Applied Numerical Mathematics 200 (2024) 286–308

Contents lists available at ScienceDirect

Applied Numerical Mathematics

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

Matrix-oriented FEM formulation for reaction-diffusion PDEs on a large class of 2D domains

Massimo Frittelli, Ivonne Sgura [*]

Department of Mathematics and Physics "E. De Giorgi", University of Salento, Via per Arnesano, Lecce, 73100, Italy

ARTICLE INFO

Article history:
Received 26 January 2023
Received in revised form 24 June 2023
Accepted 11 July 2023
Available online 16 July 2023

Keywords:
Finite elements
Sylvester equations
IMEX methods
Reaction-diffusion PDEs
Turing patterns
Battery modelling

ABSTRACT

For the spatial discretization of elliptic and parabolic partial differential equations (PDEs), we provide a Matrix-Oriented formulation of the classical Finite Element Method, called MO-FEM, of arbitrary order $k \in \mathbb{N}$. On a quite general class of 2D domains, namely *separable domains*, and even on special surfaces, the discrete problem is then reformulated as a *multiterm Sylvester matrix equation* where the additional terms account for the geometric contribution of the domain shape. By considering the IMEX Euler method for the PDE time discretization, we obtain a sequence of these equations. To solve each multiterm Sylvester equation, we apply the matrix-oriented form of the Preconditioned Conjugate Gradient (MO-PCG) method with a matrix-oriented preconditioner that captures the spectral properties of the Sylvester operator. Solving the Poisson problem and the heat equation on some separable domains by MO-FEM-PCG, we show a gain in computational time and memory occupation wrt the classical vector PCG with same preconditioning or wrt a LU based direct method. As an application, we show the advantages of the MO-FEM-PCG to approximate Turing patterns on some separable domains and cylindrical surfaces for a morphochemical reaction-diffusion PDE system for battery modelling.

1. Introduction

We are interested in the discretisation of parabolic PDEs of the form

$$u_t - \mathcal{L}(u) = f(u, \boldsymbol{x}, t), \qquad (\boldsymbol{x}, t) \in \Omega \times [0, T], \tag{1}$$

where $\Omega \subset \mathbb{R}^2$ is a compact domain, and $-\mathcal{L}(u)$ is a linear second-order elliptic differential operator, and f is a sufficiently smooth reaction term, for instance C^1 in all variables. Problem (1) is endowed with either Dirichlet or Neumann boundary conditions and suitable initial conditions. The relevance of the PDE problem (1) is well-known, as it finds numerous applications across all fields of science. In particular, we are interested in the case of parabolic reaction-diffusion systems (RDSs) where $u(\boldsymbol{x}, t), f \in \mathbb{R}^m, m \geq 2$ because they are the playground of Turing's theory of morphogenesis [45] which encompasses extremely diverse applications. Among them we recall: biological patterning [2,19], developmental biology [32], predator-prey systems [15], biomembrane modelling [13,18], tumour growth [8], metal dealloying [12], financial risk management [4], oscillating chemical reactions [46] and the recent applications to metal growth by electrodeposition [5,30] which we will consider in the present work. We focus on the approximation of Turing patterns, because, from a computational point

[*] Corresponding author.
E-mail addresses: massimo.frittelli@unisalento.it (M. Frittelli), ivonne.sgura@unisalento.it (I. Sgura).

https://doi.org/10.1016/j.apnum.2023.07.010

of view, this is a challenging task since fine meshes are required in space to capture the morphological class of the pattern itself (spots, labyrinths, etc) that must be attained as steady state of the PDE dynamics for long time of integration [9].

Among the existing methods for the spatial discretisation of (1) we mention finite differences [25], finite elements [24], spectral methods [8], kernel methods [26] and many more. The time-dependent problem (1) is commonly treated with the so-called method of lines (MOL), which consists of discretising the spatial variables with a spatial method of choice, thereby producing a continuous-in-time ODE system. Hence, in a general setting including several spatial methods, the resulting spatially discrete problem (for the single equation $m = 1$ in (1)) takes the form of the following ODE system in vector form

$$M\dot{\boldsymbol{z}} + A_u\boldsymbol{z} = \boldsymbol{b}(\boldsymbol{z}, t), \qquad \boldsymbol{z}(0) = \boldsymbol{z}_0, \qquad t \in [0, T], \tag{2}$$

with the vector $\boldsymbol{z} = \boldsymbol{z}(t) \in \mathbb{R}^d$ (where $d \in \mathbb{N}$ is the dimension of the discrete spatial function space) containing the time-dependent coefficients of the expansion of u, the matrix A_u approximating the operator $-\mathcal{L}(u)$, the vector $\boldsymbol{b}(t)$ approximating the right-hand-side of (1), and the matrix M depends on the spatial methods (e.g. the identity matrix for finite differences, mass matrix for finite elements).

The main computational challenge for the solution of (2) is dimensionality. If the discrete function space has dimension d, then $\boldsymbol{z} \in \mathbb{R}^d$ and the matrices A_u, M are of size $d \times d$. This work explores the case where such matrices possess a general Kronecker structure with $n \in \mathbb{N}$ and m terms respectively, e.g.

$$A_u = \sum_{i=1}^{n} R_i \otimes L_i, \qquad M = \sum_{i=1}^{m} P_i \otimes Q_i, \tag{3}$$

with $\otimes$ denoting the Kronecker product and R_i, L_i, P_i, Q_i being matrices of lower dimension, e.g. $\sqrt{d} \times \sqrt{d}$, see for instance [9,22,33,36,38,41]. In such cases, by considering the matrix $U \in R^{\sqrt{d}\times\sqrt{d}}$ such that $\mathtt{vec}(U) = \boldsymbol{z}$, we have

$$A_u\boldsymbol{z} = \mathtt{vec}\left(\sum_{i=1}^{n} L_i U R_i^T\right), \qquad M\dot{\boldsymbol{z}} = \mathtt{vec}\left(\sum_{i=1}^{m} Q_i \dot{U} P_i^T\right), \tag{4}$$

see [36], and the problem (2) can be reformulated as the following linear matrix ODE equation

$$\sum_{i=1}^{m} Q_i \dot{U} P_i^T + \sum_{i=1}^{n} L_i U R_i^T = B(U, t), \tag{5}$$

where $B(U, t)$ is such that $\mathtt{vec}(B(U, t)) = \boldsymbol{b}(\boldsymbol{z}, t)$. Problem (5) is called a *multiterm differential Sylvester equation*, see [44]. After time discretisation on the grid $t_n = nh_t, n = 0, \ldots, N_T := \left\lceil \frac{T}{h_t} \right\rceil$, for example by a one-step solver, at each timestep an algebraic multiterm Sylvester equation (with p terms, for suitable $p \geq 2$) must be solved:

$$\sum_{i=1}^{p} S_i U_{n+1} T_i^T = F_n, \tag{6}$$

where $U_{n+1} \approx U(t_{n+1})$, S_i, T_i are suitable matrices depending on the coefficient matrices Q_i, P_i, L_i, R_i in (4) and F_n is a matrix including all terms from the previous timestep. In this paper, we will show that the number of terms p is related to the shape of the spatial domain Ω and on the differential operator $\mathcal{L}$ in (1). In this work, for time discretisation we choose the Implicit-Explicit (IMEX) Euler because it is the simplest solver particularly effective in the simulation of Turing patterns (see e.g. [19,30]) and already applied in matrix form on square domains in [9]. However, we point out that more general and higher order time solvers can be applied. We will not face this aspect here, since we are mainly concerned with space discretization on domains as general as possible.

The solution of multiterm algebraic Sylvester equations of the type (6) is mostly uncharted territory. The vast majority of methods available for such matrix equations are iterative methods that rely on low rank approximations [43,44] which, however, are not suitable for the simulation of high-rank solutions such as Turing patterns. A special case of (6) worth mentioning is the two-term case $p = 2$, when (6) specialises to a generalised Sylvester equation

$$S_1 U_{n+1} T_1^T + S_2 U_{n+1} T_2^T = F_n, \tag{7}$$

that can be attained for instance when Ω is rectangular, see e.g. [9]. For the special form (7) multiple closed-form algorithms are available, such as the Bartels-Stewart algorithm [3] or its improvement proposed by Golub and others [21]. If L_i, R_i in (5) further fulfil suitable assumptions, even more efficient closed-form algorithms based on spectral decomposition are available, see for instance [9], as we will also discuss in the next sections.

Matrix-oriented (MO) formulations of spatial methods for PDEs were successfully carried out in some notable cases. Elliptic problems on rectangular or parallelepipedal domains were discretised via MO finite differences in [36]. The approach was then extended to more general, but still polygonal domains, see [22]. Stochastic elliptic PDEs were approximated with a

MO Galerkin method in [38]. MO isogeometric analysis was successfully applied to elliptic problems, see [1,33,41], where, on quite general domains defined through splines or NURBS, a Sylvester form can be achieved with low-rank approximations, see [33]. The work in [9] addresses time-dependent problems including RDSs on rectangular domains, solved via MO finite differences. A MO space-time method is proposed in [35] for parabolic problems and in [23] for the wave equation.

In the present work, we consider elliptic and parabolic PDE problems such as the Laplace equation, the semilinear heat equation (1), and parabolic RDSs, posed on a class of two-dimensional spatial domains Ω called *separable domains*. For such PDE problems we propose a MO Finite Element Method (FEM) for the spatial discretisation, that we will define as MO-FEM. The proposed framework advances the existing theory on MO spatial discretisation of PDEs in several directions, as listed below.

1) To the best of the author's knowledge, here we provide the first MO formulation of FEM for elliptic and parabolic PDEs. The proposed theory is general and applies to a large class of basis functions, such as Lagrangian $\mathbb{P}_k$ basis functions, $k \in \mathbb{N}$, thereby covering arbitrarily high-order convergence in space. Special focus is given to the practical special case of lumped $\mathbb{P}_1$ FEM.
2) Thanks to a suitable coordinate transformation, the proposed approach applies to domains more general than rectangles, namely *separable domains*. To the best of the authors' knowledge, our work is the first applications of a MO method to curvilinear domains without needing low rank approximations. This property is crucial in the numerical approximation of Turing patterns, which are typically high rank. Moreover, we can solve PDEs in MO form also on special surfaces, namely cylinders with curvilinear bases. In fact, a subclass of separable domains, namely *normal domains* can be wrapped around a cylinder. Elliptic PDEs on structured surfaces were considered for instance in [14,37], but the special shape of the surface was not exploited in a matrix-oriented fashion.
3) In all cases, the discrete problem takes the form of a p-terms Sylvester equation, where the additional terms for $p > 2$ account for domain shape different from rectangles. In this case, for the numerical approximation of general multiterm Sylvester equations, we propose an iterative method: a MO Preconditioned Conjugate Gradient method (MO-PCG). We prove that it always converges for the considered PDE problems, because the involved differential operators are coercive and self-adjoint, see [40, Section 6.7]. We also show theoretically that the MO-PCG strongly surpasses its classical vector counterpart in terms of memory storage and, for sufficiently high FEM polynomial order k (depending on the PDE problem) there is also an advantage in computational cost.
4) We provide numerical experiments that demonstrate (i) optimal spatial convergence (for the Poisson problem) and (ii) a significant gain in terms of memory storage and execution time for MO-PCG compared to the standard vector PCG and to the LU based direct method. We exploit this feature to face with numerical challenges in the accurate simulation of Turing patterns, obtained as asymptotic solutions of RDSs. In particular, here we solve the morphochemical reaction-diffusion PDE system for battery modelling (see e.g. [30]) on some separable domains with curvilinear boundaries.

The work is structured as follows. In Section 2, we state our model problem on a general separable domain and we show, through a reparametrisation onto a reference square domain, that the discrete differential operator possesses a multiterm Kronecker decomposition. In Section 3, we present the MO-PCG strategy to solve the discrete problem. In Section 4, we show three numerical examples to illustrate our findings on the Poisson problem and the heat equation, in Section 5 we will apply the MO-FEM approach for the approximation of Turing patterns. In the Appendix, we theoretically analyze the MO-PCG performances in terms of memory and computational cost.

2. Model and its FEM discretisation

We consider the model problem

$$\begin{cases} \dot{v} - \nabla\cdot(D\nabla v) = f(v, \boldsymbol{x}, t), & \boldsymbol{x} = (x, y) \in \Omega^S,\ t \in [0, T]; \\ v = 0 \quad \text{or} \quad (D\nabla v)\cdot \boldsymbol{n}(\boldsymbol{x}) = 0, & \boldsymbol{x} \in \partial\Omega^S,\ t \in [0, T] \end{cases} \tag{8}$$

where $\Omega^S \subset \mathbb{R}^2$ is a compact domain *as general as possible* as discussed later, $D = \{d_{ij}\} \in \mathbb{R}^{2\times 2}$ is a symmetric positive definite diffusivity matrix, $\boldsymbol{n}$ is the outward unit normal vector on $\partial\Omega^S$, and $T > 0$ is the final time. For ease of presentation, we show the derivation for the case of homogeneous boundary conditions, of Dirichlet or Neumann type. As discussed later in Remark 1 (Section 3), the proposed methodology still applies for general non-homogeneous Dirichlet or Neumann boundary conditions. In order to derive a MO-FEM discretisation of the model (8), we start by parameterising such model onto the square $\Omega := [0, 1]^2$, see [22].

2.1. Parameterisation onto the reference domain

Assume that there exists a smooth mapping $\mathcal{G}: \Omega \to \Omega^S$, $(\xi, \eta) \mapsto (x(\xi, \eta), y(\xi, \eta))$, where $\Omega = [0, 1]^2$ is the unit square. Let J be the Jacobian determinant of $\mathcal{G}$. As shown in [27], the problem (8) can be rephrased as

$$\begin{cases} J(\boldsymbol{z})\dot{u} - \nabla\cdot(E(\boldsymbol{z})\nabla u) = J(\boldsymbol{z})g(u, \boldsymbol{z}, t), & \boldsymbol{z} = (\xi, \eta) \in \Omega,\ t \in [0, T]; \\ u = 0 \quad \text{or} \quad (E(\boldsymbol{z})\nabla u)\cdot \boldsymbol{n}(\boldsymbol{z}) = 0, & \boldsymbol{z} \in \partial\Omega,\ t \in [0, T], \end{cases} \tag{9}$$

where $u := v \circ \mathcal{G}$, $g(u, \boldsymbol{z}, t) := f(u, \mathcal{G}(\boldsymbol{z}), t)$ and $E : \Omega \to \mathbb{R}^{2\times 2}$ is a symmetric matrix-valued function whose entries are defined by

$$e_{11}(\boldsymbol{z}) = \frac{d_{11}y_\eta^2(\boldsymbol{z}) - 2d_{12}y_\eta(\boldsymbol{z})x_\eta(\boldsymbol{z}) + d_{22}x_\eta^2(\boldsymbol{z})}{J(\boldsymbol{z})}; \tag{10}$$

$$e_{12}(\boldsymbol{z}) = e_{21}(\boldsymbol{z}) = -\frac{d_{11}y_\xi(\boldsymbol{z})y_\eta(\boldsymbol{z}) - d_{12}\Big(x_\xi(\boldsymbol{z})y_\eta(\boldsymbol{z}) + x_\eta(\boldsymbol{z})y_\xi(\boldsymbol{z})\Big) + d_{22}x_\xi(\boldsymbol{z})x_\eta(\boldsymbol{z})}{J(\boldsymbol{z})}; \tag{11}$$

$$e_{22}(\boldsymbol{z}) = \frac{d_{11}y_\xi^2(\boldsymbol{z}) - 2d_{12}y_\xi(\boldsymbol{z})x_\xi(\boldsymbol{z}) + d_{22}x_\xi^2(\boldsymbol{z})}{J(\boldsymbol{z})}, \tag{12}$$

where the subscripts ξ and η denote partial derivatives w.r.t. ξ and η, respectively. In particular, the boundary conditions (homogeneous Dirichlet or Neumann) of the model problem (8) are transformed accordingly as in (9). We remark that, in the more usual case when $d_{11} = d_{22} =: \tilde{d}$ and $d_{12} = d_{21} = 0$, i.e. $\mathcal{L}(u) = \tilde{d}\Delta u$, the formulas (10)-(12) reduce to

$$e_{11}(\boldsymbol{z}) = \tilde{d}\,\frac{y_\eta^2(\boldsymbol{z}) + x_\eta^2(\boldsymbol{z})}{J(\boldsymbol{z})}; \qquad e_{22}(\boldsymbol{z}) = \tilde{d}\,\frac{y_\xi^2(\boldsymbol{z}) + x_\xi^2(\boldsymbol{z})}{J(\boldsymbol{z})} \tag{13}$$

$$e_{12}(\boldsymbol{z}) = e_{21}(\boldsymbol{z}) = -\tilde{d}\,\frac{y_\xi(\boldsymbol{z})y_\eta(\boldsymbol{z}) + x_\xi(\boldsymbol{z})x_\eta(\boldsymbol{z})}{J(\boldsymbol{z})}. \tag{14}$$

Now, to derive a matrix oriented FEM formulation of (9), it is key for the entries of $E(\boldsymbol{z})$ to be finite sums of separable functions. To this end, we specialize $\mathcal{G}$ to be a separable mapping of the form

$$\begin{cases} x(\xi,\eta) = A(\xi)B(\eta), \\ y(\xi,\eta) = C(\xi)D(\eta), \end{cases} \tag{15}$$

which implies that the terms $x_\xi(\boldsymbol{z})$, $x_\eta(\boldsymbol{z})$, $y_\xi(\boldsymbol{z})$, $y_\eta(\boldsymbol{z})$ appearing in (10)-(12) are separable. We are thus left to ensure that $J(\boldsymbol{z})$ is separable. Hence, observe that

$$J(\xi,\eta) = x_\xi(\xi,\eta)y_\eta(\xi,\eta) - x_\eta(\xi,\eta)y_\xi(\xi,\eta) = A'(\xi)B(\eta)C(\xi)D'(\eta) - A(\xi)B'(\eta)C'(\xi)D(\eta). \tag{16}$$

In order for (16) to be separable, it is sufficient to ensure, for instance, that the terms depending on ξ can be grouped, i.e. there exists $\chi \in \mathbb{R}$ such that

$$\chi A'(\xi)C(\xi) = A(\xi)C'(\xi). \tag{17}$$

By solving the separable ODE (17) we obtain that

$$C(\xi) = c|A(\xi)|^\chi, \tag{18}$$

for arbitrary $c, \chi \in \mathbb{R}$. By substituting (18) into (16) we obtain the following separable form for $J(\xi,\eta)$:

$$J(\xi,\eta) = c|A(\xi)|^\chi A'(\xi)\left[B(\eta)D'(\eta) - \chi B'(\eta)D(\eta)\right] =: \sigma(\xi)\zeta(\eta). \tag{19}$$

Now, since J is separable, then from (10)-(12) it follows that each entry of E can be expressed as the sum of three or four separable terms. We can adopt the following notation for the entries e_{ij} of E:

$$e_{ij}(\xi,\eta) = \sum_{s=1}^{4-\delta_{ij}} \lambda_{ij}^s(\xi)\rho_{ij}^s(\eta), \qquad i,j \in \{1,2\}. \tag{20}$$

2.2. FEM discretisation of the parameterised problem

The weak formulation of problem (9) is: find $u \in H^*$ such that

$$\int_\Omega J\dot{u}\varphi + \int_\Omega (E\nabla u)\cdot\nabla\varphi = \int_\Omega Jg\varphi, \tag{21}$$

for all $\varphi \in H^*$, where $H^* = H^1(\Omega)$ in the presence of homogeneous Neumann boundary conditions while $H^* = H_0^1(\Omega)$ in the presence of homogeneous Dirichlet boundary conditions. The weak problem (21) is often discretised in space via the finite element method (FEM), see for instance [7]. Let $\Omega = \bigcup_{T\in\mathcal{T}_h} T$, where $\mathcal{T}_h$ is a collection of non-overlapping triangles or squares covering Ω and let $\mathbb{V}_h$ be the usual Lagrangian function space on the mesh $\mathcal{T}_h$ of given polynomial order $k \in \mathbb{N}$.

Furthermore, let $\mathbb{V}_{h,0} = \{V \in \mathbb{V}_h | V(\boldsymbol{x}) = 0 \ \forall \boldsymbol{x} \in \partial\Omega\}$ and let $\mathbb{V}^* = \mathbb{V}_h$ or $\mathbb{V}^* = \mathbb{V}_{h,0}$ depending on the boundary conditions. Let $\{\varphi_i\}_{i=1}^d$ be the usual Lagrange basis of $\mathbb{V}^*$, where $d \in \mathbb{N}$ is the dimension of $\mathbb{V}^*$. The FEM spatially discrete formulation of (21) is then: find $U \in \mathbb{V}^*$ such that

$$\int_\Omega J\dot{U}\varphi + \int_\Omega (E\nabla U)\cdot\nabla\varphi = \int_\Omega J g_h \varphi, \tag{22}$$

for all $\varphi \in \mathbb{V}^*$, where $g_h \in \mathbb{V}_h$ is the interpolant of g in the space $\mathbb{V}_h$ (and not $\mathbb{V}^*$, regardless of the boundary conditions). By expressing the numerical solution U in the Lagrange basis, i.e. $U(\boldsymbol{z},t) = \sum_{i=1}^d \mu_i(t)\varphi_i(\boldsymbol{z})$, (22) becomes: find $\boldsymbol{\mu} \in C^1([0,T];\mathbb{R}^d)$ such that

$$\sum_{i=1}^d \dot{\mu}_i(t) \int_\Omega J\varphi_i\varphi_j + \sum_{i=1}^d \mu_i(t)\int_\Omega (E\nabla\varphi_i)\cdot\nabla\varphi_j = \int_\Omega J g_h \varphi_j, \tag{23}$$

for all $j = 1,\dots,d$. The discrete problem (23) then takes the form of an ODE system:

$$\widetilde{M}\dot{\boldsymbol{\mu}}(t) + \widetilde{A}\boldsymbol{\mu}(t) = \widetilde{\boldsymbol{b}}(t), \tag{24}$$

where the mass matrix $\widetilde{M} = (\widetilde{m}_{ij}) \in \mathbb{R}^{d\times d}$, the stiffness matrix $\widetilde{A} = (\widetilde{a}_{ij}) \in \mathbb{R}^{d\times d}$ and the load term $\boldsymbol{b} = (b_j) \in \mathbb{R}^d$ are defined by

$$\widetilde{m}_{ij} = \int_\Omega J\varphi_i\varphi_j; \quad \widetilde{a}_{ij} = \int_\Omega (E\nabla\varphi_i)\cdot\nabla\varphi_j; \quad \widetilde{b}_j = \int_\Omega J g_h\varphi_j, \tag{25}$$

for all $i,j = 1,\dots,d$. The main computational issue of (24) is the curse of dimensionality. In particular, the dimension of the ODE system (24) is equal to the dimension d of the discrete space $\mathbb{V}^*$ and involves the matrices $\widetilde{A}, \widetilde{M}$ which are of size $d \times d$. In this paper, we aim at rephrasing the vector ODE system (24) as a matrix ODE system involving much smaller matrices. The main idea is to use the separability property to express the FEM matrices in tensor form and apply the properties of the Kronecker product.

Now, in (22), $J(\boldsymbol{z})$ and the entries of $E(\boldsymbol{z})$ are finite sums of separable terms, and this needs to be mirrored by the basis functions $\{\varphi_i\}_{i=1}^d$ being separable themselves, and the construction of such a separable basis is in turn made possible by the domain Ω of (22) being a square.

2.3. Square mesh and separable basis functions

On the one-dimensional reference domain: $K := [0,1]$, we consider the equally spaced mesh K_h with $(N+1)$ nodes, with $N \in \mathbb{N}$. For $m = 0,\dots,N$, we define the m-th node as $\xi_m = \frac{m}{N}$. Each element of K_h is of the form $E_m = [\xi_m, \xi_{m+1}]$ for some $m = 0,\dots,N-1$. Let $\{\psi_i\}_{i=0}^N$ be any finite element basis on the one-dimensional mesh K_h, e.g. piecewise Lagrange polynomials of any fixed degree $k \in \mathbb{N}$. On $\Omega = [0,1]^2$ we consider the Cartesian mesh Ω_h with $(N+1)\times(N+1)$ nodes. For $m,n = 0,\dots,N$, we define the nodes $\boldsymbol{z}_{mn} := (\xi_m, \eta_n) = (\frac{m}{N}, \frac{n}{N})$. Each element of Ω_h is of the form $Q_{mn} := [\xi_m, \xi_{m+1}] \times [\eta_n, \eta_{n+1}]$ for $m,n = 0,\dots,N-1$. On Ω_h we choose the tensor-product local Lagrange basis $\{\varphi_i\}_{i=1}^{(N+1)^2}$ defined as

$$\varphi_i(\xi,\eta) := \psi_\ell(\xi)\psi_r(\eta), \qquad \text{for all } \ell, r = 0,\dots,N, \tag{26}$$

where $i := \ell + (N+1)r + 1$, meaning that every 2D basis function can be uniquely decomposed as a product of 1D basis functions. The choice of the tensor-product Lagrange basis into the discrete parameterised problem (22) will yield the desired Kronecker decomposition of the mass matrix $\widetilde{M}$ and of the stiffness matrix $\widetilde{A}$ in (25), as described in the next subsection.

2.4. Kronecker decomposition of the FEM matrices

All the ingredients are ready to derive a Kronecker decomposition of the mass and stiffness matrices defined in (25). By substituting (10)-(12) and (26) into (25) we obtain

$$\widetilde{m}_{ij} = \int_\Omega J(\xi,\eta)\varphi_i(\xi,\eta)\varphi_j(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta = \int_0^1 \sigma(\xi)\psi_\ell(\xi)\psi_m(\xi)\mathrm{d}\xi \int_0^1 \zeta(\eta)\psi_r(\eta)\psi_n(\eta)\mathrm{d}\eta; \tag{27}$$

$$\widetilde{a}_{ij} = \int_\Omega (E(\xi,\eta)\nabla\varphi_i(\xi,\eta))\cdot\nabla\varphi_j(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta \tag{28}$$

$$
\begin{aligned}
&= \sum_{s=1}^{3} \int_0^1 \psi'_\ell(\xi)\psi'_m(\xi)\lambda^s_{11}(\xi)\mathrm{d}\xi \int_0^1 \psi_r(\eta)\psi_n(\eta)\rho^s_{11}(\eta)\mathrm{d}\eta \\
&+ \sum_{s=1}^{4} \int_0^1 \psi'_\ell(\xi)\psi_m(\xi)\lambda^s_{12}(\xi)\mathrm{d}\xi \int_0^1 \psi_r(\eta)\psi'_n(\eta)\rho^s_{12}(\eta)\mathrm{d}\eta \\
&+ \sum_{s=1}^{4} \int_0^1 \psi_\ell(\xi)\psi'_m(\xi)\lambda^s_{21}(\xi)\mathrm{d}\xi \int_0^1 \psi'_r(\eta)\psi_n(\eta)\rho^s_{21}(\eta)\mathrm{d}\eta \\
&+ \sum_{s=1}^{3} \int_0^1 \psi_\ell(\xi)\psi_m(\xi)\lambda^s_{22}(\xi)\mathrm{d}\xi \int_0^1 \psi'_r(\eta)\psi'_n(\eta)\rho^s_{22}(\eta)\mathrm{d}\eta,
\end{aligned}
$$

for all $i, j = 1, \dots, (N+1)^2$, where i depends on ℓ, r and j depends on m, n as in (26). Hence, the mass matrix $\widetilde{M}$ can be expressed as a single Kronecker product, while the stiffness matrix $\widetilde{A}$ can be expressed as the sum of fourteen Kronecker products, some of which can vanish depending on the domain shape, as we will see in the following. Specifically, we can rewrite (27) and (28) as

$$
\widetilde{M} = M^\eta \otimes M^\xi; \qquad \widetilde{A} = \sum_{s=1}^{14} A^\eta_s \otimes A^\xi_s, \tag{29}
$$

where $M^\xi, M^\eta \in \mathbb{R}^{(N+1)\times(N+1)}$ are defined by

$$
m^\xi_{ij} = \int_0^1 \sigma(\xi)\psi_i(\xi)\psi_j(\xi)\mathrm{d}\xi, \quad m^\eta_{ij} = \int_0^1 \zeta(\eta)\psi_i(\eta)\psi_j(\eta)\mathrm{d}\eta, \tag{30}
$$

for all $i, j = 1, \dots, N+1$. Similarly, the entries of A^η_s and A^ξ_s are defined by the univariate integrals appearing on the right hand side of (28). We omit the explicit definitions for ease of presentation.

2.5. Kronecker form of FEM matrices in the presence of lumping

The Kronecker forms (29) are general and encompass arbitrary choices of the FEM basis function, thereby including the case of Lagrangian spatial methods of any order $k \in \mathbb{N}$. This section is devoted to the special case of Lagrangian $\mathbb{P}_1$ finite elements *with mass and stiffness lumping*, see [34]. We show that such special case significantly simplifies the matrix identities (29), thereby providing a counterpart of the standard FEM of low order $k = 1$ where the matrices M^η, M^ξ, A^ξ_s and A^η_s in (29) are easier to compute.

First, we introduce the element-wise interpolant operator, see [34]. If $u : E_m \to \mathbb{R}$ is a univariate continuous function, let $I_h(u) \in \mathbb{P}_1(E_m)$ be the unique first degree polynomial on $E_m = [\xi_m, \xi_{m+1}]$ such that $u(\xi_i) = I_h u(\xi_i)$ for $i \in \{m, m+1\}$. Analogously, if $v : Q_{mn} \to \mathbb{R}$ is a bivariate continuous function, let $I_h v \in \mathbb{P}_1(E_m) \times \mathbb{P}_1(E_n)$ be the unique bivariate polynomial on Q_{mn} that matches u on the vertexes of Q_{mn}, i.e. $u(\xi_i, \eta_j) = I_h v(\xi_i, \eta_j)$ for $i, j \in \{m, m+1\}$. With slight abuse of notation, we have denoted the interpolant operator of both univariate and bivariate functions with the same symbol I_h. Moreover, we will apply I_h to functions that are defined on the whole domain rather than on a single element, in that case the interpolant is applied elementwise. We introduce the so-called lumped mass and stiffness matrices $\widehat{M} \in \mathbb{R}^{d\times d}$ and $\widehat{A} \in \mathbb{R}^{d\times d}$ with entries

$$
\widehat{m}_{ij} = \int_\Omega I_h(J\varphi_i\varphi_j); \quad \widehat{a}_{ij} = \int_\Omega I_h((E\nabla\varphi_i)\cdot\nabla\varphi_j), \qquad i, j = 1, \dots, d, \tag{31}
$$

which represent an approximation of the mass and stiffness matrices $\widetilde{M}$ and $\widetilde{A}$ defined in (25). We can see that lumping can be regarded as the usage of the element-wise bivariate trapezoidal rule for quadrature. Crucial in the next computation is that the interpolant of a separable bivariate function is separable, i.e.

$$
I_h(A(x)B(y)) = I_h(A(x))I_h(B(y)). \tag{32}
$$

Another key property is that the univariate and bivariate Lagrange basis functions $\{\psi_i\}_{i=0}^N$ and $\{\varphi_k\}_{k=1}^{(N+1)^2}$ are orthogonal with respect to the interpolant, i.e.

$$I_h(\psi_i\psi_j)=\begin{cases}\psi_i \text{ if } i=j\\ 0 \text{ otherwise}\end{cases}\qquad I_h(\varphi_k\varphi_\ell)=\begin{cases}\varphi_k \text{ if } k=\ell\\ 0 \text{ otherwise}\end{cases}\tag{33}$$

for all $i,j=0,\dots,N$ and $k,\ell=1,\dots,(N+1)^2$. Thanks to these properties, we can obtain a Kronecker decomposition of the lumped mass and stiffness matrices $\widehat{M}$ and $\widehat{A}$. By substituting (10)-(12) and (26) into (31) we obtain:

$$\begin{aligned}\widehat{m}_{ij}&=\int_\Omega I_h\Big(J(\xi,\eta)\varphi_i(\xi,\eta)\varphi_j(\xi,\eta)\Big)\mathrm{d}\xi\,\mathrm{d}\eta\\ &=\int_0^1 I_h\Big(\sigma(\xi)\psi_\ell(\xi)\psi_m(\xi)\Big)\mathrm{d}\xi\int_0^1 I_h\Big(\zeta(\eta)\psi_r(\eta)\psi_n(\eta)\Big)\mathrm{d}\eta\\ &=\sigma(\xi_\ell)\zeta(\eta_n)\int_0^1 I_h\Big(\psi_\ell(\xi)\psi_m(\xi)\Big)\mathrm{d}\xi\int_0^1 I_h\Big(\psi_r(\eta)\psi_n(\eta)\Big)\mathrm{d}\eta;\end{aligned}\tag{34}$$

$$\begin{aligned}\widehat{a}_{ij}&=\int_\Omega I_h\Big((E(\xi,\eta)\nabla\varphi_i(\xi,\eta))\cdot\nabla\varphi_j(\xi,\eta)\Big)\mathrm{d}\xi\,\mathrm{d}\eta\\ &=\sum_{s=1}^{3}\int_0^1 I_h\Big(\psi_\ell'(\xi)\psi_m'(\xi)\lambda_{11}^s(\xi)\Big)\mathrm{d}\xi\int_0^1 I_h\Big(\psi_r(\eta)\psi_n(\eta)\rho_{11}^s(\eta)\Big)\mathrm{d}\eta\\ &+\sum_{s=1}^{4}\int_0^1 I_h\Big(\psi_\ell'(\xi)\psi_m(\xi)\lambda_{12}^s(\xi)\Big)\mathrm{d}\xi\int_0^1 I_h\Big(\psi_r(\eta)\psi_n'(\eta)\rho_{12}^s(\eta)\Big)\mathrm{d}\eta\\ &+\sum_{s=1}^{4}\int_0^1 I_h\Big(\psi_\ell(\xi)\psi_m'(\xi)\lambda_{21}^s(\xi)\Big)\mathrm{d}\xi\int_0^1 I_h\Big(\psi_r'(\eta)\psi_n(\eta)\rho_{21}^s(\eta)\Big)\mathrm{d}\eta\\ &+\sum_{s=1}^{3}\int_0^1 I_h\Big(\psi_\ell(\xi)\psi_m(\xi)\lambda_{22}^s(\xi)\Big)\mathrm{d}\xi\int_0^1 I_h\Big(\psi_r'(\eta)\psi_n'(\eta)\rho_{22}^s(\eta)\Big)\mathrm{d}\eta.\end{aligned}\tag{35}$$

Since the derivatives of the ψ_i's are piecewise constant, (35) reduces to

$$\begin{aligned}\widehat{a}_{ij}&=\left[\sum_{s=1}^{3}\frac{\lambda_{11}^s(\xi_\ell)+\lambda_{11}^s(\xi_m)+(\lambda_{11}^s(\xi_{\ell-1})+\lambda_{11}^s(\xi_{\ell+1}))\delta_{\ell m}}{2+2\delta_{\ell m}}\rho_{11}^s(\eta_n)\right]\\ &\cdot\int_0^1\psi_\ell'(\xi)\psi_m'(\xi)\mathrm{d}\xi\int_0^1 I_h\Big(\psi_r(\eta)\psi_n(\eta)\Big)\mathrm{d}\eta\\ &+\left[\sum_{s=1}^{4}\lambda_{12}^s(\xi_m)\rho_{12}^s(\eta_r)\right]\int_0^1\psi_\ell'(\xi)\psi_m(\xi)\mathrm{d}\xi\int_0^1\psi_r(\eta)\psi_n'(\eta)\mathrm{d}\eta\\ &+\left[\sum_{s=1}^{4}\lambda_{21}^s(\xi_\ell)\rho_{21}^s(\eta_n)\right]\int_0^1\psi_\ell(\xi)\psi_m'(\xi)\mathrm{d}\xi\int_0^1\psi_r'(\eta)\psi_n(\eta)\mathrm{d}\eta\\ &+\left[\sum_{s=1}^{3}\lambda_{22}^s(\xi_\ell)\frac{\rho_{22}^s(\eta_r)+\rho_{22}^s(\eta_n)+(\rho_{22}^s(\eta_{r-1})+\rho_{22}^s(\eta_{r+1}))\delta_{rn}}{2+2\delta_{rn}}\right]\\ &\cdot\int_0^1 I_h\Big(\psi_\ell(\xi)\psi_m(\xi)\Big)\mathrm{d}\xi\int_0^1\psi_r'(\eta)\psi_n'(\eta)\mathrm{d}\eta.\end{aligned}\tag{36}$$

To rewrite (34) and (36) in matrix form we need some matrix notation. For any $\mathbf{v}\in\mathbb{R}^{N+1}$ we define the operators $\operatorname{diag}\mathbf{v}\in\mathbb{R}^{(N+1)\times(N+1)}$ and $\operatorname{convol}\mathbf{v}\in\mathbb{R}^{(N+1)\times(N+1)}$ as follows:

$$\operatorname{diag}\mathbf{v} = \begin{pmatrix} v_1 & & & \\ & v_2 & & \\ & & \ddots & \\ & & & v_{N+1} \end{pmatrix}; \quad \operatorname{convol}\mathbf{v} = \begin{pmatrix} \frac{v_1+v_2}{2} & \frac{v_1+v_2}{2} & & & \\ \frac{v_1+v_2}{2} & \frac{v_1+2v_2+v_3}{4} & \frac{v_2+v_3}{2} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{v_{N-1}+v_N}{2} & \frac{v_{N-1}+2v_N+v_{N+1}}{4} & \frac{v_N+v_{N+1}}{2} \\ & & & \frac{v_N+v_{N+1}}{2} & \frac{v_N+v_{N+1}}{2} \end{pmatrix}.$$

Using the operators diag and convol we define the matrices

$$W = \operatorname{diag}\begin{pmatrix} \zeta(\eta_0) \\ \vdots \\ \zeta(\eta_N) \end{pmatrix}; \; \Sigma = \operatorname{diag}\begin{pmatrix} \sigma(\xi_0) \\ \vdots \\ \sigma(\xi_N) \end{pmatrix}; \tag{37}$$

$$\Phi_{11}^s = \operatorname{diag}\begin{pmatrix} \rho_{11}^s(\eta_0) \\ \vdots \\ \rho_{11}^s(\eta_N) \end{pmatrix}; \; \Lambda_{11}^s = \operatorname{convol}\begin{pmatrix} \lambda_{11}^s(\xi_0) \\ \vdots \\ \lambda_{11}^s(\xi_N) \end{pmatrix}, \quad s = 1, \dots, 4; \tag{38}$$

$$\Phi_{22}^s = \operatorname{convol}\begin{pmatrix} \rho_{22}^s(\eta_0) \\ \vdots \\ \rho_{22}^s(\eta_N) \end{pmatrix}; \; \Lambda_{22}^s = \operatorname{diag}\begin{pmatrix} \lambda_{22}^s(\xi_0) \\ \vdots \\ \lambda_{22}^s(\xi_N) \end{pmatrix}, \quad s = 1, \dots, 4; \tag{39}$$

$$\Phi_{12}^s = \mathbf{1}\begin{pmatrix} \rho_{12}^s(\eta_0) \\ \vdots \\ \rho_{12}^s(\eta_N) \end{pmatrix}^T; \; \Lambda_{12}^s = \begin{pmatrix} \lambda_{12}^s(\xi_0) \\ \vdots \\ \lambda_{12}^s(\xi_N) \end{pmatrix}\mathbf{1}^T, \quad s = 1, \dots, 3; \tag{40}$$

$$\Phi_{21}^s = (\Phi_{12}^s)^T; \quad \Lambda_{21}^s = (\Lambda_{12}^s)^T, \quad s = 1, \dots, 3, \tag{41}$$

where $\mathbf{1} \in \mathbb{R}^{N+1}$ denotes a column vector of ones. We now define, for all $p, q = 1, 2$, the following large matrices:

$$\widehat{J} = W \otimes \Sigma, \qquad \widehat{E}_{pq} = \sum_{s=1}^{4-\delta_{pq}} \Phi_{pq}^s \otimes \Lambda_{pq}^s. \tag{42}$$

With these notations and denoting by $\circ$ the (elementwise) Hadamard product, (34) and (36) can be expressed by the following matrix decompositions

$$\widehat{M} = \widehat{J} \circ (M_0 \otimes M_0); \tag{43}$$

$$\begin{aligned} \widehat{A} &= \widehat{E}_{11} \circ (M_0 \otimes A) + \widehat{E}_{12} \circ (C \otimes C^T) + \widehat{E}_{21} \circ (C^T \otimes C) + \widehat{E}_{22} \circ (A \otimes M_0) \\ &= \widehat{E}_{11} \circ (M_0 \otimes A) + \widehat{E}_{12} \circ (C \otimes C^T) + [\widehat{E}_{12} \circ (C \otimes C^T)]^T + \widehat{E}_{22} \circ (A \otimes M_0), \end{aligned} \tag{44}$$

where $A, C, M_0 \in \mathbb{R}^{(N+1)\times(N+1)}$ are the stiffness, convection, and lumped mass matrices in 1D, defined by

$$a_{ij} = \int_0^1 \psi_i' \psi_j' \mathrm{d}\xi, \qquad c_{ij} = \int_0^1 \psi_i' \psi_j \mathrm{d}\xi, \qquad (m_0)_{ij} = \int_0^1 I_h(\psi_i \psi_j) \mathrm{d}\xi = \delta_{ij} \int_0^1 \psi_i \mathrm{d}\xi, \tag{45}$$

for all $i, j = 1, \dots, N+1$. It is evident that M_0 is a diagonal matrix. Since C is tridiagonal as a consequence of the ψ_i's being compactly supported, only the tridiagonal part of the four matrices defined in (40)-(41) is relevant, thanks to the Hadamard products in (44). Moreover, we observe from (43)-(44) that the general *lumped* discrete operators in (31) are obtained by weighting, through the matrices $\widehat{E}_{pq}$, the discrete operator

$$\widetilde{A} := M_0 \otimes A + C \otimes C^T + C^T \otimes C + A \otimes M_0, \tag{46}$$

associated to the model equation on the unit square domain

$$-\nabla \cdot \left(\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \nabla u \right) = f. \tag{47}$$

2.6. Examples of separable domains

In this section, for aim of clarity, we present several examples of separable domains Ω^S, i.e. domains that fulfil our hypotheses in (15) and (18). First of all, (15) and (18) imply that Ω^S can be characterised in parametric way as the set enclosed by the four curves

$$\begin{cases} \gamma_L : (x(t), y(t)) = (A(0)B(t), C(0)D(t)), & t \in [0,1]; \\ \gamma_R : (x(t), y(t)) = (A(1)B(t), C(1)D(t)), & t \in [0,1]; \\ \gamma_B : (x(t), y(t)) = (A(t)B(0), C(t)D(0)), & t \in [0,1]; \\ \gamma_T : (x(t), y(t)) = (A(t)B(1), C(t)D(1)), & t \in [0,1], \end{cases} \tag{48}$$

which are the images of the left, right, bottom and top edges of the reference square $\Omega = [0,1]^2$, respectively. As an illustrative example, we show in Fig. 1b the separable domain Ω^S obtained by the following choices in (48) and recalling from (18) that $C(\xi) = c|A(\xi)|^{\chi}$:

$$A(\xi) = \xi^3 + 1, \; B(\eta) = \sin\eta + 1, \; D(\eta) = e^{\eta} - \eta, \; c = -1.5, \; \chi = -1. \tag{49}$$

A remarkable subclass of separable domains is that of *x-normal domains* i.e. domains obtained when the transformation (15) is specialized to

$$\begin{cases} x(\xi,\eta) = A(\xi)B(\eta); \\ y(\xi,\eta) = D(\eta), \end{cases} \tag{50}$$

i.e. in the case when $C(\xi) = 1$, which automatically fulfils (18) for $\chi = 0$. It is possible to show that it is not a loss of generality, in terms of attainable domain shapes, to assume that $A(\xi)$ and $D(\eta)$ are non-decreasing linear functions in (50). Under these assumptions, the x-normal domain defined by the transformation (50), which we denote by Ω^N, can be characterised as

$$\Omega^N = \{(x,y) \in \mathbb{R}^2 \mid A(0)B(\eta) \le x \le A(1)B(\eta) \;\; \forall \eta \in [0,1] \quad \wedge \; D(0) \le y \le D(1)\}. \tag{51}$$

For instance, the x-normal domain defined by $A(\xi) = 2\xi - 1$, $B(\eta) = 1 + \frac{1}{2}\cos 2\pi\eta$ and $D(\eta) = 2\eta - 1$ is shown in Fig. 1c. This domain has the property that $A(0) = -A(1)$ and therefore the curves γ_L and γ_R are symmetric w.r.t. the y-axis, we then call it a *symmetric x-normal domain*. Another remarkable case of x-normal domains is when $A(0) = 0$ or $A(1) = 0$, which implies that either the curve γ_L or γ_R is a portion of the y-axis, we then call such domain a *one-sided x-normal domain*. For instance, the one-sided x-normal domain defined by $A(\xi) = \xi$, $B(\eta) = 1 + \frac{1}{2}\cos 2\pi\eta$ and $D(\eta) = \eta$ is shown in Fig. 1d.

2.7. Special surfaces

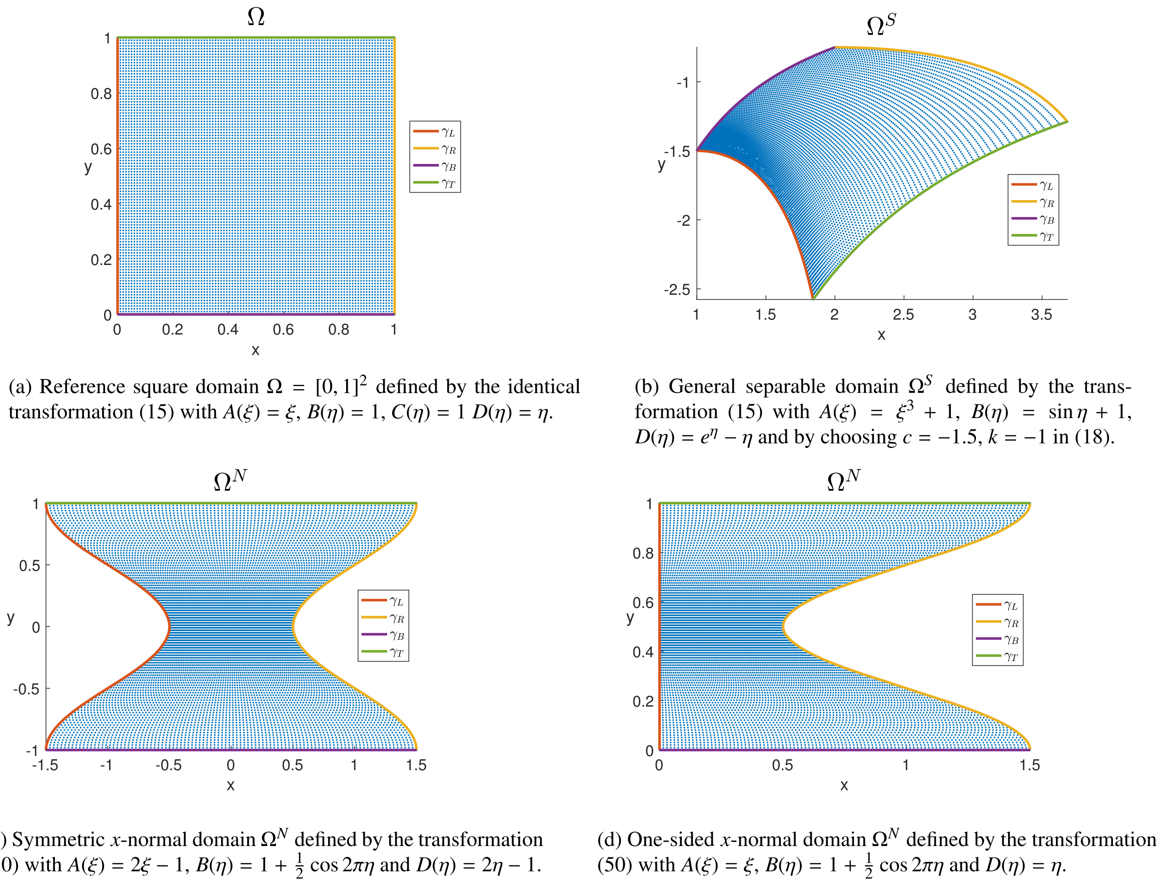
The proposed matrix-oriented FEM approach also applies to some surface PDEs, i.e. PDEs whose spatial domain is a smooth curved surface $\Gamma \subset \mathbb{R}^3$, see for instance [11]. Specifically, surface PDEs lend themselves to the matrix-oriented approach in the special case when the surface Γ is a *curvilinear cylinder*, i.e. the image of any x-normal domain Ω^N through the mapping $\mathcal{S} : \Omega^N \to \mathbb{R}^3$ defined by

$$\mathcal{S}(x,y) = \left(x, \frac{\sin 2\pi y}{2\pi}, \frac{\cos 2\pi y}{2\pi} \right), \qquad (x,y) \in \Omega^N. \tag{52}$$

We can thus define the curvilinear cylinder Γ as $\Gamma := \mathcal{S}(\Omega^N)$, see Fig. 2 for an illustration. We close this section by observing that, thanks to (10)-(12), the number $q \ge 2$ of nonzero terms in the Kronecker decompositions of the standard and lumped stiffness matrices $\widetilde{A}$ and $\widehat{A}$, respectively, depends on (i) the generality of the diffusion matrix D in the model problem (8) and (ii) the generality of the domain Ω^S. Case-by-case, the value of q is reported in Table 1. This information is relevant as it directly impacts the performances in time and memory of the MO-FEM approach.

3. The discrete problem and its solution

In this section we will show how to exploit the matrix decompositions (29) and (43)-(44) to solve the discrete problem (24). In the absence of lumping, thanks to (29), the vector trick (4) allows to rephrase the ODE system (24) in the usual vector form as the following matrix ODE system:



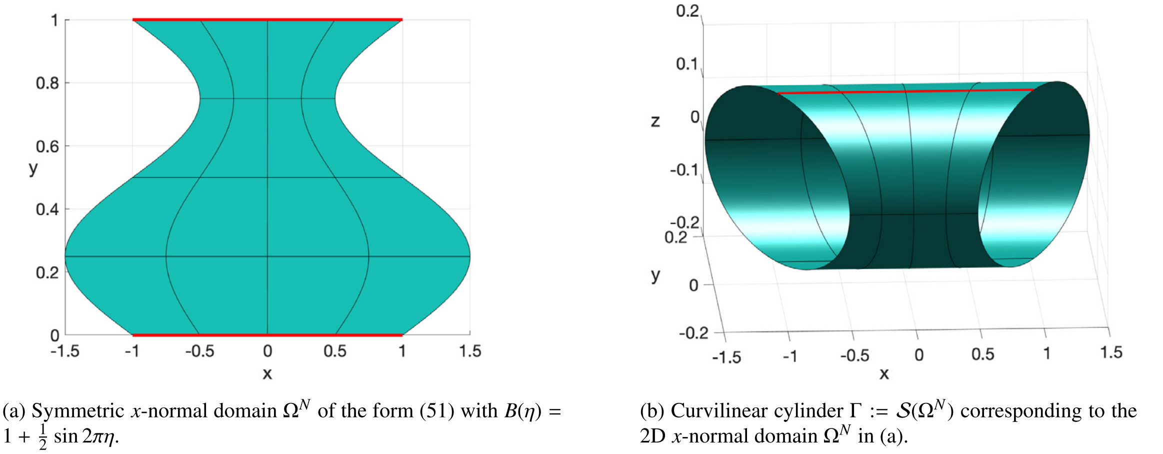
(a) Reference square domain $\Omega = [0,1]^2$ defined by the identical transformation (15) with $A(\xi) = \xi$, $B(\eta) = 1$, $C(\eta) = 1$ $D(\eta) = \eta$.

(b) General separable domain Ω^S defined by the transformation (15) with $A(\xi) = \xi^3 + 1$, $B(\eta) = \sin\eta + 1$, $D(\eta) = e^\eta - \eta$ and by choosing $c = -1.5$, $k = -1$ in (18).

(c) Symmetric x-normal domain Ω^N defined by the transformation (50) with $A(\xi) = 2\xi - 1$, $B(\eta) = 1 + \frac{1}{2}\cos 2\pi\eta$ and $D(\eta) = 2\eta - 1$.

(d) One-sided x-normal domain Ω^N defined by the transformation (50) with $A(\xi) = \xi$, $B(\eta) = 1 + \frac{1}{2}\cos 2\pi\eta$ and $D(\eta) = \eta$.

Fig. 1. Some examples of separable domains. The curves γ_L, γ_R, γ_B and γ_T are defined in (48). The blue dots inside the domains are the images through the map (15) of the corresponding blue dots inside the reference square $\Omega = [0,1]^2$. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)



(a) Symmetric x-normal domain Ω^N of the form (51) with $B(\eta) = 1 + \frac{1}{2}\sin 2\pi\eta$.

(b) Curvilinear cylinder $\Gamma := \mathcal{S}(\Omega^N)$ corresponding to the 2D x-normal domain Ω^N in (a).

Fig. 2. An x-normal domain Ω^N is transformed to the curvilinear cylinder $\Gamma := \mathcal{S}(\Omega^N)$. The transformation $\mathcal{S}$ in (52) joins the top- and bottom edges γ_T and γ_B of Ω_S, highlighted in red.

$$M^\xi \dot{U} M^{\eta^T} + \sum_{s=1}^{14} A_s^\xi U A_s^{\eta^T} = B, \tag{53}$$

where $\mathtt{vec}(U(t)) = u(t)$ and $\mathtt{vec}(B(t)) = \widetilde{\boldsymbol{b}}(t)$. In the case of lumped FEM, the matrix decompositions (43)-(44) together with the vector trick (4) allow to rephrase the discrete problem (24) as the following matrix ODE system:

Table 1
Number $q \in \mathbb{N}$ of nonzero Kronecker terms in the decompositions (29) and (44) of the standard and lumped stiffness matrices $\widetilde{A}$ and $\widehat{A}$, respectively.

Diffusion D Domain Ω^S	Diagonal ($d_{12} = d_{21} = 0$)	General
Rectangular	2	4
x-normal	5	8
Curvilinear cylinder	5	8
Separable	8	14

$$\begin{aligned}\Sigma M_0 \dot{U} M_0 W + \sum_{s=1}^{3}(\Lambda_{11}^{s} \circ A) U M_0 \Phi_{11}^{s} + \sum_{s=1}^{4}(\Lambda_{12}^{s} \circ C^T) U (C^T \circ {\Phi_{12}^{s}}^T) \\ + \sum_{s=1}^{4}({\Lambda_{12}^{s}}^T \circ C) U (C \circ \Phi_{12}^{s}) + \sum_{s=1}^{3} \Lambda_{22}^{s} M_0 U (A \circ {\Phi_{22}^{s}}^T) = B.\end{aligned} \tag{54}$$

The numerical solution of the matrix ODE systems (53) or (54) is not trivial. In order to show how we can address this problem, we first take one step back to a stationary counterpart of our model PDE in (8).

Remark 1 *(General boundary conditions).* Suppose that the model problem (8) is endowed with general non-homogeneous Dirichlet boundary conditions $v = \mathcal{B}(x, y)$ for $(x, y) \in \partial\Omega^S$. Following [39], this case is faced by considering an extension or *lifting* of $\mathcal{B}(x, y)$ that is defined for all $(x, y) \in \Omega^S$. Then, by performing the change of variable $\tilde{v}(x, y) := v(x, y) - \mathcal{B}(x, y)$ for $(x, y) \in \Omega^S$, the continuous problem (8) can be rephrased into an auxiliary problem for $\tilde{v}$, with the same differential operator, that fulfils homogeneous Dirichlet boundary conditions. Hence, our methodology still applies.
Suppose now that the model problem (8) has general non-homogeneous Neumann boundary conditions $\nabla v \cdot \boldsymbol{n}(x, y) = \mathcal{B}(x, y)$ for $(x, y) \in \partial\Omega^S$. Following [27] and references therein, it is possible to show that the corresponding parametrized problem (9), posed on the reference domain $\Omega = [0, 1]^2$, fulfils the boundary conditions $(E(\xi, \eta)\nabla u) \cdot \boldsymbol{n}(\xi, \eta) = J(\xi, \eta) \times \mathcal{B}\Big(x(\xi, \eta), y(\xi, \eta)\Big)$ for $(\xi, \eta) \in \partial\Omega$. Such boundary conditions can be inserted in the weak formulation (21) and take the form of an additional boundary integral on the right-hand-side of (21). Once again, the proposed matrix-oriented methodology illustrated so far still applies unchanged.

3.1. Multiterm Sylvester matrix equation and the matrix-oriented Preconditioned Conjugate Gradient (PCG)

In this section, we illustrate that the full PDE discretization by our MO-FEM approach leads to the solution of multiterm Sylvester matrix equations. Towards this aim, for the moment we neglect the time dependence and we consider the following linear elliptic PDE problem

$$\begin{cases} -\nabla \cdot (D\nabla v) + \gamma v = f(\boldsymbol{x}, t), & \boldsymbol{x} = (x, y) \in \Omega^S, \ t \in [0, T]; \\ v = 0 \quad \text{or} \quad \nabla v \cdot \boldsymbol{n}(\boldsymbol{x}) = 0, & \boldsymbol{x} \in \partial\Omega^S, \ t \in [0, T], \end{cases} \tag{55}$$

where Ω^S is a separable domain as described in Section 2.1, while $\gamma \geq 0$ for the case of Dirichlet boundary conditions and $\gamma > 0$ for the case of Neumann boundary conditions. By proceeding as in Section 2.2, the FEM and lumped FEM discretisations of (55) lead to the usual linear systems of dimension $(N+1)^2$ in the unknown vector μ:

$$\left(\widetilde{A} + \gamma \widetilde{M}\right) \boldsymbol{\mu} = \widetilde{\boldsymbol{b}}; \tag{56}$$

$$\left(\widehat{A} + \gamma \widehat{M}\right) \boldsymbol{\mu} = \widehat{\boldsymbol{b}}, \tag{57}$$

respectively, which we encompass with the unified notation

$$(A + \gamma M)\, \boldsymbol{\mu} = \boldsymbol{b}. \tag{58}$$

The linear system (58), which is sparse, symmetric and positive definite, is often solved with the following Preconditioned Conjugate Gradient Method (PCG):

$$\begin{cases} \boldsymbol{r}^{(0)} = \boldsymbol{b} - (A + \gamma M)\, \boldsymbol{\mu}^{(0)}; \\ \boldsymbol{z}^{(0)} = P^{-1} \boldsymbol{r}^{(0)}; \\ \boldsymbol{q}^{(0)} = \boldsymbol{z}^{(0)}; \end{cases} \tag{59}$$

$$\begin{cases} \alpha^{(p)} = \dfrac{\boldsymbol{q}^{(p)} \cdot \boldsymbol{r}^{(p)}}{((A+\gamma M)\boldsymbol{q}^{(p)}) \cdot \boldsymbol{r}^{(p)}}; \\ \boldsymbol{\mu}^{(p+1)} = \boldsymbol{\mu}^{(p)} + \alpha^{(p)}\boldsymbol{q}^{(p)}; \\ \boldsymbol{r}^{(p+1)} = \boldsymbol{r}^{(p)} - \alpha^{(p)}(A+\gamma M)\boldsymbol{q}^{(p)}; \\ \boldsymbol{z}^{(p+1)} = P^{-1}\boldsymbol{r}^{(p+1)}; \\ \beta^{(p)} = \dfrac{((A+\gamma M)\boldsymbol{q}^{(p)}) \cdot \boldsymbol{z}^{(p+1)}}{\left((A+\gamma M)\boldsymbol{q}^{(p)}\right) \cdot \boldsymbol{r}^{(p)}}; \\ \boldsymbol{q}^{(p+1)} = \boldsymbol{q}^{(p)} - \beta^{(p)}\boldsymbol{q}^{(p)}, \end{cases} \qquad p \in \mathbb{N} \cup \{0\}, \tag{60}$$

where $\boldsymbol{\mu}^{(0)} \in \mathbb{R}^{(N+1)^2}$ is an initial guess for the solution $\boldsymbol{\mu}$, and $P \in \mathbb{R}^{(N+1)\times(N+1)}$ is a suitable preconditioning matrix. In the remainder of this section we will carry out a matrix-oriented counterpart of the PCG method described above, which we will call MO-PCG. To this end, observe that the MO-FEM counterpart of the discrete problem (56) reads:

$$\sum_{s=1}^{14} A_s^{\xi} U A_s^{\eta\,T} + \gamma M^{\xi} U M^{\eta\,T} = B. \tag{61}$$

The MO-FEM counterpart of the lumped discrete problem (57) reads

$$\begin{aligned} &\sum_{s=1}^{3}(\Lambda_{11}^{s} \circ A)UM_0\Phi_{11}^{s} + \sum_{s=1}^{4}(\Lambda_{12}^{s} \circ C^T)U(C^T \circ {\Phi_{12}^{s}}^T) \\ +&\sum_{s=1}^{4}({\Lambda_{12}^{s}}^T \circ C)U(C \circ \Phi_{12}^{s}) + \sum_{s=1}^{3}\Lambda_{22}^{s}M_0U(A \circ {\Phi_{22}^{s}}^T) + \gamma\,\Sigma M_0 U M_0 W = B. \end{aligned} \tag{62}$$

Remark 2 *(Isotropic diffusion).* When $d_{11} = d_{22} =: \tilde{d}$ and $d_{12} = d_{21} = 0$, thanks to (13)-(14), the matrix equations (61)-(62) become simpler. Specifically, in (61), it holds that $A_s^{\xi} \neq 0$ only for $s \in \{1, 3, 4, 7, 8, 11, 12, 14\}$. In (62), instead, it holds that (i) $\Lambda_{11}^{s} \neq 0$ and $\Lambda_{22}^{s} \neq 0$ for $s \in \{1, 3\}$, while (ii) $\Lambda_{12}^{s} \neq 0$ and ${\Lambda_{12}^{s}}^T \neq 0$ for $s \in \{1, 4\}$. In summary, the discrete differential operator has $q = 8$ terms, see also Table 1.

Both (61) and (62) are multiterm Sylvester equations and, for ease of presentation, we will denote by $\mathcal{L}(U)$ and $\mathcal{R}$ their left- and right hand sides, so we can treat both cases using the compact notation

$$\mathcal{L}(U) = \mathcal{R}. \tag{63}$$

Because the multiterm linear matrix equation (63) is equivalent to the symmetric positive definite systems (58), we can formulate the following matrix-oriented counterpart of the PCG method, which we call MO-PCG, in terms of the above operators:

$$\begin{cases} R^{(0)} = \mathcal{R} - \mathcal{L}(U^{(0)}); \\ Z^{(0)} = \mathcal{P}^{-1}(R^{(0)}); \\ Q^{(0)} = Z^{(0)}; \end{cases} \tag{64}$$

$$\begin{cases} \alpha^{(p)} = \dfrac{Q^{(p)} : R^{(p)}}{\mathcal{L}(Q^{(p)}) : R^{(p)}}; \\ U^{(p+1)} = U^{(p)} + \alpha^{(p)}Q^{(p)}; \\ R^{(p+1)} = R^{(p)} - \alpha^{(p)}\mathcal{L}(Q^{(p)}); \\ Z^{(p+1)} = \mathcal{P}^{-1}(R^{(p+1)}); \\ \beta^{(p)} = \dfrac{\mathcal{L}(Q^{(p)}) : Z^{(p+1)}}{\mathcal{L}(Q^{(p)}) : R^{(p)}}; \\ Q^{(p+1)} = Q^{(p)} - \beta^{(p)}Q^{(p)}, \end{cases} \qquad p \in \mathbb{N} \cup \{0\}, \tag{65}$$

where $U^{(0)}$ denotes an initial guess for the solution U, while $A : B$ denotes, for any two matrices $A, B \in \mathbb{R}^{m\times n}$ of same dimension, the double tensor contraction defined by

$$A : B = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}B_{ij}, \tag{66}$$

while $\mathcal{P} : \mathbb{R}^{(N+1)\times(N+1)} \to \mathbb{R}^{(N+1)\times(N+1)}$ is a suitable preconditioning operator. The study of optimal initial guesses, optimal preconditioners, and optimal stopping criteria is outside the scope of this work. Here we will focus on the comparison, in terms of time and memory, between the PCG (59)-(60) and its MO counterpart (64)-(65), on equal initial guess, preconditioner and stopping criterion.

For the choice of the preconditioner, we address the practical case when $d_{11} = d_{22} =: \tilde{d}$ and $d_{12} = d_{21} = 0$. We choose a preconditioner such that (i) in vector form can be expressed as a single Kronecker product, so that its MO counterpart is easily invertible and (ii) aims at capturing the spectral properties of the stiffness matrix. After a heuristic experimentation, for the matrix PCG we choose

$$\widetilde{\mathcal{P}}(U) := \left(\delta M^\xi + \epsilon \sum_{s\in\mathcal{I}} A_s^\xi\right) U \left(\delta M^\eta + \epsilon \sum_{s\in\mathcal{I}} A_s^\eta\right)^T, \qquad \mathcal{I} := \{1, 3, 12, 14\}; \tag{67}$$

$$\widehat{\mathcal{P}}(U) := \left(\delta W M_0 + \epsilon \sum_{s=1,3} \left(\Lambda_{11}^s \circ A + \Lambda_{22}^s M_0\right)\right) U \left(\delta M_0 \Sigma + \epsilon \sum_{s=1,3} \left(\Phi_{11}^s M_0 + \Phi_{22}^s \circ A\right)\right)^T, \tag{68}$$

for the standard and the lumped case respectively, where $(\delta, \epsilon) = (0, 1)$ for the elliptic problem (61)-(62), while $(\delta, \epsilon) = (1, h_t)$ for the time dependent problem addressed in the next section. The above preconditioners contain only the second order terms of the discrete differential operators in (61) and (62). For this reason, in (67)-(68), the index s runs over less terms than in Remark 2. This comparison potentially extends to other linear iterative methods to solve the linear systems (56)-(57). We theoretically compare the complexity in time and memory between the vector and MO-PCG approaches in Appendix A. The numerical tests will be carried out in Section 4.

3.2. Time discretisation of the parabolic problem

Here we show how to address the MO time discretisation of the spatially discrete parabolic problems, that is the matrix ODEs (53) or (54). For ease of presentation, we will illustrate the procedure for the non-lumped version (53), since for the lumped counterpart (54) the procedure is analogous. We consider the IMplicit-EXplicit (IMEX) Euler method, which was proven to be particularly effective for the simulation of Turing patterns in reaction-diffusion systems [17] and also applied successfully in matrix form on square domains in [9]. Let $h_t > 0$ be the timestep. For all $n = 0, \ldots, N_T$, $N_T := \left\lceil \frac{T}{h_t} \right\rceil$, let $t_n := nh_t$ the time grid and let $U_n \in \mathbb{R}^{(N+1)\times(N+1)}$, $U_n \approx U(t_n)$ be the nodal matrix of the fully discrete solution at time t_n. By applying the IMEX Euler time discretisation to problem (53) we obtain

$$M^\xi U_{n+1} {M^\eta}^T + h_t \sum_{s=1}^{14} A_s^\xi U_{n+1} {A_s^\eta}^T = M^\xi U_n {M^\eta}^T + h_t B_n. \tag{69}$$

If $\mathcal{S}(U_{n+1})$ and $\mathcal{T}_n$ denote the left and right hand sides of (69), respectively, the IMEX-Euler-MOFEM full discretisation of the PDE problem (9) consists of the following sequence of multiterm Sylvester equations

$$\mathcal{S}(U_{n+1}) = \mathcal{T}_n, \qquad n = 0, \ldots, N_T - 1. \tag{70}$$

Hence, our idea is to solve (70) at each time step t_n via the MO-PCG algorithm as explained in Section 3.1. For completeness, we report the IMEX Euler method in the classical vector form for the approximation of the semi-discrete ODE system in (24):

$$(M + d_u h_t A)\, \boldsymbol{z}_{n+1} = M\boldsymbol{z}_n + h_t \boldsymbol{b}_n, \qquad n = 0, \ldots, N_T - 1, \tag{71}$$

where $\boldsymbol{z}_n = \texttt{vec}(U_n)$, such that at each timestep a large sparse linear system must be solved.

4. Numerical examples

In this section, we present some classical stationary and time dependent PDEs on separable domains, to highlight the performance of the matrix-oriented approach for FEM of different orders. In the next section, we will apply the MO-FEM-PCG to simulate the formation of Turing patterns in a morphochemical reaction-diffusion system (RDS) for metal growth in batteries, where a high level of spatial resolution is required.

4.1. Stationary PDEs: Poisson equation

In this section, we consider the following Poisson equation with Dirichlet boundary conditions:

$$\begin{cases} -\Delta v = f(\boldsymbol{x}), \qquad \boldsymbol{x} = (x, y) \in \Omega^S; \\ v = 0, \qquad \boldsymbol{x} \in \partial\Omega^S, \end{cases} \tag{72}$$

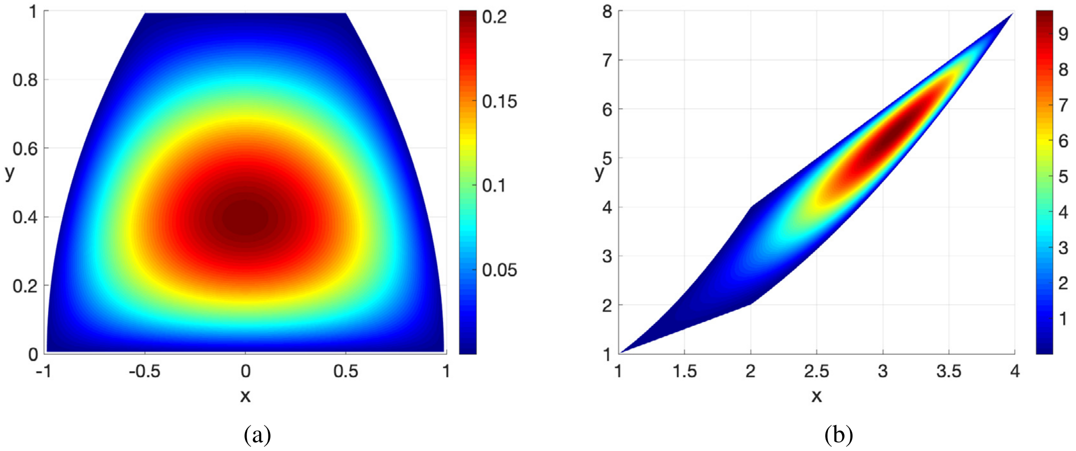


Fig. 3. Numerical solution of the Poisson equation (72) obtained by MO-PCG FEM on a Cartesian mesh on the unit square with $N = 384$. (a) Test 1 on x-normal domain Ω^N (73). (b) Test2 on separable domain Ω^S (76).

for two choices of the spatial domain, x-normal in Test 1 and separable in Test 2. Our goals are (i) to compare the performances of the *vector* PCG algorithm (59)-(60) with the MO-PCG (64)-(65) in terms of execution time and memory storage and (ii) to show optimal convergence in space for both methods. All the experiments in this section are carried out in MATLAB R2019a on a MacBook Pro 2,3 GHz 8-Core Intel Core i9.

Test 1: x-normal domain

Let be $\Omega^S = \Omega^N$ the symmetric x-normal domain defined by (50) with:

$$A(\xi) = \xi - \frac{1}{2}; \quad B(\eta) = 2 - \eta^2; \quad D(\eta) = \eta, \tag{73}$$

that corresponds to the cap-shaped domain

$$\Omega^S := \left\{ (x, y) \in \mathbb{R}^2 \;\middle|\; 0 \le y \le 1,\ |x| \le 1 - \frac{y^2}{2} \right\}. \tag{74}$$

$f(\boldsymbol{x})$ is such that the exact solution is the lowest-order non-zero polynomial that fulfils the boundary conditions:

$$u(x, y) = \left(-\frac{y^2}{2} + x + 1 \right) \left(\frac{y^2}{2} + x - 1 \right) y(y - 1). \tag{75}$$

We discretise the reference unit square Ω with a Cartesian grid of N intervals along each dimension, with $N = 48, 96, 192, 384$ and we apply both the vector (59)-(60) and MO (64)-(65) FEM PCG algorithms to solve the toy model (72). We recall that, since here we are considering Dirichlet boundary conditions, by the iterative scheme, in the first case we solve a huge linear system of $(N-1)^2$ equations and in the second one a multiterm Sylvester matrix equation of dimension $N-1$ (for the internal nodes) and q terms, where the possible values of q are shown in Table 1. In both cases, the PCG iterations are stopped when the residual in Frobenius norm reaches a tolerance `tol` =1e-14, close to machine precision. Moreover, we choose the preconditioners defined in (67)-(68). The numerical solution obtained on the finest mesh ($N = 384$) is shown in Fig. 3a. The obtained results are reported in Table 2. For each N, we compute: i) the relative $L^2(\Omega^N)$ error of the numerical solution on the original domain Ω^N; ii) the execution time and memory occupation required by both variants of the PCG; iii) the experimental order of convergence. We observe that, for all orders $k = 1, 2, 3, 4$, the MO-FEM method shows at least optimal convergence of order $k + 1$, thereby assessing the correctness of the matrix decompositions (29) and (43)-(44). For $k = 2$, we observe superconvergence of order $k + 2 = 4$, probably due to the symmetries of the problem.

Compared to the classical vector PCG, MO-PCG shows a strong advantage both in execution time (in seconds) and in memory storage (reported in MegaBytes) for all N and k, as also evident from the experimental ratios Ratio_m and Ratio_t, for memory and time, respectively. It is worth noting that the *lumped case* $k = 1$ is the cheapest in terms of execution time for all N. Moreover, we note that Ratio_t increases with N and it is almost constant wrt k, while, as predicted in the Appendix A, Ratio_m increases with k and slightly changes wrt to N. As consequence, the advantages in time are relevant for $N > 100$ and increase with the order k, for example for $N = 384, k = 4$, MO-PCG requires about 28 seconds and the vector PCG about 13 minutes. (For the lumped case we have 5 seconds versus 5 minutes.)

Test 2: separable domain

We consider the separable domain defined by (50) with:

$$A(\xi) = \xi + 1; \quad B(\eta) = \eta + 1; \quad C(\xi) = (\xi + 1)^2; \quad D(\eta) = \eta + 1. \tag{76}$$

Table 2
Test 1: Poisson problem (72) **on x-normal domain**, solved through FEM by the PCG method in vector and matrix form on a sequence of meshes with increasing number N. For each polynomial order $k = 1L, 1, 2, 3, 4$, where the L stands for "lumped", the experimental convergence rate in $L^2(\Omega^N)$ norm is approximately $k + 1$ (except for $k = 2$ that superconverges of order 4), as expected. The performances of MO-FEM-PCG show significant gains in memory storage and execution times (sec) wrt the analogous vector formulation, as evident from the ratios Ratio$_m$ and Ratio$_t$, respectively. In particular, the advantages in time are relevant for $N > 100$ and increase with the order k, for example for $N = 384, k = 4$, MO-PCG requires about 28 seconds and the vecPCG about 13 minutes. It is worth noting that the lumped case $k = 1$ is the cheapest in terms of execution time for all N.

N	k	PCG	Time (sec)			Memory (MB)			Rel error	Conv
		it	Vec	Mat	Ratio$_t$	Vec	Mat	Ratio$_m$	Mat	rate
48	1L	246	0.4947	0.023969	20.639	0.94728	0.12479	7.5912	3.7973e-04	1.9991
	1	381	0.75526	0.082341	9.1724	1.3211	0.12772	10.344	2.0767e-04	1.9998
	2	438	0.98111	0.10091	9.7227	2.1872	0.13846	15.797	3.7761e-07	3.9907
	3	443	1.1095	0.11787	9.4124	3.2896	0.14969	21.976	3.1155e-07	3.9909
	4	494	1.3906	0.11196	12.42	4.6284	0.16066	28.808	2.5574e-08	4.9903
96	1L	464	4.3869	0.1793	24.467	3.8494	0.4243	9.0723	9.4946e-05	1.9998
	1	712	6.8265	0.58746	11.62	5.4512	0.43021	12.671	5.1921e-05	1.9999
	2	839	9.3508	0.81421	11.484	9.1489	0.45267	20.211	2.3683e-08	3.9950
	3	827	8.4087	0.86241	9.7502	13.88	0.4761	29.153	1.9497e-08	3.9981
	4	919	11.31	1.1227	10.074	19.644	0.49904	39.363	8.0045e-10	4.9977
192	1L	834	37.929	1.1226	33.787	15.594	1.5508	10.056	2.3737e-05	2.0000
	1	1283	55.577	2.9653	18.742	22.149	1.5625	14.175	1.2980e-05	2.0000
	2	1517	70.876	4.219	16.799	37.416	1.6084	23.262	1.4829e-09	3.9974
	3	1464	72.817	4.5802	15.898	56.997	1.6563	34.413	1.2189e-09	3.9996
	4	1616	103.21	5.7497	17.95	80.893	1.7031	47.497	2.5023e-11	4.9995
384	1L	1440	298.87	5.2272	57.176	62.639	5.9131	10.593	5.9344e-06	2.0000
	1	2224	465.93	16.18	28.796	89.294	5.9366	15.041	3.2451e-06	2.0000
	2	2628	610.87	21.669	28.191	151.33	6.0293	25.098	9.3294e-11	3.9905
	3	2480	609.05	23.156	26.303	230.98	6.126	37.705	7.6202e-11	3.9996
	4	2754	806.81	28.565	28.245	328.27	6.2207	52.77	1.771e-12	3.8206

Table 3
Test 2: Poisson equation (72) **on separable domain** Ω^S (76). The results are similar to the previous experiment (see Table 2), the expected convergence order is obtained, the MO-FEM-PCG occupies significantly less memory and exhibits extremely lower computational times wrt to its vector counterpart, as become also more evident for $N > 100$ and all FEM orders. The ratios reported indicate the classical vector approach is more than ten times expensive than the new proposed MO approach, both in memory and execution time.

N	k	PCG	Time (sec)			Memory (MB)			Rel error	Conv
		it	Vec	Mat	Ratio$_t$	Vec	Mat	Ratio$_m$	Mat	rate
48	1L	297	0.61462	0.050788	12.102	0.9487	0.12971	7.3137	1.6370e-03	1.9736
	1	471	0.97949	0.099224	9.8715	1.2214	0.12978	9.4118	1.5785e-03	2.0001
	2	693	1.7546	0.14222	12.337	2.056	0.14186	14.493	1.1940e-05	3.8887
	3	845	1.932	0.18696	10.334	3.2377	0.1559	20.768	3.8359e-06	4.0853
	4	931	2.7134	0.22742	11.931	4.6284	0.1689	27.403	2.1537e-07	5.0525
96	1L	541	4.968	0.27623	17.985	3.8637	0.4344	8.8944	4.1145e-04	1.9923
	1	860	8.5345	0.69158	12.341	5.041	0.43446	11.603	3.9464e-04	1.9999
	2	1288	13.851	1.2557	11.031	8.6053	0.45973	18.718	7.7465e-07	3.9461
	3	1616	15.984	1.9447	8.2192	13.659	0.48891	27.937	2.3495e-07	4.0291
	4	1754	20.967	2.406	8.7142	19.644	0.51607	38.064	6.6394e-09	5.0196
192	1L	987	43.476	1.7436	24.935	15.6	1.5711	9.9293	1.0302e-04	1.9978
	1	1562	66.749	3.8078	17.53	20.485	1.5712	13.038	9.8664e-05	2.0000
	2	2351	110.36	6.9562	15.865	35.204	1.6228	21.693	4.9273e-08	3.9747
	3	2936	148.67	9.7303	15.279	56.087	1.6823	33.34	1.4602e-08	4.0081
	4	3185	210.28	12.28	17.124	80.893	1.7377	46.551	2.0669e-10	5.0055
384	1L	1780	399.41	9.0593	44.088	62.698	5.9539	10.53	2.5765e-05	1.9994
	1	2800	599.5	21.207	28.268	82.591	5.954	13.872	2.4666e-05	2
	2	4249	1034.7	36.476	28.366	142.4	6.0584	23.505	3.1048e-09	3.9882
	3	5168	1308.6	54.13	24.174	227.29	6.1784	36.788	9.1204e-10	4.001
	4	5689	1729.7	65.345	26.47	328.27	6.2905	52.185	6.7072e-12	4.9456

The exact solution is given by

$$u(x, y) = (y - x)(y - 2x)(y - x^2)(2y - x^2). \tag{77}$$

The numerical solution obtained on the finest mesh ($N = 384$) is shown in Fig. 3b. Table 3 shows the convergence and the performance benchmark in time and memory. The expected convergence order is obtained. In this case the Kronecker decompositions (29) and (44) have $q = 8$ nonzero terms, as opposed to the x-normal case where the nonzero terms are $q = 5$, see Table 1. The strong gain in memory storage and computational execution time of MO-PCG versus the vector

PCG method are still valid as evident from the results in Table 3. The MO-FEM-PCG occupies significantly less memory and exhibits extremely lower computational times wrt to its vector counterpart. In particular, for $N < 100$ both approaches require few seconds and at most less than 20 MB, while for $N > 100$ and all FEM orders the advantage of the MO approach increases. For example, for $N = 384, k = 1$, we compute the solution by MO-PCG in about 21 secs versus 10 minutes, for $k = 4$ in about 1 minute versus about 29 minutes. It is worth noting that the lumped case $k = 1L$ is also cheaper than the unlumped case (same accuracy, but about half time). The ratios reported indicate that, in general, the classical vector approach is much more than ten times as expensive as the new proposed MO approach, both in memory and execution time.

4.2. Time-dependent PDEs: semilinear heat equation on x-normal domain

We consider the following heat equation with zero Dirichlet boundary conditions on the cap-shaped domain Ω^N defined in (74) for the previous Test 1:

$$\begin{cases} u_t - d_u \Delta u = f(x,y,t), \qquad (x,y) \in \Omega^S, \quad t \in [0,1]; \\ u(x,y,t) = 0, \qquad (x,y) \in \partial\Omega^S, \quad t \in [0,1]; \\ u(x,y,0) = y(y-1)\left(-\frac{y^2}{2} + x + 1\right)\left(\frac{y^2}{2} + x - 1\right), \qquad (x,y) \in \Omega^S, \end{cases} \tag{78}$$

where $d_u = 0.1$ and $f(x,y,t)$ is chosen in such a way that the exact solution is $u(x,y,t) = u(x,y,0)\exp(t)$, we omit the cumbersome expression of such $f(x,y,t)$. We consider both $\mathbb{P}_k$, $k = 1,2,3,4$, and lumped $\mathbb{P}_1$ elements. In this case, the matrix oriented approach will require the solution of a sequence of multiterm Sylvester equations like (69). Stemming from the results obtained in the previous section, where only one discrete problem was solved, we prefer to abandon the comparisons with the *vector* PCG method and solve the sequence of linear systems in the classical vector formulation of the IMEX Euler method in (71) via the MATLAB direct solver `mldivide`, with only one preliminary LU-decomposition and reordering of unknowns by `symamd`. Therefore, we will compare these results with the MO-PCG approach solving (65) with the operator $\mathcal{S}$ in (70) with preconditioner (67) (and (68) for the lumped case). Moreover, here we focus only on the computational execution times. We use the following stopping criterion for MO-PCG: at each timestep, we stop the iterations of (65) when the truncated solution $U^{(s)}$ fulfils $\|R^{(s)}\| \le 0.01\|R^{(0)}\|$, where $R^{(s)}$ is the residual defined by $R^{(s)} := \mathcal{S}(U^{(s)}) - \mathcal{T}(U^{(s)})$, with the corresponding operators as defined in (70). This and all the following time-dependent experiments are carried out in MATLAB R2019a on a HP Z230 Tower Workstation with Intel Core i7-440 CPU and 16 GB RAM.

To highlight the computational advantages of the MO-PCG approach we consider $N = 480, 960, 1920$ and fixed timestep $\tau = 1\text{e-}2$, as representative test of typical user-case scenarios where high spatial resolution is required. The obtained results indicate a significant advantage of MO-PCG and are shown in Table 4 for all k and for all N. In all cases, the MO-PCG approach solving a multiterm Sylvester matrix equation at each timestep is quicker than the vector (direct) approach, especially for $k = 1$ and $k = 2$. The gap increases with N, as shown by comparing the time ratios. At each timestep, MO-PCG converges with just one iteration, except with lumped $\mathbb{P}_1$ elements, where up to two PCG iterations per timestep are required.

5. Applications to pattern formation in battery modelling

We now consider the following reaction-diffusion model in two variables $\eta : \Omega \times [0,T] \to \mathbb{R}$ and $\theta : \Omega \times [0,T] \to [0,1]$, endowed with zero Neumann boundary conditions, on an arbitrary compact domain $\Omega \subset \mathbb{R}^2$:

$$\begin{cases} \eta_t - \Delta\eta = \rho f(\eta,\theta), \qquad (x,y,t) \in \Omega \times [0,T]; \\ \theta_t - d_\theta \Delta\theta = \rho g(\eta,\theta), \qquad (x,y,t) \in \Omega \times [0,T]; \\ \nabla\eta \cdot \boldsymbol{n} = \nabla\theta \cdot \boldsymbol{n} = 0, \qquad (x,y,t) \in \partial\Omega \times [0,T]; \\ \eta(x,y,0) = \eta_0(x,y), \quad \theta(x,y,0) = \theta_0(x,y), \qquad (x,y) \in \Omega, \end{cases} \tag{79}$$

where $d_\theta > 0$ is the diffusion coefficient, $\rho > 0$ is a space-time rescaling factor and the kinetics are

$$f(\eta,\theta) := A_1(1-\theta)\eta - A_2\eta^3 - B(\theta - \alpha); \tag{80}$$

$$g(\eta,\theta) := C(1 + k_2\eta)(1-\theta)[1 - \gamma(1-\theta)] - D\theta(1 + \gamma\theta)(1 + k_3\eta), \tag{81}$$

with $\alpha, \gamma, A_1, A_2, B, C, D, k_2, k_3$ positive parameters.

The PDE system (79)-(81) is known as DIB model and has been introduced for the first time in [5] to describe electrodeposition processes. Under suitable choices of the parameters and of the domain Ω, this model was shown to possess a variety of spatially structured solutions, known as Turing patterns, see for example [28,29]. An interesting application in

Table 4

Semilinear heat equation (78) **on the cap-shaped domain** (74). Computational execution time (sec) for $\mathbb{P}_k$ and lumped FEMs in space and IMEX Euler method in time with timestep $\tau = 0.01$. In all cases, the MO-PCG approach solving a multiterm Sylvester matrix equation at each timestep is quicker than the vector (direct) approach, especially for $k = 1$. The gap increases with N, as shown by comparing the time ratios Ratio$_t$.

N	k	Time (sec) Vec direct	Time (sec) MO-PCG	Ratio$_t$ Vec/Mat	PCG Iter
480	1L	16.79	4.089	4.106	2
	1	10.10	4.313	2.342	1
	2	10.21	4.571	2.233	1
	3	13.42	4.886	2.746	1
	4	17.30	4.094	4.228	1
960	1L	129.1	18.58	6.949	2
	1	131.7	16.41	8.023	1
	2	54.98	19.10	2.879	1
	3	58.79	20.25	2.902	1
	4	89.18	21.62	4.126	1
1920	1L	1269	77.50	16.37	2
	1	4337	74.35	58.33	1
	2	497.9	87.01	5.722	1
	3	473.0	81.17	5.828	1
	4	3311	87.27	37.93	1

battery modelling is reported in [30,31]. Turing patterns are obtained as stationary solutions of (79)-(81) and then their numerical approximation requires highly spatial accuracy for longtime integration, this motivates the development of efficient solvers. In this direction, a first work based on matrix oriented formulation of (79)-(81) is [9] where finite differences and several time solvers have been proposed on square domains. In [9], the Sylvester matrix equations obtained at each time step have been approximated by the *reduced approach*. For example, the IMEX Euler yielded the *rEuler* method, that revealed much more efficient than its classical vector approach (like (71)). On the other hand, domain geometry was also proven to play an important role in pattern selection, as shown also in [30,31], for this reason efficient solvers that can be applied on domains as general as possible are needed. Towards this aim, here we propose the MO-FEM spatial approximation and the MO-PCG approach presented in Section 4 with preconditioners (68) and (68) to deal with some separable domains. We will present three kinds of simulations: i) first on the (x-normal) cap shaped domain introduced in (74); ii) on a new separable domain; iii) on the jar-domain shown in Fig. 2a that corresponds to the curvilinear cylinder in Fig. 2b. In cases (i) and (iii) we will consider domains of increasing *effective domain size* given by $\mathcal{A} = \rho|\Omega|$, where $|\Omega|$ is the area of the domain in (79). In fact, as shown in [30,31] there exists a sufficiently large $\mathcal{A}^*$ such that for $\mathcal{A} > \mathcal{A}^*$ the *intrinsic* Turing pattern corresponding to the given model parameter set arises (see also [30] for more details), otherwise only a portion of it can be approximated, giving rise to doubts about its morphological classification.

To solve on domains of large size, here we exploit the meaning and the role of the parameter ρ in (79), as follows. By introducing new variables $\widetilde{t} := \rho t$ and $(\widetilde{x}, \widetilde{y}) := \sqrt{\rho}(x, y)$, the chain rule yields

$$\frac{\partial \eta}{\partial t} = \rho \frac{\partial \eta}{\partial \widetilde{t}}, \qquad \nabla_{(x,y)}\eta = \sqrt{\rho}\nabla_{(\widetilde{x},\widetilde{y})}\eta, \qquad \Delta_{(x,y)}\eta = \rho\Delta_{(\widetilde{x},\widetilde{y})}\eta; \tag{82}$$

$$\frac{\partial \theta}{\partial t} = \rho \frac{\partial \theta}{\partial \widetilde{t}}, \qquad \nabla_{(x,y)}\theta = \sqrt{\rho}\nabla_{(\widetilde{x},\widetilde{y})}\theta, \qquad \Delta_{(x,y)}\theta = \rho\Delta_{(\widetilde{x},\widetilde{y})}\theta. \tag{83}$$

Hence, if we define $\Omega_\rho := \sqrt{\rho}\Omega$ and $T_\rho := \rho T$, ρ acts as a rescaling parameter in space and time. In the following simulations we always solve the DIB model in the reference domains Ω (cap shaped and jar shaped) with final time T and timestep τ such that $T_\rho = 300$ and $\tau_\rho = \rho\tau = 5\text{e-}3$ which guarantees the stability of the IMEX Euler method. The corresponding numerical solutions will be plotted in the rescaled domain Ω_ρ for $T_\rho = 300$. In all the experiments, we fix the following model parameters:

$$\alpha = 0.5, \gamma = 0.2, A_1 = 10, k_2 = 2.5, k_3 = 1.5, d_\theta = 20, \tag{84}$$

and $D = C(1-\alpha)(1-\gamma+\gamma\alpha)/(\alpha(1+\gamma\alpha))$. The initial data in (79) are given by $\theta_0(x, y) = \theta_e + 10^{-4}\texttt{rand}(x, y)$ and $\eta_0(x, y) = \eta_e + 10^{-4}\texttt{rand}(x, y)$ as spatially random perturbations of the homogeneous equilibrium $(\eta_e, \theta_e) := (0, 0.5)$ (`rand` is the Matlab command for generating uniformly distributed random numbers).

5.1. Cap-shaped (x-normal) domain

In this example, we consider the cap-shaped domain (74) and we choose $A_2 = 30$, $B = 25$, $C = 7$, that, according to the segmentation results in [42], can yield mixed spots-worms Turing patterns at the steady state.

Table 5

DIB model (79)-(81) **on the cap-shaped domain for** $A_2 = 30, B = 25, C = 7$: numerical setting used for the simulations reported in Fig. 4 and computational execution times for the vector-direct approach based on LU decomposition and the MO-PCG method. The last column shows the amount of iterations required by MO-PCG for each of the two PDEs of the model. When the time ratio Ratio_t is close to 1, the methods take approximately the same cost; $\text{Ratio}_t > 1$ indicates that MO-PCG is quicker. MO-PCG is less expensive as N_x and N_y increase, i.e. in Case (c) when the *intrinsic* Turing pattern for this model parameter choice (see main text for details) is attained. The best performances are obtained by the MO-PCG for $k = 1$ (lumped is quicker) and for $k = 4$.

	$\mathcal{A} = \rho\|\Omega\|$	ρ	N_x	N_y	k	Time (sec) Vec direct	Time (sec) MO-PCG	Ratio_t (V/M)	PCG iter (η, θ)
(a)	$\mathcal{A}_a = \frac{2000}{3}$	400	100	50	1L	57.18	93.76	0.6099	(1, 6)
					1	62.76	94.60	0.6635	(2, 3)
					2	75.60	182.0	0.4159	(2, 4)
					3	226.6	193.4	1.172	(2, 3)
					4	270.2	207.7	1.301	(2, 3)
(c)	$50\mathcal{A}_a$	20000	400	200	1L	1590	879.8	1.807	(1, 4)
					1	1623	1249	1.299	(2, 3)
					2	1809	1934	0.9352	(1, 3)
					3	1973	2143	0.9209	(2, 3)
					4	6323	2266	2.791	(2, 3)

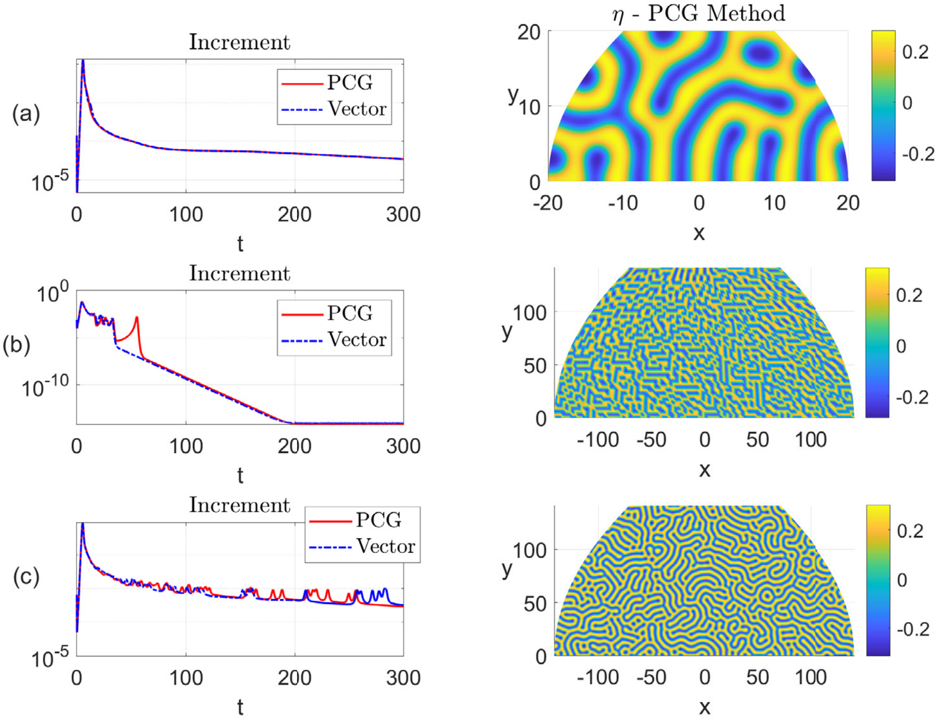


Fig. 4. DIB model (79)-(81) **on the cap-shaped domain for** $A_2 = 30, B = 25, C = 7$. Solution for η by the lumped $\mathbb{P}_1$ MO-FEM. Each row corresponds to the $(\mathcal{A}, \rho, N_x, N_y)$ combination in Table 5. For $\rho = 400$ in (a) only few structures arise in the pattern. For $\rho = 20000$: in (b) $N_x = 200, N_y = 100$ are not sufficient to identify the pattern structure, that instead is well-resolved for $N_x = 400$ and $N_y = 200$ in (c). Smaller values of N_x and N_y do not provide sufficient spatial approximation on the larger domain and the pattern appears grainy as in (b). The patterns obtained by the MO-PCG method and the vector (direct) method are very similar, they are the stationary solutions obtained at $T_\rho = 300$, as shown by the increment dynamics (left subplots). Computational execution times are reported in Table 5.

We solve both equations for η and θ of the RD-PDE system by $\mathbb{P}_k$ elements ($k = 1, 2, 3, 4$) and lumped $\mathbb{P}_1$ elements in space and by IMEX Euler in time. As in the previous section, here we compare the vector approach solving for each PDE equation the sequence of linear systems like (71) by the direct method (that we will call “vec direct”) with the MO-PCG approach (65), solving the multiterm ($q = 5$) Sylvester equations (69) for this choice of the domain.

We solve the DIB model (79) with different combinations of ρ, N_x and N_y as listed in Table 5, that is for domains of increasing effective area $\mathcal{A}$. In all the computations, we consider $N_x = 2N_y$ nodes to reflect the aspect ratio of the domain. In Fig. 4, for each simulation, we report the final patterns obtained for the variable η by the MO-PCG (those by the vec

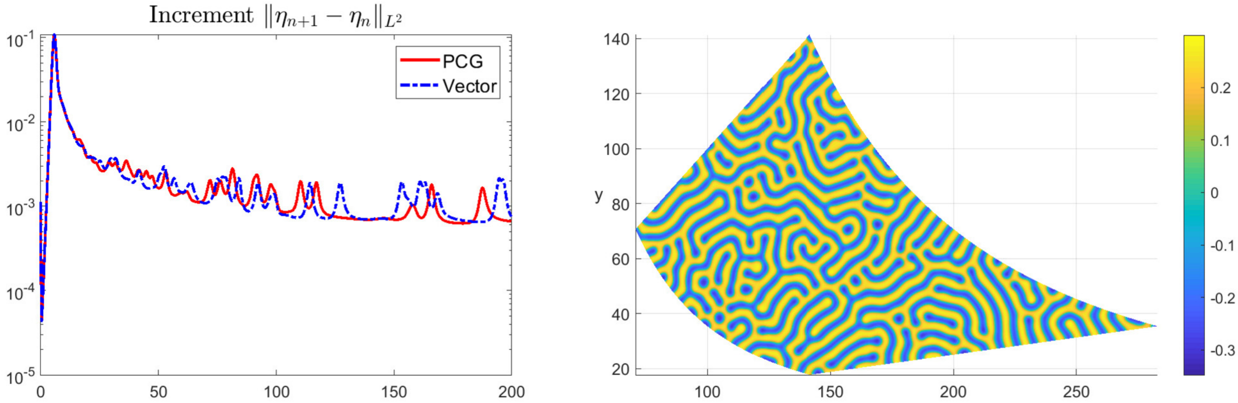


Fig. 5. DIB model on separable domain (85) for $A_2 = 30$, $B = 25$, $C = 7$, $\rho = 5000$. Right: Turing pattern for η obtained by the MO-PCG IMEX Euler method at final time $T = 200$ on the separable domain Ω^S approximated with $N_x = N_y = 300$. Left: time dynamics of the increment.

approach are very similar), together with the increments $\|\eta_{n+1} - \eta_n\|_F$, $n = 0, \ldots, N_T - 1$ along the time dynamics, with $\|\cdot\|_F$ Frobenius norm. If such increment decreases over time and tends to an almost small stationary value, we deduce that the numerical solution is converging to a steady state.

For increasing values of the *effective domain size* $\mathcal{A}$ the solution morphology changes and a pattern with more structures is attained, as shown for the unknown η in Fig. 4(a)–(c) corresponding to the values (a)–(c) in Table 5, respectively. In simulation (a) a good pattern is attained by both methods, but its morphology is not completely expressed. In terms of computational times, the vector approach is more convenient for $k \leq 2$, the matrix approach is cheaper for $k = 3, 4$ (see Table 5). To capture the true Turing morphology a larger domain and then a *sufficiently* fine mesh is required, otherwise *phantom patterns* could be obtained. This is exactly what happens for the simulation (b) in the second row of Fig. 4, where the same mesh of case (a) yields a stationary “pixelated” pattern. For this reason, for this same domain, in simulation (c) a finer mesh is used and both methods are finally able to attain a “complete” more structured pattern. Note that, for each equation of the RD system (79)-(80)-(81), in case (c) we solve a sequence of $N_t = 6 \cdot 10^4$ discrete problems, at each time step by the vector-direct we solve a linear system of $N_x \cdot N_y = 400 \cdot 200 = 8 \cdot 10^4$ equations, by the MO-PCG instead we solve a rectangular multiterm Sylvester equation of size $N_x \times N_y = 400 \times 200$. Moreover, this example shows that the MO-PCG algorithm (65)-(67) can successfully solve also rectangular Sylvester equations. In Table 5, we do not report case (b), since a “wrong” solution is attained. For case (c), when the more significant pattern is attained, the time ratios indicate that MO-PCG is quicker than the vector-direct approach with significant advantage for $k = 4$ and for lumped and no lumped $k = 1$ FEM. For $k = 2, 3$, very similar performances are obtained and we believe that a different preconditioner could improve the results of the MO-PCG.

5.2. Separable domain

In this example, we solve the DIB morphochemical model on a new separable Ω^S domain defined by setting

$$A(\xi) = \xi + 1; \quad B(\eta) = \eta + 1; \quad C(\xi) = (\xi + 1)^{-2}; \quad D(\eta) = \eta + 1 \tag{85}$$

in (50). In this case we consider the space approximation by $k = 1$ lumped FEM and we compare the MO-PCG with the vector-direct approach for $\rho = 5000$ and the other model parameters as in the previous subsection, here $\mathcal{A} = 11250$. We take $N_x = N_y = 300$ gridpoints and timestep $\tau = 0.01$. The numerical solution at the final time $T = 200$ (by MO-PCG) is reported in Fig. 5 (right) together with the dynamics of the increment (left). For the vector direct approach, we register a storage of about 461.98 MB, which is remarkably reduced to just 5.199 MB with the MO-PCG approach. The computational execution times in seconds are 2197.4 for the MO-PCG and 2636.6 for the direct method. Also in this case, on a more general domain, the matrix approach is significantly more competitive.

5.3. Jar-shaped domain and cylindrical surface

Thanks to the results in the previous cases, here we solve the model only with $\mathbb{P}_1$ elements, with and without lumping. To further explore the robustness of the matrix PCG approach w.r.t. *domain complexity and mesh distortion*, we consider the *jar-shaped domain* in Fig. 2a. We fix the parameters $A_2 = 1$, $B = 30$, $C = 3$ for the DIB model which are known to produce Turing patterns with holes (also called reversed spots) [42] and again we solve for increasing $\mathcal{A}$ with different combinations of ρ, N_x and N_y as listed in Table 6. In all the computations, we consider $N_x = 3N_y$ to reflect the aspect ratio of the domain. The timestep and the final time are as in the previous test on the cap-shaped domain. We show the $\mathbb{P}_1$ solution obtained

Table 6
DIB model (79)-(81) **on the jar-shaped domain (cylindrical surface)**, for $A_2 = 1, B = 30, C = 3$, other parameters as in Section 5.3: performance comparison between the vector (direct) method (71) with LU decomposition and the MO-PCG method (69). As evident by the ration Ratio_t of the obtained execution times (in seconds), the two approaches are almost similar on the smallest domain, but matrix-PCG becomes quicker than the vector approach as N_x and N_y increase, when the intrinsic Turing pattern is attained.

	$\mathcal{A}$	ρ	N_x	N_y	k	Time (sec) Vec direct	Time (sec) MO-PCG	Ratio_t	PCG Iter (η, θ)
(a)	$\mathcal{A}_a = 800$	400	150	50	1L	309.6	306.6	1.010	$(2, 16)$
					1	328.7	278.4	1.181	$(3, 7)$
(c)	$50\mathcal{A}_a$	20000	600	200	1L	7995	2316.4	3.451	$(1, 9)$
					1	8067	3028	2.664	$(2, 7)$

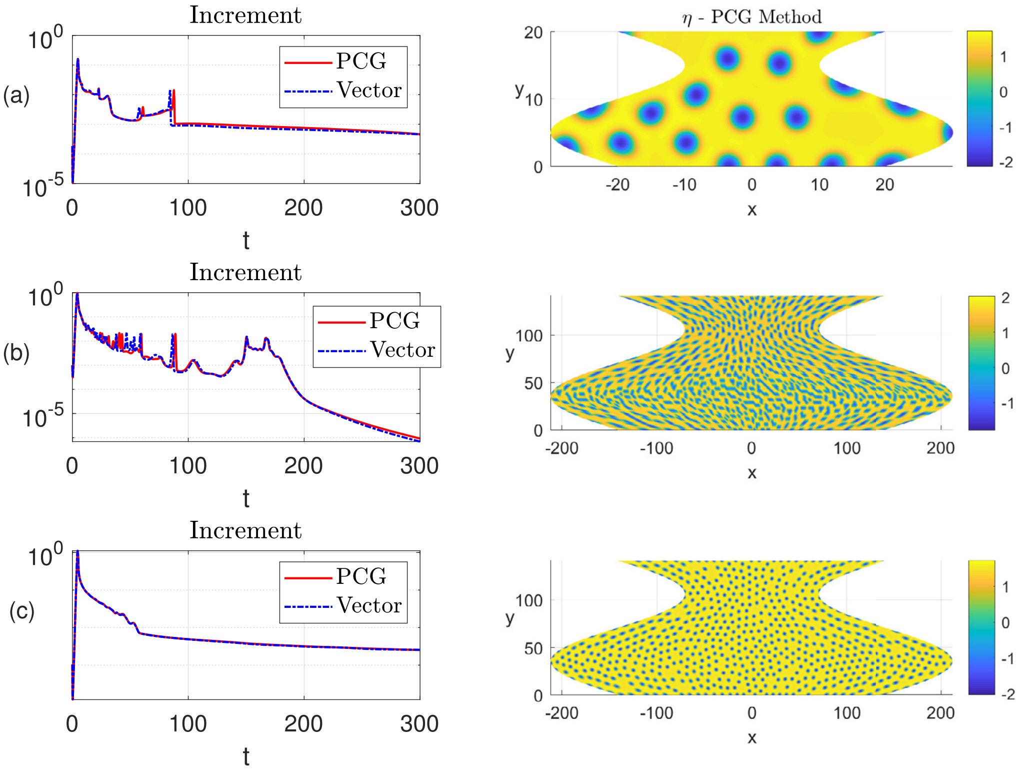


Fig. 6. DIB model (79)-(81) **on the jar-shaped domain** for $A_2 = 1, B = 30, C = 3$: lumped $\mathbb{P}_1$ solutions for η by the MO-PCG (right) and time increments (left). Values of (ρ, N_x, N_y) are given in Table 6. (a): on the smallest domain only few holes appear in the pattern. (b): for $\rho = 20000$ on the larger domain the mesh is too coarse and a *phantom pattern* arises. (c): the intrinsic Turing pattern is well-resolved for $N_x = 600$ and $N_y = 200$. The MO-PCG and vector solutions are very similar, but the MO approach converges in significant less time (see Table 6).

by the MO-PCG in the case with lumping in Fig. 6. The stationary patterns are the same by the other schemes in vector form. As we can see in Fig. 6, in case (a) on the smallest domain only few holes arise in the pattern. In case (b) on the larger domain, more structures arise in the pattern, but the mesh is too coarse and a *phantom pattern* arises. In case (c), the intrinsic Turing pattern is well-resolved for $N_x = 600$ and $N_y = 200$. The MO-PCG and the vector solutions are very similar, but the MO approach converges in significant less time (see Table 6). We can say that for the smaller area, all methods perform similarly, on the largest area (case (c), more structured pattern) the direct method requires a simulation of more than 2 hours, the MO-PCG instead in any case less than 1 hour, about 45 minutes by the lumped FEM.

We conclude this section by remarking that, since the jar-shaped domain in Fig. 2a can be transformed to the cylinder Γ shown in Fig. 2b, the solutions shown in Fig. 6 can be interpreted, after the coordinate transformation (52), as solutions to the *surface DIB model*, that is (79)-(81) where the Laplace operator Δ is replaced by the Laplace-Beltrami operator Δ_Γ on the cylinder Γ, see for example [30]. As an example, we report in Fig. 7 the solution of the above cases (a) and (c) wrapped on a curvilinear cylinder. The application of the model on a cylindrical surface can be of applicative interest as shown in [6], in which the authors consider the use of cylindrical Zn sponges as a means of limiting the shape change and dendrite formation issues in Zn-based rechargeable batteries.

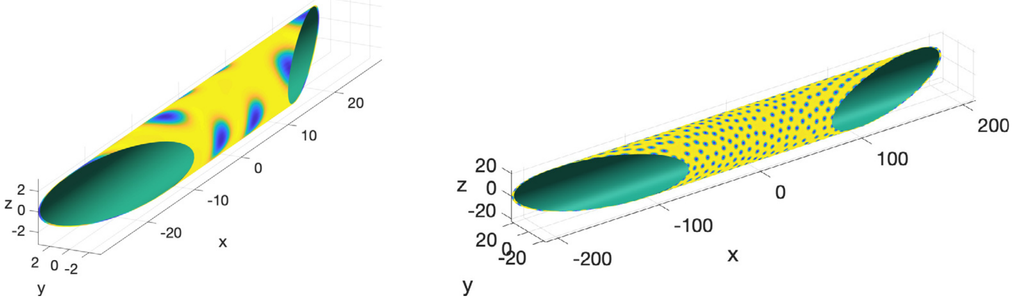


Fig. 7. Numerical solution of Fig. 6 for $\rho = 400$ (left) and $\rho = 20000$ (right) mapped onto the cylinder Γ in Fig. 2b. These results can be interpreted as the solution of the DIB *surface reaction-diffusion model* on Γ.

6. Conclusions

In this work we have provided a matrix-oriented formulation for Lagrangian finite elements of arbitrarily high order $k \in \mathbb{N}$ on a general class of separable 2D domains. The proposed approach applies to both elliptic and parabolic PDE problems. The discrete problems take the form of a multiterm Sylvester matrix equation (or a sequence of these equations in the time-dependent case) of much smaller dimension that is mathematically equivalent to the much larger standard linear system in Kronecker form. Since the special case of square domains yields a generalised two-term Sylvester equation, our work extends the findings in [9] (based on classical finite differences) to the case of high order FEM in space. Moreover, our proposed approach adopts a curvilinear structured mesh that eliminates geometric boundary error and, through a coordinate transformation, it applies also to special surface domains, namely cylinders with arbitrary curved boundaries.

We solve each *multiterm* Sylvester equation through a matrix-oriented PCG method with a matrix-oriented preconditioner that is quick to evaluate thanks to its single-term form. We show that such solver is always quicker than the classical PCG in vector form for the Poisson elliptic PDE on x-normal and separable domains, both in terms of memory storage and execution time. For time-dependent PDEs we compare our approach with the optimised direct solver `mldivide` of MATLAB. In terms of computational time, the matrix-oriented PCG method always improves on the direct solver that relies on the full storage of the Kronecker matrix, and the gap increases both with the number of gridpoints N and the polynomial order k of the method.

Special consideration deserves the application to reaction-diffusion systems, where the simulation of fine-grained solutions requires high spatial resolution, which translates into computationally intense simulations both in time and memory. Our experiments for the approximation of Turing patterns in the DIB morphochemical RD system for battery modelling provide encouraging results in this direction and support the matrix approach in terms of execution times and storage. The best performance gains were observed with $\mathbb{P}_1$ elements, both lumped and not lumped. We believe that further improvements can be attained through the development of more efficient preconditioners and more efficient solvers for multiterm Sylvester equations, such as a truncated PCG [43]. Moreover, the proposed approach is deemed to be suitable for the extension to three space dimensions by using structured cubic meshes [20] and/or evolving domains and surfaces [16]. These aspects will be addressed in future studies.

CRediT authorship contribution statement

Massimo Frittelli: Original draft preparation, Conceptualization, Methodology, Investigation, Software, Data Curation, Writing – Reviewing and Editing, Supervision. **Ivonne Sgura:** Original draft preparation, Conceptualization, Methodology, Investigation, Formal analysis, Writing – Reviewing and Editing, Supervision.

Acknowledgements

The work of MF was funded by Regione Puglia (Italy) through the research programme REFIN-Research for Innovation (protocol code 901D2CAA, project number UNISAL026) and by the research project "Metodi numerici innovativi per lo studio delle batterie" (INdAM-GNCS project CUP_E55F22000270001, U-UFMBAZ-2022-000858 26-05-2022). The work of IS is supported by the MIUR Project PRIN 2020, "Mathematics for Industry 4.0", Project No. 2020F3NCPX, by the ICSC – Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing, funded by European Union – NextGenerationEU, Project code CN00000013 and by the research project "Tecniche avanzate per problemi evolutivi: discretizzazione, algebra lineare numerica, ottimizzazione" (INdAM-GNCS project CUP_E55F22000270001, U-UFMBAZ-2022-000858 26-05-2022). MF and IS are members of the INdAM-GNCS activity group (Italian National Group of Scientific Computing).

Appendix A. Performance analysis in time and memory

In this Appendix we theoretically compare the performances in time and memory of the vector (60) and matrix PCG approaches (65). For ease of presentation, we will consider the decompositions (29) in the absence of lumping for general FEM order $k \in \mathbb{N}$ and Dirichlet boundary conditions ($\gamma = 0$ in (61)).
On a general separable domain Ω^S, we start by observing that the $2q$ matrices A_s^η and A_s^ξ appearing in (61) have a band structure with no more than $2k+1$ nonzero diagonals, amounting to $2q(2k+1)(N-1)$ entries. On the other hand, the huge sparse classical stiffness matrix $\widetilde{A}$ has a band structure with $(2k+1)^2$ nonzero diagonals, amounting to $(2k+1)^2(N-1)^2$ entries. Hence the ratio between the memory storage of the stiffness operator for the vector and MO approaches, respectively, is given by

$$\text{Ratio}_m := \frac{(2k+1)(2N-1)}{2q} = O(kN), \tag{A.1}$$

i.e. MO-PCG requires a memory storage for $\widetilde{A}$ linearly smaller, in terms of N and k, than vector PCG. The storage of the numerical solution $\boldsymbol{\mu}^{(s)}$ and the other auxiliary variables $\boldsymbol{r}^{(s)}, \boldsymbol{z}^{(s)}, \boldsymbol{q}^{(s)}$ in (59) is equivalent for both approaches. For the performances in time (number of operations), each matrix-vector product involving the $(2k+1)^2$-banded matrix $\widetilde{A}$ costs $(2k+1)^2(N-1)^2$ multiplications. In the MO counterpart, this operation translates to $2q$ matrix-matrix multiplications of size $(N-1) \times (N-1)$. In each of these multiplications, one matrix is full (e.g. $U^{(s)}$ in (65)), while the other is $(2k+1)$-banded (one of the A_s^η or A_s^ξ). Hence, the MO counterpart of a matrix-vector multiplication costs $2q(2k+1)(N-1)^2$. It follows that the ratio in computational cost, for each large matrix-vector product in (60), between matrix and MO-PCG is

$$\text{Ratio}_t := \frac{(2k+1)}{2q} = O(k), \tag{A.2}$$

which means that the vector approach is more costly for sufficient high order $k \geq q$. For instance, for an x-normal domain, we expect a computational advantage when $k \geq 5$, because $q = 5$, see Table 1. It is worth remarking that, for $k = 1$, the Kronecker factors of the lumped stiffness matrix in (44) are more sparse (some are diagonal or bidiagonal instead of tridiagonal) than their non-lumped counterparts A_s^ξ and A_s^η in (29). Hence, a suitable usage of lumping in the higher order case $k > 1$ [10] might improve the ratio (A.2), resulting in a lower threshold k above which the MO approach is more convenient in terms of computations, other than memory. This forms part of our current investigations. We close this section by remarking that the matrix-vector and matrix-matrix multiplications in MATLAB (which we use for our numerical experiments) are performed through BLAS routines, whose computational cost might differ from our theoretical analysis.

References

[1] P. Antolin, A. Buffa, F. Calabró, M. Martinelli, G. Sangalli, Efficient matrix computation for tensor-product isogeometric analysis: the use of sum factorization, Comput. Methods Appl. Mech. Eng. 285 (2015) 817–828, https://doi.org/10.1016/j.cma.2014.12.013.

[2] R. Barreira, C.M. Elliott, A. Madzvamuse, The surface finite element method for pattern formation on evolving biological surfaces, J. Math. Biol. 63 (2011) 1095–1119, https://doi.org/10.1007/s00285-011-0401-0.

[3] R.H. Bartels, G.W. Stewart, Solution of the matrix equation AX + XB = C, Commun. ACM 15 (1972) 820–826, https://doi.org/10.1145/361573.361582.

[4] D. Becherer, M. Schweizer, et al., Classical solutions to reaction–diffusion systems for hedging problems with interacting Itô and point processes, Ann. Appl. Probab. 15 (2005) 1111–1144, https://doi.org/10.1214/105051604000000846.

[5] B. Bozzini, D. Lacitignola, I. Sgura, Spatio-temporal organization in alloy electrodeposition: a morphochemical mathematical model and its experimental validation, J. Solid State Electrochem. 17 (2013) 467–479, https://doi.org/10.1007/s10008-012-1945-7.

[6] B. Bozzini, C. Mele, A. Veneziano, N. Sodini, G. Lanzafame, A. Taurino, L. Mancini, Morphological evolution of Zn-sponge electrodes monitored by in situ X-ray computed microtomography, ACS Appl. Energy Mater. 3 (2020) 4931–4940, https://doi.org/10.1021/acsaem.0c00489.

[7] S.C. Brenner, L.R. Scott, The Mathematical Theory of Finite Element Methods, vol. 3, Springer, 2008.

[8] M.A.J. Chaplain, M. Ganesh, I.G. Graham, Spatio-temporal pattern formation on spherical surfaces: numerical simulation and application to solid tumour growth, J. Math. Biol. 42 (2001) 387–423, https://doi.org/10.1007/s002850000067.

[9] M.C. D'Autilia, I. Sgura, V. Simoncini, Matrix-oriented discretization methods for reaction–diffusion PDEs: comparisons and applications, Comput. Math. Appl. 79 (2020) 2067–2085, https://doi.org/10.1016/j.camwa.2019.10.020.

[10] S. Duczek, H. Gravenkamp, Critical assessment of different mass lumping schemes for higher order serendipity finite elements, Comput. Methods Appl. Mech. Eng. 350 (2019) 836–897, https://doi.org/10.1016/j.cma.2019.03.028.

[11] G. Dziuk, C.M. Elliott, Finite element methods for surface PDEs, Acta Numer. 22 (2013) 289–396, https://doi.org/10.1017/S0962492913000056.

[12] C. Eilks, C.M. Elliott, Numerical simulation of dealloying by surface dissolution via the evolving surface finite element method, J. Comput. Phys. 227 (2008) 9727–9741, https://doi.org/10.1016/j.jcp.2008.07.023.

[13] C.M. Elliott, B. Stinner, Modeling and computation of two phase geometric biomembranes using surface finite elements, J. Comput. Phys. 229 (2010) 6585–6612, https://doi.org/10.1016/j.jcp.2010.05.014.

[14] M. Frittelli, I. Sgura, Virtual Element Method for the Laplace-Beltrami equation on surfaces, ESAIM: Math. Model. Numer. Anal. 52 (2018) 965–993, https://doi.org/10.1051/m2an/2017040.

[15] M. Frittelli, A. Madzvamuse, I. Sgura, C. Venkataraman, Lumped finite elements for reaction-cross-diffusion systems on stationary surfaces, Comput. Math. Appl. 74 (2017) 3008–3023, https://doi.org/10.1016/j.camwa.2017.07.044.

[16] M. Frittelli, A. Madzvamuse, I. Sgura, C. Venkataraman, Numerical preservation of velocity induced invariant regions for reaction–diffusion systems on evolving surfaces, J. Sci. Comput. 77 (2018) 971–1000, https://doi.org/10.1007/s10915-018-0741-7.

[17] M. Frittelli, A. Madzvamuse, I. Sgura, C. Venkataraman, Preserving invariance properties of reaction–diffusion systems on stationary surfaces, IMA J. Numer. Anal. 39 (2019) 235–270, https://doi.org/10.1093/imanum/drx058.

[18] M. Frittelli, A. Madzvamuse, I. Sgura, Bulk-surface virtual element method for systems of PDEs in two-space dimensions, Numer. Math. 147 (2021) 305–348, https://doi.org/10.1007/s00211-020-01167-3.

[19] M. Frittelli, A. Madzvamuse, I. Sgura, The bulk-surface virtual element method for reaction-diffusion PDEs: analysis and applications, Commun. Comput. Phys. 33 (2023) 733–763, https://doi.org/10.4208/cicp.OA-2022-0204.

[20] M. Frittelli, A. Madzvamuse, I. Sgura, Virtual element method for elliptic bulk-surface PDEs in three space dimensions, Numer. Methods Partial Differ. Equ. (2023) 1–27, https://doi.org/10.1002/num.23040.

[21] G. Golub, S. Nash, C. Van Loan, A Hessenberg-Schur method for the problem AX + XB = C, IEEE Trans. Autom. Control 24 (1979) 909–913, https://doi.org/10.1109/tac.1979.1102170.

[22] Y. Hao, V. Simoncini, Matrix equation solving of PDEs in polygonal domains using conformal mappings, J. Numer. Math. 29 (3) (2021) 221–244, https://doi.org/10.1515/jnma-2020-0035.

[23] J. Henning, D. Palitta, V. Simoncini, K. Urban, An ultraweak space-time variational formulation for the wave equation: analysis and efficient numerical solution, ESAIM: Math. Model. Numer. Anal. 56 (2022) 1173–1198, https://doi.org/10.1051/m2an/2022035.

[24] T.J.R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Courier Corporation, 2012.

[25] C. Jordan, K. Jordán, Calculus of Finite Differences, vol. 33, American Mathematical Soc., 1965.

[26] E.J. Kansa, Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics – II solutions to parabolic, hyperbolic and elliptic partial differential equations, Comput. Math. Appl. 19 (1990) 147–161, https://doi.org/10.1016/0898-1221(90)90271-k.

[27] P. Knupp, S. Steinberg, Fundamentals of Grid Generation, CRC Press, 2020.

[28] D. Lacitignola, B. Bozzini, I. Sgura, Spatio-temporal organization in a morphochemical electrodeposition model: analysis and numerical simulation of spiral waves, Acta Appl. Math. 132 (2014) 377–389, https://doi.org/10.1007/s10440-014-9910-3.

[29] D. Lacitignola, B. Bozzini, I. Sgura, Spatio-temporal organization in a morphochemical electrodeposition model: Hopf and Turing instabilities and their interplay, Eur. J. Appl. Math. 26 (2015) 143–173, https://doi.org/10.1017/s0956792514000370.

[30] D. Lacitignola, B. Bozzini, M. Frittelli, I. Sgura, Turing pattern formation on the sphere for a morphochemical reaction-diffusion model for electrodeposition, Commun. Nonlinear Sci. Numer. Simul. 48 (2017) 484–508, https://doi.org/10.1016/j.cnsns.2017.01.008.

[31] D. Lacitignola, I. Sgura, B. Bozzini, T. Dobrovolska, I. Krastev, Spiral waves on the sphere for an alloy electrodeposition model, Commun. Nonlinear Sci. Numer. Simul. 79 (2019) 104930, https://doi.org/10.1016/j.cnsns.2019.104930.

[32] D. Lacitignola, M. Frittelli, V. Cusimano, A. De Gaetano, Pattern formation on a growing oblate spheroid. An application to adult sea urchin development, J. Comput. Dyn. 9 (2022) 185–206, https://doi.org/10.3934/jcd.2021027.

[33] A. Mantzaflaris, B. Jüttler, B.N. Khoromskij, U. Langer, Low rank tensor methods in Galerkin-based isogeometric analysis, Comput. Methods Appl. Mech. Eng. 316 (2017) 1062–1085, https://doi.org/10.1016/j.cma.2016.11.013.

[34] Y.-Y. Nie, V. Thomée, A lumped mass finite-element method with quadrature for a non-linear parabolic problem, IMA J. Numer. Anal. 5 (1985) 371–396, https://doi.org/10.1093/imanum/5.4.371.

[35] D. Palitta, Matrix equation techniques for certain evolutionary partial differential equations, J. Sci. Comput. 87 (2021) 99, https://doi.org/10.1007/s10915-021-01515-x.

[36] D. Palitta, V. Simoncini, Matrix-equation-based strategies for convection–diffusion equations, BIT Numer. Math. 56 (2016) 751–776, https://doi.org/10.1007/s10543-015-0575-8.

[37] R.G. Plaza, F. Sanchez-Garduno, P. Padilla, R.A. Barrio, P.K. Maini, The effect of growth and curvature on pattern formation, J. Dyn. Differ. Equ. 16 (2004) 1093–1121, https://doi.org/10.1007/s10884-004-7834-8.

[38] C.E. Powell, D. Silvester, V. Simoncini, An efficient reduced basis solver for stochastic Galerkin matrix equations, SIAM J. Sci. Comput. 39 (2017) A141–A163, https://doi.org/10.1137/15m1032399.

[39] A. Quarteroni, Numerical Models for Differential Problems, vol. 16, Springer, 2017.

[40] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, 2003.

[41] G. Sangalli, M. Tani, Isogeometric preconditioners based on fast solvers for the Sylvester equation, SIAM J. Sci. Comput. 38 (2016) A3644–A3671, https://doi.org/10.1137/16m1062788.

[42] I. Sgura, A.S. Lawless, B. Bozzini, Parameter estimation for a morphochemical reaction–diffusion model of electrochemical pattern formation, Inverse Probl. Sci. Eng. 27 (2019) 618–647, https://doi.org/10.1080/17415977.2018.1490278.

[43] S.D. Shank, V. Simoncini, D.B. Szyld, Efficient low-rank solution of generalized Lyapunov equations, Numer. Math. 134 (2016) 327–342, https://doi.org/10.1007/s00211-015-0777-7.

[44] V. Simoncini, Computational methods for linear matrix equations, SIAM Rev. 58 (2016) 377–441, https://doi.org/10.1137/130912839.

[45] A.M. Turing, The chemical basis of morphogenesis, Philos. Trans. R. Soc. Lond. B, Biol. Sci. 237 (1952) 37–72, https://doi.org/10.1098/rstb.1952.0012.

[46] V.K. Vanag, Waves and patterns in reaction–diffusion systems. Belousov–Zhabotinsky reaction in water-in-oil microemulsions, Phys. Usp. 47 (2004) 923, https://doi.org/10.1070/pu2004v047n09abeh001742.