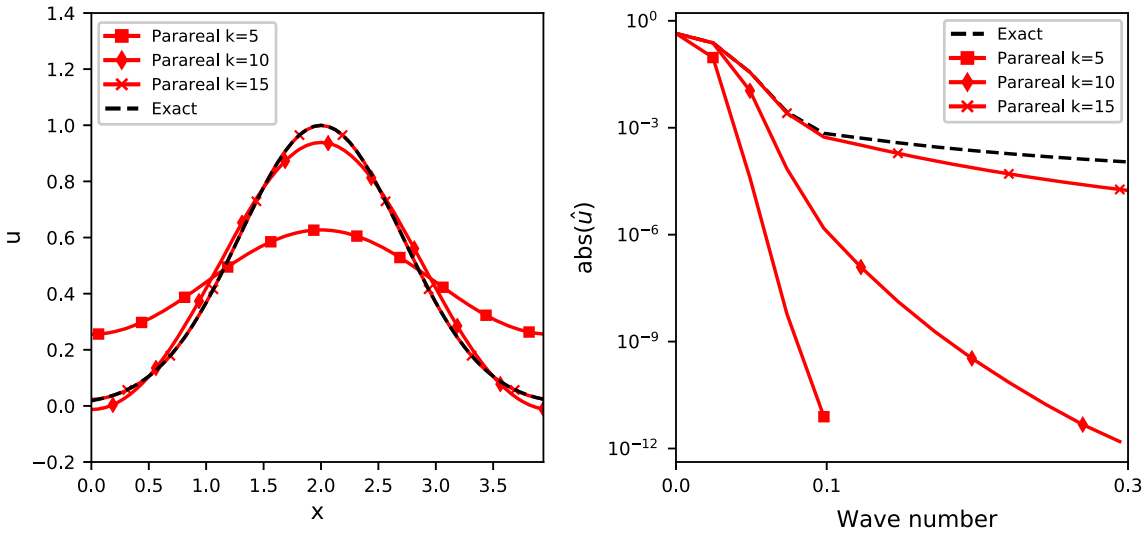


**Fig. 8** The same Gauss peak (left) and corresponding spectrum (right) as in Fig. 4 but now computed with the $R_1$ coarse propagator with exact phase speed

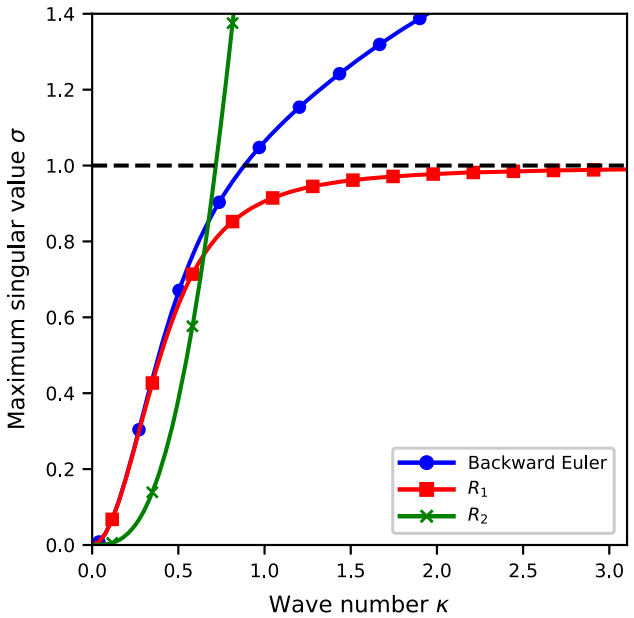


**Fig. 9** Maximum singular value $\sigma$ of error propagation matrix $\mathbf{E}$ depending on the wave number of three choices of coarse propagator. $R_1$ has exact phase speed while $R_2$ has exact amplification factor

**Observation 3** *The instability in Parareal seems to be caused by phase errors in the coarse propagator while amplitude errors are quickly corrected by the iteration.*

#### 4.3.1 Relation to asymptotic Parareal

It is interesting to point out how the $R_1$ propagator with exact phase speed is related to the asymptotic Parareal method developed by Haut et al. [18]. The exact propagator for (25) is given by

$$R_{\text{exact}} = e^{\delta(U,\kappa,\nu)} = e^{-\nu\kappa^2 t} e^{-Ui\kappa}. \tag{45}$$

Therefore, we have

$$|R_{\text{exact}}| = e^{-\nu\kappa^2} \tag{46}$$

and

$$\theta(R_{\text{exact}}) = -U\kappa. \tag{47}$$

Equivalent to the use of a coarse propagator $R_1$ with exact phase propagation would be solving a transformed coarse level problem instead by setting

$$\tilde{u}(t) := e^{Ui\kappa t}\hat{u}(t). \tag{48}$$

This leads to the purely diffusive coarse level problem

$$\tilde{u}_t(t) = -\nu\kappa^2\tilde{u}(t) \tag{49}$$

with restriction operator $e^{Ui\kappa t}$ and interpolation $e^{-Ui\kappa t}$ taking care of the propagation part. This is precisely the strategy pursued in the nonlinear case by "asymptotic Parareal" where they factor out a fast term with purely imaginary eigenvalues, related to acoustic waves. In a sense, their approach can be understood as an attempt to construct a coarse method with minimal phase speed error. Of course, evaluation of the transformation is not trivial for more complex problems and requires a sophisticated approach [30], in contrast to the here studied linear advection–diffusion equation where the transformation is simply multiplication by $e^{-Ui\kappa t}$ and $e^{Ui\kappa t}$.

### 4.3.2 Phase error or mismatch

So far, we have always assumed that the fine method is exact. This leaves the question whether the instability is triggered by phase errors in the coarse method or simply by a mismatch in phase speeds between fine and coarse level. In order to see if the instability arises if both fine and coarse level have the same large phase error, we replace the fine propagator stability function by

$$R_3 = |R_{\text{fine}}|\, e^{i\theta(R_{\text{coarse}})}. \tag{50}$$

Now, the fine propagator is a method with exact amplification factor but a discrete phase speed that is as inaccurate as the coarse method. While such an integrator would not make for a very useful method in practice it is valuable for illustrative purposes. The coarse method is again the standard implicit Euler.

Figure 10 shows the phase speed (upper) and amplification factor (lower) of Parareal for this configuration. Since the fine and coarse method have the same (highly inaccurate) phase speed, Parareal matches the fine method's phase speed exactly from the first iteration and all lines coincide. The amplification factor converges quickly to the correct value and looks almost identical to the case where a coarse propagator with exact phase speed was used, compare for Fig. 7 (upper). No instability occurs and amplification factors are below one across the whole spectrum for all iterations.

Figure 11 shows how Parareal converges for this configuration in physical and spectral space. Because both fine and coarse method now have substantial phase error, the Gauss peak is at a completely wrong position. However, for $k = 10$, Parareal already approximates it reasonably well and shows no sign of instability. Convergence looks again very similar to the results shown in Fig. 8 except for the wrong position of the Gauss peak. While making the fine method as inaccurate as the coarse method is clearly not a useful strategy to stabilise Parareal, this experiment nevertheless demonstrates that the instability is triggered by different discrete phase speeds in the coarse and fine method.

**Observation 4** *Analysing further the issue of phase errors shows that the instability seems to arise from mismatches between the phase speed of coarse and fine propagator.*

It is interesting to note that a very similar observation was made by Ernst and Gander for multi-grid methods for the Helmholtz equation. There, the "coarse grid correction fails because of the incorrect dispersion relation (phase error) on coarser and coarser grids […]" [9]. They find that adjusting the wave number of the coarse level problem in relation to the mesh size leads to rapid convergence of their multi-grid solver. Investigating if and how their approach might be applied to Parareal (which can also be considered as a multi-grid in time method [17]) would be a very interesting direction for future research. Furthermore, it seems possible that parallel-in-time methods with more than two levels like MGRIT [10] or PFASST [8] could yield some improvement, because they would allow for less drastic changes in resolution compared to two-level Parareal.

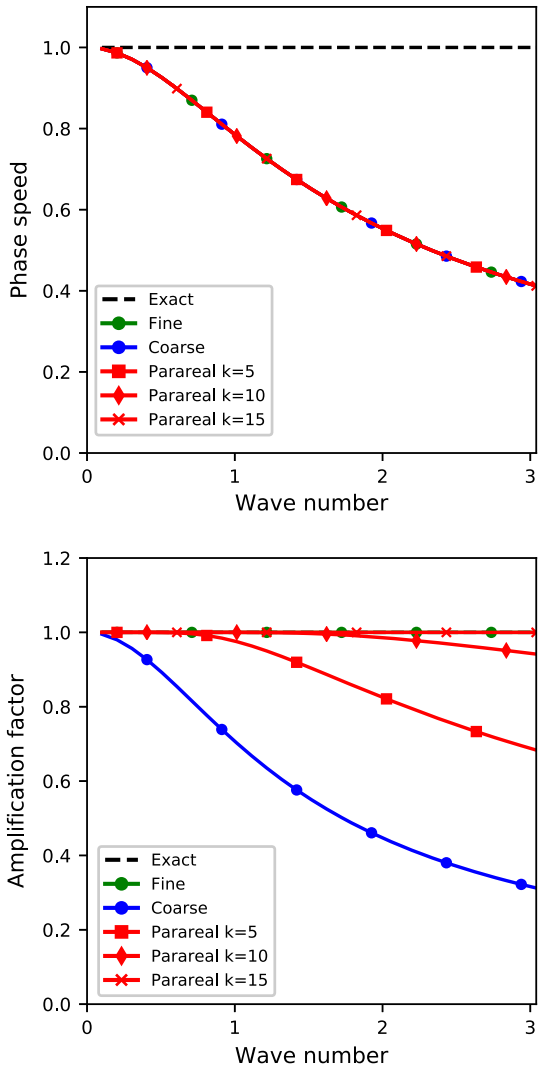


**Fig. 10** Phase speed (upper) and amplification factor (lower) for an artificially constructed fine propagator with the same phase error as the implicit Euler coarse propagator

## 4.4 Coarse time step

Using a smaller time step for the coarse method will obviously reduce its phase error and can thus be expected to benefit Parareal convergence. Figure 12 shows that this is indeed true. It shows discrete phase speed (left) and amplification factor (right) for the same configuration as used for Fig. 3, except now using two coarse step per time slice instead of one. Since the coarse propagator alone is now

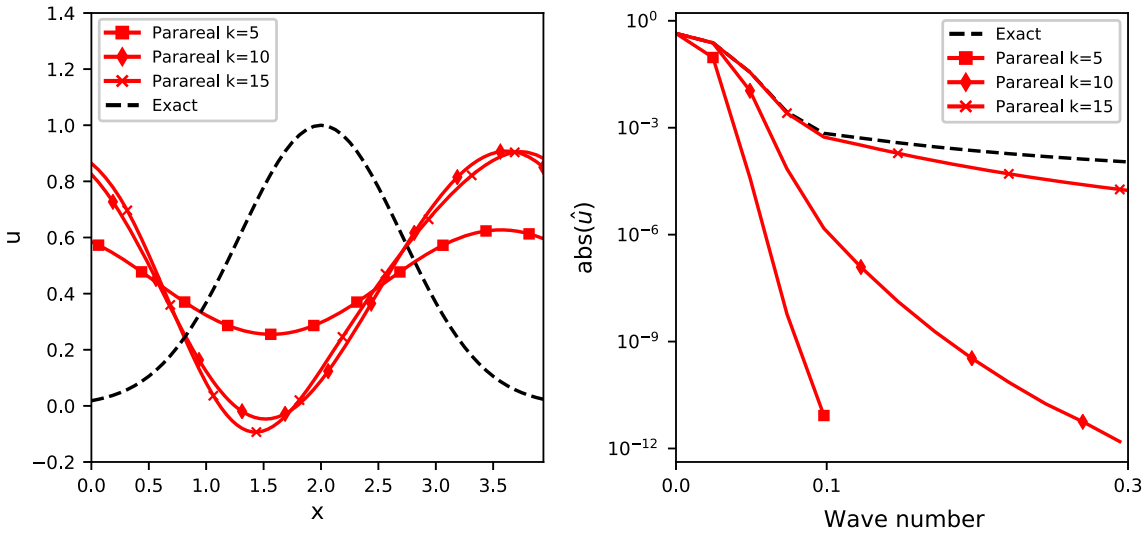


**Fig. 11** Gauss peak computed with the $R_3$ fine propagator. The incorrect phase speed of the fine method puts the peak at a completely wrong position (left), but there is no instability and the spectrum (right) converges as quickly as for the exact phase speed coarse propagator in Fig. 8

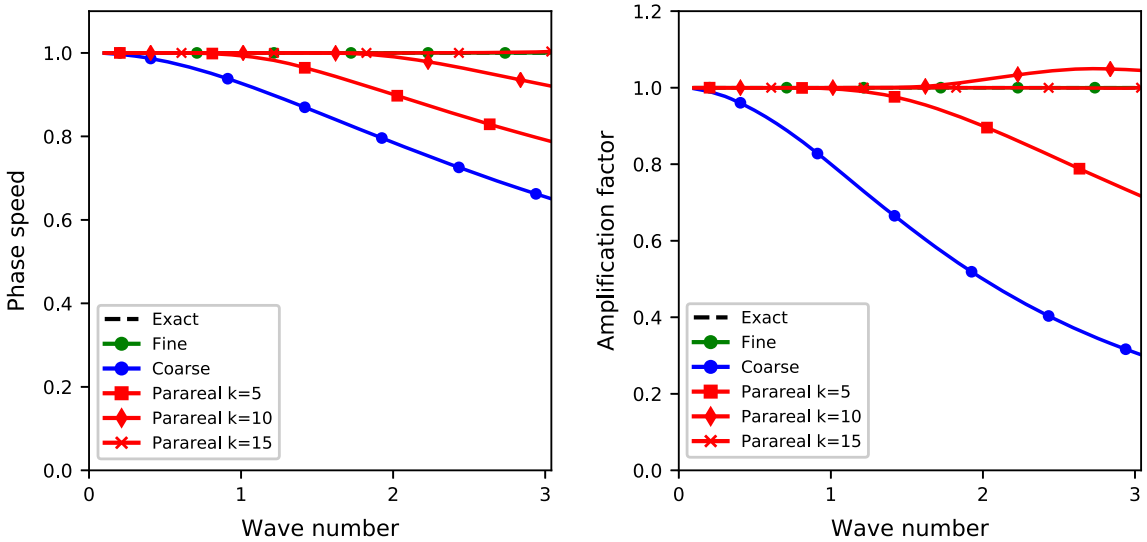


**Fig. 12** Phase speed (left) and amplification factor (right) for standard Parareal with the same configuration as for Fig. 3 except using two coarse time steps per time slice

already significantly more accurate, Parareal with $k = 5$ and $k = 10$ iterations provides more accurate phase speeds and, for $k = 15$, reproduces the exact value exactly, The reduced phase errors translate into a milder instability. For $k = 10$, some wave numbers have amplification factors above one, but both the range of unstable wave numbers and the severity of the instability are much smaller than if only a single coarse time step is used. This explains why configurations can be quite successful where both $\mathcal{F}$ and $\mathcal{G}$ use nearly identical time steps and the difference in runtime is achieved by other means, e.g. an expensive high order spatial discretisation for the fine and a cheap low order discretisation on the coarse level.

**Observation 5** *Since phase errors of the coarse method obviously depend on its time step size, reducing the coarse time step helps to reduce the range of unstable wave numbers and the severity of the instability.*

### 4.5 Number of time slices

All examples so far only considered $P = 16$ time slices and processors. To illustrate the effect of increasing $P$, Fig. 13 shows the discrete dispersion relation for standard Parareal for $P = 64$ time slices or processors (same configuration as in Fig. 3 except for $P$). Even for $k = 15$ iterations, Parareal reproduces the correct phase speed (left figure) very

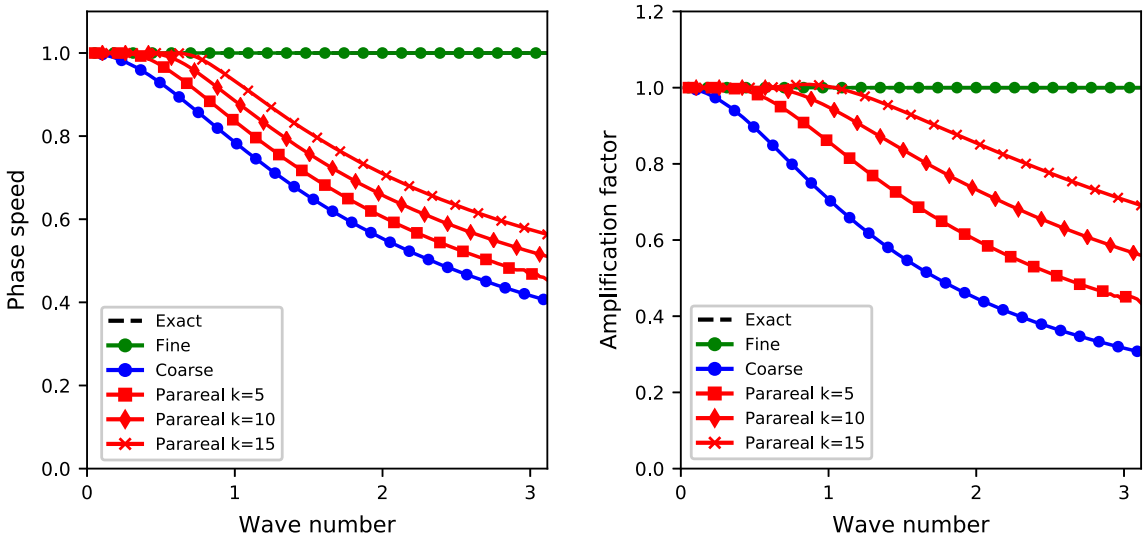


**Fig. 13** Phase error (left) and amplification factor for (right) $P = 64$ in contrast to $P = 16$ as in Fig. 3

poorly—waves across large parts of the spectrum suffer from substantial numerical slowdown. Also, converge is slow and there is only marginal improvement from $k = 5$ to $k = 15$ iterations. Convergence is somewhat faster for the amplification factor (right figure) with more substantial improvement for $k = 15$ over the coarse method. However, there also remains significant numerical attenuation of the upper half of the spectrum. If integrating the Gauss peak with this configuration, the result at $T = 64$ after $k = 5$ iterations is essentially a straight line (not shown) as almost all modes beyond $\kappa = 0$ are strongly damped. A small overshoot at around $\kappa = 1$ is noticeable for $k = 15$ iterations and this will worsen as $k$ increases. In general, as $P$ increases, it takes more iterations to trigger the instability since the slow convergence requires longer to correct for the strong diffusivity of the coarse method.

These results suggest that the high wave numbers are the slowest to converge and that convergence deteriorates as $P$ increases. This is confirmed by Fig. 14, showing the maximum singular value for three wave numbers plotted against $P$. Convergence generally gets worse as $P$ increases, but note that even for $P = 64$ the low wave number mode (blue circles) still converges monotonically while the high wave number mode (green crosses) might already converge non-monotonically for only $P = 4$ processors. There also seems to be a limit for $\sigma$ as $P$ increases, with higher wave numbers levelling off at higher values of $\sigma$.

Therefore, Parareal could provide some speedup for linear hyperbolic problems if the solution consists mainly of very low wave number modes and/or numerical diffusion in the fine propagator is sufficiently strong to remove higher wave number modes. This also explains why divergence damping in the fine propagator can accelerate convergence of

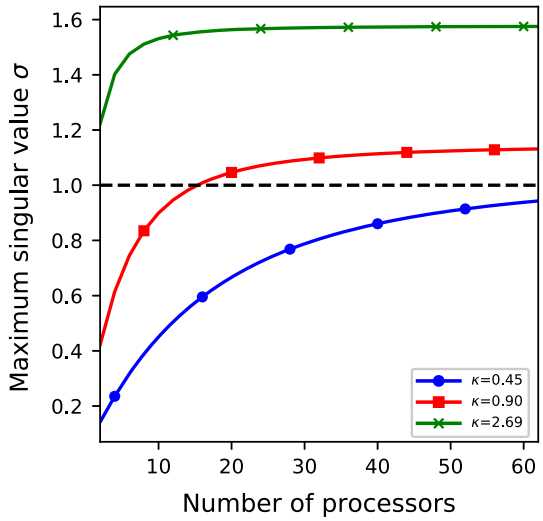


**Fig. 14** Maximum singular value of $\mathbf{E}$ depending on the number of processors $P$ for three different wave numbers $\kappa$

Parareal [29], as it removes exactly the high wave number modes that converge the slowest.

**Observation 6** *While convergence becomes slower as the number of processors $P$ increases, low wave numbers converge monotonically even for large numbers of $P$ but high wave numbers might not do so already for $P = 4$.*

### 4.6 Wave number

The analysis above showed that higher wave numbers converge slower and are more susceptible to instabilities. This is confirmed in Fig. 15 showing the difference between the

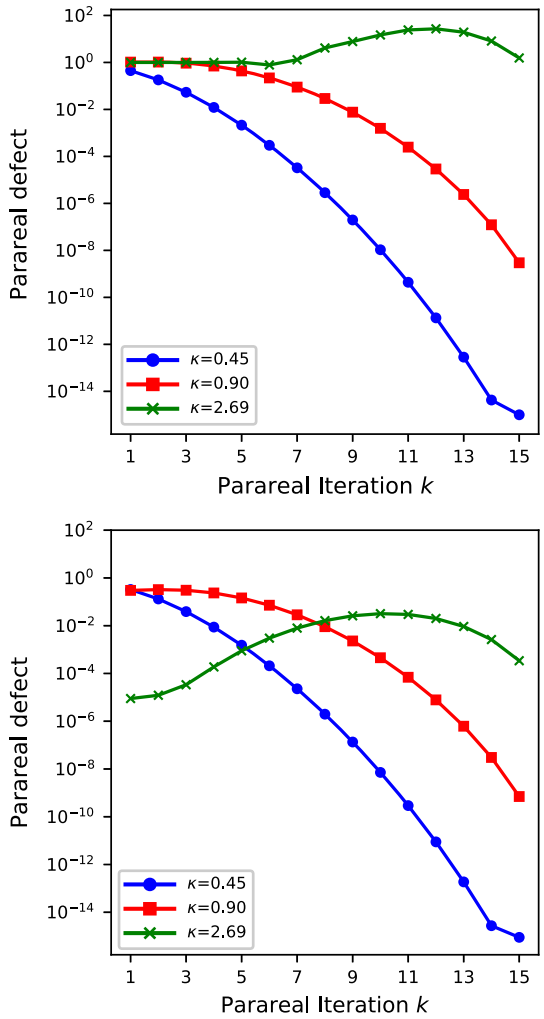


**Fig. 15** Parareal defect versus number of iterations for $\nu = 0.0$ (upper) and $\nu = 0.1$ (lower)

Parareal and fine integrator stability function

$$d(k) := |R_{\text{Parareal}}(k) - R_{\text{fine}}| . \tag{51}$$

The smallest wave number, $\kappa = 0.45$, converges quickly in the hyperbolic (upper) and diffusive case (lower). For $\kappa = 0.9$, both cases show some initial stalling before the mode converges. Finally, $\kappa = 2.69$ shows first a significant increase in the defect before convergence sets in as $k$ comes close to $P = 16$. Interestingly, this is the case for both $\nu = 0$ and $\nu = 0.1$. While the "bulge" is even more pronounced in the diffusive case, since modes decay proportional to $e^{-\nu\kappa^2 t}$ in amplitude, the absolute values of the defect are orders of magnitude smaller. Therefore, at least until $k = 7$ iterations, it is no longer the high wave number mode $\kappa = 2.69$ that will restrict performance, but rather the lower wave number $\kappa = 0.9$. Then, the instability for the high wave number kicks in and for $k \geq 8$ wave number $\kappa = 2.69$ is again causing

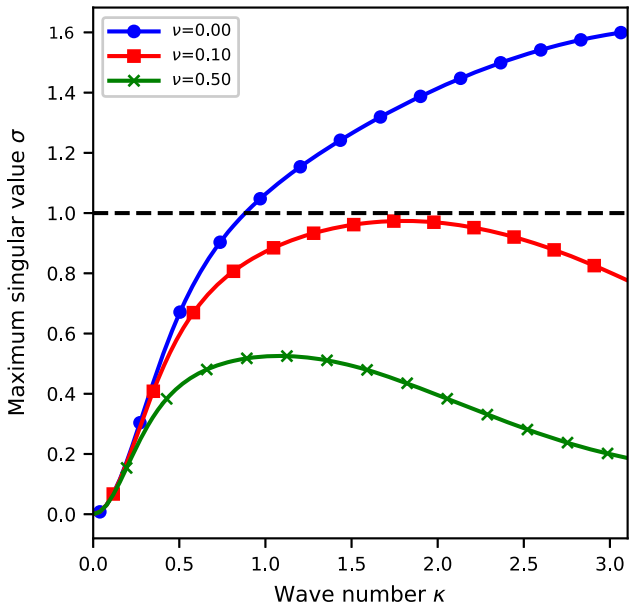


**Fig. 16** Maximum singular value of $\mathbf{E}$ depending on the wave number $\kappa$ for three values of diffusivity $\nu$

the largest defect. As $\nu$ increases, however, the defects for $\kappa = 2.69$ will reduce further, the cross-over point will move to later iterations and eventually lower wave numbers will determine convergence for all iterations. In a sense, in line with the analysis above, Parareal propagates high wave number modes very wrongly in both cases, but since high wave number modes are quickly attenuated if $\nu > 0$, it does not matter very much in the diffusive case.

Figure 16 confirms this for a wider range of wave numbers $\kappa$. It shows the maximum singular value $\sigma$ of the error propagation matrix $\mathbf{E}$ over the whole spectrum for three different values of $\nu$. For $\nu = 0$ (hyperbolic case), $\sigma$ increases monotonically with $\kappa$ and the highest wave number converges the slowest. After around $\kappa \geq 0.8$, the singular values are larger than one and convergence becomes potentially non-monotone. For $\nu = 0.1$, $\sigma$ increases until around $\kappa = 1.8$ and then decreases again. Therefore, the slowest converging mode is no longer at the end but in the middle of the spectrum. Also, we now have $\sigma < 1$ for all $\kappa$ so that all modes will converge, even though some potentially very slowly. Increasing diffusion further to $\nu = 0.5$ greatly improves convergence for all modes, the largest $\sigma$ across the whole spectrum is now below 0.5. The worst converging mode has also moved "further down" the spectrum and is now at around $\kappa = 1.0$. This shows how the strong natural damping of high wave numbers from diffusion counteracts Parareal's tendency to amplify them and thus stabilises it.

**Observation 7** *Since diffusion naturally damps higher wave numbers, it removes the issue of slow or no convergence at the end of the spectrum. Therefore, as the diffusivity parameter*

*v increases, the wave number that converges the slowest and determines performance of Parareal becomes smaller.*

## 5 Conclusions

Efficient parallel-in-time integration of hyperbolic and advection-dominated problems has been shown to be problematic. This prevents application of a promising new parallelisation strategy to many problems in computational fluid dynamics, despite the urgent need for better utilisation of massively parallel computers. For the Parareal parallel-in-time method, mathematical theory has shown that the algorithm is either unstable or inefficient when applied to hyperbolic equations, but so far no detailed analysis exists of how exactly the instability emerges.

The paper presents the first detailed analysis of how Parareal propagates waves and the ways in which the instability is triggered. It uses a formulation of Parareal as a preconditioned fixed point iteration for linear problems to derive its stability function. From there, a discrete dispersion relation is obtained that allows to study the phase speed and amplitude errors from Parareal when computing wave-like solutions. To deal with issues arising from increasing the time interval together with the number of processors, a simple procedure is introduced to normalise the stability function to the unit interval.

Analysis of the discrete dispersion relation and the maximum singular value of the error propagation matrix then allows to make a range of observations, illustrating where the issues of Parareal for wave problems originate. A key finding is that the source of the instability are different discrete phase speeds on the coarse and fine level, which cause instability of higher wave number modes. Interestingly, the overestimation of high wave number amplitudes is present in diffusive problems, too, but since there these amplitudes are naturally strongly damped, it does not trigger instabilities. Further analysis addresses the role of the number of processors, the coarse time step size and comments on possible connections to asymptotic Parareal and multi-grid methods for the Helmholtz equation.

The analysis presented here will be useful to interpret and understand performance of Parareal for more complex problems in computational fluid dynamics. A natural line of future research would be to attempt to develop a new, more stable, parallel-in-time method for hyperbolic problems based on the provided observations. For example, the update in Parareal proceeds component wise. That means that if the coarse propagator moves a wave at the wrong speed, the update will not know that a simple shift of entries could provide a good correction. Attempting to somehow modify the Parareal update to take into account this type of information seems promising, even though probably challenging to do in 3D. Extending the analysis presented here to systems with multiple waves, e.g. the shallow water equations, or to nonlinear problem where wave numbers interact would be another interesting line of inquiry. Furthermore, the framework used here to analyse Parareal is straightforward to adopt for other parallel-in-time integration methods as long as a matrix representation for them is available.

**Acknowledgements** I would like to thank Martin Gander, Martin Schreiber and Beth Wingate for the very interesting discussions at the 5th Workshop on Parallel-in-time integration at the Banff International Research Station (BIRS) in November 2016, which led to the comments about asymptotic Parareal and multi-grid for 1D Helmholtz equation in the paper. Parts of the pseudo-spectral code used to illustrate the effect of numerical dispersion come from David Ketcheson's short course PseudoSpectralPython [21]. The pyParareal code written for this paper relies heavily on the open Python packages `NumPy` [34], `SciPy` [20] and `Matplotlib` [19].



## References

1. Amodio, P., Brugnano, L.: Parallel solution in time of ODEs: some achievements and perspectives. Appl. Numer. Math. **59**(3–4), 424–435 (2009). https://doi.org/10.1016/j.apnum.2008.03.024
2. Bal, G.: On the convergence and the stability of the Parareal algorithm to solve partial differential equations. In: Kornhuber, R., et al. (eds.) Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering, vol. 40, pp. 426–432. Springer, Berlin (2005). https://doi.org/10.1007/3-540-26825-1_43
3. Chen, F., Hesthaven, J.S., Zhu, X.: On the use of reduced basis methods to accelerate and stabilize the Parareal method. In: Quarteroni, A., Rozza, G. (eds.) Reduced Order Methods for Modeling and Computational Reduction, MS&A—Modeling, Simulation and Applications, vol. 9, pp. 187–214. Springer, Berlin (2014). https://doi.org/10.1007/978-3-319-02090-7_7
4. Dai, X., Maday, Y.: Stable Parareal in time method for first- and second-order hyperbolic systems. SIAM J. Sci. Comput. **35**(1), A52–A78 (2013). https://doi.org/10.1137/110861002
5. Dongarra, J., et al.: Applied mathematics research for exascale computing. Technical Report LLNL-TR-651000, Lawrence Livermore National Laboratory (2014). http://science.energy.gov/%7E/media/ascr/pdf/research/am/docs/EMWGreport.pdf
6. Durran, D.R.: Numerical Methods for Fluid Dynamics, Texts in Applied Mathematics, vol. 32. Springer, New York (2010). https://doi.org/10.1007/978-1-4419-6412-0
7. Eghbal, A., Gerber, A.G., Aubanel, E.: Acceleration of unsteady hydrodynamic simulations using the Parareal algorithm. J. Comput. Sci. (2016). https://doi.org/10.1016/j.jocs.2016.12.006
8. Emmett, M., Minion, M.L.: Toward an efficient parallel in time method for partial differential equations. Commun. Appl. Math. Comput. Sci. **7**, 105–132 (2012). https://doi.org/10.2140/camcos.2012.7.105
9. Ernst, O.G., Gander, M.J.: Multigrid methods for Helmholtz problems: a convergent scheme in 1D using standard components. In:

Direct and Inverse Problems in Wave Propagation and Applications. De Gruyer (2013). https://doi.org/10.1515/9783110282283.135
10. Falgout, R.D., Friedhoff, S., Kolev, T.V., MacLachlan, S.P., Schroder, J.B.: Parallel time integration with multigrid. SIAM J. Sci. Comput. **36**, C635–C661 (2014). https://doi.org/10.1137/130944230
11. Falgout, R.D., Manteuffel, T.A., O'Neill, B., Schroder, J.B.: Multigrid reduction in time for nonlinear parabolic problems: A case study. SIAM J. Sci. Comput. 39(5), S298–S322. (2016). https://doi.org/10.1137/16M1082330
12. Farhat, C., Chandesris, M.: Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid–structure applications. Int. J. Numer. Methods Eng. **58**(9), 1397–1434 (2003). https://doi.org/10.1002/nme.860
13. Farhat, C., Cortial, J., Dastillung, C., Bavestrello, H.: Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses. Int. J. Numer. Methods Eng. **67**, 697–724 (2006). https://doi.org/10.1002/nme.1653
14. Fischer, P.F., Hecht, F., Maday, Y.: A Parareal in time semi-implicit approximation of the Navier–Stokes equations. In: Kornhuber, R., et al. (eds.) Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering, vol. 40, pp. 433–440. Springer, Berlin (2005). https://doi.org/10.1007/3-540-26825-1_44
15. Gander, M.J., Petcu, M.: Analysis of a Krylov subspace enhanced Parareal algorithm for linear problem. ESAIM: Proc. **25**, 114–129 (2008). https://doi.org/10.1051/proc:082508
16. Gander, M.J., Vandewalle, S.: Analysis of the Parareal time-parallel time-integration method. SIAM J. Sci. Comput. **29**(2), 556–578 (2007). https://doi.org/10.1137/05064607X
17. Gander, M.J., Vandewalle, S.: On the superlinear and linear convergence of the parareal algorithm. In: Widlund, O.B., Keyes, D.E. (eds.) Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering, vol. 55, pp. 291–298. Springer, Berlin (2007). https://doi.org/10.1007/978-3-540-34469-8_34
18. Haut, T., Wingate, B.: An asymptotic parallel-in-time method for highly oscillatory PDEs. SIAM J. Sci. Comput. **36**(2), A693–A713 (2014). https://doi.org/10.1137/130914577
19. Hunter, J.D.: Matplotlib: a 2D graphics environment. Comput. Sci. Eng. **9**(3), 90–95 (2007). https://doi.org/10.1109/MCSE.2007.55
20. Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: open source scientific tools for python (2001–). http://www.scipy.org/. Accessed 28 May 2018
21. Ketcheson, D.I.: Pseudo spectral python (2015). https://github.com/ketch/PseudoSpectralPython. Accessed 28 May 2018
22. Kiehl, M.: Parallel multiple shooting for the solution of initial value problems. Parallel Comput. **20**(3), 275–295 (1994). https://doi.org/10.1016/S0167-8191(06)80013-X
23. Kooij, G., Botchev, M., Geurts, B.: A block Krylov subspace implementation of the time-parallel Paraexp method and its extension for nonlinear partial differential equations. J. Comput. Appl. Math. **316**, 229–246 (2017). https://doi.org/10.1016/j.cam.2016.09.036.. (Selected papers from NUMDIFF-14)
24. Kreienbuehl, A., Naegel, A., Ruprecht, D., Speck, R., Wittum, G., Krause, R.: Numerical simulation of skin transport using Parareal. Comput. Vis. Sci. **17**, 99–108(2015). https://doi.org/10.1007/s00791-015-0246-y
25. Lions, J.L., Maday, Y., Turinici, G.: A "Parareal" in time discretization of PDE's. Comptes Rendus l'Acad. Sci. Ser. I Math. **332**, 661–668 (2001). https://doi.org/10.1016/S0764-4442(00)01793-6
26. Minion, M.L.: A hybrid Parareal spectral deferred corrections method. Commun. Appl. Math. Comput. Sci. **5**(2), 265–301 (2010). https://doi.org/10.2140/camcos.2010.5.265
27. Nievergelt, J.: Parallel methods for integrating ordinary differential equations. Commun. ACM **7**(12), 731–733 (1964). https://doi.org/10.1145/355588.365137
28. Ruprecht, D.: v2.0 Parallel-in-time/pypareareal: wave propagation characteristics of parareal (2017). https://doi.org/10.5281/zenodo.1012274
29. Ruprecht, D., Krause, R.: Explicit parallel-in-time integration of a linear acoustic–advection system. Comput. Fluids **59**, 72–83 (2012). https://doi.org/10.1016/j.compfluid.2012.02.015
30. Schreiber, M., Peixoto, P.S., Haut, T.,Wingate, B.: Beyond spatial scalability limitations with a massively parallel method for linear oscillatory problems. Int. J. High Perform. Comput. Appl. (2017). https://doi.org/10.1177/1094342016687625
31. Staff, G.A., Rønquist, E.M.: Stability of the Parareal algorithm. In: Kornhuber, R., et al. (eds.) Domain Decomposition Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering, vol. 40, pp. 449–456. Springer, Berlin (2005). https://doi.org/10.1007/3-540-26825-1_46
32. Steiner, J., Ruprecht, D., Speck, R., Krause, R.: Convergence of Parareal for the Navier–Stokes equations depending on the Reynolds number. In: Abdulle, A., Deparis, S., Kressner, D., Nobile, F., Picasso, M. (eds.) Numerical Mathematics and Advanced Applications—ENUMATH 2013, Lecture Notes in Computational Science and Engineering, vol. 103, pp. 195–202. Springer, Berlin (2015). https://doi.org/10.1007/978-3-319-10705-9_19
33. Trindade, J.M.F., Pereira, J.C.F.: Parallel-in-time simulation of the unsteady Navier–Stokes equations for incompressible flow. Int. J. Numer. Methods Fluids **45**(10), 1123–1136 (2004). https://doi.org/10.1002/fld.732
34. van der Walt, S., Colbert, S.C., Varoquaux, G.: The numpy array: a structure for efficient numerical computation. Comput. Sci. Eng. **13**(2), 22–30 (2011). https://doi.org/10.1109/MCSE.2011.37