Appl. Numer. Math. 218 (2025) 91–108

Contents lists available at ScienceDirect

Applied Numerical Mathematics

journal homepage: www.elsevier.com/locate/apnum

Research Paper

Space-time parallel solvers for reaction-diffusion problems forming Turing patterns ☆

Andrés Arrarás [a], Francisco J. Gaspar [b], Iñigo Jimenez-Ciga [a], Laura Portero [a],*

[a] *Institute for Advanced Materials and Mathematics (INAMAT^2), Department of Statistics, Computer Science and Mathematics, Public University of Navarre, Edificio de Las Encinas, Campus Arrosadia, Pamplona, 31006, Spain*

[b] *University Institute of Mathematics and Applications (IUMA), Department of Applied Mathematics, University of Zaragoza, Pedro Cerbuna 12, Zaragoza, 50009, Spain*

ARTICLE INFO

MSC:
35K57
65M55
65Y05

Keywords:
Parareal algorithm
Reaction-diffusion problems
Space-time parallel methods
Splitting methods
Turing pattern formation

ABSTRACT

In recent years, parallelization has become a strong tool to avoid the limits of classical sequential computing. In the present paper, we introduce four space-time parallel methods that combine the parareal algorithm with suitable splitting techniques for the numerical solution of reaction-diffusion problems. In particular, we consider a suitable partition of the elliptic operator that enables the parallelization in space by using splitting time integrators. Those schemes are then chosen as the propagators of the parareal algorithm, a well-known parallel-in-time method. Both first- and second-order time integrators are considered for this task. The resulting space-time parallel methods are applied to integrate reaction-diffusion problems that model Turing pattern formation. This phenomenon appears in chemical reactions due to diffusion-driven instabilities, and rules the pattern formation for animal coat markings. Such reaction-diffusion problems require fine space and time meshes for their numerical integration, so we illustrate the usefulness of the proposed methods by solving several models of practical interest.

1. Introduction

In the last decades, the development and improvement of complex computer architectures has led to the construction of more and more sophisticated supercomputers. The aim of this phenomenon is to increase the number of connected computing cores, thus enabling more powerful parallel implementations. Although it is an ongoing process, most of future costly computations will definitely be performed significantly faster in parallel.

In the field of numerical analysis, parallelization has become a strong tool for the numerical integration of evolutionary differential problems. In particular, problems with very fine spatial and time meshes, which imply a high computational cost, are the main focus of this technique.

☆ The work of Iñigo Jimenez-Ciga was supported by Public University of Navarre (PhD Grant). The work of all the authors was supported by Grant PID2019-105574GB-I00 funded by MCIN/AEI/10.13039/501100011033 and Grant PID2022-140108NB-I00 funded by MCIN/AEI/10.13039/501100011033 and by “ERDF A way of making Europe”. Open access funding provided by Universidad Pública de Navarra.

* Corresponding author.
E-mail address: laura.portero@unavarra.es (L. Portero).

https://doi.org/10.1016/j.apnum.2025.07.012
Received 29 March 2024; Received in revised form 12 May 2025; Accepted 17 July 2025

Available online 23 July 2025

Many methods have been proposed in order to implement parallelization in space, but the increasing number of available connected processors might also require numerical techniques that further permit parallelization in time. Research on parallel-in-time methods goes back to the 1960s, when Nievergelt proposed the first parallel time integrator (cf. [1]). For a general survey on the existing parallel-in-time integrators, we refer the reader to [2]. Among these methods, we highlight the parareal algorithm. First introduced by Lions, Maday, and Turinici in [3], this rather simple but efficient method is based on two propagators: one of them should be cheap and inaccurate, whereas the other one is considered to be expensive but accurate. The key point of the algorithm is that a parallel implementation of the latter propagator is permitted.

Several interpretations of this algorithm have been provided in the literature along the years. It was defined as a predictor-corrector method in [4], and Gander and Vandewalle proved that it is, in fact, a multiple shooting method (see [5] for further details). Finally, it can also be considered as a two-level multigrid method (cf. [6]).

In this paper, we present a new family of space-time parallel methods based on the parareal algorithm, which can be applied to integrate reaction-diffusion problems. Parallelization in space is performed by partitioning the elliptic operator into simpler terms. Subsequently, using time-splitting integrators, the original problem can be reduced to the solution of a set of uncoupled systems at each internal stage. More information about this class of time integrators can be found in the monograph [7].

In this work, we consider two splitting techniques, namely, dimensional (cf. [8]) and domain decomposition (cf. [9]) splittings. While the former is simpler, the latter is more flexible since it can be implemented for complex spatial domains and discretizations. The main contribution of our proposal lies in using such integrators as propagators within the parareal algorithm, thereby enabling parallelization in both space and time. Consequently, both the fine and coarse time integration steps can exploit spatial parallelism. As a result, the fine propagator is computed more efficiently, and the coarse propagator becomes significantly cheaper. In particular, processors that remain unused during the coarse step in the classical parareal scheme can now be used to parallelize it in space. Furthermore, unlike other space-time parallel methods based on parareal, our strategy avoids the need for spatial iterations to ensure convergence (only the corresponding iterations of the parareal time-parallelization framework are required).

The proposed methods will be tested for the integration of reaction-diffusion problems. In particular, Turing pattern models will be considered. These models, first introduced by Turing in [10], describe phenomena that involve chemical reactions with spatial diffusion, giving rise, in the presence of diffusion-driven instabilities, to animal coat patterns (see [11] for more details). More specifically, we will consider some well-known Turing pattern formation problems from the literature, as the Gray-Scott, Thomas, and Schnakenberg models. A general survey on these models can be found in [11] and [12]. Such particular models typically require the use of fine spatial grids in order to capture the referred patterns (cf. [13]). In these situations, the implementation of classical explicit schemes might be overly expensive. That is not the case for the so-called implicit-explicit (IMEX) methods, which consider an explicit contribution of the reactive part and the implicit treatment of the diffusive part, usually containing stiff terms. In this setting, the time integrators involved in the newly proposed parareal solvers are related to classical IMEX methods. For further details on such methods and their application to reaction-diffusion problems, we refer to [8,14,15].

Remarkably, for reaction-diffusion problems posed on simple geometries that admit a Cartesian grid, some efficient matrix-oriented methods (cf. [16]) and exponential time-differencing schemes (cf. [13,17]) were recently developed. Nevertheless, there exist many applications in which the solutions cannot be approximated with the aforementioned techniques, since the spatial domain has an irregular shape (see, e.g., [18] for a series of examples from developmental biology). In this setting, the new algorithms provide an alternative for irregular geometries that may require general non-Cartesian spatial grids.

The rest of the paper is organized as follows. In Section 2, we introduce the model problem and recall two splitting techniques that permit parallelization in space. Moreover, we present the four splitting time integrators that will be used to solve the partitioned problem. In Section 3, we describe the parareal algorithm and adapt the main theoretical results from the classical literature for the case under study. In Section 4, we formulate four specific space-time parallel methods, which are subsequently used in Section 5 for solving reaction-diffusion problems stemming from Turing pattern models. Numerical experiments include a convergence analysis of the new solvers, the study of splitting errors for the time integrators under consideration, a series of robustness tests with respect to the spatial discretization parameter, scalability results that show the speedup of the methods for an increasing number of processors, and a comparison of CPU runtimes and global errors with classical IMEX methods from the literature. Different spatial domains are considered, including a circular domain for the Schnakenberg model that requires non-Cartesian meshes and suitable spatial discretizations for its solution. Finally, Section 6 contains some concluding remarks based on the performance of the methods for the preceding experiments.

2. Model problem and splitting techniques

In this section, we first introduce the model problem under consideration and then provide a brief description of two partitioning strategies for the elliptic operator. Following the method of lines, the partitioned problem is first discretized in space. Next, suitable splitting time integrators are introduced in order to integrate the resulting initial value problem. As we will see, each internal stage of the fully discrete problem consists of a set of uncoupled linear systems, whose solution can be parallelized in space.

2.1. Model problem and partitioning strategies

Let us consider the following reaction-diffusion model problem

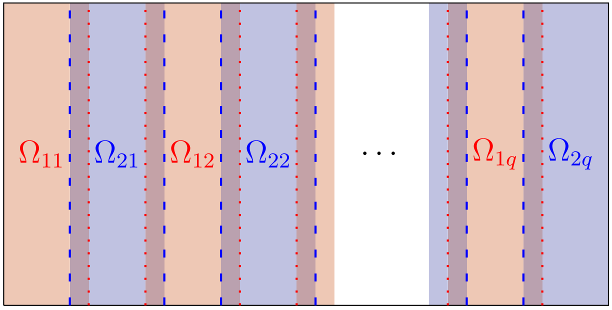


Fig. 1. The overlapping domain decomposition formed by two subdomains as the union of q vertical strips.

$$\begin{cases} u_t = Lu + r(u) + f, & \text{in } \Omega \times (0,T], \\ \nabla u \cdot \mathbf{n} = 0, & \text{on } \Gamma \times (0,T], \\ u = u_0, & \text{in } \overline{\Omega} \times \{0\}, \end{cases} \tag{1}$$

where $\Omega \subset \mathbb{R}^2$ is a bounded connected spatial domain, with boundary $\Gamma = \partial\Omega$. L and $r(\cdot)$ denote the diffusive and reactive operators, respectively, the latter being a possibly nonlinear function. We denote by $\mathbf{n}$ the outward unit normal on Γ. Let us assume sufficient smoothness on $u = u(\mathbf{x},t)$, $r(u)$, $f = f(\mathbf{x},t)$, and $u_0 = u_0(\mathbf{x})$, and suitable compatibility conditions. The diffusive operator considered hereafter has the form $Lu = \nabla \cdot (D\nabla u)$, where D is a 2×2 symmetric positive definite diffusion tensor.

In the sequel, we adopt a classical approach to split the elliptic operator by separately treating the diffusion and reaction terms. The former requires an implicit treatment, whereas the latter will be treated explicitly. In addition, let us consider a suitable partition of the diffusion operator into M terms, i.e., $L = L_1 + \cdots + L_M$. In this work, we set $M = 2$, although the extension to a larger number of split terms is straightforward. We propose two partitioning strategies that allow for space parallelization, namely, dimensional and domain decomposition splittings.

On the one hand, dimensional splitting is a classical partitioning technique that decomposes the diffusive operator into $M = d$ terms, d being the dimension of the spatial domain. In this case, each suboperator involves partial derivatives with respect to just one of the spatial variables. (cf. [8]). This strategy has certain evident drawbacks. If the operator contains mixed partial derivatives, dimensional splitting cannot be applied in a straightforward way. Moreover, complex spatial domains and spatial discretizations can also hinder the implementation of this partitioning technique. Assuming that $D = \operatorname{diag}(d_1, d_2)$, the dimensional splitting of the diffusive operator L for a two-dimensional problem can be defined as

$$L_1 u = (d_1 u_x)_x \qquad \text{and} \qquad L_2 u = (d_2 u_y)_y.$$

On the other hand, domain decomposition splitting stands out as a more versatile type of splitting. A first version of this technique was introduced in [19], and later analyzed in [20], by Vabishchevich. Henceforth, we define a simple decomposition of the spatial domain into two overlapping subdomains, following the ideas described in [9]. For that purpose, let us consider two overlapping subdomains, $\{\Omega_1, \Omega_2\}$, that cover the spatial domain Ω. In particular, each subdomain consists of the union of q disjoint vertical strips, as shown in Fig. 1. Note that every strip from a certain subdomain overlaps only with its neighboring strips, which belong to the other subdomain. The overlapping size is assumed to be equal for all the overlapping regions, and it will be denoted by β in the sequel. Further details on how to construct similar domain decompositions are described in [21].

Next, let us define a family of partition of unity functions, $\{\rho_1(\mathbf{x}), \rho_2(\mathbf{x})\}$, subordinate to the aforementioned domain decomposition. Both functions should belong to $C^\infty(\overline{\Omega})$ and satisfy the following conditions (see, e.g., [9]): $\operatorname{supp}(\rho_j(\mathbf{x})) \subseteq \overline{\Omega}_j$, for every $j \in \{1,2\}$; $0 \le \rho_j(\mathbf{x}) \le 1$, for every $j \in \{1,2\}$ and $\mathbf{x} \in \overline{\Omega}$; and $\rho_1(\mathbf{x}) + \rho_2(\mathbf{x}) = 1$, for every $\mathbf{x} \in \overline{\Omega}$. As noted in [9], for numerical purposes, $\rho_j(x)$ need not be $C^\infty(\overline{\Omega})$ functions, but just continuous and piecewise smooth functions. For the domain decomposition defined in Fig. 1, the following functions satisfy such conditions

$$\rho_1(\mathbf{x}) = \begin{cases} 0, & \text{in } \overline{\Omega} \setminus \overline{\Omega}_1, \\ h(x), & \text{in } \overline{\Omega}_1 \cap \overline{\Omega}_2, \\ 1, & \text{in } \overline{\Omega}_1 \setminus \overline{\Omega}_2, \end{cases}$$

and $\rho_2(\mathbf{x}) = 1 - \rho_1(\mathbf{x})$, where $0 \le h(x) \le 1$ is defined through suitable translations of the function $(1 \pm \cos(\pi x/\beta))/2$. Similar partition of unity functions will be considered in the numerical experiments below.

Once the partition of unity functions are defined, we consider the following suboperators for the splitting of the diffusive term

$$L_j u = \nabla \cdot (\rho_j D \nabla u), \quad j \in \{1,2\}.$$

2.2. Method of lines and splitting time integrators

Let us consider the reaction-diffusion problem (1), together with a splitting of the diffusion term, $L = L_1 + L_2$, based on one of the partitioning strategies introduced in Subsection 2.1. We first define a suitable mesh Ω_h covering the spatial domain Ω, where h refers to the maximal grid spacing. Then, the spatial semidiscretization of such a problem (using, e.g., finite differences, finite elements, or finite volumes) yields an initial value problem of the form

$$\begin{cases} U'(t) = (A_1 + A_2)U(t) + R(U(t)) + F(t), & t \in (0,T], \\ U(0) = \mathcal{P}u_0, \end{cases}$$

where $U(t)$ denotes the semidiscrete approximation to $u(\mathbf{x},t)$, A_1, A_2, and $R(\cdot)$ are semidiscrete analogues of the operators L_1, L_2, and $r(\cdot)$, respectively, and $F(t)$ comprises the contribution of the boundary conditions and the source term $f(\mathbf{x},t)$. Finally, $\mathcal{P}$ denotes a suitable projection or restriction operator acting on the initial condition.

We now recall certain time-splitting methods suitable for the integration of partitioned reaction-diffusion problems. For that purpose, we will consider first- and second-order schemes. Since the splitting of the diffusive operator consists of two terms, we will restrict the description of the schemes to $M = 2$.

Let us first divide the time interval $[0,T]$ into N subintervals $[t_n, t_{n+1}]$, with stepsize $\tau = t_{n+1} - t_n = T/N$, for $n = 0, 1, \dots, N-1$. We further define the fully discrete solution $U_n \approx U(t_n)$ and source term $F_n = F(t_n)$ at times $t_n = n\tau$, for $n = 0, 1, \dots, N$.

The fractional implicit Euler (FIE) method, also known as the locally one-dimensional (LOD) backward Euler scheme (see [8]), is a classical time-splitting method based on the Lie operator splitting. This method integrates each resulting subproblem by the implicit Euler scheme. Remarkably, it is L-stable and first-order convergent (see [22] for further details). In this work, we use a variant of the method that considers an explicit contribution of the reactive term and preserves its first-order consistency. The scheme is defined as follows: given $U_0 = \mathcal{P}u_0$, for $n \in \{0, 1, \dots, N-1\}$,

$$\begin{cases} (I - \tau A_1)U_{n,1} = U_n + \tau F_{n+1}, \\ (I - \tau A_2)U_{n,2} = U_{n,1}, \\ U_{n+1} = U_{n,2} + \tau R(U_{n,2}). \end{cases} \tag{2}$$

In order to introduce the stability function of splitting time integrators, it is usual to consider the partitioned Dahlquist test problem

$$\begin{cases} y'(t) = (\lambda_0 + \lambda_1 + \dots \lambda_M)\, y(t), & t \in (0,T], \\ y(0) = y_0, \end{cases} \tag{3}$$

where $\lambda_i \in \mathbb{C}$, for $i \in \{0, 1, \dots, M\}$. In particular, λ_0 represents the reactive term, whereas λ_j, with $j \geq 1$, are related to the diffusive suboperators. Then, considering $M = 2$ and denoting $z_j = \tau\lambda_j$, the application of scheme (2) to problem (3) yields the stability function

$$R_{\mathtt{FIE}}(z_0, z_1, z_2) = \frac{1 + z_0}{(1 - z_1)(1 - z_2)}. \tag{4}$$

A slightly modified version of the previous scheme is also considered: the Douglas-Rachford (DR) method, first introduced in [23] for the cases $M = 2$ and $M = 3$, and subsequently generalized to an arbitrary number of split terms in [24] and [25]. The method is defined as follows: given $U_0 = \mathcal{P}u_0$, for $n \in \{0, 1, \dots, N-1\}$,

$$\begin{cases} (I - \tau A_1)U_{n,1} = (I + \tau^2 A_1 A_2)U_n + \tau F_{n+1}, \\ (I - \tau A_2)U_{n,2} = U_{n,1}, \\ U_{n+1} = U_{n,2} + \tau R(U_{n,2}). \end{cases} \tag{5}$$

In this case, the diffusive part of the scheme is no longer L-stable, although it is still A-stable and first-order convergent (cf. [26]). The underlying benefit of this scheme is that, even though both FIE and DR schemes approximate the implicit Euler method, the former has first-order splitting error, whereas the latter has second-order splitting error (cf. [27]). Thus, the diffusive part is approximated more accurately using the DR method, although its stability properties are worse. Finally, the stability function of scheme (5) is given by

$$R_{\mathtt{DR}}(z_0, z_1, z_2) = \frac{(1 + z_0)(1 + z_1 z_2)}{(1 - z_1)(1 - z_2)}. \tag{6}$$

For the sake of completeness, we also consider two second-order splitting time integrators, namely, the modified Douglas (MD) method and the linearly implicit Peaceman-Rachford (LIPR) method. The former scheme was initially presented in [28], and later analyzed for reaction-diffusion problems in [29]. The MD method is defined as follows: given $U_0 = \mathcal{P}u_0$, for $n \in \{0, 1, \dots, N-1\}$,

$$\begin{cases} U_{n,1} = (I + \tau A)U_n + \tau R(U_n) + \tau F_n, \\ U_{n,2} = U_{n,1} + \frac{\tau}{2}(R(U_{n,1}) - R(U_n)), \\ (I - \frac{\tau}{2}A_1)U_{n,3} = U_{n,2} - \frac{\tau}{2}A_1 U_n + \frac{\tau}{2}(F_{n+1} - F_n), \\ (I - \frac{\tau}{2}A_2)U_{n+1} = U_{n,3} - \frac{\tau}{2}A_2 U_n. \end{cases} \tag{7}$$

It is based on the Douglas method (cf. [28]), but considers a second-order explicit treatment for the reactive term by means of the explicit trapezoidal rule. As shown in [29], the method is second-order convergent. If no reactive term is considered (i.e., $R(\cdot) \equiv 0$), the scheme is A-stable (see [8] for further details). However, when the reactive term is non-zero, only A_0-stability is guaranteed under certain restrictions on $R(\cdot)$ (as detailed in [29]). It is worth noting that this scheme belongs to the family of stabilizing correction methods (see, for instance, [30]), whose main advantage is that steady state solutions are also stationary points of the methods. The stability function of the MD method is given by the expression

$$R_{\mathrm{MD}}(z_0, z_1, z_2) = 1 + \frac{(1 + z_0/2)(z_0 + z_1 + z_2)}{(1 - z_1/2)(1 - z_2/2)}. \tag{8}$$

Finally, let us consider the second-order time-splitting integrator given by the linearly implicit Peaceman-Rachford (LIPR) method. This scheme is based on the well-known Peaceman-Rachford method, first introduced in [31], and belongs to a family of time integrators called linearly implicit fractional step methods (cf. [32]). For the implementation of the scheme for reaction-diffusion problems, we refer the reader to [33]. Recall that the original Peaceman-Rachford method is A-stable, although its linearly implicit variant no longer satisfies such a property. Considering two stiff split terms and a reactive term, the LIPR method is defined as follows: given $U_0 = \mathcal{P}u_0$, for $n \in \{0, 1, \dots, N-1\}$,

$$\begin{cases} (I - \frac{\tau}{2}A_2)U_{n,1} = (I + \frac{\tau}{2}A_1)U_n + \frac{\tau}{2}R(U_n) + \frac{\tau}{2}F_n, \\ (I - \frac{\tau}{2}A_1)U_{n+1} = (I + \frac{\tau}{2}A_2)U_{n,1} - \frac{\tau}{2}R(U_n) + \tau R(U_{n,1}) + \frac{\tau}{2}F_{n+1}. \end{cases} \tag{9}$$

The scheme is second-order consistent (see [32] for further details) and can be generalized to M arbitrary split terms preserving its second-order consistency. The stability function of this method is given by

$$R_{\mathrm{LIPR}}(z_0, z_1, z_2) = \frac{(1 + z_1/2)(1 + z_2/2) + z_0(1 + z_0/2 + z_1/2 + z_2/2)}{(1 - z_1/2)(1 - z_2/2)}. \tag{10}$$

The numerical solution of semilinear reaction-diffusion problems on fine meshes using linearly implicit methods, such as (2), (5), (7), and (9), usually requires solving numerous large linear systems. The proposed partitioning techniques permit to reduce such systems to a set of uncoupled linear subsystems per internal stage. For dimensional splitting, each internal stage consists of essentially one-dimensional uncoupled subsystems to be solved. If domain decomposition splittings are considered, since each function ρ_j vanishes outside subdomain Ω_j, each internal stage now contains an independent subsystem for each disjoint vertical strip belonging to Ω_j. If each subdomain consists of the union of q vertical strips, parallelization in space can be implemented using up to q processors.

At first glance, one could think that dimensional splitting offers a stronger parallelization power, since more independent subsystems are obtained. However, once the subsystems are solved, all the processors need to communicate among them before starting the solution of the next stage. This is not the case for domain decomposition splittings, since only data from the overlapping regions need to be communicated between no more than two processors. As a consequence, the overall process can be faster, even if a smaller number of calculation cores is involved.

3. The parareal algorithm

Since the 1960s, there has been a rapid and promising development of parallel-in-time integrators. In the current section, we describe a classical method of this kind, known as the parareal algorithm. First introduced in [3], it is a rather simple method that guarantees strong parallelization properties. In the last two decades, equivalent formulations of the algorithm have been derived (see, for instance, [34] and [35]).

Let us consider two different partitions of the time interval $[0, T]$. First, we uniformly divide it into N_c sets of the form $[T_n, T_{n+1}]$, for $n \in \{0, 1, \dots, N_c - 1\}$. We define a coarse time step $\Delta T = T/N_c$, and denote $T_n = n\Delta T$, for every $n \in \{0, 1, \dots, N_c\}$. Next, each set $[T_n, T_{n+1}]$ is uniformly partitioned into s finer subintervals $[t_{ns+j}, t_{ns+j+1}]$, for $j \in \{0, 1, \dots, s-1\}$. In this case, a fine time step $\delta t = \Delta T/s = T/(sN_c)$ is introduced.

In this framework, the parareal algorithm considers two time integrators, also known as coarse and fine propagators. While the former is inexpensive and inaccurate, the latter is expensive but accurate. In fact, the coarse and fine partitions of the time interval introduced above are the basis for the coarse and fine propagators, respectively. The underlying advantage of the algorithm is that the fine propagator can be computed in parallel.

Let $\mathcal{G}_{\Delta T}(T_n, T_{n+1}, v)$ denote the approximation at T_{n+1}, obtained after a single time step ΔT of the coarse propagator, considering the initial value v at time T_n. Similarly, $\mathcal{F}_{\delta t}(t_{ns+j}, t_{ns+j+1}, v)$ denotes the approximation at t_{ns+j+1}, obtained after a single time step δt of the fine propagator, with initial value v at t_{ns+j}. Then, if we consider the following initial guess for the parareal algorithm,

$$U^0_{n+1} = \mathcal{G}_{\Delta T}(T_n, T_{n+1}, U^0_n), \quad n \in \{0, 1, \dots, N_c - 1\},$$

where $U^0_0 = \mathcal{P}u_0$, the algorithm is defined as follows

$$\begin{aligned} U^{k+1}_0 &= \mathcal{P}u_0, \\ U^{k+1}_{n+1} &= \mathcal{G}_{\Delta T}(T_n, T_{n+1}, U^{k+1}_n) + \mathcal{F}^s_{\delta t}(T_n, T_{n+1}, U^k_n) - \mathcal{G}_{\Delta T}(T_n, T_{n+1}, U^k_n), \end{aligned} \tag{11}$$

for $n \in \{0, 1, \dots, N_c - 1\}$ and $k \in \{0, 1, 2, \dots\}$. The term $\mathcal{F}^s_{\delta t}(T_n, T_{n+1}, U^k_n)$ from (11) denotes the approximation $\tilde{U}_{(n+1)s}$ computed by the recurrence relation

$$\tilde{U}_{ns+j+1} = \mathcal{F}_{\delta t}(t_{ns+j}, t_{ns+j+1}, \tilde{U}_{ns+j}), \quad j \in \{0, 1, \dots, s-1\},$$

where $\tilde{U}_{ns} = U_n^k$. Note that $\tilde{U}_{(n+1)s}$ can be computed independently for every $n \in \{0, 1, \dots, N_c - 1\}$, since the initial values $\tilde{U}_{ns} = U_n^k$ are known from the previous iteration. Therefore, the computation of $\mathcal{F}^s_{\delta t}(T_n, T_{n+1}, U_n^k)$ can be implemented in parallel using up to N_c processors.

The breakthrough of this work is to consider the previously defined splitting time integrators as propagators of the parareal algorithm, in such a way that space-time parallel methods are obtained. It is worth noting that the only required iterations are those related to the time parallel process.

Among the properties of the parallel-in-time method, we highlight the fact that, even though it is defined as an iterative algorithm, convergence is guaranteed in a finite number of steps (cf. [36]). In fact, the algorithm needs at most N_c iterations to converge to the fine approximation (i.e., the approximation computed sequentially by the fine propagator). However, in order to be competitive, a smaller number of iterations should be enough to find a good approximation. Concerning stability issues, the authors in [37] advise considering an L-stable coarse propagator in order to avoid instabilities of the parareal algorithm, especially when stiff problems are solved. For a deeper discussion on the stability properties of the algorithm, we refer the reader to [4].

To conclude this section, we will adapt the definition of the convergence factor, first presented in [5], to the case of considering a partitioned equation. For that purpose, let us consider again the partitioned Dahlquist test problem (3). In this framework, the convergence factor of the parareal algorithm is defined as

$$\mathcal{K}(z_0, \dots, z_M, s) = \frac{|r(z_0/s, \dots, z_M/s)^s - R(z_0, \dots, z_M)|}{1 - |R(z_0, \dots, z_M)|}, \tag{12}$$

where R and r are the stability functions of the coarse and fine propagators, respectively. Then, if y_n^F denotes the approximation to the solution of problem (3) at $t = T_n$ computed by the fine propagator, and y_n^k is the approximation obtained at the k-th iteration of the parareal algorithm at $t = T_n$, the following bound holds (cf. [5])

$$\sup_{n>0} |y_n^F - y_n^k| \leq (\mathcal{K}(\lambda_0 \Delta T, \dots, \lambda_M \Delta T, s))^k \sup_{n>0} |y_n^F - y_n^0|.$$

Thus, if $\mathcal{K}(\lambda_0 \Delta T, \dots, \lambda_M \Delta T, s) < 1$ is fulfilled, the parareal algorithm converges to the fine approximation at each iteration.

These results can be generalized to more complex problems as long as certain additional conditions are imposed. In particular, for matrix differential problems of the form

$$\begin{cases} y'(t) = (A_0 + A_1 + \ldots + A_M) y(t), & t \in (0, T], \\ y(0) = y_0, \end{cases}$$

with $A_i \in \mathbb{R}^{m \times m}$, for $i \in \{0, 1, \dots, M\}$, if the set of matrices $\{A_0, A_1, \dots, A_M\}$ is simultaneously diagonalizable, that is, if there exists an invertible matrix $V \in \mathbb{R}^{m \times m}$ such that $V^{-1} A_i V$ is a diagonal matrix, then it holds

$$\sup_{n>0} \|V e_n^k\|_\infty \leq \left(\max_{\substack{\lambda_i \in \sigma(A_i) \\ i \in \{0,1,\dots,M\}}} \mathcal{K}(\Delta T \lambda_0, \dots, \Delta T \lambda_M, s) \right)^k \sup_{n>0} \|V e_n^0\|_\infty,$$

where $e_n^k = y_n^F - y_n^k$ and $\sigma(A_i)$ denotes the set of eigenvalues of the matrix A_i, for $i \in \{0, 1, \dots, M\}$.

We should remark that these results cannot be directly applied to reaction-diffusion problems with nonlinear reactive terms. Nevertheless, as suggested in [38], the qualitative behavior of the algorithm usually goes hand in hand with the theoretical convergence condition. The numerical experiments in Section 5 will confirm this claim.

4. Formulation of the space-time parallel solvers

In this section, we combine the parareal algorithm and the described splitting time integrators in order to obtain space-time parallel methods suitable for evolutionary problems. As explained before, the solvers proposed in this work consider the time-splitting integrators as the coarse and fine propagators of the parareal algorithm.

We choose the FIE scheme (2) as the coarse propagator for all the proposed solvers, due to its simplicity and inexpensiveness. Recall that (2) further applies an L-stable integration of the stiff diffusion term, thus representing a suitable choice that preserves the overall convergence of the parareal method (cf. [4] and [37]). Second-order splitting time integrators could also be considered as coarse propagators. However, if the diffusive part was solved by a second-order L-stable scheme, the computational cost would drastically increase, which would be counterproductive in terms of efficiency. Finally, the four aforementioned time-splitting methods are considered as fine propagators of the algorithm.

In summary, the following methods are proposed:

- the Parareal/FIE-FIE method: the parareal algorithm with the FIE scheme (2) as the coarse and fine propagators;
- the Parareal/FIE-DR method: the parareal algorithm with the FIE scheme (2) as the coarse propagator and the DR method (5) as the fine propagator;
- the Parareal/FIE-MD method: the parareal algorithm with the FIE scheme (2) as the coarse propagator and the MD method (7) as the fine propagator; and

- the Parareal/FIE-LIPR method: the parareal algorithm with the FIE scheme (2) as the coarse propagator and the LIPR method (9) as the fine propagator.

Before numerically testing the performance of the proposed solvers, we formulate a general convergence result that applies to diffusion problems.

Proposition 1. *Let $\mathcal{K}_{FIE}(z_0, z_1, z_2, s)$ denote the convergence factor (12) of the Parareal/FIE-FIE method for $M = 2$. Analogously, let $\mathcal{K}_{DR}$, $\mathcal{K}_{MD}$, and $\mathcal{K}_{LIPR}$ denote the convergence factors of the Parareal/FIE-DR, Parareal/FIE-MD, and Parareal/FIE-LIPR methods, respectively. Then, the bounds*

$$\mathcal{K}_{FIE}(0, z_1, z_2, s) \le 1/3, \qquad \mathcal{K}_{DR}(0, z_1, z_2, s) < 1,$$

$$\mathcal{K}_{MD}(0, z_1, z_2, s) < 1, \qquad \mathcal{K}_{LIPR}(0, z_1, z_2, s) < 1$$

hold for every $(z_1, z_2) \in (-\infty, 0)^2$ and $s \ge 2$ even.

Proof. The convergence factors for each method are obtained by substituting the amplification functions (4), (6), (8), and (10), respectively, into formula (12). The proof for the results corresponding to the first-order methods is thoroughly detailed in [39]. To establish the results for the second-order methods, it suffices to extend the arguments used for the Parareal/FIE-DR method.

When an explicit term is included, the convergence of the parareal algorithm becomes sensitive to the choice of coarse and fine time steps. Therefore, an alternative approach is adopted to identify suitable time step combinations that ensure convergence of the proposed solvers. This strategy is analyzed below in the context of several well-known reaction-diffusion problems.

5. Numerical experiments

This section is devoted to the numerical illustration of the newly proposed space-time parallel methods when applied to the solution of nonlinear reaction-diffusion problems. Both dimensional and domain decomposition splittings will be studied.

In the sequel, we consider three different models governing chemical reactions with spatial diffusion, which, in the presence of diffusion-driven instabilities (also known as Turing instabilities), give rise to the biological phenomenon of pattern formation (cf. [10,40]). In particular, we deal with the following reaction-diffusion model problem

$$\begin{cases} u_t = d_1 \Delta u + r_1(u, v) + f_1, & \text{in } \Omega \times (0, T], \\ v_t = d_2 \Delta v + r_2(u, v) + f_2, & \text{in } \Omega \times (0, T], \\ \nabla u \cdot \mathbf{n} = 0, \ \nabla v \cdot \mathbf{n} = 0, & \text{on } \Gamma \times (0, T], \\ u = u_0, \ v = v_0, & \text{in } \overline{\Omega} \times \{0\}, \end{cases} \tag{13}$$

where two chemical species $u = u(\mathbf{x}, t)$ and $v = v(\mathbf{x}, t)$ (called morphogens in the original work [10] by Turing) are involved. For each particular model, the diffusion coefficients d_1 and d_2, the nonlinear reaction kinetics functions $r_1(u, v)$ and $r_2(u, v)$, and the source terms $f_1 = f_1(\mathbf{x}, t)$ and $f_2 = f_2(\mathbf{x}, t)$ will be specified. Note that, in general, f_1 and f_2 will be null, since the chemical processes under consideration do not usually involve an external addition or removal of concentration of species. Taking into account the geometry of the spatial domains, the first two models, due to Gray-Scott and Thomas, will be discretized in space by using second-order finite differences, while the third one, due to Schnakenberg, will be discretized by means of piecewise linear finite elements.

Together with the evolution of the error for an increasing number of iterations, we also introduce a rather heuristic procedure that will help us analyze the convergence of the methods. Taking into account that the elliptic operator is partitioned into three terms, the first one corresponding to the reactive part and the last two to the diffusive part, we will consider $M = 2$ in the expression (12) for the convergence factor. In this framework, the convergence region is described as the set of every $(z_0, z_1, z_2) \in \mathbb{C}^3$ that satisfies the condition $\mathcal{K}(z_0, z_1, z_2) < 1$. In our case, let us denote by λ_1 and λ_2 the eigenvalues of the two diffusive terms that generate the smallest convergence region if $z_1 = \lambda_1 \Delta T$ and $z_2 = \lambda_2 \Delta T$ are considered. Then, for those values of z_1 and z_2, a convergence region with respect to z_0 can be defined as the set of all $z_0 \in \mathbb{C}$ that fulfills the bound $\mathcal{K}(z_0, z_1, z_2) < 1$. In the sequel, when we refer to the convergence region, it will mean this one, that is defined in $\mathbb{C}$ and can be visually analyzed. In order to estimate the value of z_0 for each reaction-diffusion problem, we take a similar approach to that presented in [41], which is based on the ideas introduced in [42] and [43]. The key point is to obtain the Jacobian of the reaction function $r(u, v) = (r_1(u, v), r_2(u, v))$, to evaluate it at the steady state solution for u and v, and to compute its eigenvalues. Such eigenvalues, denoted by $\lambda_{0,1}$ and $\lambda_{0,2}$, are an estimation of the spectrum of the Jacobian of $r(u, v)$. If $z_{0,1} = \lambda_{0,1} \Delta T$ and $z_{0,2} = \lambda_{0,2} \Delta T$ are inside the convergence region, we can conclude, at least heuristically, that the methods will converge. Although not theoretically proven, in practice this criterion works as a quite good estimator for the convergence behavior of the presented methods, as we will see in the following subsections.

5.1. Gray-Scott model

Let us consider a reaction-diffusion problem based on the Gray-Scott model with known exact solution. The Gray-Scott model (also called cubic autocatalytic model) was described in [44] as the two reaction scheme $A + 2B \rightarrow 3B$ and $B \rightarrow C$, where B is a

Table 1
Global errors for the u-component obtained with schemes (2), (5), (7) and (9), considering dimensional splitting and a set of increasing values for N.

	$N = 25$	$N = 50$	$N = 100$	$N = 200$
FIE	2.3874e-01	1.4272e-01	7.9667e-02	4.2383e-02
DR	1.1008e-01	4.8758e-02	2.2751e-02	1.1003e-02
MD	1.6684e-02	3.9796e-03	9.6877e-04	2.3895e-04
LIPR	9.8714e-03	2.2934e-03	5.5673e-04	1.3739e-04

Table 2
Global errors for the u-component obtained with schemes (2), (5), (7) and (9), considering domain decomposition splitting ($q = 2$ and $\beta = 1/8$) and a set of increasing values for N.

	$N = 25$	$N = 50$	$N = 100$	$N = 200$
FIE	2.1868e-01	1.3281e-01	7.8068e-02	4.3634e-02
DR	1.0032e-01	4.4609e-02	2.1613e-02	1.0732e-02
MD	1.8707e-02	4.4801e-03	1.1134e-03	2.7639e-04
LIPR	5.8203e-02	1.3719e-02	3.3448e-03	8.2361e-04

catalyst and C some inert product. Based on the mass-balance equations and considering diffusion, the model has the form (13) with

$$r_1(u,v) = \theta\,(1-u) - uv^2, \qquad r_2(u,v) = uv^2 - (\theta+\eta)\,v. \tag{14}$$

In this experiment, we consider the spatial domain $\Omega = (-1,1)\times(-1,1)$ and the final time $T = 2$. The parameters are set equal to $d_1 = d_2 = \theta = 1$ and $\eta = 0$. The source terms and initial conditions are such that the exact solution of the problem (13)-(14) is

$$u(x,y,t) = \cos(2\pi x)\cos(\pi y)\cos(2t),$$
$$v(x,y,t) = \cos(\pi x)\cos(2\pi y)\cos(2t).$$

This same experiment was performed in [14] using second-order IMEX methods.

Let us first analyze the convergence behavior of the splitting time integrators introduced in Subsection 2.2. Tables 1 and 2 show the evolution of the global errors for the u-component when considering the spatial and time discretization parameters $h = \tau = 2/N$, for a set of increasing values of N. As expected, both FIE and DR methods behave like first-order schemes. Essentially, the difference in the size of the errors is due to the size of the splitting error for the diffusion part of both methods. While the FIE method shows a first-order splitting error with respect to the first-order IMEX scheme, the DR scheme has a second-order splitting error. As a consequence, global errors are bigger for the FIE scheme for both partitioning techniques. A detailed description of splitting errors for first- and second-order methods, as well as some techniques to improve the accuracy of splitting methods when considering ADI and DD techniques, can be found in [45] and [21], respectively. Note that, if we modify the FIE method in order to reduce its splitting error following the strategy proposed in [45], we obtain the DR scheme. Later in this section (cf. Fig. 10) we will see the implications of the splitting errors and the parallel capabilities on the CPU time. On the other hand, both MD and LIPR schemes show a second-order convergent behavior. Depending on the scheme, either dimensional or domain decomposition splittings can yield more accurate approximations. Therefore, we will focus on the parallelization capabilities and the geometry of the spatial domain in order to choose one technique or the other.

Fig. 2 shows an accurate approximation of problem (13)-(14) computed by the MD method with dimensional splitting, considering $h = 0.01$ and a time step $\tau = 2.5\cdot 10^{-3}$. The results are consistent with those obtained in [14].

Next, we analyze the convergence behavior of the proposed iterative methods applied to the model problem (13)-(14) with $T = 1$. In this case, we consider the discretization parameters $h = 0.025$, $\Delta T = 0.05$, and $\delta t = 2.5\cdot 10^{-3}$. Two vertical strips and an overlapping size $\beta = 1/8$ are set for the decomposition of the spatial domain. In Fig. 3, we represent the convergence region of the Parareal/FIE-FIE for the reactive part of the problem. The red diamond in this plot represents the eigenvalue of the reactive part of problem (13)-(14) multiplied by ΔT (in this case, there is a single eigenvalue $\lambda_{0,1} = \lambda_{0,2} = -1$ with multiplicity 2). Note that $z_{0,1} = z_{0,2} = -0.05$ is real and negative, and it is located inside the convergence region. Even though we only show the convergence region for the Parareal/FIE-FIE method, the rest of the proposed solvers have similar regions, and the concluding qualitative results are the same.

Fig. 4 shows the evolution of the errors for the u-component with respect to the fine approximation. The errors of all four methods, considering both dimensional and domain decomposition splittings, are shown for an increasing number of iterations. Remarkably, for this model problem, we observe a faster convergence when dimensional splitting is used. Moreover, second-order methods show a slightly larger error with respect to the fine approximation. It must be noted, however, that the fine approximation of these methods is significantly more accurate than that of first-order schemes. Therefore, it seems to be preferable to choose second-order methods since, when compared with first-order solvers, the number of iterations to reach a certain tolerance is similar, but the corresponding approximations are more accurate.

In addition, space-time parallelization becomes particularly advantageous when dealing with large time intervals. In such cases, explicit methods are generally impractical due to the prohibitively large number of time steps required, and parallelization in time becomes crucial. In order to show the performance of the proposed space-time parallel solvers in this setting, we consider the Gray-

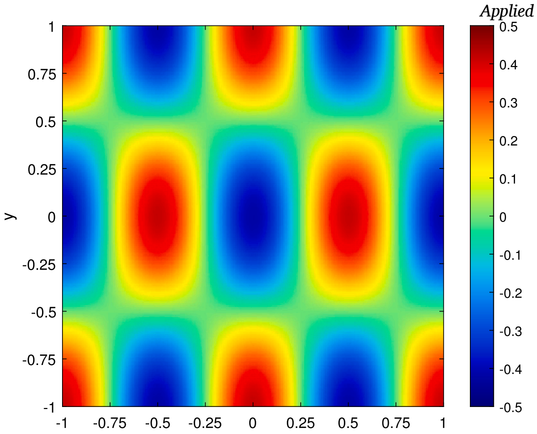


Fig. 2. Approximation of the u-component for the Gray-Scott problem at $t = 1$ computed by the MD method, considering dimensional splitting with $h = 0.01$ and $\tau = 2.5 \cdot 10^{-3}$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

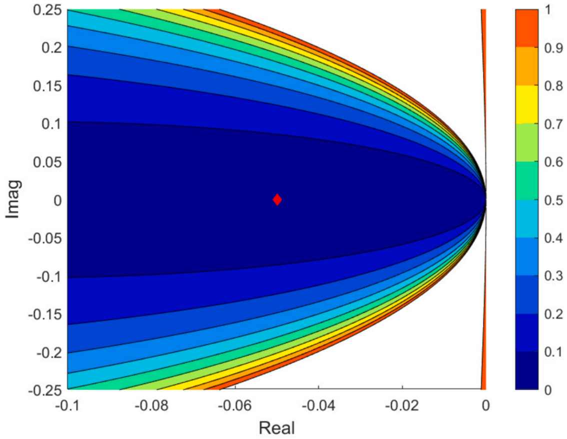
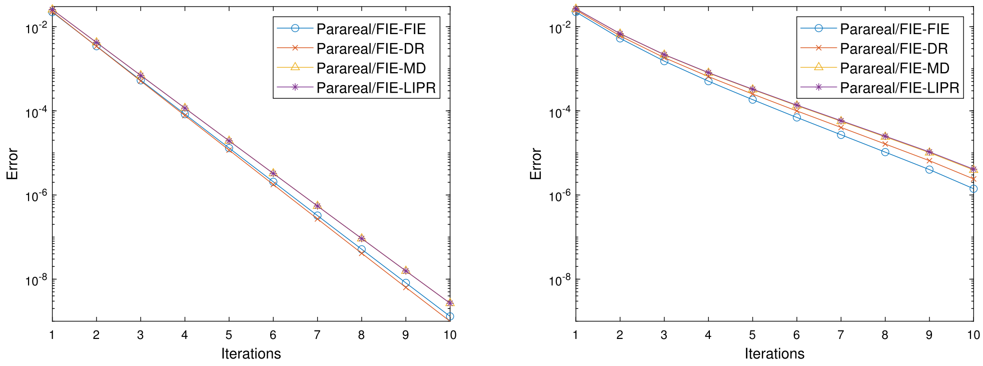


Fig. 3. Convergence region of the Parareal/FIE-FIE method and scaled eigenvalue of the reactive part (red diamond) for the Gray-Scott model, for $h = 0.025$, $\Delta T = 0.05$ and $\delta t = 2.5 \cdot 10^{-3}$.

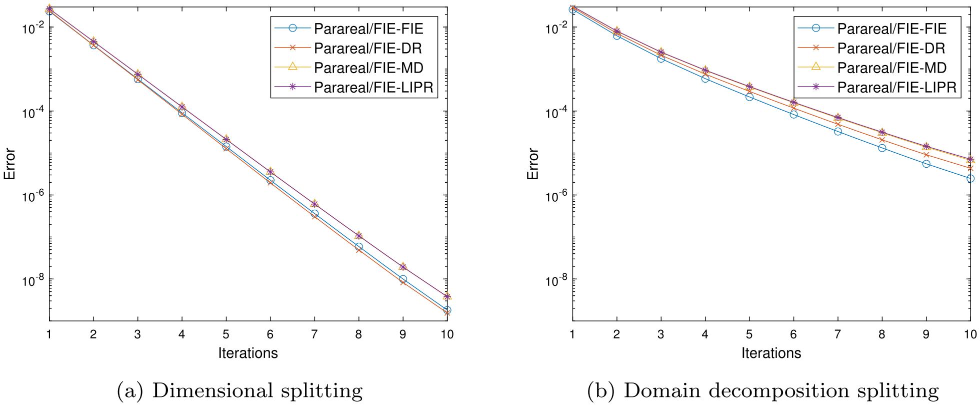


(a) Dimensional splitting

(b) Domain decomposition splitting

Fig. 4. Evolution of the error for the u-component of the Parareal/FIE-FIE, Parareal/FIE-DR, Parareal/FIE-MD, and Parareal/FIE-LIPR methods with respect to the fine approximation for the Gray-Scott problem, considering (a) dimensional splitting or (b) domain decomposition splitting, and choosing $h = 0.025$, $\Delta T = 0.05$ and $\delta t = 2.5 \cdot 10^{-3}$.

Scott problem with an extended final time $T = 200$, and we analyze the evolution of the error for an increasing number of iterations. The results, shown in Fig. 5, correspond to the four solvers under consideration with both partitioning strategies. A comparison between Figs. 4 and 5 reveals a closely similar convergence behavior, which indicates that increasing the time domain does not affect



(a) Dimensional splitting

(b) Domain decomposition splitting

Fig. 5. Evolution of the error for the u-component of the Parareal/FIE-FIE, Parareal/FIE-DR, Parareal/FIE-MD, and Parareal/FIE-LIPR methods with respect to the fine approximation for the Gray-Scott problem, considering (a) dimensional splitting or (b) domain decomposition splitting, and choosing $h = 0.025$, $\Delta T = 0.05$, $\delta t = 2.5 \cdot 10^{-3}$ and $T = 200$.

the convergence of the proposed algorithms. Consequently, given that larger time intervals allow for enhanced parallelization, the solvers presented here constitute a highly effective option for problems with a large final time T.

Next, in order to discuss the scalability properties of the proposed methods and compare them with their sequential counterparts, we perform several scalability tests. We set $h = 0.025$, $\Delta T = 1/32$ and $\delta t = 1/640$, so that $s = 20$. Two and three iterations are computed, respectively, for the first- and second-order solvers, thus obtaining global errors with a similar order of magnitude. In this experiment, we specially focus on the speedup obtained from the parallelization of the coarse propagator, since the classical version of the parareal algorithm only considers parallelization of the fine propagator.

In this framework, we assign 2, 4, or 8 processors for the spatial parallelization of the sequential coarse propagator. The computations required for the fine propagator are parallelized in space and time as follows. Let $j \in \{2, 4, 8, 16, 32, 64\}$ denote the total number of processors under consideration. For each value of j, we will consider $j/2$ large time slots for the time parallelization of the fine propagator. Then, inside each time slot, two processors will be used for the spatial parallelization. For instance, if 16 processors are available, 8 large time slots will be considered for the time parallelization (each one with length $T/8$), and two processors will be assigned to each of them in order to perform the spatial parallelization.

In addition, we also compare both partitioning strategies. In the case of domain decomposition splittings, we define two different settings, each corresponding to one of the propagators. In particular, for the coarse propagator, we consider $q = 2$, $q = 4$ or $q = 8$ strips, and an overlapping size $\beta = 1/32$, while, for the fine propagator, we choose $q = 2$ strips and $\beta = 1/16$.

On the other hand, for the sequential counterparts of the proposed methods, we consider the first-order IMEX-θ method (cf. [46]), with $\theta = 1$, and the second-order IMEX Runge-Kutta method proposed in [8, section IV.4.3]. This latter scheme results from the combination of the implicit and explicit trapezoidal rules, and is given by

$$\begin{cases} \left(I - \frac{\tau}{2}A\right) U_{n,1} = U_n + \tau R(U_n) + \frac{\tau}{2}(AU_n + F(t_n) + F(t_{n+1})), \\ U_{n+1} = U_n + \frac{\tau}{2}(A(U_n + U_{n,1}) + R(U_n) + R(U_{n,1}) + F(t_n) + F(t_{n+1})), \end{cases} \tag{15}$$

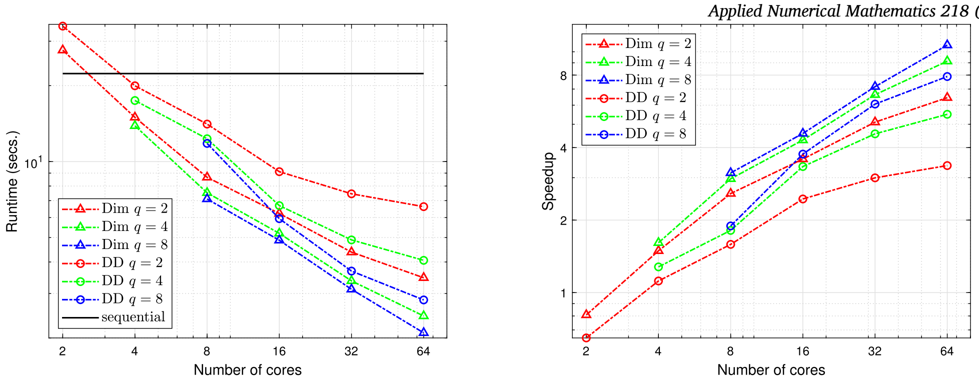
for $n \in \{0, 1, \dots, N-1\}$. This choice is based on the stability and convergence properties of the splitting time integrators under study. Although more sophisticated solvers could be considered (see, e.g., [47]), they would be more expensive in terms of computational cost.

In this framework, we recall the definition of the speedup as

$$S(q) = \frac{T_s}{T_p(q)},$$

where T_s is the runtime for a sequential integration of the problem at hand, while $T_p(q)$ denotes the runtime for its parallel solution using q calculation cores. The computations were performed on a shared-memory machine with 256 GB of RAM and 192 AMD CPU Epyc7513 2.6 GHz processors, hosted by CESAR (Supercomputing Center of Aragón, Spain).

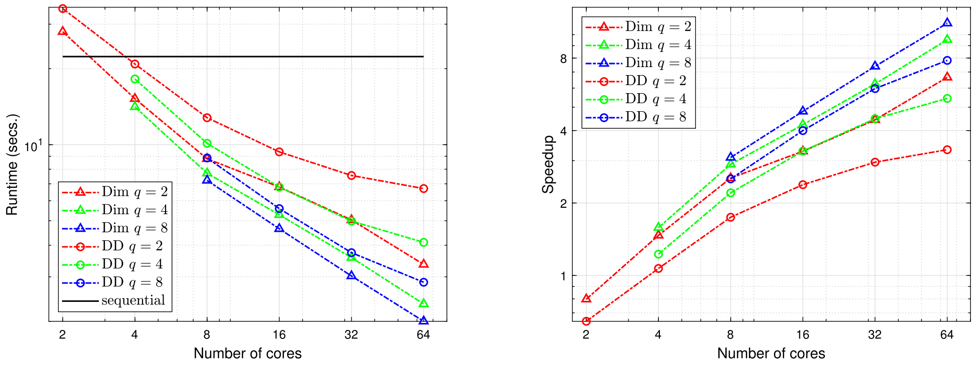
Figs. 6-9 show the CPU runtime (in seconds) and the corresponding speedup for an increasing number of processors. Overall, at most eight calculation cores suffice to outperform the sequential implementation in all the cases. In addition, dimensional splitting seems to be faster than domain decomposition splitting, since the former strategy yields tridiagonal systems, while pentadiagonal systems are obtained with the latter. Note that considering an increasing number of processors $q \in \{2, 4, 8\}$ for the coarse solver permits to reduce the total runtime and to increase the speedup of the method. This is a key advantage of the new proposal as compared with the classical parareal algorithm, which restricts the time parallelization to the fine propagator. Observe that the first-order solvers perform faster than the second-order ones, due to the fact that an extra iteration is considered for the latter. As a result, we can conclude that, if a large number of processors is available, the new algorithms show very good scalability properties for problems with high computational demands.



(a) Runtime in seconds

(b) Speedup

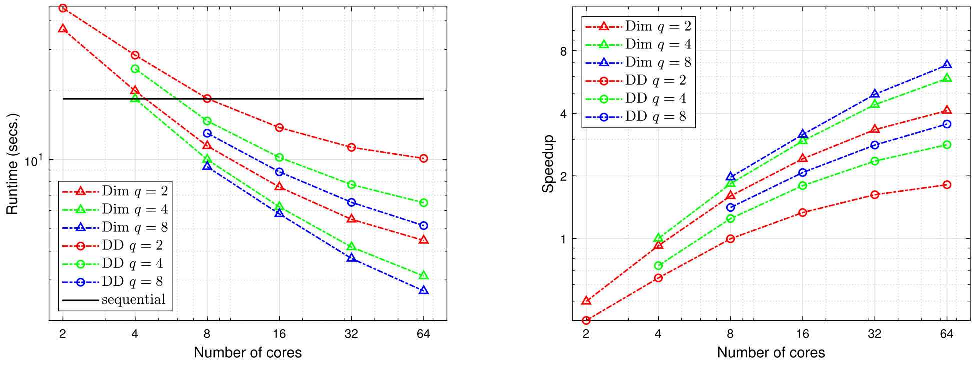
Fig. 6. Scalability results for the Parareal/FIE-FIE method applied to the Gray-Scott model for both dimensional (Dim) and domain decomposition (DD) splittings, with $h = 0.025$, $\Delta T = 1/32$, and $\delta t = 1/640$. The IMEX-θ method, with $\theta = 1$, is considered as the sequential integrator.



(a) Runtime in seconds

(b) Speedup

Fig. 7. Scalability results for the Parareal/FIE-DR method applied to the Gray-Scott model for both dimensional (Dim) and domain decomposition (DD) splittings, with $h = 0.025$, $\Delta T = 1/32$, and $\delta t = 1/640$. The IMEX-θ method, with $\theta = 1$, is considered as the sequential integrator.

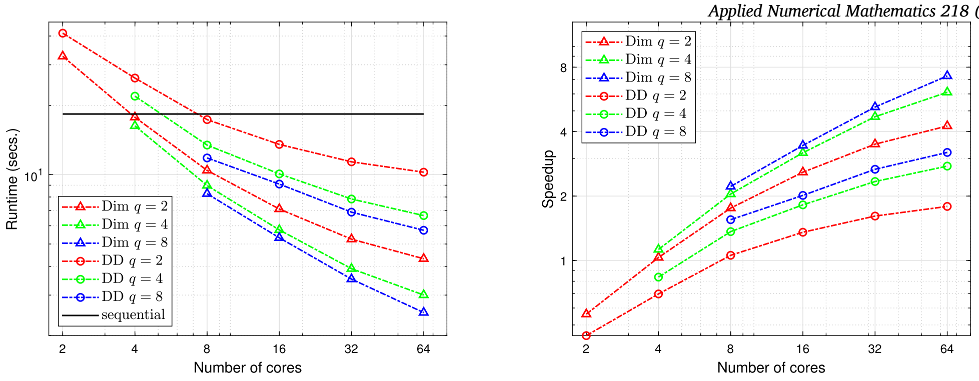


(a) Runtime in seconds

(b) Speedup

Fig. 8. Scalability results for the Parareal/FIE-MD method applied to the Gray-Scott model for both dimensional (Dim) and domain decomposition (DD) splittings, with $h = 0.025$, $\Delta T = 1/32$, and $\delta t = 1/640$. The IMEX Runge-Kutta method (15) is considered as the sequential integrator.

Finally, we illustrate the computational savings provided by the combined parareal splitting solvers with respect to splitting methods without parareal and to classical unsplit IMEX methods. More specifically, we show the CPU runtime required to achieve a prescribed error tolerance for the Parareal/FIE-FIE, the Parareal/FIE-DR, the FIE, the DR, and the IMEX-θ methods, with $\theta = 1$. First, we discuss whether the partitioning techniques improve the IMEX-θ method, since the former can be parallelized at the cost of assuming a splitting error. For that purpose, we fix the spatial discretization parameter to $h = 0.02$, and implement the aforementioned methods for an increasing number of time steps, ensuring that a prescribed accuracy is reached. Dimensional splitting is considered as



(a) Runtime in seconds

(b) Speedup

Fig. 9. Scalability results for the Parareal/FIE-LIPR method applied to the Gray-Scott model for both dimensional (Dim) and domain decomposition (DD) splittings, with $h = 0.025$, $\Delta T = 1/32$, and $\delta t = 1/640$. The IMEX Runge-Kutta method (15) is considered as the sequential integrator.

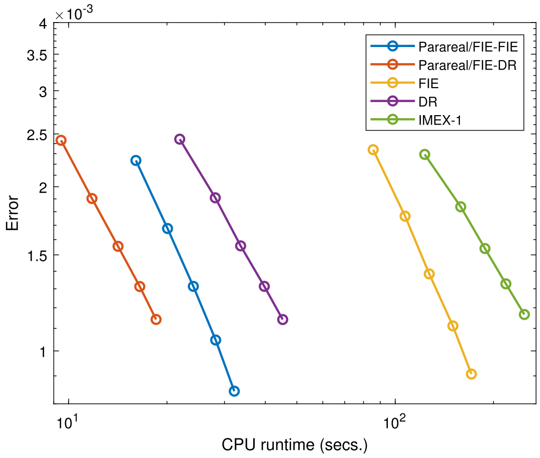


Fig. 10. CPU runtime for certain error tolerances performed by the Parareal/FIE-FIE, the Parareal/FIE-DR, the FIE, the DR, and the IMEX-1 for fixed $h = 0.02$ and applied to the Gray-Scott model.

the partitioning strategy. Fig. 10 shows the results of the described analysis. As expected, due to the smaller size of its splitting error, the DR method displays similar errors for lower runtimes than the FIE method, and both of them outperform the IMEX-θ method, which is shown to be the most time-consuming scheme. A similar behavior can also be observed when comparing the Parareal/FIE-DR with the Parareal/FIE-FIE method, and both of them with the IMEX-θ method. Remarkably, the combined parareal splitting solvers are the fastest methods, and even the Parareal/FIE-FIE scheme outperforms the DR method. Thus, we conclude that space-time parallelization can make a difference in the search for faster computations for complex problems demanding relatively fine spatial and temporal meshes.

5.2. Thomas model

In this subsection, we consider a Turing pattern model governing the reactions between oxygen and urid acid. It is known as the Thomas model and it was first introduced in [48] to describe activator-substrate kinetics with uncompetitive inhibition mechanisms. The model considers null source terms, while the nonlinear reactive terms are given by

$$r_1(u,v) = \gamma\left(a - u - \frac{\rho u v}{1 + u + K u^2}\right), \tag{16}$$

$$r_2(u,v) = \gamma\left(\alpha b - \alpha v - \frac{\rho u v}{1 + u + K u^2}\right). \tag{17}$$

In the sequel, we perform an experiment proposed by Murray in [11], considering the spatial domain $\Omega = (-2,2) \times (-2,2)$ and $T = 4$. We set the parameter values $\rho = 18.5$, $a = 92$, $b = 64$, $\gamma = 25$, $\alpha = 1.5$, $K = 0.1$, $d_1 = 1$, and $d_2 = 10$, and the following initial conditions

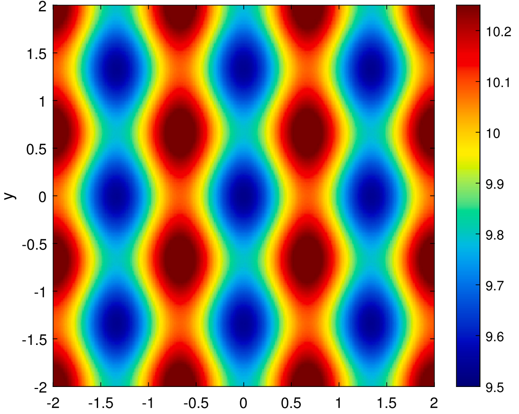


Fig. 11. Approximation of the u-component for the Thomas problem computed by the MD method, considering dimensional splitting with $h = 0.02$ and $\tau = 10^{-5}$.

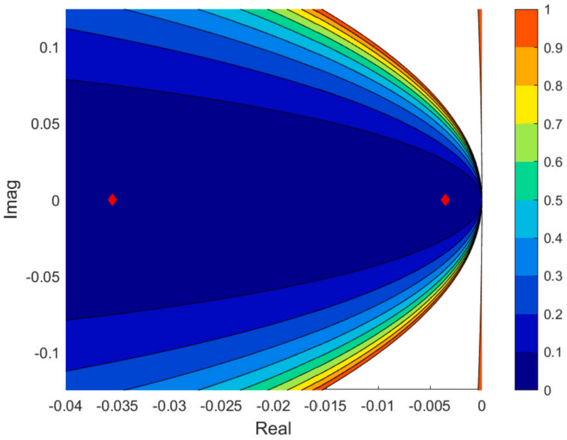


Fig. 12. Convergence region of the Parareal/FIE-FIE method and scaled eigenvalues of the reactive part (red diamonds) for the Thomas model, for $h = 0.02$, $\Delta T = 2 \cdot 10^{-4}$ and $\delta t = 10^{-5}$.

$$u_0(x, y) = 10 - 0.05 \sum_{k=1}^{9} \cos(\pi k x/2),$$

$$v_0(x, y) = 9 + 0.05 \sum_{k=1}^{9} \cos(\pi k y/2).$$

Let us first illustrate the pattern described by the reaction-diffusion problem given by (13), (16) and (17), with the preceding initial conditions. Fig. 11 shows the approximation of the u-component computed by the MD method, considering dimensional splitting, with $h = 0.02$ and $\tau = 10^{-5}$.

In order to test the proposed space-time parallel methods, we consider a spatial mesh size $h = 0.02$, and the coarse and fine time steps $\Delta T = 2 \cdot 10^{-4}$ and $\delta t = 10^{-5}$, respectively. In this setting, the convergence region of the Parareal/FIE-FIE method and the eigenvalues of the reactive part multiplied by ΔT are plotted in Fig. 12. Note that, once again, both eigenvalues $\lambda_{0,1} \approx -177.532425$ and $\lambda_{0,2} \approx -17.784571$ are real and negative, while the scaled eigenvalues $z_{0,1} \approx -0.035506$ and $z_{0,2} \approx -0.003557$ are within the convergence region of the Parareal/FIE-FIE method. As compared with the previous model, the eigenvalues are significantly larger in this case. Therefore, a finer time step ΔT is chosen so that $z_{0,1}$ and $z_{0,2}$ are within the convergence region. Note that both values are also inside the convergence regions of the rest of the newly proposed methods.

According to the previous analysis and given the simple geometry of the spatial domain, dimensional splitting seems to be the best choice to decompose the elliptic operator into split terms. We first study the convergence behavior of the four given solvers, considering a fixed mesh size $h = 0.02$ that guarantees a stiff regime, and then justify the use of these methods rather than explicit schemes.

Fig. 13 depicts the evolution of the error for the u-component with respect to the fine approximation, for all four methods. A similar behavior is observed for the v-component. The solvers under consideration converge, with the first-order methods showing a slightly higher convergence rate.

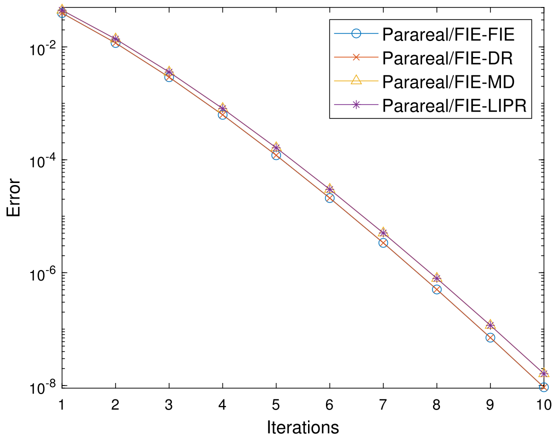


Fig. 13. Evolution of the error for the u-component of the Parareal/FIE-FIE, Parareal/FIE-DR, Parareal/FIE-MD, and Parareal/FIE-LIPR methods with respect to the fine approximation for the Thomas problem, considering dimensional splitting, and choosing $h = 0.02$, $\Delta T = 2 \cdot 10^{-4}$ and $\delta t = 10^{-5}$.

Table 3
Number of iterations required by the solvers to achieve a fixed tolerance of 10^{-6}, with $h = 0.5/2^j$, $\Delta T = 2 \cdot 10^{-4}$ and $\delta t = 10^{-5}$, for increasing values of j.

	$j = 0$	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$
Parareal/FIE-FIE	9	9	8	8	8	8
Parareal/FIE-DR	9	9	8	8	8	8
Parareal/FIE-MD	9	9	9	8	8	8
Parareal/FIE-LIPR	9	9	9	8	8	8

Finally, we will compare the newly proposed algorithms with classical explicit methods. In this experiment, we considered the discretization parameters $h = 0.02$, $\Delta T = 2 \cdot 10^{-4}$ and $\delta t = 10^{-5}$. As pointed out above, the reactive term demands a small time step in order to be located inside the stability region (cf. [8]), but it does not depend on the spatial mesh size. Therefore, if the time step is sufficiently small, considering finer spatial grids does not affect stability. This is not true for explicit methods. A simple stability analysis concludes that, if we set $h = 0.02$ for the Thomas model, a time step smaller than $3 \cdot 10^{-6}$ is required to ensure the stability of classical explicit methods. This situation becomes even worse if we consider successive refinements of the spatial grid, since the time step needs to be reduced accordingly.

To conclude, we perform a robustness analysis of the proposed space-time parallel solvers with respect to the spatial mesh size h. As shown in Table 3, when the grid is successively refined, the methods require the same (or even a smaller) number of iterations to achieve a prescribed tolerance. This property, together with the fact that finer spatial meshes do not require smaller time steps, implies that the new algorithms outperform explicit time integrators for fine grids, while the latter may be preferable for coarser ones.

5.3. Schnakenberg model

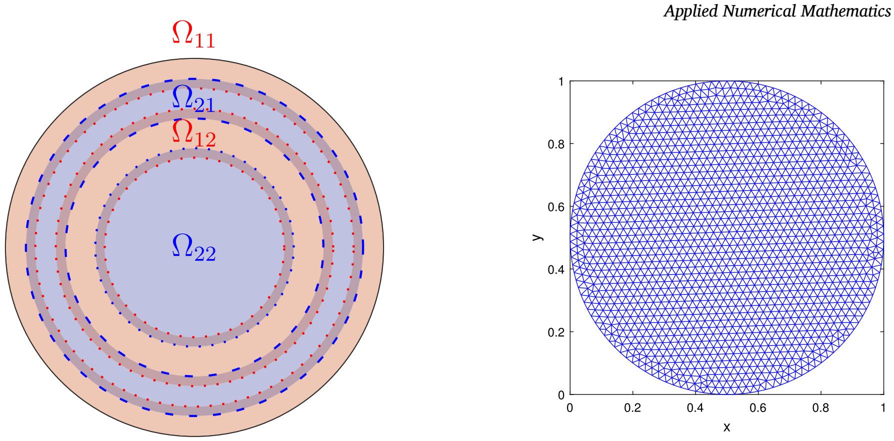
So far, we have considered reaction-diffusion models posed on square domains discretized by means of finite difference methods. As mentioned in the introduction, there exist specific algorithms specially designed for such domains, e.g., matrix-oriented techniques or exponential time-differencing methods (see [13,16,17] for further details). These methods, however, cannot be applied when the spatial domain does not admit a transformation into a rectangular domain (cf. [49]). In this setting, the newly proposed solvers can deal with complex geometries and more general discretizations preserving their efficiency.

In order to illustrate this idea, let us consider a classical activator-substrate model due to Schnakenberg (cf. [50]). It is an autocatalytic system involving two reactants in the kinetics with at least three reactions. In this model, lateral inhibition is achieved via substrate consumption. The corresponding reaction-diffusion problem considers null source terms and the nonlinear reaction functions

$$r_1(u, v) = \gamma(a - u + u^2 v), \qquad r_2(u, v) = \gamma(b - u^2 v). \tag{18}$$

The spatial domain Ω is the circle centered at $(0.5, 0.5)$ with radius 0.5, and the final time is $T = 0.25$ (see [18] for similar experiments). The values of the parameters are set according to [15, Example 5] (clearer figures for the same example are depicted in [51] for a rectangular domain). In particular, we choose $a = 0.126779$, $b = 0.792366$, $\gamma = 1000$, $d_1 = 1$, and $d_2 = 10$, and the initial conditions are defined as suitable perturbations of the steady state solution

$$u_0(x, y) = 0.919145 + 0.0016 \cos(2\pi(x + y)) + 0.01 \sum_{k=1}^{8} \cos(2\pi k x),$$



(a) Domain decomposition

(b) Triangular mesh

Fig. 14. (a) Overlapping domain decomposition for a circle formed by two subdomains as the union of two concentric rings and (b) triangular mesh covering the spatial domain.

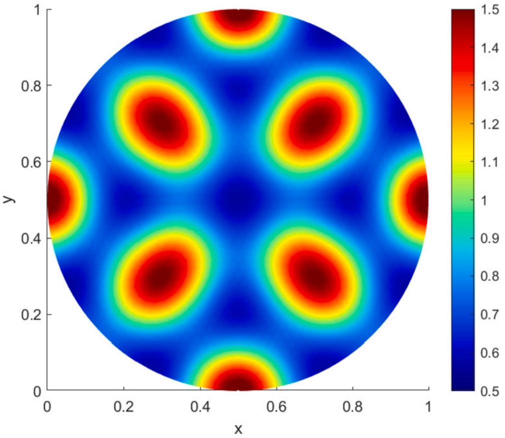


Fig. 15. Approximation of the u-component for the Schnakenberg problem computed by the MD method, considering domain decomposition splitting with $h = 0.005$ and $\tau = 10^{-6}$.

$$v_0(x, y) = 0.937903 + 0.0016\cos(2\pi(x + y)) + 0.01\sum_{k=1}^{8}\cos(2\pi kx).$$

As explained in Subsection 2.1, the circular domain does not allow for a direct implementation of dimensional splitting and, as a consequence, we are required to use a domain decomposition partitioning for this example. In the sequel, we consider a slight modification of the decomposition defined in Fig. 1 in order to account for the circular domain. Fig. 14(a) depicts a possible decomposition which satisfies the properties established in Subsection 2.1. The radii of the rings are defined in such a way that the area of each ring is the same. For $M = 2$ subdomains, each of which is formed by $q = 2$ rings, the radii are $r_1 = 1/4$, $r_2 = \sqrt{2}/4$, $r_3 = \sqrt{3}/4$, and $r_4 = 1/2$. In the following experiments, we will consider $q = 2$ and $\beta = 0.025$. Thus, the overlapping region between Ω_{11} and Ω_{21} is defined as $\{(x, y) \in \mathbb{R}^2 : (r_1 - \beta/2)^2 < (x - 0.5)^2 + (y - 0.5)^2 < (r_1 + \beta/2)^2\}$. Accordingly, the overlapping regions between Ω_{21} and Ω_{12}, and between Ω_{12} and Ω_{22} are defined using the values of r_2 and r_3, respectively. On the other hand, Fig. 14(b) represents a triangular mesh with a maximum element size $h = 0.025$. Remarkably, the mesh can be defined to be independent of the domain decomposition.

In this setting, Fig. 15 shows an accurate approximation of the u-component for the preceding values of the parameters and initial conditions. The approximation is computed using the MD method with domain decomposition splitting, taking a time stepsize $\tau = 10^{-6}$. Piecewise linear finite elements are considered for the spatial discretization with $h = 0.005$.

Now, let $h = 0.025$, $\Delta T = 2 \cdot 10^{-5}$ and $\delta t = 10^{-6}$ be the discretization parameters for the proposed space-time solvers. In the next experiment, the time steps under consideration are really fine due to two reasons: on the one hand, the model requires small stepsizes in order to obtain qualitatively close approximations to the exact solution (see [15]); on the other, the chosen parameter ΔT is such that the eigenvalues of the reaction term multiplied by ΔT are located within the convergence region of the parareal algorithm.

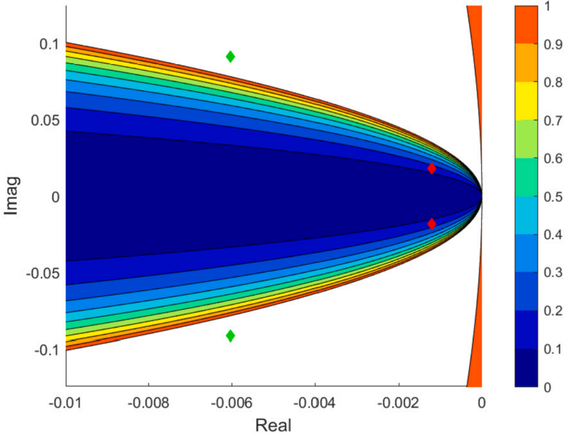


Fig. 16. Convergence region of the Parareal/FIE-FIE method and eigenvalues multiplied by ΔT (red diamonds) or ΔT^* (green diamonds) of the reactive part for the Schnakenberg model, for $h = 0.025$, $\Delta T = 2 \cdot 10^{-5}$, $\Delta T^* = 10^{-4}$ and $\delta t = 10^{-6}$.

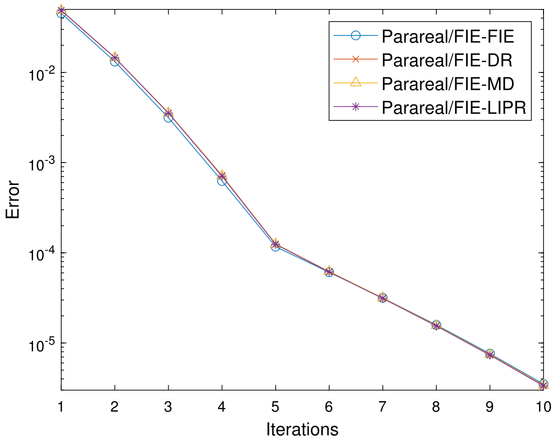


Fig. 17. Evolution of the error for the u-component of the Parareal/FIE-FIE, Parareal/FIE-DR, Parareal/FIE-MD, and Parareal/FIE-LIPR methods with respect to the fine approximation for the Schnakenberg problem, considering domain decomposition splitting, and choosing $h = 0.025$, $\Delta T = 2 \cdot 10^{-5}$, and $\delta t = 10^{-6}$.

Fig. 16 illustrates this latter motivation by plotting the convergence region of the Parareal/FIE-FIE method and the eigenvalues of the reactive term multiplied by ΔT (red diamonds). Note that, unlike in previous models, such eigenvalues are no longer real for the Schnakenberg model. Their values are $\lambda_{0,1} \approx -60.34521 + 917.16192\,i$ and $\lambda_{0,2} \approx -60.34521 - 917.16192\,i$, that is, their imaginary parts are significantly larger than their real parts, which remain negative. In this way, $z_{0,1} \approx -0.001207 + 0.018343i$ and $z_{0,2} \approx -0.001207 - 0.018343i$ lie within the convergence region, so the method should converge as shown later. Observe that, if we consider a coarser time step $\Delta T^* = 10^{-4}$, the scaled eigenvalues obtained when multiplying the eigenvalues by ΔT^* are $z^*_{0,1} \approx -0.006035 + 0.091716i$ and $z^*_{0,2} \approx -0.006035 - 0.091716i$ (green diamonds). These values lie outside the convergence region, which means that the iterative method will fail to converge. In fact, considering such a value for the coarse time step, the errors of the successive iterations drastically increase. As in the previous experiments, similar regions can be obtained for the rest of the proposed solvers. This simple example illustrates the fact that, although the study of the convergence regions does not have a formal theoretical support, it can help us determine a suitable size for the coarse time step ΔT.

Finally, Fig. 17 shows the evolution of the error for the u-component of the proposed methods, when applied to solve the problem given by (13) and (18) with the discretization parameters indicated above. As expected, all the methods have a convergent behavior and, remarkably, their first 5 iterations converge extremely quickly. This is a matter of interest, since a small number of iterations is usually enough to obtain close approximations to the fine solution. In fact, the error of the fine approximation is larger than 10^{-2} for the four solvers, concluding that at most three iterations are sufficient to obtain an approximation whose order of magnitude is similar to that of the error. Note that the curves corresponding to the Parareal/FIE-MD and the Parareal/FIE-LIPR methods overlap in Fig. 17. Numerical experiments suggest that second-order space-time parallel solvers offer more accurate approximations without a significant extra number of iterations. All in all, the experiment shows that the proposed methods are also perfectly implementable for complex geometries and discretizations, with similar performance.

6. Concluding remarks

In this work, we have introduced four space-time parallel solvers based on the combination of the parareal algorithm and different splitting techniques for reaction-diffusion problems. In the resulting methods, the diffusive part has an implicit contribution, while the reactive term is treated explicitly. Since the methods are parallelizable both in space and time, they can be implemented using a large number of processors.

After providing some theoretical convergence results for diffusion problems, the newly proposed algorithms have been applied to the numerical solution of several reaction-diffusion pattern formation models, where fine spatial and temporal discretizations are essential for achieving accurate results. Based on the collection of numerical experiments provided, the methods represent a useful tool to simulate Turing phenomena governed by different models.

The convergence properties of the iterative strategy are quite promising, since just a few iterations are required to get close approximations to the solution provided by a sequential implementation of the fine propagator. The coarse time step is sometimes required to be small, especially for problems with complex eigenvalues in their reactive part. In this context, however, this condition is not a demanding requirement, since small stepsizes are needed to obtain accurate approximations for the models at hand. In addition, it is relevant to note that the number of iterations required for convergence does not increase for second-order splitting methods. Hence, the overall convergence properties are preserved when increasing the order of the time integrators. Remarkably, it has also been shown that increasing the size of the time interval does not impact the convergence behavior of the methods, further reinforcing their suitability for long-time simulations (precisely where parallelization becomes increasingly advantageous).

Suitable robustness and scalability analyses have been performed, obtaining compelling results. On the one hand, the solvers are robust with respect to the spatial discretization parameter, resulting in excellent performance on fine spatial grids. On the other hand, the speedup of parallel computations performed in the scalability test shows an optimal exploitation of the available processors.

Regarding the choice of dimensional or domain decomposition splittings as partitioning techniques, we conclude that while the former generally shows better convergence properties, the latter is significantly more flexible concerning the geometry of the domain and the spatial discretization under consideration. In particular, domain decomposition splittings can be easily extended to finite element or finite volume discretizations on complex domains, as illustrated for the Schnakenberg model on a circular domain.

The ideas exposed in this work for the combination of parallel-in-time methods and time-splitting integrators can be further explored in order to develop more efficient schemes for evolutionary problems. In particular, since the parareal algorithm can be regarded as a two-level multigrid method (cf. [5]), we plan to extend the algorithms proposed here to a multilevel framework that considers the multigrid-reduction-in-time (MGRIT) approach (cf. [6]).

CRediT authorship contribution statement

Andrés Arrarás: Writing – original draft, Visualization, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization, Writing – review & editing, Validation, Supervision. **Francisco J. Gaspar:** Project administration, Writing – review & editing, Validation, Supervision, Methodology, Investigation, Funding acquisition, Conceptualization. **Iñigo Jimenez-Ciga:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Laura Portero:** Writing – review & editing, Writing – original draft, Visualization, Formal analysis, Conceptualization, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition.

References

[1] J. Nievergelt, Parallel methods for integrating ordinary differential equations, Commun. ACM 7 (1964) 731–733.
[2] M.J. Gander, 50 years of time parallel time integration, in: T. Carraro, et al. (Eds.), Multiple Shooting and Time Domain Decomposition Methods, in: Contrib. Math. Comput. Sci., vol. 9, Springer, Cham, 2015, pp. 69–113.
[3] J. Lions, Y. Maday, G. Turinici, Résolution d'EDP par un schéma en temps "pararéel", C. R. Acad. Sci. Paris, Ser. I Math. 332 (2001) 661–668.
[4] G.A. Staff, E.M. Rønquist, Stability of the parareal algorithm, in: T.J. Barth, et al. (Eds.), Domain Decomposition Methods in Science and Engineering, in: Lect. Notes Comput. Sci. Eng., vol. 40, Springer, Berlin/Heidelberg, 2005, pp. 449–456.
[5] M.J. Gander, S. Vandewalle, Analysis of the parareal time-parallel time-integration method, SIAM J. Sci. Comput. 29 (2007) 556–578.
[6] R.D. Falgout, S. Friedhoff, T.V. Kolev, S.P. MacLachlan, J.B. Schroder, Parallel time integration with multigrid, SIAM J. Sci. Comput. 36 (2014) C635–C661.
[7] N.N. Yanenko, The Method of Fractional Steps. The Solution of Problems of Mathematical Physics in Several Variables, Springer-Verlag, Berlin/Heidelberg/New York, 1971.
[8] W. Hundsdorfer, J. Verwer, Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations, Springer Ser. Comput. Maths., vol. 33, Springer-Verlag, Berlin/Heidelberg, 2003.
[9] T.P. Mathew, P.L. Polyakov, G. Russo, J. Wang, Domain decomposition operator splittings for the solution of parabolic equations, SIAM J. Sci. Comput. 19 (1998) 912–932.
[10] A.M. Turing, The chemical basis of morphogenesis, Philos. Trans. R. Soc. Lond. 237 (1952) 37–72.
[11] J.D. Murray, Mathematical Biology II. Spatial Models and Biomedical Applications, Springer-Verlag, New York, 2003.
[12] J.D. Murray, Mathematical Biology I. An Introduction, Springer-Verlag, New York, 2002.
[13] M. Caliari, F. Cassini, A second order directional split exponential integrator for systems of advection-diffusion-reaction equations, J. Comput. Phys. 498 (2024) 112640.
[14] K. Zhang, J.C.F. Wong, R. Zhang, Second-order implicit-explicit scheme for the Gray-Scott model, J. Comput. Appl. Math. 213 (2008) 559–581.
[15] S.J. Ruuth, Implicit-explicit methods for reaction-diffusion problems in pattern formation, J. Math. Biol. 34 (1995) 148–176.
[16] M.C. D'Autilia, I. Sgura, V. Simoncini, Matrix-oriented discretization methods for reaction-diffusion PDEs: comparison and applications, Comput. Math. Appl. 79 (2020) 2067–2085.

[17] E.O. Asante-Asamani, A. Kleefeld, B.A. Wade, A second-order exponential time differencing scheme for non-linear reaction-diffusion systems with dimensional splitting, J. Comput. Phys. 415 (2020) 109490.

[18] J. Zhu, Y.T. Zhang, S.A. Newman, M. Alber, Application of discontinuous Galerkin methods for reaction-diffusion systems in developmental biology, J. Sci. Comput. 40 (2009) 391–418.

[19] P.N. Vabishchevich, Difference schemes with domain decomposition for solving non-stationary problems, USSR Comput. Math. Math. Phys. 29 (1989) 155–160.

[20] P.N. Vabishchevich, Domain decomposition methods with overlapping subdomains for the time-dependent problems of mathematical physics, Comput. Methods Appl. Math. 8 (2008) 393–405.

[21] A. Arrarás, L. Portero, Improved accuracy for time-splitting methods for the numerical solution of parabolic equations, Appl. Math. Comput. 267 (2015) 294–303.

[22] A.A. Samarskii, On an economical difference method for the solution of a multidimensional parabolic equation in an arbitrary region, USSR Comput. Math. Math. Phys. 2 (1963) 894–926.

[23] J. Douglas Jr., H.H. Rachford, On the numerical solution of heat conduction problems in two and three space variables, Trans. Am. Math. Soc. 82 (1956) 421–439.

[24] J. Douglas Jr., J.E. Gunn, Alternating direction methods for parabolic systems in m space variables, J. Assoc. Comput. Mach. 9 (1962) 450–456.

[25] J. Douglas Jr., J.E. Gunn, A general formulation of alternating direction methods. Part I. Parabolic and hyperbolic problems, Numer. Math. 6 (1964) 428–453.

[26] P.L. Lions, B. Mercier, Splitting algorithms for the sum of two nonlinear operators, SIAM J. Numer. Anal. 16 (1979) 964–979.

[27] E. Hansen, A. Ostermann, K. Schratz, The error structure of the Douglas-Rachford splitting method for stiff linear problems, J. Comput. Appl. Math. 303 (2016) 140–145.

[28] W. Hundsdorfer, Accuracy and stability of splitting with stabilizing corrections, Appl. Numer. Math. 42 (2002) 213–233.

[29] A. Arrarás, K.J. in 't Hout, W. Hundsdorfer, L. Portero, Modified Douglas splitting methods for reaction-diffusion equations, BIT Numer. Math. 57 (2017) 261–285.

[30] G.I. Marchuk, Splitting and alternating direction methods, in: P.G. Ciarlet, J.L. Lions (Eds.), Handbook of Numerical Analysis I, vol. 1, North-Holland, Amsterdam, 1990, pp. 197–462.

[31] D.W. Peaceman, H.H. Rachford Jr., The numerical solution of parabolic and elliptic differential equations, J. Soc. Ind. Appl. Math. 3 (1955) 28–41.

[32] L. Portero, J.C. Jorge, A new class of second order linearly implicit fractional step methods, J. Comput. Appl. Math. 218 (2008) 603–615.

[33] A. Arrarás, L. Portero, J.C. Jorge, Linearly implicit domain decomposition methods for nonlinear time-dependent reaction-diffusion problems, in: M. Bercovier, M.J. Gander, R. Kornhuber, O. Widlund (Eds.), Domain Decomposition Methods in Science and Engineering XVIII, in: Lect. Notes Comput. Sci. Eng., vol. 70, Springer-Verlag, Berlin, 2009, pp. 267–274.

[34] L. Baffico, S. Bernard, Y. Maday, G. Turinici, G. Zérah, Parallel-in-time molecular-dynamics simulations, Phys. Rev. E 66 (2002) 057701.

[35] G. Bal, Y. Maday, A "parareal" time discretization for non-linear PDE's with application to the pricing of an American put, in: L. Pavarino, A. Toselli (Eds.), Recent Developments in Domain Decomposition Methods, in: Lect. Notes Comput. Sci. Eng., vol. 23, Springer, Berlin/Heidelberg, 2002, pp. 189–202.

[36] M.J. Gander, F. Kwok, H. Zhang, Multigrid interpretations of the parareal algorithm leading to an overlapping variant and MGRIT, Comput. Vis. Sci. 19 (2018) 59–74.

[37] S. Friedhoff, B.S. Southworth, On "Optimal" h-independent convergence of Parareal and multigrid-reduction-in-time using Runge-Kutta time integration, Numer. Linear Algebra Appl. 28 (2021) e2301.

[38] T. Buvoli, M. Minion, Exponential Runge-Kutta Parareal for non-diffusive equations, J. Comput. Phys. 497 (2024) 112623.

[39] A. Arrarás, F.J. Gaspar, I. Jimenez-Ciga, L. Portero, Space-time parallel iterative solvers for the integration of parabolic problems, arXiv, math.NA, https://arxiv.org/abs/2502.08370, 2025.

[40] R.I. Fernandes, G. Fairweather, An ADI extrapolated Crank-Nicolson orthogonal spline collocation method for nonlinear reaction-diffusion systems, J. Comput. Phys. 231 (2012) 6248–6267.

[41] S. González-Pinto, S. Pérez-Rodríguez, A variable time-step-size code for advection-diffusion-reactions PDEs, Appl. Numer. Math. 62 (2012) 1447–1462.

[42] P.J. van der Houwen, B.P. Sommeijer, Approximate factorization for time-dependent partial differential equations, J. Comput. Appl. Math. 128 (2001) 447–466.

[43] S. Singh, R.C. Mittal, S.R. Thottoli, M. Tamsir, High-fidelity simulations for Turing pattern formation in multi-dimensional Gray-Scott reaction-diffusion system, Appl. Math. Comput. 452 (2023) 128079.

[44] P. Gray, S.K. Scott, Autocatalytic reactions in the isothermal, continuous stirred tank reactor: oscillations and instabilities in the system $A + 2B \to 3B$, $B \to C$, Chem. Eng. Sci. 39 (1984) 1087–1097.

[45] J. Douglas Jr, S. Kim, Improved accuracy for locally one-dimensional methods for parabolic equations, Math. Models Methods Appl. Sci. 11 (2001) 1563–1579.

[46] D. Hoff, Stability and convergence of finite difference methods for systems of nonlinear reaction-diffusion equations, SIAM J. Numer. Anal. 15 (1978) 1161–1177.

[47] X. Zhong, Additive semi-implicit Runge-Kutta methods for computing high-speed nonequilibrium reactive forms, J. Comput. Phys. 128 (1996) 19–31.

[48] D. Thomas, Artificial enzyme membranes, transport, memory and oscillatory phenomena, in: D. Thomas, J.P. Kernevez (Eds.), Analysis and Control of Immobilized Enzyme Systems, Springer-Verlag, Berlin/Heidelberg/New York, 1975, pp. 115–150.

[49] P.M. Knupp, S. Steinberg, Fundamentals of Grid Generation, CRC Press, Boca Raton, 1993.

[50] J. Schnakenberg, Simple chemical reaction systems with limit cycle behaviour, J. Theor. Biol. 81 (1979) 389–400.

[51] A. Madzvamuse, Time-stepping schemes for moving grid finite elements applied to reaction-diffusion systems on fixed and growing domains, J. Comput. Phys. 214 (2006) 239–263.