Journal of Computational Physics 274 (2014) 370–392

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Asynchronous finite-difference schemes for partial differential equations

Diego A. Donzis *, Konduri Aditya

Department of Aerospace Engineering, Texas A&M University, College Station, TX 77843, United States

ARTICLE INFO

Article history:
Received 25 March 2014
Received in revised form 5 June 2014
Accepted 11 June 2014
Available online 17 June 2014

Keywords:
Asynchronous schemes
Partial differential equations
Massive computations

ABSTRACT

Current trends in massively parallel computing systems suggest that the number of processing elements (PEs) used in simulations will continue to grow over time. A known problem in this context is the overhead associated with communication and/or synchronization between PEs as well as idling due to load imbalances. Simulation at extreme levels of parallelism will then require an elimination, or at least a tight control of these overheads. In this work, we present an analysis of common finite difference schemes for partial differential equations (PDEs) when no synchronization between PEs is enforced. PEs are allowed to continue computations regardless of messages status and are thus asynchronous. We show that while stability is conserved when these schemes are used asynchronously, accuracy is greatly degraded. Since message arrivals at PEs are essentially random processes, so is the behavior of the error. Within a statistical framework we show that average errors drop always to first-order regardless of the original scheme. The value of the error is found to depend on both grid spacing as well as characteristics of the computing system including number of processors and statistics of the delays. We propose new schemes that are robust to asynchrony. The analytical results are compared against numerical simulations.

1. Introduction

Many natural and engineered systems and processes can be accurately described by partial differential equations (PDEs). This includes fluid mechanics, electromagnetism, quantum mechanics as well as common simple processes in continuum media such as diffusion and wave propagation. In a number of real applications, however, due to the complexity of the equations themselves as well as the geometrical aspects of the problem, analytical solutions are not known and numerical simulations provide invaluable information to understand these systems.

Even with numerical simulations, the complexity of systems at realistic conditions typically requires massive computational resources. In the last few decades, this computational power has been realized through increasing levels of parallelism. When a problem is decomposed into a number of processing elements (PEs), solving the PDE typically requires communication between PEs to compute spatial derivatives. As the number of PEs increases, this communication becomes more challenging (e.g. [1,2]). In fact, this may well be a major bottleneck at the next generation of computing systems [3] which may comprise an extremely large number of PEs. At those extreme levels of parallelism, even small imbalances due to noise [4] in otherwise perfectly balanced codes can represent enormous penalties as PEs idle waiting to receive data from other

* Corresponding author.
E-mail addresses: donzis@tamu.edu (D.A. Donzis), akonduri@tamu.edu (K. Aditya).

http://dx.doi.org/10.1016/j.jcp.2014.06.017
0021-9991/

PEs. This is especially critical when, as commonly done, a global synchronization is imposed at each time step to finalize all communications as well as to obtain information to determine the time-step size in unsteady calculations subjected to a so-called Courant–Friedrichs–Lewy condition.

Thus, in order to take advantage of computational systems at extreme levels of parallelism, relaxing all (especially global) synchronizations is of prime necessity [3]. This is the main thrust of this work. In particular, our objective here is to study widely employed finite difference schemes but used in such a way to avoid all synchronizations as well as potential idle time. While it is currently possible to implement asynchronous communications using modern hardware and software, current numerical schemes still require synchronizations at the mathematical level. In this work we propose schemes which are asynchronous at a mathematical level and thus relax this strong constraint. These new schemes do not require updated values from other PEs and can proceed with calculations without interruption. Studying the numerical properties of these schemes including stability, consistency and accuracy as well as suggesting a framework for devising new formulations resilient to asynchrony are the main objectives of this work.

Substantial efforts have been devoted to asynchronous algorithms in different contexts such as linear systems of equations or more general fixed-point formulations [5,6]. Asynchronous linear solvers have been used in the context of PDEs to solve linear systems required in the computations but global synchronizations are typically still required. Due to the importance of increasing parallelism, other approaches have also been investigated such as explicit–implicit methods (e.g. [7]) though, again, some type of synchronization at some point during the calculations is typically unavoidable or so-called discrete event-driven simulations (e.g. [8]). Work exploiting asynchrony to solve directly problems governed by time-dependent PDEs, however, has been more limited. For example, some studies that focused on PDEs [9,10] were limited to a particular class of PDE (heat equation) and the order of accuracy of the resulting schemes remained low. Further extensions to higher orders has also been limited (e.g. [11] for second order schemes). We study the behavior of general finite differences when used in an asynchronous manner. We provide general results on stability and accuracy that can be applied to a broader class of problems. Unlike previous studies, our analysis incorporates stochastic characteristics of the performance of the parallel computing system which allows us to determine accuracy based on network performance. We show, for example, that accuracy depends on how the number of PEs is increased as the problem size increases as well as how the time step size is determined based on common stability conditions or physical considerations (some preliminary related results have been presented in Ref. [12]). We also provide a framework to devise new asynchronous schemes.

In Section 2 we introduce the concept behind asynchronous schemes using finite differences. The stability of such schemes is analyzed in Section 3 and the resulting accuracy is studied in Section 4. Numerical simulations to support the theoretical developments in previous sections are shown in Section 5. Due to the loss in accuracy when asynchrony is present, we develop new schemes that can preserve some desired order of accuracy which is presented in Section 6. Conclusions and further discussion are included in Section 7.

2. Concept

Our interest is in the general linear PDE:

$$\frac{\partial u}{\partial t} = \sum_{d=1,D} \beta_d \frac{\partial^d u}{\partial x^d} \tag{1}$$

with D being the highest derivative in the PDE and the constants β_d's determine the characteristic of the different physical processes represented by the different terms. Particular cases of interest are the wave equation ($D=1$ with $\beta_1 \neq 0$), the heat equation ($D=2$ with $\beta_1 = 0$ and $\beta_2 \neq 0$), and the advection–diffusion equation ($D=2$ with $\beta_1 \neq 0$ and $\beta_2 \neq 0$).

For illustration purposes consider the unsteady one-dimensional heat (or diffusion) equation

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \tag{2}$$

where $u(x,t)$ is the temperature at a spatial location $x \in [0,l]$ and time t and α is the thermal diffusivity of the medium. With N uniformly distributed grid points, Eq. (2) can be discretized using a second-order central difference in space and first-order forward difference in time to obtain a numerical scheme with well-know characteristics [13]:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + \mathcal{O}(\Delta t, \Delta x^2) \tag{3}$$

where u_i^n is the temperature at a point $x = x_i$ and time level n. Here $x_i = i\Delta x$ with $\Delta x = l/N$ being the grid spacing and $i = 1, \ldots, N$. We will assume, unless explicitly mentioned, periodic boundary conditions. The time step size is Δt. The last term represents the order of the truncation error in time and space for this approximation.

The scheme in Eq. (3) can be rewritten in the following form

$$u_i^{n+1} = u_i^n + \frac{\alpha \Delta t}{\Delta x^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right) \tag{4}$$

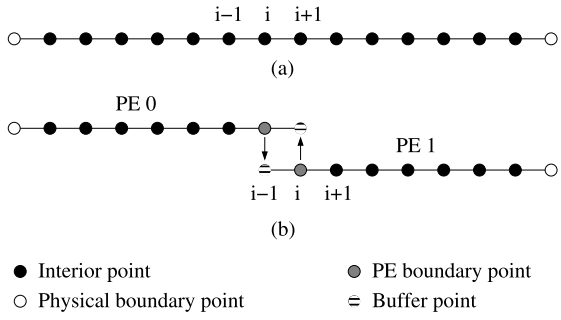

Fig. 1. Discretized one-dimensional domain. (a) Domain in serial codes. (b) Same domain decomposed into two PEs.

which shows that to advance the solution from time level n to $n+1$, one needs the value of the function at neighboring points at time level n. This is trivially implemented in a serial code where all the values u_i^n are available in the PE's memory (Fig. 1(a)).

Consider, however, the case where the discretized domain is divided among a specified number of PEs (say 2, as in Fig. 1(b)). Computations at interior points remain trivial as the required information is available locally to the PEs. Updating the values at grid points close to PE boundaries, however, require values from other PEs, that is, either u_{i-1}^n or u_{i+1}^n from the corresponding neighboring PEs. These values are typically communicated over the network into buffer (or "ghost") arrays. Computations are halted until all PEs receive data in these so-called halo exchanges.

For later use, we define I_I to be the set of all interior grid points; that is, if $i \in I_I$ then computing derivatives at x_i does not require data from other PEs. If, on the other hand, computing the derivative at x_i does make use of data from other PEs, then i is a boundary point and we have $i \in I_B$. Obviously $I_I \cap I_B = \emptyset$. The number of elements in I_I and I_B will be denoted by N_I and N_B respectively.

In applications, especially at extreme scales, the computing time may be much smaller than the communication time [1,2]. This may result in a significant waiting time in large computing systems and have a profound effect in the scalability of the code. Furthermore, even if computation time is comparable to or larger than communication time, a global synchronization such as that described above, could cause substantial idling time if additional tasks (even if small) are required by one or a subset of PEs forces all PEs to wait for a single (the slowest) PE to finish its computations. This may also be an issue with respect to system noise at very large scales [4].

These problems can be avoided by relaxing these synchronizations and allowing all PEs to continue calculations regardless of the status of the messages that are to be received by the corresponding PEs. In the context of Fig. 1, PE 1 is not required to wait for the most updated value u_{i-1}^n, but instead, can compute derivatives using $u_{i-1}^{\tilde{n}}$ where $\tilde{n}$ is the latest time level available to PE 1 at x_{i-1}. This modifies the finite difference equation (4) for points close to PEs boundaries. In particular, the expressions for the leftmost and rightmost grid points in each PE are, respectively,

$$u_i^{n+1} = u_i^n + \frac{\alpha \Delta t}{\Delta x^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^{\tilde{n}}\right) \tag{5}$$

$$u_i^{n+1} = u_i^n + \frac{\alpha \Delta t}{\Delta x^2}\left(u_{i+1}^{\tilde{n}} - 2u_i^n + u_{i-1}^n\right). \tag{6}$$

In the most general case, $\tilde{n}$ could be n, $(n-1)$, $(n-2)$, etc. Furthermore, these delays can be different at different PEs boundaries and time levels. The occurrence of a particular level for $\tilde{n}$ at a particular location and time depends on how fast the communications take place, which in turn depends on a number of factors like hardware, network topology, network traffic, message size, etc. some of which are unpredictable and turn the process into a random one. Indeed, it is known that communication times possess random characteristics in real systems (e.g. [14]).

Since $\tilde{n}$ will be essentially a random variable,[1] we associate the occurrence of each time level with a probability. While in principle $\tilde{n}$ could take any value, it is convenient for the analysis (and necessary in terms of accuracy as we show below) to limit the number of past time levels the scheme could use. If the number of allowable time levels is L, then $\tilde{n} \in \{n, n-1, \ldots, n-L+1\}$. Let $p_{k[i]}$ be the probability of having $\tilde{n} = n - \tilde{k}_i$ at a grid point i, where the random delay $\tilde{k}_i$ can take the values $\tilde{k}_i = 0, 1, \ldots, L-1$, then the probabilities at any grid point x_i are obviously related by

$$\sum_{k[i]=0}^{L-1} p_{k[i]} = 1. \tag{7}$$

The probability of having asynchronous computations at a buffer point i is thus $(1 - p_{0[i]})$.

[1] In order to distinguish random from deterministic variables, we will use a tilde (˜) over the variable for the former.

Table 1
Parameters used in the study of the properties of asynchronous schemes. N: number of grid points; P: number of processing elements; L: maximum level allowed for delays; $p_{k[i]}$: probability of observing a delay of k at grid point i; S: number of grid points on each side of a grid point used to compute a derivative. For a symmetric finite difference the number of grid points in the stencil is $2S+1$.

Parameter	Symbol
Grid resolution	N
Number of PEs	P
Maximum number of time levels	L
Probability of $\tilde{n} = (n-k)$ at x_i	$p_{k[i]}$
Number of points in stencil	$2S+1$

Such asynchronous numerical schemes will be a viable option only if they are shown to be stable, consistent and accurate. Interestingly, the computed solution will not only depend on grid resolution and timestep, but also on the parameters that influence $\tilde{n}$. Table 1 lists the parameters we will use to study the properties of finite differencing schemes under asynchronous conditions.

3. Stability

An obvious requirement for any scheme to be usable in practice is that it has to be stable. For deterministic schemes well-known techniques such as von Neumann analysis have been used extensively to prove stability [15]. The main idea is to take advantage of the linear nature of the governing PDE in which case errors due to, for example, finite precision arithmetic, evolve according to the same PDE. By using a Fourier series to represent this error, one can easily determine under what conditions a particular mode (characterized by its wavenumber) will grow unbounded in time in which case the scheme is said to be unstable. If all the modes are either damped or conserve their amplitude in time, the schemes is said to be stable. The important parameter here is the amplification factor G which is defined as the ratio of the error between successive iterations. In the case of Eq. (2) discretized as in Eq. (4), it is readily shown [15] that the condition for stability is $r_\alpha \leq 1/2$ where $r_\alpha \equiv \alpha \Delta t/\Delta x^2$.

When asynchrony is allowed across PE boundaries, however, there are two difficulties that prevent us from utilizing von Neumann stability analysis. First, von Neumann analysis requires the discretized equation to be the same across the entire domain. This is not the case when $\tilde{k}_i \neq 0$ at some $i \in I_B$. Second, the specific value of $\tilde{k}_i$ (and therefore $\tilde{n}$) at those points is essentially a random variable which complicates further the applicability of von Neumann analysis.

The so-called matrix formulation, on the other hand, provides the flexibility needed to incorporate these complexities in a unified framework that allows us to compare stability characteristics with the original synchronous scheme. This is the approach we will follow in this work.

3.1. Synchronous schemes

Consider again Eq. (6) written in the following form

$$u_i^{n+1} = r_\alpha u_{i-1}^n + (1-2r_\alpha)u_i^n + r_\alpha u_{i+1}^{\tilde{n}}. \tag{8}$$

If the scheme is completely synchronous (i.e. $\tilde{n} = n$ always) and we assume a periodic domain, we can write

$$\boldsymbol{V}^{n+1} = \boldsymbol{A}^n \boldsymbol{V}^n \tag{9}$$

where $\boldsymbol{V}^n = [u_1^n \; u_2^n \; \ldots \; u_i^n \; \ldots \; u_N^n]^T$ (superscript T stands for transpose) is a vector of size N and the $N \times N$ (cyclic tridiagonal) matrix $\boldsymbol{A}^n$ is given by

$$\boldsymbol{A}^n = \begin{bmatrix} (1-2r_\alpha) & r_\alpha & 0 & . & . & r_\alpha \\ r_\alpha & (1-2r_\alpha) & r_\alpha & 0 & . & . \\ 0 & r_\alpha & (1-2r_\alpha) & r_\alpha & 0 & . \\ . & . & \ddots & \ddots & \ddots & . \\ 0 & . & . & r_\alpha & (1-2r_\alpha) & r_\alpha \\ r_\alpha & 0 & . & . & r_\alpha & (1-2r_\alpha) \end{bmatrix}. \tag{10}$$

Although the elements of this array are clearly independent of the value of n, the superscript in $\boldsymbol{A}^n$ is maintained for later analysis.

It is well known [15] that the stability of such a scheme is determined by the spectral characteristics of the matrix $\boldsymbol{A}^n$ (in particular its largest eigenvalue). However, as will be clear momentarily a more general treatment is needed for our asynchronous schemes. We, thus proceed by noting that the evolution in time can be represented by

$$\boldsymbol{V}^{n+1} = \boldsymbol{A}^n \boldsymbol{V}^n = \boldsymbol{A}^n \boldsymbol{A}^{n-1} \boldsymbol{V}^{n-1} = \cdots = \boldsymbol{A}^n \boldsymbol{A}^{n-1} \ldots \boldsymbol{A}^0 \boldsymbol{V}^0 \tag{11}$$

where $\boldsymbol{V}^0$ is the initial condition. In order for the scheme to be stable one requires a bounded solution with no amplification of perturbations. Due to the linear nature of the governing equation, the error evolves according to Eq. (11) as well and, thus, the system is stable when

$$\left\| \boldsymbol{V}^{n+1} \right\| / \left\| \boldsymbol{V}^0 \right\| \leq 1 \tag{12}$$

where $\|\cdot\|$ is an appropriate norm. Using the property that $\|\boldsymbol{A}\boldsymbol{B}\| \leq \|\boldsymbol{A}\|\|\boldsymbol{B}\|$ for any matrix norm and matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, it is easy to see that Eq. (12) is satisfied if

$$\left\| \boldsymbol{A}^n \right\| \left\| \boldsymbol{A}^{n-1} \right\| \ldots \left\| \boldsymbol{A}^0 \right\| \leq 1. \tag{13}$$

Since in our case $\boldsymbol{A}^n$ is independent of n (i.e. $\boldsymbol{A}^n = \boldsymbol{A}^{n-1} = \cdots = \boldsymbol{A}^0$), it follows that the system is stable when the matrix in Eq. (10) satisfies $\|\boldsymbol{A}^n\| \leq 1$. To track the largest perturbation, we use the ∞-norm which is defined, for any matrix $\boldsymbol{B}$ with elements b_{ij}, as $\|\boldsymbol{B}\|_\infty = \max_i \sum_j |b_{ij}|$. In our present case the condition

$$\left\| \boldsymbol{A}^n \right\|_\infty \leq 1 \tag{14}$$

for stability is readily shown to reduce to

$$|r_\alpha| + |1 - 2r_\alpha| + |r_\alpha| \leq 1. \tag{15}$$

Trivially, this is satisfied for $0 < r_\alpha \leq 1/2$, a well-known result [15] for Eq. (8).

3.2. Asynchronous schemes

Consider again Eq. (8) but now with $\tilde{n}$ being a random variable. For simplicity in the exposition, we will consider the case where $\tilde{n}$ can only take the values n or $n-1$. The generalization to arbitrary time lags is discussed later in this section. If we define the vector $\boldsymbol{W}^n$ as the concatenation of the values of u_i^n for all i's and the values of u_i^{n-1} also for all i's we can write

$$\boldsymbol{W}^{n+1} = \tilde{\boldsymbol{C}}^n \boldsymbol{W}^n \tag{16}$$

where $\boldsymbol{W}^n$ is of size $2N$ and given by

$$\boldsymbol{W}^{n+1} = \begin{bmatrix} \boldsymbol{V}^{n+1} \\ \boldsymbol{V}^n \end{bmatrix} \tag{17}$$

or explicitly,

$$\boldsymbol{W}^{n+1} = \left[u_1^{n+1} \quad u_2^{n+1} \quad \ldots \quad u_i^{n+1} \quad \ldots \quad u_N^{n+1} \,\middle|\, u_1^n \quad u_2^n \quad \ldots \quad u_i^n \quad \ldots \quad u_N^n \right]^T \tag{18}$$

and the $2N \times 2N$ random matrix $\tilde{\boldsymbol{C}}^n$ can be divided into four $N \times N$ blocks as

$$\tilde{\boldsymbol{C}}^n = \left[\begin{array}{c|c} \tilde{\boldsymbol{A}}_0^n & \tilde{\boldsymbol{A}}_1^n \\ \hline \boldsymbol{I} & \boldsymbol{0} \end{array} \right] \tag{19}$$

with $\boldsymbol{I}$ and $\boldsymbol{0}$ being the identity and zero matrices respectively, both of size $N \times N$. The matrices $\tilde{\boldsymbol{A}}_0^n$ and $\tilde{\boldsymbol{A}}_1^n$ contain a random component due to $\tilde{n}$ which can be introduced as follows. In previous sections, we have defined $\tilde{k}_i$ to be the (random) delay seen by a PE which requires the value of the function at that point $i \in I_B$. This is related to $\tilde{n}$ as $\tilde{n} = n - \tilde{k}_i$. In our simplified case, $\tilde{k}_i$ can take the value of 0 or 1 and the probabilities associated with each of these outcomes is $p_{0[i]}$ and $p_{1[i]}$, respectively. Then we can write:

$$\tilde{\boldsymbol{A}}_0^n = \begin{bmatrix} (1-2r_\alpha) & (1-\tilde{k}_2)r_\alpha & 0 & . & . & r_\alpha \\ r_\alpha & (1-2r_\alpha) & (1-\tilde{k}_3)r_\alpha & 0 & . & . \\ 0 & r_\alpha & (1-2r_\alpha) & (1-\tilde{k}_4)r_\alpha & 0 & . \\ . & . & \ddots & \ddots & \ddots & . \\ 0 & . & . & r_\alpha & (1-2r_\alpha) & (1-\tilde{k}_N)r_\alpha \\ (1-\tilde{k}_1)r_\alpha & 0 & . & . & r_\alpha & (1-2r_\alpha) \end{bmatrix} \tag{20}$$

which is identical to Eq. (10) except for the factors $(1-\tilde{k}_i)$ in the elements above the diagonal. The matrix $\tilde{\boldsymbol{A}}_1^n$ is:

$$\tilde{\boldsymbol{A}}_1^n = \begin{bmatrix} 0 & \tilde{k}_2 r_\alpha & 0 & . & . & 0 \\ 0 & 0 & \tilde{k}_3 r_\alpha & 0 & . & . \\ 0 & 0 & 0 & \tilde{k}_4 r_\alpha & 0 & . \\ . & . & \ddots & \ddots & \ddots & \tilde{k}_N r_\alpha \\ \tilde{k}_1 r_\alpha & 0 & . & . & 0 & 0 \end{bmatrix}. \tag{21}$$

Note that the random elements appearing only above the diagonal is due to the use of Eq. (8) which has delays only on the right boundary for simplicity. The general case is considered in the next section.

From the definitions above it is easy to see the effect of $\tilde{k}_i$ on the scheme. For a given i, if $\tilde{k}_i = 0$ then the element $(i-1, i)$ in $\tilde{\boldsymbol{A}}_1^n$ will be zero and the scheme will be governed by $\tilde{\boldsymbol{A}}_0^n$—that is a classical synchronous scheme with $\tilde{n} = n$. If on the other hand $\tilde{k}_i = 1$, then the element $(i-1, i)$ in $\tilde{\boldsymbol{A}}_0^n$ will be zero, and the corresponding element $(i-1, i)$ of $\tilde{\boldsymbol{A}}_1^n$ will be r_α—that is an asynchronous scheme with $\tilde{n} = n-1$. Thus, the random variable $\tilde{k}_i$ is responsible for switching between the two different schemes that result from $\tilde{n} = n$ and $\tilde{n} = n-1$.

We can now analyze the stability of the scheme described by Eq. (8). Just like Eq. (11) for synchronous schemes, we can write the evolution of the solution as

$$\boldsymbol{W}^{n+1} = \tilde{\boldsymbol{C}}^n \boldsymbol{W}^n = \tilde{\boldsymbol{C}}^n \tilde{\boldsymbol{C}}^{n-1} \boldsymbol{W}^{n-1} = \cdots = \tilde{\boldsymbol{C}}^n \tilde{\boldsymbol{C}}^{n-1} \ldots \tilde{\boldsymbol{C}}^0 \boldsymbol{W}^0. \tag{22}$$

The stability criterion

$$\left\| \boldsymbol{W}^{n+1} \right\| / \left\| \boldsymbol{W}^0 \right\| \leq 1 \tag{23}$$

is now satisfied if

$$\left\| \tilde{\boldsymbol{C}}^n \tilde{\boldsymbol{C}}^{n-1} \ldots \tilde{\boldsymbol{C}}^0 \right\| \leq \left\| \tilde{\boldsymbol{C}}^n \right\| \left\| \tilde{\boldsymbol{C}}^{n-1} \right\| \ldots \left\| \tilde{\boldsymbol{C}}^0 \right\| \leq 1. \tag{24}$$

Obviously, this inequality holds if the norm of each individual matrix on the left-hand-size is less than unity.

Anticipating the use of the ∞-norm, we note that the $(i-1)$-th row of $\tilde{\boldsymbol{C}}^n$ is given by

$$\begin{bmatrix} 0 & \ldots & r_\alpha & (1-2r_\alpha) & (1-\tilde{k}_i) r_\alpha & 0 & \ldots & \vert & 0 & \ldots & \tilde{k}_i r_\alpha & 0 & \ldots \end{bmatrix}. \tag{25}$$

Thus, the absolute row sum to be considered to compute the ∞-norm from this row is

$$|r_\alpha| + \left|(1-2r_\alpha)\right| + \left|(1-\tilde{k}_i) r_\alpha\right| + |\tilde{k}_i r_\alpha|. \tag{26}$$

Now it is readily seen that since $\tilde{k}_i$ is 0 or 1, then the contribution for the sum of the last two terms is always $|r_\alpha|$ independent of $\tilde{k}_i$. Furthermore, since all the rows in the upper half of $\tilde{\boldsymbol{C}}^n$ have identical structure (with the same absolute row sum) and the contribution from rows in the lower half is, trivially, $|1|$, we find that the scheme will be stable when

$$|r_\alpha| + \left|(1-2r_\alpha)\right| + |r_\alpha| \leq 1, \tag{27}$$

which is identical to Eq. (15). Stability for the asynchronous scheme is then also assured when $0 < r_\alpha \leq 1/2$.

We thus arrive at the conclusion that if the base synchronous scheme is stable (in the norm sense described in Section 3.1), the asynchronous equivalent will also be stable (in the same sense). The property that enables this is the invariance of the norm of $\tilde{\boldsymbol{C}}^n$ to random switching between numerical schemes.

While the results above were derived for the heat equation with a particular space and time discretization, the main conclusion can be generalized to a much broader class of problems and conditions. This is the topic of the next section.

3.3. The general case

We first note that we have included the random variable $\tilde{k}_i$ in every row of $\tilde{\boldsymbol{A}}_0^n$ and $\tilde{\boldsymbol{A}}_1^n$ in Eq. (20) and Eq. (21). If the source of randomness is due to the communication between PEs, this will represent an extreme case where every grid point is in I_B, that is a PE boundary point. In practical terms this would imply that each PE contains only one grid point which is typically of little practical relevance. More common is the case where each PE contains a number of grid points in each direction such that only a (typically small) fraction of the grid points will need data from neighboring PEs. This is easily accommodated in our formulation by removing the factors $(1-\tilde{k}_i)$ in the corresponding non-boundary rows (i.e. $i \in I_I$) of $\tilde{\boldsymbol{A}}_0^n$ and setting to zero the only non-zero elements in $\tilde{\boldsymbol{A}}_1^n$ in that row.

Alternatively, one could still use the matrices in Eq. (20) and Eq. (21) with Dirac delta PDFs for $\tilde{k}_i$ when $i \in I_I$ (i.e. interior points). This means that $\tilde{k}_i = 0$ always which, in turn, implies $\tilde{n} = n$ always too. This approach can be generalized further. Every entry in the matrix could be multiplied by random variables that turn that particular entry on or off randomly. For interior points the PDFs of these random variables will always be Dirac deltas centered at zero (that is, no delay).

For both approaches, only a subset of rows would need the special treatment of Eq. (26). Stability will then be determined by the worst case between random (PE boundary grid points) and deterministic (interior grid points) rows in $\tilde{\boldsymbol{C}}^n$. Since in our example, the original deterministic scheme is stable, the issue of stability will depend only on the random rows, which according to Eq. (27) lead to a stable scheme as well. We now proceed to formalize these ideas.

Formally, for a two-level temporal scheme we write Eq. (1) in a discretized form as

$$u_i^{n+1} = \sum_{j=-S}^{+S} c_j u_{i+j}^n \tag{28}$$

where $2S+1$ is the stencil size to compute spatial derivatives and c_j the coefficients that result from a particular discretization of the time and space derivatives in the governing equation. This system can be rewritten as Eq. (9) where $\boldsymbol{A}^n$ is a banded matrix (with band size $2S+1$). The only non-zero coefficients in row i are simply the c_j's with $j=-S\ldots 0\ldots S$. When the same differentiation scheme is used uniformly across the domain, these coefficients are independent of the row number (i.e. i). Thus, if, as in Section 3.1, we use the ∞-norm, then it is sufficient to show that the absolute row sum of any row is less or equal than unity to assure stability (Eq. (14)). Formally,

$$\sum_{j=-S}^{S} |c_j| \leq 1 \tag{29}$$

is sufficient (but not necessary) to ensure a stable scheme. A practical example of an application of this criterion was given in Eq. (15). Note that this condition may be, as discussed below, stricter than needed for stability which is known for synchronous schemes [15].

In the case asynchrony is present, all the superscripts n on the right-hand-side become $\tilde{n}=n-\tilde{k}_{i+j}$ in the most general case. If we further allow $\tilde{k}_i$ to assume values $\tilde{k}_i=0,1,2,\ldots,M$, we can write

$$u_i^{n+1}=\sum_{m=0}^{M}\sum_{j=-S}^{+S} c_j \tilde{L}_{i+j}^{m} u_{i+j}^{n-m}, \tag{30}$$

where the random function $\tilde{L}_i^m$ is the Lagrange polynomial of order M defined as

$$\tilde{L}_i^m=\prod_{\substack{l=0\\ l\neq m}}^{M}\frac{(\tilde{k}_i-l)}{(m-l)}. \tag{31}$$

By construction, we can see that $\tilde{L}_i^m=1$ when the random variable $\tilde{k}_i$ assumes the value m (i.e. a delay of m steps) and $\tilde{L}_i^m=0$ for any other allowable value of $\tilde{k}_i$ which provides a formal way of generalizing the coefficients in Eq. (20) and Eq. (21). Explicitly Eq. (30) can be written as

$$\begin{aligned}
u_i^{n+1}=&\cdots+c_{-1}\tilde{L}_{i-1}^{0}u_{i-1}^{n}+c_0\tilde{L}_i^{0}u_i^{n}+c_1\tilde{L}_{i+1}^{0}u_{i+1}^{n}+\cdots+\\
&\cdots+c_{-1}\tilde{L}_{i-1}^{1}u_{i-1}^{n-1}+c_0\tilde{L}_i^{1}u_i^{n-1}+c_1\tilde{L}_{i+1}^{1}u_{i+1}^{n-1}+\cdots+\\
&\cdots+c_{-1}\tilde{L}_{i-1}^{2}u_{i-1}^{n-2}+c_0\tilde{L}_i^{2}u_i^{n-2}+c_1\tilde{L}_{i+1}^{2}u_{i+1}^{n-2}+\cdots+\\
&\vdots\\
&\cdots+c_{-1}\tilde{L}_{i-1}^{M}u_{i-1}^{n-M}+c_0\tilde{L}_i^{M}u_i^{n-M}+c_1\tilde{L}_{i+1}^{M}u_{i+1}^{n-M}
\end{aligned} \tag{32}$$

where only a few terms around i are shown. Note also that the random coefficients $\tilde{L}_j^n$ are applied to all grid points for generality, though in practical calculations on current machines, at least the case $j=i$ would be always deterministic with $\tilde{L}_i^0=1$ and $\tilde{L}_i^m=0$ for $m>0$.

Following Eq. (17), we can define

$$\boldsymbol{W}^{n+1}=\begin{bmatrix}\boldsymbol{V}^{n+1}\\ \boldsymbol{V}^{n}\\ \boldsymbol{V}^{n-1}\\ \vdots\\ \boldsymbol{V}^{n-M}\end{bmatrix} \tag{33}$$

where $\boldsymbol{V}^{n+1}=[u_1^{n+1}\ u_2^{n+1}\ \ldots\ u_N^{n+1}]^T$ and write the evolution equation as

$$\boldsymbol{W}^{n+1}=\tilde{\boldsymbol{C}}^{n}\boldsymbol{W}^{n}. \tag{34}$$

The matrix $\tilde{\boldsymbol{C}}^n$ is now

$$\tilde{\boldsymbol{C}}^{n}=\begin{bmatrix}\tilde{\boldsymbol{A}}_0^n & \tilde{\boldsymbol{A}}_1^n & \tilde{\boldsymbol{A}}_2^n & \ldots & \tilde{\boldsymbol{A}}_M^n\\ \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \ldots & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \ldots & \boldsymbol{0}\\ & & \ddots & & \\ \boldsymbol{0} & \ldots & \ldots & \boldsymbol{I} & \boldsymbol{0}\end{bmatrix} \tag{35}$$

where each $N \times N$ matrix $\tilde{\boldsymbol{A}}_m^n$ contains the terms in Eq. (32) with the factors $\tilde{L}_j^m$ for all available j's. Since the stability conditions in Eq. (22)–(24) apply in the general case too, we now turn to the condition (using the infinity norm):

$$\left\|\tilde{\boldsymbol{C}}^n\right\|_\infty \leq 1, \tag{36}$$

which assures stability as discussed above. From the structure of $\tilde{\boldsymbol{C}}^n$, we can see that except for the first N rows, the absolute row sum is always 1. Thus to ensure Eq. (36), the absolute row sum of the first N rows must be less or equal than unity. For the i-th row, that is,

$$\begin{aligned}
&\left|c_{-S}\tilde{L}_{i-S}^0\right| + \cdots + \left|c_0\tilde{L}_i^0\right| + \cdots + \left|c_S\tilde{L}_{i+S}^0\right| + \\
&\left|c_{-S}\tilde{L}_{i-S}^1\right| + \cdots + \left|c_0\tilde{L}_i^1\right| + \cdots + \left|c_S\tilde{L}_{i+S}^1\right| + \\
&\vdots \\
&\left|c_{-S}\tilde{L}_{i-S}^M\right| + \cdots + \left|c_0\tilde{L}_i^M\right| + \cdots + \left|c_S\tilde{L}_{i+S}^M\right| \leq 1
\end{aligned} \tag{37}$$

or in more compact form

$$\sum_{m=0}^{M}\sum_{j=-S}^{+S}\left|c_j\tilde{L}_{i+j}^m\right| \leq 1. \tag{38}$$

Since $\tilde{L}_{i+j}^m$ can only take non-negative values (0 or 1), we can further rewrite this as

$$\sum_{m=0}^{M}\sum_{j=-S}^{+S}|c_j|\tilde{L}_{i+j}^m = \sum_{j=-S}^{+S}|c_j|\sum_{m=0}^{M}\tilde{L}_{i+j}^m \leq 1 \tag{39}$$

and because, by construction, the Lagrange polynomials $\tilde{L}_i^m$ satisfy the property of $\sum_{m=0}^{M}\tilde{L}_i^m = 1$, it follows that Eq. (36) is satisfied when

$$\sum_{j=-S}^{+S}|c_j| \leq 1. \tag{40}$$

This condition is identical to Eq. (29). Thus we have shown that if the original synchronous scheme is stable (in the sense determined by Eq. (29)), the corresponding asynchronous scheme is stable as well. A particular application of this has been presented in Eq. (27).

Another example of interest which will be used later is the advection–diffusion equation given by Eq. (1) for $D = 2$ with $\beta_1 = -c$, $\beta_2 = \alpha$. Using second-order central differences for both first and second derivatives we can write

$$u_i^{n+1} = u_{i+1}^n(r_\alpha - r_c/2) + u_i^n(1 - 2r_\alpha) + u_{i-1}^n(r_\alpha + r_c/2). \tag{41}$$

The stability condition (40) then can be written in this particular case as

$$|r_\alpha - r_c/2| + |1 - 2r_\alpha| + |r_\alpha + r_c/2| \leq 1, \tag{42}$$

which implies the following when $r_c < 1$:

$$r_c/2 \leq r_\alpha \leq 1/2, \tag{43}$$

for both synchronous and asynchronous schemes. (An additional particular combination of parameters that satisfy Eq. (42), is when r_c and r_α are exactly 1 and 1/2 respectively, though this is known to present consistency issues [13].) The entire stability region just obtained (which provides sufficient conditions) will be discussed further in the context of the generality of these results below.

3.4. Other considerations on the generality of the results

A first remark on the generality of the result is that stability results are independent of the statistics of the delay $\tilde{k}_i$. Therefore, independently of the statistical characteristics of the communications in a particular computing system, stability is assured. This is also relevant in the context of the domain decomposition on parallel machines since the delays at different boundary points may be different depending on whether neighboring PEs are in the same node or far apart in the network topology, for example. Furthermore, while randomness is included in every entry of $\tilde{\boldsymbol{C}}^n$ (see Eq. (32)), as mentioned above interior points can be treated as possessing a Dirac delta PDF around zero and boundary points with delays governed by arbitrary statistics. An important implication from our results, then, is that stability is independent of number of PEs. As

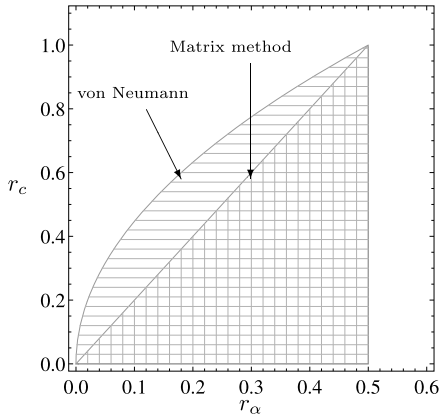

Fig. 2. Stability region for the advection–diffusion equation. Horizontal dashed pattern: von Neumann method (necessary and sufficient conditions). Vertical dashed pattern: matrix method with bounded ∞-norm (sufficient conditions).

we will see below, however, accuracy does depend on both number of PEs as well as the statistical characteristics of the delays.

A second remark is regarding the use of different norms in our results. While we have used the ∞-norm, any element-wise induced norm will result in similar conclusions. The reason for this is that the randomness in the scheme is essentially a switching mechanism between two schemes with the same coefficients. The result is a relocation of that coefficient in the matrix but its value is unchanged. Also note that a consistent choice has to be made as to which norm is used. Our results show that if the original synchronous scheme is stable in the ∞-norm, the asynchronous scheme is stable too in that norm. However, classical inequalities between norms would allow us to express some stability conditions in terms of different norms. For example, because $\|\boldsymbol{A}\|_\infty/\sqrt{N} \le \|\boldsymbol{A}\|_2 \le \|\boldsymbol{A}\|_F$, ($\|\cdot\|_F$ is the Frobenius norm) we can guarantee stability in the ∞-norm sense when $\|\tilde{\boldsymbol{C}}^n\|_2 \le 1/\sqrt{N}$ or $\|\tilde{\boldsymbol{C}}^n\|_F \le 1/\sqrt{N}$, which are, obviously, only sufficient conditions.

Third, we note that by requiring each individual matrix in Eq. (24) to possess a norm bounded by one, we are also requiring that amplification does not occur at any single step. This is typically a stronger condition that simply requiring that perturbations (error) decrease to zero as $n \to \infty$. It is well known that in some cases, the latter can lead to temporary growth of perturbations before it is asymptotically damped [16]. Since with asynchronous schemes random factors affect the solution, assuring no amplification at every step seems a sensible choice. However, depending on the accuracy of the solution needed in particular application, this restriction may be relaxed and the stability range may be extended. In fact for some problems, Eq. (36) could lead to overly restrictive stability conditions, even failing to predict the existence of stability regions. The smaller stability region predicted by the matrix method requiring no growth uniformly in n is known [15]. For example, for a advection–diffusion equation, necessary and sufficient conditions for stability can be obtained exactly from a von Neumann analysis. The result is $r_c^2 \le 2r_\alpha \le 1$ which ought to be compared to Eq. (43). The different stability regions covered by these two methods are shown in Fig. 2. Clearly, the matrix method stability region is contained within the von Neumann stability region, as expected. Sharper bounds on the inequalities leading to Eq. (40) is part of our own ongoing research.

In terms of the time step size, the stability condition Eq. (43) leads to overly restrictive time steps and a restriction on the grid size which, although it leads to stable results (sufficient condition), in reality does not exists [17]. In terms of the necessary and sufficient stability criterion, we can obtain the conditions in terms of the time step as:

$$\Delta t \le \min\left(\frac{2\alpha}{c^2}, \frac{\Delta x^2}{2\alpha}\right). \tag{44}$$

Fourth, boundary conditions were assumed periodic in our analysis. However, since our analysis is based on a matrix description of the finite difference approximation, it is straightforward to include other boundary conditions as is commonly done in standard stability analysis of synchronous schemes. For example, Dirichlet boundary conditions at both ends of the domain will change the first and N-th rows of $\tilde{\boldsymbol{C}}^n$ to contain only a 1 at the first and N-th columns, respectively. Clearly, from a ∞-norm based analysis (i.e. Eq. (36)), this has no effect on the stability regions obtained for periodic domains.

Finally, we comment of the extension of these results to time discretizations involving more than two steps. In these cases, Eq. (28) has to be extended to multiple time levels leading to an expression similar to Eq. (30) but the term inside of the summations will contain both deterministic and random terms stemming from the original scheme and the delayed information respectively. The characteristics of such schemes are more complicated and the stability analysis may depend also on how the scheme is implemented in practice as values at different time lags may be used in different ways. The discussion of such schemes will, thus, be published elsewhere.

4. Consistency and accuracy

4.1. General considerations

We now turn to the issue of the consistency and accuracy. A finite difference equation (FDE) at a point in the discretized domain, is an approximation of a partial differential equation (PDE) with associated truncation error (E), i.e., PDE = FDE + E. Typically the truncation error is conveniently studied using a Taylor expansion of each term in the FDE about a grid point i and time level n.

Let us consider again Eq. (8) with $\tilde{n} = n$ (synchronous) and rewrite it by substituting the expansions $u_{i+1}^n = u_i^n + u'\Delta x + u''\Delta x^2/2! + u'''\Delta x^3/3! + \ldots$, $u_{i-1}^n = u_i^n - u'\Delta x + u''\Delta x^2/2! - u'''\Delta x^3/3! + \ldots$, and $u_i^{n+1} = u_i^n + \dot{u}\Delta t + \ddot{u}\Delta t^2/2! + \dddot{u}\Delta t^3/3! + \ldots$, where space derivatives (denoted by primes) and time derivatives (denoted by a dot over the variable) are always evaluated at (i, n) and therefore the subscripts and superscripts are omitted for convenience in the notation. After rearrangement, it is easy to show that, to leading order, the truncation error is

$$E_i^n = -\frac{\ddot{u}}{2}\Delta t + \frac{\alpha u''''}{12}\Delta x^2 + \mathcal{O}\left(\Delta t^2, \Delta x^4\right). \tag{45}$$

When $\Delta x \to 0$ and $\Delta t \to 0$, we can see that the truncation error has a limiting behavior $E_i^n \to 0$, showing that the FDE is consistent with the corresponding PDE at point i. The formal accuracy of a given scheme is defined as the power law exponent of the leading order terms in the truncation error. From Eq. (45), it is clear that the scheme Eq. (4) is first order in time and second order in space.

When asynchronous computations are allowed, $\tilde{n}$ is a random variable which can take values n, $n-1$, $n-2$, ... The truncation error at a particular point and time step, then, becomes also a random variable. If, for example, $\tilde{n} = n-1$ in Eq. (8), (i.e. $\tilde{k}_{i+1} = 1$), the truncation error is given by

$$\begin{aligned}\tilde{E}_i^n\big|_{\tilde{k}_{i+1}=1} = &-\frac{\ddot{u}}{2}\Delta t + \frac{\alpha u''''}{12}\Delta x^2 - \alpha\dot{u}\frac{\Delta t}{\Delta x^2} + \alpha\dot{u}'\frac{\Delta t}{\Delta x} - \frac{\alpha\dot{u}''}{2}\Delta t \\ &+ \mathcal{O}\left(\Delta x^3, \Delta t^2, \Delta x^p \Delta t^q\right)\end{aligned} \tag{46}$$

where we have extended the notation on the truncation error to indicate the location and value of the delay ($\tilde{k}_{i+1} = 1$). As indicated in Eq. (46), higher order terms involving the product of Δx and Δt are present due to data now being used at multiple time levels. These are here indicated by the variables p and q which are, in this case, bounded from below by -2 and 1, respectively.

The first two terms in Eq. (46) are identical to the leading order terms for the synchronous scheme (Eq. (45)). The additional terms are the result of the appearance of delayed values at $i+1$. It is interesting to observe that these contributions could actually *grow* as fast as $\Delta x^{-2} \sim N^2$ for fixed Δt as one refines the grid. Thus, consistency with the original governing equation requires careful considerations of how the time step size and grid spacing are reduced for more accurate solutions. We will expand on this issue momentarily.

For an arbitrary delay k at grid point $i+1$ (i.e. $\tilde{k}_{i+1} = k$) we can also find

$$\begin{aligned}\tilde{E}_i^n\big|_{\tilde{k}_{i+1}=k} = &-\frac{\ddot{u}}{2}\Delta t + \frac{\alpha u''''}{12}\Delta x^2 - \alpha k\dot{u}\frac{\Delta t}{\Delta x^2} + \alpha k\dot{u}'\frac{\Delta t}{\Delta x} - \frac{\alpha k\dot{u}''}{2}\Delta t \\ &+ \mathcal{O}\left(\Delta x^3, \Delta t^2, \Delta x^p \Delta t^q\right),\end{aligned} \tag{47}$$

which obviously reduces to Eq. (45) for $k = 0$ (no delay). Since, as discussed in Section 3, stability is governed by the parameter $r_\alpha = \alpha\Delta t/\Delta x^2$, it is convenient to rewrite the above equation as

$$\begin{aligned}\tilde{E}_i^n\big|_{\tilde{k}_{i+1}=k} = &-\frac{r_\alpha\ddot{u}}{2\alpha}\Delta x^2 + \frac{\alpha u''''}{12}\Delta x^2 - r_\alpha k\dot{u} + r_\alpha k\dot{u}'\Delta x - \frac{r_\alpha k\dot{u}''}{2}\Delta x^2 \\ &+ \mathcal{O}\left(\Delta x^3, \Delta t^2, \Delta x^p \Delta t^q\right),\end{aligned} \tag{48}$$

which shows the drastic effect of utilizing standard schemes in an asynchronous manner. Specifically, if r_α is kept constant in refining the discretization in time and space, then the third term on the right-hand-side represents a zeroth order contribution. That is, errors will not decrease under grid refinement. The scheme is thus not consistent with the original PDE.

This decrease in accuracy is not specific to the second-order space discretization used as an example. As discussed in more detail below, for any order of a (finite-difference) discretization of Eq. (1), asynchrony will result in truncation errors containing terms proportional to $\tilde{k}_i\Delta x^{-d}\Delta t$ where d goes from 1 to D in the most general case. If, because of a stability constraint, the time step size is determined according to a condition $\Delta t \sim \Delta x^p$ where $p < D$, then terms that grow with increasing resolution (i.e. terms proportional to $1/\Delta x^{D-p}$) will appear. If $p = D$, then zeroth-order terms will be present.

We note however, that the remarks above correspond to the truncation error at a point with a given value of $\tilde{n}$. However, delays are only expected at PE boundaries ($i \in I_B$) and not at interior points. Furthermore, even for grid points in I_B, delays

are random and will depend on the particular realization of the simulation. A more general description of accuracy is thus needed to study these asynchronous schemes. In particular, accuracy will depend, in principle, on the original scheme, the number of PEs (number of boundaries where delays are expected), and the statistics of $\tilde{k}_i$ at each PE boundary which in turn will typically depend on architectural details of computing nodes, processors, network, etc. as well as the specific resource usage (e.g. by other users) at the time of the simulations. We thus now proceed to introduce some definitions in which all these elements can be taken into account.

4.2. Statistical description of truncation error

Consider a one-dimensional periodic domain with N grid points which is decomposed into P PEs leading also to P boundaries between processors. Our objective here is to quantify the error in order to obtain the order of accuracy. In doing so, we recognize that there are two elements that need to be considered in defining these concepts. First, the truncation error structure is not homogeneous in space. As discussed above for the heat equation, zeroth-order terms may appear at PE boundary points but the scheme is still second-order accurate in interior points. Second, even at boundary points the appearance of these terms is random. In particular, these additional terms due to asynchrony will appear with a probability $(1-p_{0[i+1]})$, where $p_{0[i+1]}$ is the probability of having no delay at point $i+1$, that is $\tilde{k}_{i+1}=0$. Thus, a probabilistic definition of the error seems appropriate in order to define convergence properties.

We first define two types of averages for a variable f: a space average and an ensemble average. In general, the space average could be taken over the entire domain or a subset of grid points (e.g. PE boundary points). If the average is over the entire domain the space average is $\langle f\rangle=\sum_{i=1,N} f_i/N$. If, instead, the average is taken over PE boundary points or interior points, the average is $\langle f\rangle_B=\sum_{i\in I_B} f_i/N_B$ or $\langle f\rangle_I=\sum_{i\in I_I} f_i/N_I$, respectively (the subscript in the angular brackets denotes the subset of grid points over which the average is taken). Ensemble averages, which take into account the stochastic nature of the delays, will be denoted by an overline $\overline{f}$.

Consider the (space and ensemble) average error over the entire domain at time step n:

$$\langle\overline{E}\rangle=\frac{1}{N}\sum_{i=1,N}\overline{E_i^n}. \tag{49}$$

Since the truncation error for $i\in I_B$ is random and for $i\in I_I$ is not, it is convenient to split the sum into interior and boundary points

$$\langle\overline{E}\rangle=\frac{1}{N}\left[\sum_{i\in I_I}E_i^n+\sum_{i\in I_B}\overline{\tilde{E}_i^n\big|_{\tilde{k}_{i+1}}}\right] \tag{50}$$

where ensemble averages are only needed for the asynchronous regions at PE boundaries (second term on the right-hand-side). We now proceed to estimate the two different terms inside the brackets.

For interior points, there is no random component and the error at each i is simply given by Eq. (45). Thus, to leading order we can write

$$\sum_{i\in I_I}E_i^n\approx\sum_{i\in I_I}\left(-\frac{\ddot{u}}{2}\Delta t+\frac{\alpha u''''}{12}\Delta x^2\right), \tag{51}$$

or in terms of r_α

$$\begin{aligned}\sum_{i\in I_I}E_i^n&\approx\sum_{i\in I_I}\left(-\frac{\ddot{u}r_\alpha}{2\alpha}+\frac{\alpha u''''}{12}\right)\Delta x^2\\&\approx\Delta x^2\sum_{i\in I_I}K_s\\&\approx N_I\langle K_s\rangle_I\Delta x^2.\end{aligned} \tag{52}$$

For the last step we have used the fact that for the space average of $K_s\equiv-(\ddot{u}r_\alpha/2\alpha)+(\alpha u''''/12)$, which is defined here for convenience (subscript s stands for synchronous), the sum contains N_I terms corresponding to the number of interior grid points. For a central-difference scheme with a stencil size $2S+1$, the number of grid points in the domain where delays can be observed (if delays are expected only on one side of the stencil) is SP where P is, as before, the number of PEs which, for periodic boundary conditions, is equal to the number of PE boundaries. Thus the number of grid points in I_I is $N_I=(N-SP)$ which appears multiplying $\langle K\rangle_I$ in Eq. (52). The assumption of delays only on one side does not affect the generality of the results and is made only for clarity in the exposition. If delays are present on both sides, some expressions will have different prefactors but all conclusions regarding the accuracy of the scheme will still be valid.

For the second sum over I_B in Eq. (50), the truncation error is given by Eq. (47). Clearly when $\tilde{k}_{i+1}=0$, which happens with probability $p_{0[i+1]}$, we recover a synchronous scheme and the second sum is similar to Eq. (52) with the difference

that the prefactor is now the number of grid points at boundaries, which is $N_B = SP$. If, on the other hand $\tilde{k}_{i+1} > 0$, then the other terms in the truncation error appear. The ensemble average, to leading order, can be then estimated as

$$\begin{aligned}
\overline{\tilde{E}_i^n|_{\tilde{k}_{i+1}}} &\approx \sum_{k=0}^{L-1} p_{k[i+1]} \tilde{E}_i^n|_{\tilde{k}_{i+1}=k} \\
&\approx \sum_{k=0}^{L-1} p_{k[i+1]} \left(-\frac{\ddot{u}}{2}\Delta t + \frac{\alpha u''''}{12}\Delta x^2 - \alpha k \dot{u}\frac{\Delta t}{\Delta x^2} + \alpha k \dot{u}'\frac{\Delta t}{\Delta x} - \frac{\alpha k \dot{u}''}{2}\Delta t \right) \\
&\approx \left(-\frac{\ddot{u}}{2}\Delta t + \frac{\alpha u''''}{12}\Delta x^2 \right) + \sum_{k=0}^{L-1} p_{k[i+1]} \left(-\alpha k \dot{u}\frac{\Delta t}{\Delta x^2} + \alpha k \dot{u}'\frac{\Delta t}{\Delta x} - \frac{\alpha k \dot{u}''}{2}\Delta t \right) \\
&\approx \left(-\frac{\ddot{u}}{2}\Delta t + \frac{\alpha u''''}{12}\Delta x^2 \right) + \left(-\alpha \dot{u}\frac{\Delta t}{\Delta x^2} + \alpha \dot{u}'\frac{\Delta t}{\Delta x} - \frac{\alpha \dot{u}''}{2}\Delta t \right) \sum_{k=0}^{L-1} p_{k[i+1]} k
\end{aligned} \tag{53}$$

where we have used the fact that $\sum_{k=0,L} p_{k[i+1]} = 1$. Finally, we can write

$$\overline{\tilde{E}_i^n|_{\tilde{k}_{i+1}}} \approx \left(-\frac{\ddot{u}}{2}\Delta t + \frac{\alpha u''''}{12}\Delta x^2 \right) + \left(-\alpha \dot{u}\frac{\Delta t}{\Delta x^2} + \alpha \dot{u}'\frac{\Delta t}{\Delta x} - \frac{\alpha k \dot{u}''}{2}\Delta t \right) \overline{\tilde{k}_{i+1}} \tag{54}$$

where $\overline{\tilde{k}_{i+1}}$ is the ensemble average of the delay given by $\overline{\tilde{k}_{i+1}} = \sum_{k=0,L} p_{k[i+1]} k$. The first and second parenthesis contain the synchronous and asynchronous terms, respectively. Clearly, for synchronous conditions we have $\tilde{k}_{i+1} = 0$ always, and thus $\overline{\tilde{k}_{i+1}} = 0$ making the last term in Eq. (54) equal to zero. It is also interesting to note that the average error increases linearly with the mean delay.

In terms of r_α we can rewrite Eq. (54) as

$$\overline{\tilde{E}_i^n|_{\tilde{k}_{i+1}}} \approx \left(-\frac{r_\alpha \ddot{u}}{2\alpha}\Delta x^2 + \frac{\alpha u''''}{12}\Delta x^2 \right) + \left(-r_\alpha \dot{u} + r_\alpha \dot{u}'\Delta x - \frac{r_\alpha \dot{u}''}{2}\Delta x^2 \right) \overline{\tilde{k}_{i+1}}. \tag{55}$$

If we now take the sum over I_B following Eq. (52), we obtain

$$\sum_{i \in I_B} \overline{\tilde{E}_i^n|_{\tilde{k}_{i+1}}} \approx N_B \langle K_s \rangle_B \Delta x^2 + \left(-r_\alpha N_B \langle \dot{u} \rangle_B + r_\alpha N_B \langle \dot{u}' \rangle_B \Delta x - N_B \frac{\langle \dot{u}'' \rangle_B}{2}\Delta x^2 \right) \bar{\tilde{k}}. \tag{56}$$

For simplicity, we have assumed that the statistics of the delays are homogeneous in space such that $\overline{\tilde{k}_{i+1}}$ is independent of i (i.e. the $\tilde{k}_{i+1}$ are i.i.d.) and thus the subscript $i+1$ has been dropped. If the statistical behavior of $\tilde{k}_{i+1}$ does depend on i, one can still take this into account in the above formulation by using the space average $\langle \bar{\tilde{k}} \rangle_B$ instead of simply $\bar{\tilde{k}}$.

Finally we substitute Eq. (56) and Eq. (52) into Eq. (50), and using the fact that $N_I \langle f \rangle_I + N_B \langle f \rangle_B = N \langle f \rangle$ for any quantity f, we obtain

$$\langle \bar{E} \rangle \approx \langle K_s \rangle \Delta x^2 + \frac{N_B}{N} \bar{\tilde{k}} \left(-r_\alpha \langle \dot{u} \rangle_B + r_\alpha \langle \dot{u}' \rangle_B \Delta x - r_\alpha \frac{\langle \dot{u}'' \rangle_B}{2}\Delta x^2 \right). \tag{57}$$

A number of interesting observations can be made from this result. First, for zero delays we recover second order convergence from the first term of Eq. (57). Second, we note that even a very small amount of asynchrony (which results in $\bar{\tilde{k}} > 0$) decreases the order of convergence significantly. In particular as Δx decreases to zero, the leading order term is the first one in parenthesis. Asymptotically, we then have

$$\begin{aligned}
\langle \bar{E} \rangle &\approx -\frac{N_B}{N} \bar{\tilde{k}} r_\alpha \langle \dot{u} \rangle_B \\
&\approx -S \frac{P}{N} \bar{\tilde{k}} r_\alpha \langle \dot{u} \rangle_B,
\end{aligned} \tag{58}$$

where we have used $N_B = SP$. For all other parameters constant, we can now write (using $\Delta x = l/N$)

$$\langle \bar{E} \rangle \sim \frac{P}{N} \sim P \Delta x, \tag{59}$$

showing that the original second-order scheme drops to first order (for the average error $\langle \bar{E} \rangle$) when asynchrony is present for a constant number of PEs, P. However, it is interesting to note the situation in which the number of PEs grow with the size of the problem N, which would correspond to so-called weak scaling when the parallel performance of a code is investigated. This is indeed, a common situation in applications where an increase in computational power available

is typically used to increase computational work through finer grids to achieve more realistic conditions or include more physical content in the simulations. In such a case, P/N is constant and Eq. (59) shows that the error does not depend on Δx (that is a zeroth-order term) rendering the scheme inconsistent with the original equations. This highlights an interesting aspect of asynchronous schemes (and perhaps a number of numerical algorithms designed to run at so-called exascale [3]), namely, the tight link between the numerical scheme and the computational system on which they are run. This is explored in more detail later on.

The decrease in formal accuracy we just explored, while asymptotically true, might not be an issue in some practical simulations. From Eq. (57), it can be seen that if simulations are performed with a large number of grid points per PE (small N_B/N) and on a fast network (such that $\bar{\tilde{k}}$ is very small), then for realistic grids, the asynchronous contribution to the error (the entire second term on the right-hand-side) may be small and errors follow the synchronous second-order accuracy. The exact transition from the formal accuracy of the base scheme to zeroth or first order in asynchronous schemes cannot, in principle, be determined a priori. In Section 5 we shows actual numerical experiments where these issues are explored.

4.3. Generalizations

The results in the previous section were derived for a particular discretization scheme of the heat equation. In this section we show how the conclusions found in that case indeed extend to a much wider set of situations.

First, we note that while we have used the average error $\langle \bar{E} \rangle$, the asymptotic result Eq. (58) also holds if one uses an L_1 norm to characterize the error (i.e. the space average of the absolute value of the error). This is clear if one considers that one of the terms in Eq. (55) (the zeroth-order term) is much larger than all the others for large N. After space averaging, we obtain

$$\left\langle |\overline{E}| \right\rangle \approx (N_B/N)\overline{|\tilde{k}|} r_\alpha \left\langle |\dot{u}| \right\rangle_B \sim P/N \sim P\Delta x. \tag{60}$$

This measure is typically more sensible in practice than the simple mean value since this eliminates the possibility of error cancellations during the averaging procedure.

The decrease of accuracy when common finite difference schemes are used asynchronously, is in fact very general. For example, if the second derivative on the right-hand-side of the heat equation is discretized using a longer stencil (higher order of accuracy), the truncation error pertaining to asynchronous terms retain the same form of the second parenthesis in Eq. (54), the only difference being the numerical coefficients for the different terms. For example, for a fourth-order central discretization of the second derivative $\partial^2 u/\partial x^2 = (-u^{\tilde{n}}_{i+2} + 16u^{\tilde{n}}_{i+1} - 30u^n_i + 16u^n_{i-1} - u^n_{i-2})/12\Delta x^2 + \mathcal{O}(\Delta x^4)$, where both values at $i+1$ and $i+2$ are delayed, we obtain an analogous to Eq. (54)

$$\overline{\tilde{E}^n_i|_{\tilde{k}_{i+1}=\tilde{k}_{i+2}=k}} \approx \left(-\frac{\ddot{u}}{2}\Delta t - \frac{\alpha u''''}{90}\Delta x^4\right) + \left(-\frac{5}{4}\alpha k\dot{u}\frac{\Delta t}{\Delta x^2} - \frac{7}{6}\alpha k\dot{u}'\frac{\Delta t}{\Delta x} - \frac{\alpha k\dot{u}''}{2}\Delta t\right)\overline{\tilde{k}_{i+1}} \tag{61}$$

with the first parenthesis being identical to the truncation error for the synchronous scheme, that is $\mathcal{O}(\Delta t, \Delta x^4)$. Again, asymptotically the first term in the second parenthesis will dominate (if one uses a constant r_α for stability) leading to Eq. (58) with a prefactor 5/4.

The generality of this results can be easily seen if one of the terms in the spatial derivative of a consistent scheme is delayed. In that case, the Taylor expansion for that particular term will contain the standard spatial error related to the derivative and terms in Δt due to the necessary expansion in time. More formally, consider an approximation to the d-th derivative at location i as

$$\frac{\partial^d u}{\partial x^d} \approx \sum_{j=-S}^{S} \frac{b_j u^n_{i+j}}{\Delta x^d}. \tag{62}$$

Assume that the value of the function experiences a delay of k at $i+1$, that is $\tilde{k}_{i+1} = k$. Then,

$$\frac{\partial^d u}{\partial x^d} \approx \frac{\cdots + b_1\, u^{n-k}_{i+1} + \ldots}{\Delta x^d}, \tag{63}$$

where b_1 is the appropriate coefficient at $i+1$ for the given discretization. The Taylor expansion of the only term explicitly written in the numerator is $u^{n-k}_{i+1} = u^n_i + [u'\Delta x + u''\Delta x^2/2! + u'''\Delta x^3/3! + \ldots] + [-k\dot{u}\Delta t + \ddot{u}k^2\Delta t^2/2! - k^3\dddot{u}/3! + \ldots] + [-k\dot{u}'\Delta x\Delta t - k\dot{u}''\Delta x^2\Delta t/2 + \ldots]$. The first bracket is identical to the expansion for synchronous schemes. The last bracket contains some cross terms involving both Δt and Δx. It is the second bracket, however, that contains the more problematic contributions. It comprises terms in powers of Δt starting with $k\dot{u}\Delta t$. When introduced into Eq. (63), the result is the term

$$C_d k\dot{u}\frac{\Delta t}{\Delta x^d}, \tag{64}$$

where C_d is a combination of numerical prefactors determined by the original PDE, and the discretization scheme used. In general, the leading order terms in the local truncation error (i.e. at a particular point) of the discretized general equation (1) due to asynchrony will be

Table 2
Summary of leading terms in the local (for $i \in I_B$), and average truncation error due to asynchronicity for different scenarios. First column indicates the conditions under which grid refinement (i.e. $\Delta x \to 0$) is conducted. The variable $r_D = \beta_D \Delta t/\Delta x^D$ corresponds to the resulting parameter from the highest derivative in the problem (for the heat, wave and advection–diffusion equations it will correspond to r_c, r_α and r_α respectively).

Parameters held constant	Local E_i^n	$\langle \bar{E} \rangle$ or $\langle \overline{\lvert E \rvert} \rangle$
P, r_D	1	Δx
P, Δt	$1/\Delta x^D$	$1/\Delta x^{D-1}$
P/N, r_D	1	1
P/N, Δt	$1/\Delta x^D$	$1/\Delta x^D$
P, $\Delta t/\Delta x^q$	Δx^{q-D}	Δx^{q-D+1}
P/N, $\Delta t/\Delta x^q$	Δx^{q-D}	Δx^{q-D}

$$\sum_{d=1}^{D} C_d \beta_d k\dot{u} \frac{\Delta t}{\Delta x^d}. \tag{65}$$

Note that this local truncation error will appear only for the subset of grid points $i \in I_B$.

When Δt is chosen as a power law in Δx (due to, e.g., stability constraints) clearly the highest derivative in the original PDE will be the leading order term in the truncation error. In the case of the advection–diffusion equation, for example, the original PDE contains first and second order derivatives, and then the two terms $\Delta t/\Delta x$ and $\Delta t/\Delta x^2$ will be present (actually the former will be present even if the original equation contained only a second derivative because, as shown under Eq. (63), higher-order terms also include $\Delta x \Delta t$ which when divided by Δx^2 yields a term $\Delta t/\Delta x$).

A few important conclusions can be made about the truncation error induced by asynchrony. First, if the space resolution is fixed (i.e. fixed Δx), the *local* truncation error will be first order in time regardless of the order of accuracy of the time integration. Second, if a stability criterion requires a condition on $\beta_D \Delta t/\Delta x^D$, then the local truncation error will always have zeroth-order terms which will not vanish as resolution increases. As clear from our analysis leading to Eq. (60), however, if P is constant, the space average over I_B leads to an additional factor Δx. Thus, the *average* truncation error will be first order.

The stability criterion based on $\beta_D \Delta t/\Delta x^D$ can be analytically shown for simple model equations: for the wave equation one needs $r_c = c\Delta t/\Delta x \leq 1$ and for the heat equation $r_\alpha = \alpha \Delta t/\Delta x^2 \leq 1/2$. For more complex equations such as the advection-diffusion equation which contain both convective as well as diffusive terms, if the time step is selected according to the diffusive condition $\Delta t = r_\alpha \Delta x^2/\alpha$ then the local truncation error will contain terms of order $r_\alpha \Delta x$ and r_α (i.e. first and zeroth order, respectively). If on the other hand the time step is chosen according to a convective condition $\Delta t = r_c \Delta x/c$ then the truncation error will contain terms of the form r_c and $r_c/\Delta x$. The magnitude of the last term actually increases under grid refinement; when averaged in space, though, the truncation error due to asynchrony becomes zeroth order. However, in Section 3.4 we mentioned that stability for this equation requires Δt to be controlled by r_α asymptotically (Eq. (44)) and thus the average truncation error is asymptotically first order.

Due to the number of possible combinations of parameters, we include a summary of the results in this section in Table 2 for the general equation (1) with an arbitrary finite difference discretization in space and a two-level discretization in time. The second and third columns indicate the leading order term in the local (i.e. at a grid point) and the average truncation error, respectively for the conditions listed in the first column. The first two rows correspond to the case of a fixed number of PEs; the third and fourth correspond to the case of P/N constant (i.e. weak scaling); the last two rows correspond to a more general setting with $\Delta t \sim \Delta x^q$ (to be discussed below).

We conclude this section by noting that an important finding of our analysis, is that the numerical characteristics of schemes used in an asynchronous fashion to exploit extreme levels of parallelism, are intrinsically linked to architectural details of the computational system as well as the manner in which the problem is scaled up. For example, for fixed number of PEs, we have shown the scheme is first-order accurate on the average error. However, if both problem size as well as PE count grow in proportion to each other, the scheme is inconsistent. This, can clearly be alleviated if Δt is chosen in a different way than what a stability condition would suggest. For example, for the heat equation with a first-order forward difference in time and a second-order central difference in space, choosing $\Delta t \propto \Delta x^3$, would ensure a second-order formal order of convergence for the average error. If a general relation $\Delta t \sim \Delta x^q$ is used, then the resulting schemes will have the convergence properties described in the last two rows of Table 2. Obviously, for realistic large-scale problems, a value of q beyond that required for stability reasons, may likely lead to prohibitively small time steps which may render the entire simulation unfeasible.

5. Numerical simulations

To verify the properties of the asynchronous schemes analyzed in previous sections, we now turn to actual applications of the schemes. Our focus will be on the advection–diffusion equation as it contains both first and second derivatives.

The interest of this equation is well known in fluid mechanics as it governs the time evolution of fluid flow phenomena involving convection and diffusion processes. It is also of interest here since analytical solutions are known against which we can compare our theoretical developments. The equation is

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \alpha \frac{\partial^2 u}{\partial x^2} \tag{66}$$

where $u = u(x,t)$ is the velocity, c is the constant convection speed and α the viscosity coefficient.

5.1. Implementation details

We assume an initial condition given by a sum of sinusoidal waves in a periodic domain of length 2π:

$$u(x,0) = \sum_{\kappa} A(\kappa)\sin(\kappa x + \phi_\kappa) \tag{67}$$

where κ is the wavenumber, and $A(\kappa)$ and ϕ_κ are the amplitude and phase angle corresponding to a particular wavenumber κ. The analytical solution of Eq. (66) and Eq. (67), indicated by a subscript a, is readily found to be

$$u_a(x,t) = \sum_{\kappa} e^{-\alpha\kappa^2 t} A(\kappa)\sin(\kappa x + \phi_\kappa - ct). \tag{68}$$

The reason for including the phase ϕ_κ is to avoid situations in which the PE boundaries coincide with specific features of the solution which may lead to very particular conditions in the truncation error. As an example, suppose we solve the heat equation with initial conditions given by a single sine wave and that the number of processors P is such that the PE boundaries are co-located with the zero crossings of this initial condition. From Eq. (68) with $c = 0$, it is clear that the solution at the zero crossings (and PE boundaries) will not change with time, that is $\dot{u} = 0$. Thus, the average over all boundary points will also vanish: $\langle \dot{u} \rangle_B = 0$. By looking at Eq. (57), we see that the leading order term vanishes and thus the resulting scheme is of higher order that it would in a more general condition. To avoid these very particular cases, the phases are chosen randomly for different wavenumbers and the results averaged over this space too.

We discretize Eq. (66) using second order central differences in space and a forward first order difference in time. For grid points close to a PE boundary on the right, we have

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_{i+1}^{\tilde{n}} - u_{i-1}^n}{2\Delta x} = \alpha \frac{u_{i+1}^{\tilde{n}} - 2u_i^n + u_{i-1}^n}{\Delta x^2}. \tag{69}$$

A similar expression can be written for grid points close the PE boundary on the left, which will have delays on $i-1$, instead of $i+1$.

In order to have complete control over the statistics of message delays and compare against the theoretical predictions in previous sections, we use random number generators to simulate these delays. In particular, the delays on both sides are drawn from uniform distributions in the interval $[0,1]$ with some initial random seed. For a given L, we arbitrarily set the probabilities $\{p_{0[i+1]}, p_{1[i+1]}, \ldots, p_{L-1[i+1]}\}$ corresponding to delays $\tilde{k}_{i+1} = 0, 1, 2, \ldots, L-1$ respectively by comparing each random number to the corresponding partition of $[0,1]$ into L bins. Since in our numerical experiments we use i.i.d. random sequences at the different PE boundaries, there is no dependence on location and we will thus drop the subscript $i+1$ for simplicity and write $\{p_0, p_1, \ldots, p_{L-1}\}$. For example, for $L = 3$, the set $\{p_0, p_1, p_2\} = \{0.6, 0.3, 0.1\}$ represents the situation where the probability of having $\tilde{k}_{i+1} = 0$ (i.e. no delay), $\tilde{k}_{i+1} = 1$ and $\tilde{k}_{i+1} = 2$ is 0.6, 0.3 and 0.1, respectively. In this case the mean $\overline{\tilde{k}_{i+1}}$ is easily found to be $\sum_{k=0,2} p_k k = 0.5$. Ensemble averages in practice are taken by running multiple simulations with different initial seeds for the random number generator.

To compare against our theoretical predictions in previous sections, we define the error at grid point i and time level n as $E_i^n = u_i^n - u_a(x_i, t_n)$ with $u_a(x,t)$ from Eq. (68) and take the different averages presented in Section 4.

5.2. Results

In Fig. 3 we show a typical realization of a simulation with $N = 128$, $P = 4$, $L = 2$ and $\{p_0, p_1\} = \{0.3, 0.7\}$ and an initial condition given by a single wave with $\kappa = 2$. In part (a) we show the time evolution of velocity obtained with synchronous (dashed red line) and asynchronous (solid black line) schemes. As expected, the initial wave is convected to the right at a speed of c with a decreasing amplitude due to the effect of viscosity. From this figure the difference between the two solutions is hardly seen. However, this is apparent in part (b) where we show the error for both schemes. As expected the asynchronous solution presents higher errors. We also see that errors seem to be larger at PE boundaries (vertical dashed lines) and spread to inner regions of the domain as time proceeds. This is not surprising as the evolution of error is, for linear systems, governed by the same PDE as u itself.

We note that since these errors appear at PE boundaries and are typically localized in space, the effect on statistical features of the solution may be small. For some applications, these localized (and bounded) errors may not affect specific quantities of interest. For example, in high-Reynolds number turbulence, a great deal of interest is in statistical features of

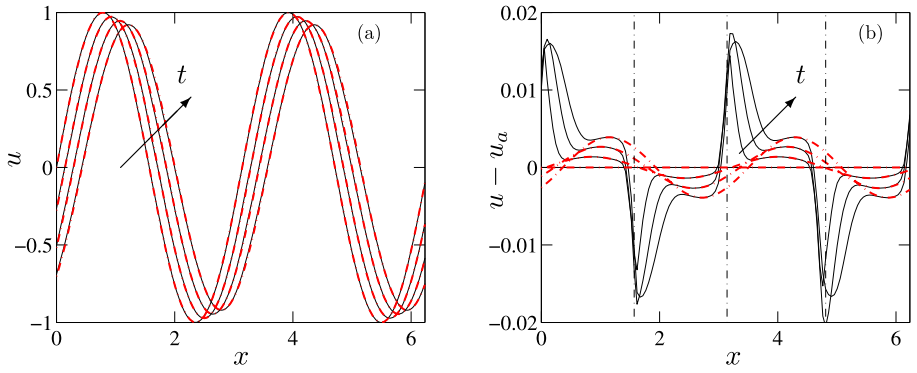

Fig. 3. Typical time evolution of the numerical solution of the advection–diffusion equation using synchronous (dashed red lines) and asynchronous (solid black lines) schemes. (a) The velocity field. (b) Error $E_i^n = u_i^n - u_a(x_i, t_n)$. Vertical dash-dotted lines correspond to PE boundaries. Simulation parameters: $N = 128$, $P = 4$, $L = 2$, with $\{p_0, p_1\} = \{0.3, 0.7\}$ for the asynchronous scheme. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

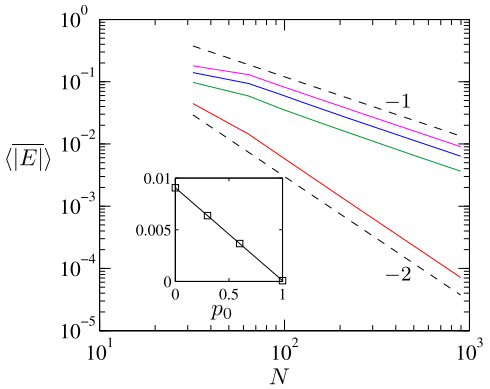

Fig. 4. Average error for the advection–diffusion equation with constant $r_\alpha = 0.1$. Different lines correspond to $p_0 = 1$ (red), 0.6 (green), 0.3 (blue), and 0.0 (magenta). Dashed lines with slopes -2 and -1 are included for reference. Inset: $\langle|\overline{E}|\rangle$ versus p_0 for the largest resolution in the main figure. Symbols are from simulations and the solid line is a linear best fit. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the flows due to the tremendous complexity of instantaneous fields. A few examples, include the calculation of the mean turbulent kinetic energy and dissipation, scaling of spectra and structure functions all of which are typically insensitive to (numerical) perturbations at scales smaller than the so-called Kolmogorov scale (e.g. [18,19]). High-order moments of velocity gradients, on the other hand, could become a challenge [20]. As we will show momentarily, however, these errors can be made smaller with finer grids with specifically designed schemes. Numerical experiments related to these issues will be presented in Section 6.

We now turn to a statistical description of the error with more realistic initial conditions. The initial condition comprises a range of wavenumbers (typically spanning a decade or so) with random phases and for simplicity only two possible delays are allowed ($L = 2$): $\tilde{k}_{i+1} = 0$ and 1. Note that in this case the probabilities are given by $\{p_0, p_1\}$ where $p_1 = 1 - p_0$. Thus, p_0 is enough to characterize the simulations.

The ensemble and space average error $\langle|\overline{E}|\rangle$ is shown in Fig. 4 as a function of resolution N for fixed r_α, and different set of the probabilities. The error was calculated at a normalized time of $tc/l = 0.08$ (l is the length of the domain). We have verified that the conclusions below are the same for longer times as well.

The synchronous (deterministic) case corresponds to $p_0 = 1$ (solid red line) which is seen to be second order. When asynchrony is introduced we have $p_0 < 1$ and we clearly see that the formal order of accuracy drops to one as predicted by the theory for constant values of P and r_α (Eq. (59)). For a given resolution, we can also see that the numerical value of the error increases as p_0 decreases, that is as the probability of having delayed values increases. The inset shows the same error as a function of p_0 (squares) at the largest resolution available. The good agreement between the data and the best linear fit (solid line through data points) support a linear relation between $\langle|\overline{E}|\rangle$ and p_0. This is consistent with Eq. (58) since the mean delay in the present case ($L = 2$) is simply $\bar{\tilde{k}} = 1 - p_0$.

To test the more general claim in Eq. (58) that the error is linearly dependent on the average value of the delay $\tilde{k}$, we have performed simulations with different values of L and probability sets with the same mean value of $\bar{\tilde{k}}$. A set of results is shown in Fig. 5 with different colors for different values of $\bar{\tilde{k}}$. The dependence on $\bar{\tilde{k}}$ instead of L or $\{p_0, p_1, \ldots\}$ is seen by

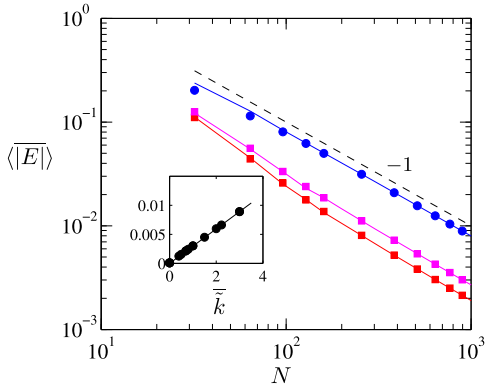

Fig. 5. Effect of mean value of delay on the average error for the advection–diffusion equation with constant $r_\alpha = 0.1$ and $P = 32$. Different colors correspond to different values of the average delay: $\bar{\bar{k}} = 0.7$ (red), 1.0 (magenta) and 3 (blue). Squares, solid lines and circles correspond to $L = 2$, 4 and 8 respectively. Dashed line with slope -1 is included for reference. Inset: $\langle|\overline{E}|\rangle$ as a function of $\bar{\bar{k}}$ for the largest resolution in the main figure with $P = 32$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

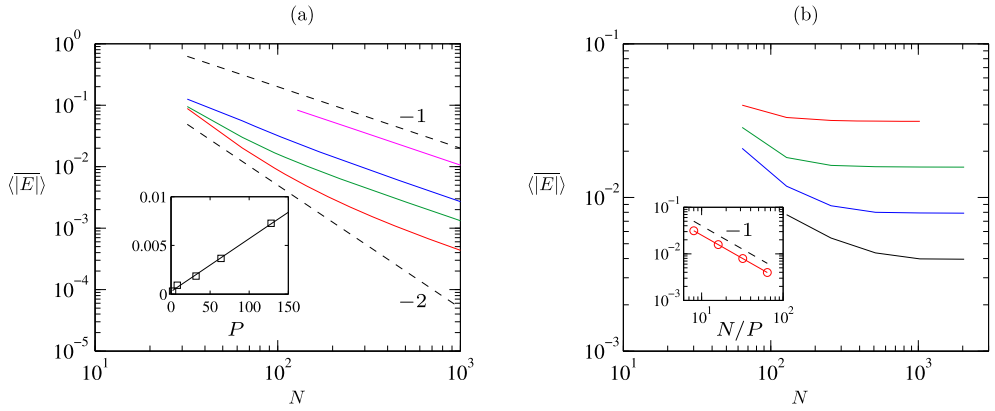

Fig. 6. Effect of number of PEs on the average error for the advection–diffusion equation with constant $r_\alpha = 0.1$. (a) Cases with constant P. Different lines correspond to $P = 2$ (red), 8 (green), 32 (blue) and 128 (magenta). Inset: $\langle|\overline{E}|\rangle$ versus P for the largest resolution ($N = 1024$) in the main figure. (b) Cases with constant P/N. Different lines correspond to $P/N = 1/64$ (black), 1/32 (blue), 1/16 (green) and 1/8 (red). Inset: $\langle|\overline{E}|\rangle$ versus N/P for the largest resolution ($N = 1024$) in the main figure. Dashed lines with slope -2 and -1 are included for reference. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

observing the collapse of, for example, the red squares ($L = 2$, $\{0.3, 0.7\}$) and the red solid line ($L = 4$, $\{0.6, 0.2, 0.1, 0.1\}$) both with $\bar{\bar{k}} = 0.7$. The same observation can be made for the other set of simulations at $\bar{\bar{k}} = 1$ (magenta) and 3 (blue). The inset shows a larger set of simulations with a wide range of values of $\bar{\bar{k}}$ (with different combinations of L and probabilities). The solid line is a best linear fit through the datapoints essentially confirming the linear dependence in Eq. (58).

Since in future computing systems, the number of PEs may be extremely large, it is also important to understand the behavior with P. Eq. (59) predicts that the average error is proportional to P/N. Thus, in an exercise of so-called strong scaling (that is, keeping the problem size, N, constant and progressively increasing the number of PEs, P) the average error $\langle|\overline{E}|\rangle$ is first order in Δx and grows linearly with P. These two predictions are indeed supported by our simulations. In Fig. 6(a) we clearly see asymptotic first order accuracy. For small PE counts (red line), we can see the transition from second order to first order which is also predicted by Eq. (57). As P increases, the numerical value of the error increases for fixed N. In the inset, we show again $\langle|\overline{E}|\rangle$ as a function of P at the largest resolution in the main figure ($N = 1024$). The linear increase with P is clearly seen as symbols appear to be very well aligned with the best linear fit shown as the solid line.

As discussed in Section 4.2, in a number of applications the number of PEs is increased proportionally to the work load (weak scaling). In this case Eq. (59) shows that the error is constant. This result is also supported by our simulations in Fig. 6(b) where grid refinement does not lead to any decrease in the average error. The inset shows that the asymptotic value of $\langle|\overline{E}|\rangle$ increase linearly with P/N as predicted by Eq. (59).

There is another important consequence of using common finite differences in an asynchronous fashion. One of the objectives of using asynchronous schemes is to avoid global synchronizations. However, in many realistic simulations where diffusion effects are small, the time step is determined by a convective stability condition which is computing from the maximum velocity in the entire domain. In such a case, obtaining a consistent time step across all PEs, requires a global

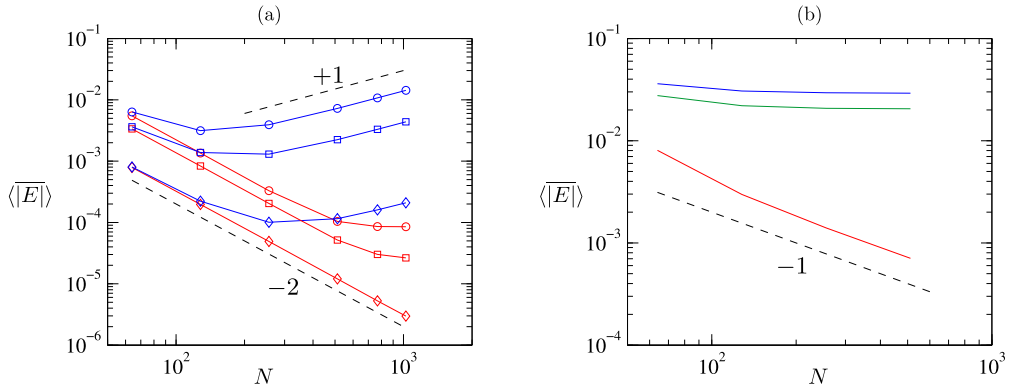

Fig. 7. Convergence properties with different relations between time step size and grid spacing for the advection–diffusion equation. (a) Cases with constant Δt. Different lines correspond to $\Delta t = 1 \times 10^{-5}$ (circles), 5×10^{-6} (squares), and 1×10^{-6} (diamonds). (Note that time steps are small because they need to be chosen such that the stability condition discussed in Section 3.3 is satisfied at the largest resolution.) Red and blue lines correspond to $p_0 = 1.0$ (synchronous) and 0.3, respectively. (b) Cases with constant $r_c = c\Delta t/\Delta x$. Red, green and blue lines correspond to $p_0 = 1.0$ (synchronous), 0.3, and 0.0, respectively. Dashed lines in both panels are reference power laws with slope noted next to the corresponding line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

synchronization. If a good estimate of the maximum velocity in the problem is known a priori, however, then a constant Δt can be set based on this knowledge to satisfy the convective condition everywhere. While this avoids a global synchronization, the accuracy of the scheme is strongly modified. This can be seen from Eq. (65) and Table 2 which, with $D = 2$ at constant P, results in an average error scaling as $1/\Delta x^{D-1} = 1/\Delta x$. These predictions are in fact consistent with the numerical data shown in Fig. 7(a) where convergence transitions from Δx^2 to $1/\Delta x$ for asynchronous schemes (blue lines). For synchronous schemes, that is $p_0 = 1.0$, the transition is from Δx^2 to a constant (red lines) since the truncation error stemming from the time discretization is proportional to Δt and thus remains constant as the space discretization error decreases with N. The effect of decreasing Δt in both cases is, as expected, a reduced numerical value of the error.

Even if a global maximum velocity can be determined efficiently, the use of a convective condition (that is $r_c = c\Delta t/\Delta x$ kept constant in our linear case) results in higher derivatives providing divergent terms in the truncation error. This is also seen from the last two rows in Table 2. For the case of the advection–diffusion equation ($D = 2$) with a convective condition ($q = 1$ in Table 2) and constant P, the average error is also constant as the grid is refined. Fig. 7(b) shows again this theoretical result to be consistent with the numerical data. The figure also shows the standard synchronous scheme which is seen to be first order. This is expected since the time discretization which is $O(\Delta t)$, becomes $O(\Delta x)$ when r_c is kept constant.

6. Order recovery

As shown in previous sections, while common finite differences used in an asynchronous fashion retain stability characteristics, its accuracy can be degraded significantly making these schemes potentially unusable for real applications. Thus, alternative strategies are needed to make asynchronous approaches feasible for accurate solutions of general PDEs.

Since leading order terms in the truncation error due to asynchrony are proportional to Δt (see e.g. Eq. (65)), the simplest strategy is to select the time step according to $\Delta t \sim \Delta x^q$ with appropriate values of q. For $D = 2$ (i.e. heat or advection–diffusion equations) a value of $q = 3$ will recover the original average truncation error of Δx^2 when the number of PEs is kept constant according to Table 2. This is indeed observed from the numerical data presented in Fig. 8(a). However, as noted in Section 4.3, this may render the simulations prohibitively expensive as time step size decreases rapidly with increasing space resolution, especially when high-order schemes are utilized. Thus, we now present a different approach which retains a desired order of accuracy without the need to decrease the time step size with a steep power law on the grid spacing.

The idea is simply to try to eliminate the leading order terms in the truncation error associated with asynchrony. This procedure is analogous to that used to derive regular finite differences: that is, assume an approximation of the form Eq. (62), expand each term in Taylor series and impose conditions on the different terms on the truncation error (that is on the unknown coefficients b_i's) to produce an approximation of the d-derivative and eliminate as many terms in the truncation error as possible. As an example, consider the five-point approximation of a second derivative:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{b_{-2}u^n_{i-2} + b_{-1}u^n_{i-1} + b_0 u^n_i + b_1 u^{\tilde{n}}_{i+1} + b_2 u^{\tilde{n}}_{i+2}}{\Delta x^2} \tag{70}$$

where we have assumed asynchrony appears only on the right boundary of the PE. If $\tilde{n} = n$ (i.e. synchronous) by expanding the terms on the right-hand-side in Taylor series, one can show that it is possible to obtain an approximation of order $O(\Delta x^4)$. Omitting the details which can be found in textbooks [13] one finds

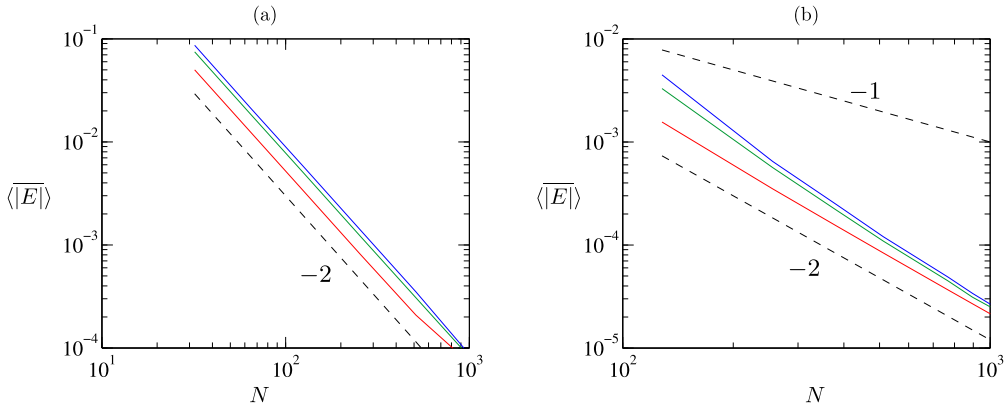

Fig. 8. Recovery of convergence properties for the advection–diffusion equation. (a) Standard scheme equation (69) with time step determined as $\Delta t \sim \Delta x^3$. (b) Scheme equation (77) with a fixed r_α of 0.1. Dashed lines in both panels are reference power laws with slope noted next to the corresponding line. Red, green and blue lines correspond to $p_0 = 1.0$ (synchronous), 0.3, and 0.0, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Truncation error representation of Eq. (70) with an arbitrary delay $\tilde{n} = n - k$. The first column contains the actual grid spacing and time step sizes in the truncation error. The second column contains the resulting order in Δx if a constant value of r_α is assumed (that is $\Delta t \sim \Delta x^2$). The third column contains the coefficients multiplying powers of Δx or Δt. The last column contains the row number for reference (see text).

Order	Order with constant r_α	Coefficient	Row
$1/\Delta x^2$	$1/\Delta x^2$	$(b_{-2} + b_{-1} + b_0 + b_1 + b_2)u$	1
$\Delta t/\Delta x^2$	1	$-(b_1 + b_2)\dot{u}$	2
$\Delta t^2/\Delta x^2$	Δx^2	$(b_1 + b_2)k^2\ddot{u}/2$	3
$1/\Delta x$	$1/\Delta x$	$(-2b_{-2} - b_{-1} + b_1 + 2b_2)u'$	4
$\Delta t/\Delta x$	Δx	$-(b_1 + 2b_2)k\dot{u}'$	5
$\Delta t^2/\Delta x$	Δx^3	$(b_1 + 2b_2)k^2\ddot{u}'/2$	6
1	1	$(4b_{-2} + b_{-1} + b_1 + 4b_2)u''/2$	7
Δt	Δx^2	$(b_1 + 4b_2)k\dot{u}''/2$	8
Δx	Δx	$(-8b_{-2} - b_{-1} + b_1 + 8b_2)ku'''/6$	9

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{-u^n_{i-2} + 16u^n_{i-1} - 30u^n_i + 16u^n_{i+1} - u^n_{i+2}}{12\Delta x^2} + \mathcal{O}(\Delta x^4) \tag{71}$$

where the coefficients b_i's have been set to eliminate low powers in Δx; in this case, terms containing powers of less than four.

If, on the other hand, $\tilde{n} = n - k$, then as seen in Eq. (47), terms involving both Δt and Δx appear. In fact, some of these error terms may even grow with increasing resolution as discussed in previous sections. Therefore, these terms must be eliminated in order for the resulting scheme to be usable. This can be accomplished by imposing conditions on the coefficients b_i's. With five coefficients, it is possible to impose five conditions. Using Taylor series for each of the contributions in Eq. (70), we find that the first few terms in the truncation error are those in Table 3.

It is clear that five conditions are not enough to eliminate all the terms necessary to obtain a fourth-order approximation as in Eq. (71). However, our aim here will be to devise a scheme that can maintain the first-order in time, second-order in space accuracy of Eq. (3). Since the average error contains a factor N_B/N which is proportional to Δx and $\Delta t \sim \Delta x^2$ is set by a fixed r_α, then, second order in space is maintained if terms like $\Delta t^p \Delta x^q \sim \Delta x^{2p+q}$ with $2p + q < 1$ are eliminated; that is, we would like to maintain terms of $O(\Delta x)$ or higher.

The resulting order in Δx for each term is also shown in Table 3 (second column) when Δt is written in terms of Δx as described above. Clearly rows one, two and four need to be eliminated. We can also see that the second and third rows present the same conditions on the coefficients b_i's and, therefore, by eliminating one, the other is automatically eliminated. The same linear dependence is observed for rows five and six. The term given in row seven should equal u'' to obtain an approximation for the second derivative.

Thus, one further condition may be imposed on the five unknown coefficients. The next terms in Δx are rows five and nine. The schemes resulting from zeroing either one of these terms are different. If one eliminates row five, one obtains an approximation with $b_1 = b_2 = 0$, that is a one-sided finite difference for the derivative towards the side without delay which would effectively disconnect one part of the domain with the other. Eliminating row nine, on the other hand, leads to a symmetric approximation. We also note that the resulting symmetric scheme is the same as one which would be obtained if one imposed a symmetry condition such as $b_{-1} = b_1$ as the fifth constraint.

Thus, the five equations for the five unknowns come from the following conditions in Table 3: (i) row seven equal to u'', and (ii) rows one, two, four and nine equal zero (to eliminate truncation error terms). The resulting system is

$$\begin{bmatrix} 2 & 1/2 & 0 & 1/2 & 2 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ -2 & -1 & 0 & 1 & 2 \\ -4/3 & -1/6 & 0 & 1/6 & 4/3 \end{bmatrix} \begin{bmatrix} b_{-2} \\ b_{-1} \\ b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{72}$$

whose solution leads to the following approximation of the second derivative:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+2}^{\tilde{n}} - u_{i+1}^{\tilde{n}} - u_{i-1}^{n} + u_{i-2}^{n}}{3\Delta x^2} + \mathcal{O}\big(k\Delta t, \Delta x^2, k\Delta t/\Delta x\big), \tag{73}$$

where we have included also the dependency on the delay k in the truncation error. When used synchronously, we have $k = 0$ and the scheme is second order. Under asynchronous conditions, if r_α is constant, then $\Delta t \sim \Delta x^2$ and the above approximation is $\mathcal{O}(\Delta x)$. When the average $\langle|\overline{E}|\rangle$ is computed, the resulting order is $\mathcal{O}(\Delta x^2)$.

This new scheme can be used to solve, for example, the advection–diffusion equation which now reads:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_{i+1}^{\tilde{n}} - u_{i-1}^{n}}{2\Delta x} = \alpha \frac{u_{i+2}^{\tilde{n}} - u_{i+1}^{\tilde{n}} - u_{i-1}^{n} + u_{i-2}^{n}}{3\Delta x^2}, \tag{74}$$

or, after some rearrangement,

$$u_i^{n+1} = \frac{r_\alpha}{3}u_{i+2}^{\tilde{n}} - \left(\frac{r_c}{2} + \frac{r_\alpha}{3}\right)u_{i+1}^{\tilde{n}} + u_i^n + \left(\frac{r_c}{2} - \frac{r_\alpha}{3}\right)u_{i-1}^n + \frac{r_\alpha}{3}u_{i-2}^n. \tag{75}$$

Obviously, the new scheme must be stable in order to be useful in practice. One can attempt to obtain the limits of stability by using the main result in Section 3.3 which states that an asynchronous scheme is stable if its synchronous counterpart is stable. However, in this case, the condition given in Eq. (40), is never satisfied. While this does not imply an unstable scheme since Eq. (40) is only a sufficient condition, a detailed von Neumann stability analysis of Eq. (75), does reveal that the scheme is indeed unstable for any value of r_c and r_α at high wavenumbers. Thus, even under synchronous conditions the scheme equation (75) is unstable.

However, other schemes can be formed using the five-point stencil in Eq. (70). For example, a non-symmetric scheme can be established if the fifth equation (last row in Eq. (72)) is replaced by a condition such as $b_{-2} = -b_2$ or $b_{-1} = -b_1$. The former leads to an approximation of the form $\partial^2 u/\partial x^2 \approx (u_{i+2}^{\tilde{n}} - u_{i+1}^{\tilde{n}} - 2u_i^n + 3u_{i-1}^n - u_{i-2}^n)/\Delta x^2 + \mathcal{O}(k\Delta t, \Delta x, k\Delta t/\Delta x)$, which after a detailed von Neumann analysis of the advection–diffusion equation can be shown to be unstable too. The latter, on the other hand, leads to

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+2}^{\tilde{n}} - u_{i+1}^{\tilde{n}} - u_i^n + u_{i-1}^n}{2\Delta x^2} + \mathcal{O}(k\Delta t, \Delta x, k\Delta t/\Delta x), \tag{76}$$

which, when used in the advection–diffusion equation results in the following scheme

$$u_i^{n+1} = \frac{r_\alpha}{2}u_{i+2}^{\tilde{n}} - \left(\frac{r_c}{2} + \frac{r_\alpha}{2}\right)u_{i+1}^{\tilde{n}} + u_i^n\left(1 - \frac{r_\alpha}{2}\right) + \left(\frac{r_c}{2} - \frac{r_\alpha}{2}\right)u_{i-1}^n. \tag{77}$$

The ∞-norm criterion Eq. (36) for this scheme is too conservative and cannot assure stability. However, a von Neumann stability analysis for $\tilde{n} = n$ yields an amplification factor given by $G = (1/2)(2 - r_\alpha + r_\alpha\cos(2\theta) - 2\iota(r_c + r_\alpha - r_\alpha\cos(\theta))\sin(\theta))$ where $\iota = \sqrt{-1}$ and $\theta = k\Delta x$ with k being the wavenumber associated with the perturbation introduced. Detailed analysis of this amplification factor shows that the condition for stability $|G| \le 1$ is satisfied for $r_\alpha < 1/2$ and $r_c \le \sqrt{2r_\alpha} - 2r_\alpha$. Since there is no reason at present to believe that the claim (in Section 3) that synchronous stable schemes remain stable under asynchronous conditions when stability is based on the amplification factor from a von Neumann analysis, we proceed to verify the properties of this scheme numerically.

Typical results from simulations using Eq. (77) are shown in Fig. 9. Comparison of Fig. 9(b) and Fig. 3(b) reveals smaller errors for the new scheme. As predicted by our analysis, the scheme recovers second-order convergence asymptotically even under asynchronous conditions as seen in Fig. 8(b). At lower resolutions, a dependence on the probability p_0 is seen but weakens as N increases. These cases were run at $r_\alpha = 0.1$ and r_c in the range 0.035–0.245 which were found to provide a stable solution.

The selection of these parameters was guided by the von Neumann analysis above for the synchronous scheme which assures stability for $r_\alpha < 1/2$ and $r_c \le \sqrt{2r_\alpha} - 2r_\alpha$. The analysis also shows that the most unstable modes are at high wavenumbers. Thus, our numerical experiments started from initial conditions with non-zero high-wavenumber content ($\theta > 2$) as a worst case scenario. For $r_\alpha = 0.1$, we have progressively increased r_c until instability (solutions that grow in time) is observed. Instability was indeed seen at high wavenumbers when r_c becomes greater than 0.25, consistent with the criterion $r_c \le \sqrt{2r_\alpha} - 2r_\alpha = 0.24721$ for $r_\alpha = 0.1$. We stress again, however, that this agreement cannot be taken as a general result, and has to be established on a more formal basis. This is part of our own ongoing efforts. The main conclusion from this section, though, is that it is possible to construct schemes that under asynchronous conditions can retain some desired order of accuracy.

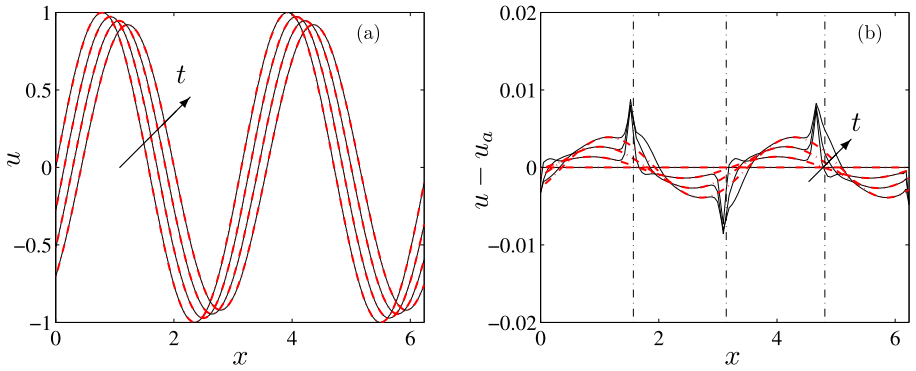

Fig. 9. Typical time evolution of the numerical solution of the advection–diffusion equation using the standard synchronous scheme Eq. (69) with $\tilde{n} = n$ (dashed red lines) and the new asynchronous scheme equation (77) (solid black lines). (a) The velocity field. (b) Error $E_i^n = u_i^n - u_a(x_i, t_n)$. Vertical dash-dotted lines correspond to PE boundaries. Simulation parameters: $N = 128$, $P = 4$, $L = 2$, with $\{p_0, p_1\} = \{0.3, 0.7\}$ for the asynchronous scheme. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

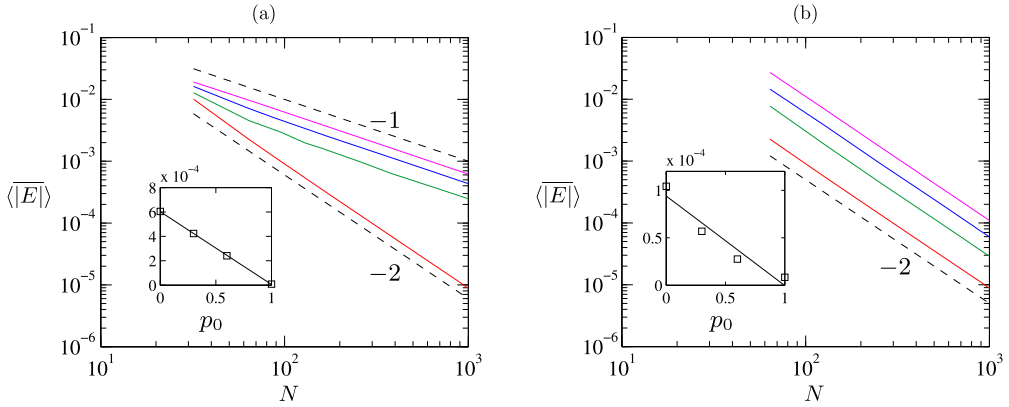

Fig. 10. Convergence for non-linear Burgers equation. (a) Results using Eq. (78). (b) Results using Eq. (79). Different lines correspond to $p_0 = 1.0$ (red), 0.6 (green), 0.3 (blue), and 0.0 (magenta). Dashed lines are reference power laws with the slope noted next to the corresponding line. Inset: $\langle\overline{|E|}\rangle$ as a function of p_0 for the highest resolution in the main figure and $P = 32$ and $L = 2$. Solid line is a best linear fit. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

We conclude this section by noting that although the derivations above were done for the advection–diffusion equation, there is obvious interest also in its non-linear version, that is Burgers' equation. This is so because of this equation resembles a one-dimensional (and more analytically amenable) version of the full equations describing the motion of fluid flows in important applications such as high-Reynolds numbers turbulence. In Fig. 10(a) we show an example of a scheme like Eq. (69). Specifically,

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + u_i^n \frac{u_{i+1}^{\tilde{n}} - u_{i-1}^n}{2\Delta x} = \alpha \frac{u_{i+1}^{\tilde{n}} - 2u_i^n + u_{i-1}^n}{\Delta x^2}. \tag{78}$$

In part (a) of the figure we show results with $r_\alpha = 0.1$ and initial conditions given by Eq. (67) with different ranges of wavenumbers and random phases. Conclusions below are insensitive to those parameters in the initial conditions as long as the scheme remains stable. Just as in the linear case, we observe that the accuracy drops from second to first order and that the effect of decreasing p_0 is to increase the error linearly (inset). If the new scheme equation (76) is applied to the non-linear equation, the scheme is now

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + u_i^n \frac{u_{i+1}^{\tilde{n}} - u_{i-1}^n}{2\Delta x} = \alpha \frac{u_{i+2}^{\tilde{n}} - u_{i+1}^{\tilde{n}} - u_i^n + u_{i-1}^n}{2\Delta x^2}. \tag{79}$$

A typical result using this scheme is shown in Fig. 10(b). All the conclusions we arrived at with the linear equation appear to apply equally well in the non-linear case. However, more thorough testing as well as a deeper fundamental understanding of the non-linear case is required to make a more generally valid claim. This is especially so for small values of α, where increasingly large gradients (and numerical instabilities) are known to appear.

As mentioned in Section 5.2, since errors may be localized in space, some statistical features of the solution may be insensitive to the numerical perturbations due to asynchrony. An example of this, is shown in Fig. 11 where we can see the decay of the space-averaged kinetic energy $K = \langle u^2 \rangle / 2$ and its dissipation rate $\epsilon = \alpha \langle (\partial u / \partial x)^2 \rangle$ as a function of time.

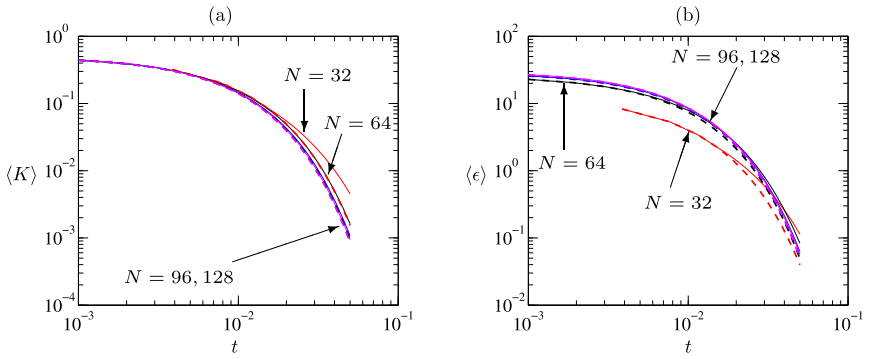

Fig. 11. Time evolution of overall (a) kinetic energy, $\langle K \rangle$ and (b) dissipation rate, $\langle \epsilon \rangle$ of the flow at different grid resolutions. Red, black, blue, and magenta correspond to $N = 32$, 64, 96, and 128 respectively. Synchronous (solid lines) and asynchronous results (dashed lines) were obtained with Eq. (78) and Eq. (79). Initial conditions: Eq. (67) with wavenumbers $\kappa = 6, 7, 8$. Other simulation parameters: $P = 16$, $L = 4$, $\{p_0, p_1, p_2, p_3\} = \{0.4, 0.2, 0.3, 0.1\}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

A grid convergence study for the synchronous schemes (also seen in Fig. 11) show that K becomes grid-independent at $N = 96$ — that is, results are indistinguishable from results at $N = 128$ and higher. The same behavior is observed for asynchronous computations (dashed lines) which practically overlap with synchronous results when the grid is fine enough for (synchronous) results to have converged.

The decay of the dissipation rate is shown in part (b). As expected this is more sensitive to resolution than K because ϵ depends on gradients. Thus, it is not surprising that larger errors appear. Still grid-independent results are also attained at N larger than 96. Again, asynchronous and synchronous results are indistinguishable from each other in the figure.

We then conclude that, in this case, the decay of both K and ϵ are virtually unaffected by asynchrony for grid-converged simulations. Thus, if the interest is, for example, in decay rates, then asynchronous computations would provide results which are as accurate as those using synchronous schemes. Obviously, extending this conclusion to three-dimensional, high-Reynolds number turbulence will require careful examination of numerical data from asynchronous simulations. In this endeavor it would also be important to study the behavior of other methods more suitable for the non-linear Burgers equation when asynchrony is present.

7. Conclusions and discussion

Due to the massive levels of parallelism already available on petascale systems, communication between PEs presents significant challenges in practical applications for sustained performance at scale. In future exascale systems, parallelism is expected to increase further in a significant way. Communication, global synchronizations as well as minor load imbalances could represent major bottlenecks to scalability on future systems.

To avoid these problems, we have investigated the behavior of widely used finite difference schemes applied in an asynchronous fashion such that all synchronizations and communication overheads are avoided. Not only this allows for overlap of computation and communication but it also relaxes expensive synchronizations due to small but common delays across PEs which are already present in current systems due to e.g. system noise.

We first showed that commonly used synchronous schemes remain stable when the value of the function at one or more grid points correspond to time steps previous to the most current one. This property was shown using bounds on the infinity-norm of the matrix representing the evolution of the solution at every step. The use of a matrix method to analyze stability is necessary because of the random nature of the delays as well as the localized nature of these delays (at PEs boundaries) which renders a classical von Neumann analysis inappropriate in this case. Our analysis can only give sufficient conditions, though, which in some cases may be too restrictive. Our numerical tests show that the conservation of stability under asynchronous conditions is still observed when the infinity norm of the evolution matrix (e.g. Eq. (36)) is not bounded by unity. Developing tighter stability regions are of obvious interest as it will expand the range of conditions in which one can compute solutions with greater confidence.

We have also analyzed the issues of consistency and accuracy of asynchronous schemes. We have shown that when used asynchronously widely used finite differences may present consistency issues, which can, in principle, be mitigated by the way in which time step size and grid spacings are reduced simultaneously. By studying the details of the truncation error, it is clear that a zeroth-order term always appears. However, these errors will only be present at PE boundaries. Due to the random nature of the delays expected at PE boundaries, a statistical approach has been used to study the convergence properties of these asynchronous schemes. In particular we used the average of the error and the absolute value of the error (corresponding to the 1-norm of the error) across the domain. We showed that this error always drops to first order when asynchrony is present regardless of the order of the original scheme. Furthermore, we found that the error is also proportional to the number of PEs (P) and the mean delay $|\overline{\tilde{k}}|$, that is $\langle |\overline{E}| \rangle \sim |\overline{\tilde{k}}| P \Delta x$. This is both an interesting and important result. It states that the error of the solution depends not only on grid spacing but also on the characteristics

of the system it is run on. At extremely large levels of parallelism due to the enormous challenges in data movement it is indeed expected that scalable codes are able to trade some accuracy for computational performance. This scaling of the error has been verified by thorough numerical experiments.

Since first-order methods are rarely of practical use, we have shown that it is possible to construct new numerical schemes robust to asynchrony. By using larger stencils, one can mitigate the errors incurred by delays in messages and recover higher orders of accuracy. We have derived such schemes and used them for the linear advection–diffusion and the non-linear Burgers equation. Numerical results confirm the theoretical predictions.

We conclude by noting that the results presented here can, in principle, be extended to a variety of numerical schemes for PDEs based on finite differences. Since our developments were based on one-dimensional linear equations, it is of obvious interest to develop extensions to two and three-dimensional domains. It will also be important to generate tighter bounds on stability regions (in any number of dimensions) which, as mentioned above, is part of our ongoing research. Asynchrony-tolerant schemes like those shown here could also be exploited on complex geometries using other discretization schemes such as finite volumes or elements. This is also part of future research.

Acknowledgements

The authors gratefully acknowledge NSF (Grant nos. OCI-1054966 and CCF-1349100) for financial support. The authors also thank NERSC and XSEDE for computer time on their systems. The authors benefited from discussions with Lawrence Rauchwerger, Raktim Bhattacharya and Jacqueline H. Chen.

References

[1] S. Jagannathan, D.A. Donzis, Massively parallel direct numerical simulations of forced compressible turbulence: a hybrid MPI/OpenMP approach, in: Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment, 2012, article no. 23.
[2] M. Lee, N. Malaya, R.D. Moser, Petascale direct numerical simulation of turbulent channel flow on up to 786k cores, in: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, SC '13, ACM, New York, NY, USA, 2013, pp. 61:1–61:11.
[3] J. Dongarra, P. Beckman, T. Moore, P. Aerts, G. Aloisio, J.-C. Andre, D. Barkai, J.-Y. Berthou, T. Boku, B. Braunschweig, F. Cappello, B. Chapman, X. Chi, A. Choudhary, S. Dosanjh, T. Dunning, S. Fiore, A. Geist, B. Gropp, R. Harrison, M. Hereld, M. Heroux, A. Hoisie, K. Hotta, Z. Jin, Y. Ishikawa, F. Johnson, S. Kale, R. Kenway, D. Keyes, B. Kramer, J. Labarta, A. Lichnewsky, T. Lippert, B. Lucas, B. Maccabe, S. Matsuoka, P. Messina, P. Michielse, B. Mohr, M.S. Mueller, W.E. Nagel, H. Nakashima, M.E. Papka, D. Reed, M. Sato, E. Seidel, J. Shalf, D. Skinner, M. Snir, T. Sterling, R. Stevens, F. Streitz, B. Sugar, S. Sumimoto, W. Tang, J. Taylor, R. Thakur, A. Trefethen, M. Valero, A. van der Steen, J. Vetter, P. Williams, R. Wisniewski, K. Yelick, The international exascale software project roadmap, Int. J. High Perform. Comput. Appl. 25 (2011) 3–60.
[4] T. Hoefler, T. Schneider, A. Lumsdaine, Characterizing the influence of system noise on large-scale applications by simulation, in: International Conference for High Performance Computing, Networking, Storage and Analysis (SC'10), 2010.
[5] D.P. Bertsekas, J.N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.
[6] A. Frommer, D. Szyld, On asynchronous iterations, J. Comput. Appl. Math. 123 (2000) 201–216.
[7] R. Tavakoli, P. Davami, New stable group explicit finite difference method for solution of diffusion equation, Appl. Math. Comput. 181 (2006) 1379–1386.
[8] H. Karimabadi, J. Driscoll, Y. Omelchenko, N. Omidi, A new asynchronous methodology for modeling of physical systems: breaking the curse of courant condition, J. Comput. Phys. 205 (2005) 755–775.
[9] D. Amitai, A. Averbuch, S. Itzikowitz, M. Israeli, Parallel adaptive and time-stabilizing schemes for constant-coefficient parabolic PDEs, Comput. Math. Appl. 24 (1992) 33–53.
[10] D. Amitai, A. Averbuch, S. Itzikowitz, E. Turkel, Asynchronous and corrected-asynchronous finite difference solutions of PDEs on MIMD multiprocessors, Numer. Algorithms 6 (1994) 275–296.
[11] D. Amitai, A. Averbuch, M. Israeli, S. Itzikowitz, On parallel asynchronous high-order solutions of parabolic PDEs, Numer. Algorithms 12 (1996) 159–192.
[12] K. Aditya, D.A. Donzis, Asynchronous computing for partial differential equations at extreme scales, in: Proceedings of the 2012 SC Companion: High Performance Computing, Networking Storage and Analysis, SCC '12, IEEE Computer Society, Washington, DC, USA, 2012, p. 1444.
[13] J.C. Tannehill, D.A. Anderson, R.H. Pletcher, Computational Fluid Mechanics and Heat Transfer, Taylor & Francis, 1997.
[14] T. Hoefler, T. Schneider, A. Lumsdaine, Accurately measuring collective operations at massive scale, in: Proceedings of the 22nd IEEE International Parallel & Distributed Processing Symposium, PMEO'08 Workshop, 2008.
[15] C. Hirsch, Numerical Computation of Internal and External Flows, vol. 1, Wiley, New York, 1994.
[16] A.C. Hindmarsh, P.M. Gresho, D.F. Griffiths, The stability of explicit Euler time-integration for certain finite-difference approximations of the multi-dimensional advection diffusion equation, Int. J. Numer. Methods Fluids 4 (1984) 853–897.
[17] H.D. Thompson, B.W. Webb, J.D. Hoffman, The cell Reynolds-number myth, Int. J. Numer. Methods Fluids 5 (1985) 305–310.
[18] T. Watanabe, T. Gotoh, Inertial-range intermittency and accuracy of direct numerical simulation for turbulence and passive scalar turbulence, J. Fluid Mech. 590 (2007) 117–146.
[19] T. Ishihara, T. Gotoh, Y. Kaneda, Study of high-Reynolds number isotropic turbulence by direct numerical simulation, Annu. Rev. Fluid Mech. 41 (2009) 165–180.
[20] D.A. Donzis, P.K. Yeung, K.R. Sreenivasan, Dissipation and enstrophy in isotropic turbulence: scaling and resolution effects in direct numerical simulations, Phys. Fluids 20 (2008) 045108.