A level-set multigrid technique for nonlinear diffusion in the numerical simulation of marble degradation under chemical pollutants

A. Coco †, M. Semplice ‡ and S. Serra Capizzano §

February 20, 2019

Abstract

Having in mind the modelling of marble degradation under chemical pollutants, e.g. the sulfation process, we consider governing nonlinear diffusion equations and their numerical approximation. The space domain of a computation is the pristine marble object and, in order to accurately discretize it while maintaining the simplicity of finite difference discretizations, the domain is implicitly defined using a level-set approach. A uniform Cartesian grid is laid over a box containing the domain, but the solution is defined and updated only in the grid nodes that lie inside the domain, the level-set being employed to select them and to impose accurately the boundary conditions. We use a Crank-Nicolson in time, while for the space variables the discretization is performed by a standard Finite-Difference scheme for grid points inside the domain and by a ghost-cell technique on the ghost points (by using boundary conditions). The solution of the large nonlinear system is obtained by a Newton-Raphson procedure and a tailored multigrid technique is developed for the inner linear solvers. The numerical results, which are very satisfactory in terms of reconstruction quality and of computational efficiency, are presented and discussed at the end of the paper.

1 Introduction

Quantitative forecasts of damage by gaseous pollutants to monuments are becoming more and more important, since they allow to schedule monitoring, preservation and, when needed, restoration activities in the management of cultural heritage [21]. Phenomena involving chemical reactions of the constitutive material of a work of art with chemicals in the surrounding environment have long been recognized very important for the damage to cultural heritage sites. In a recent review of the related mathematical models [48], the employment of models based on partial differential equations has been advocated for the next generation of models with regulatory powers.

Differential models in this field typically involve a, possibly nonlinear, diffusion term describing the penetration of the gas in the bulk material, coupled with reaction terms modelling the chemistry of the interaction. A typical example is the sulfation process that turns marble into gypsum, for which a model was proposed in [4]. More recent models include the effects of the Darcy velocity [2], of the surface rugosity [7]; a kinetic approach to the derivations of the models of sulfation was proposed in [1]. More complex models including free boundaries can take into account the swelling of the material [44] or treat appropriately the heterogeneity of the crust layer [45]. Modelling of layered material with moving interfaces has been exploited also for copper corrosion in [13].
In this paper we focus on the model of [4], since it is quite simple but yet contains the more relevant numerical difficulties. The model considers a closed domain $\Omega$ that represents the pristine marble piece and two scalar variables defined in $\Omega$: $c(t, x)$ represents the marble concentration, initially set to 1, and $s(t, x)$ is the gas concentration, initially set to 0. The evolution equations (1) are characterized by a diffusion term for $s$, that is nonlinear since the diffusion coefficient depends on $c$, and by reaction terms coupling the two variables. Boundary conditions are of Dirichlet type and impose a value of $s$ on the boundary of $\Omega$, representing the pollution level of the surrounding air. It is assumed that, the gypsum concentration is $1 - c(t, x)$ and so, as time goes by, a gypsum crust forms on the outer shell of the monument, whose thickness and evolution is of interest to the managers of cultural heritage.

Previous numerical work on this model have considered the one-dimensional version of the model [32] or Cartesian grids in two space dimensions [49]. Of course this kind of meshes cannot stand one of the main difficulties of this kind of computations, which is the accurate discretization of the domain. This is quite relevant, since, despite the simplicity of the numerical techniques in [49], the computational model have shown the importance of two-dimensional effects near corners and other sharp features of the domain. In a real case $\Omega$ should coincide with the pristine work of art and only rarely this can be accurately represented on a Cartesian grid.

One of the simplest methods to overcome this difficulty consists of approximating $\Omega$ by small cuboids, whose size and shape are eventually adapted close to the boundary of $\Omega$ in such as way that the most external corners lie on the boundary. This approach is adopted by the Shortley-Weller discretization [50], providing a simple first-order accurate method that cannot be easily extended to obtain higher order methods (especially for Neumann boundary conditions). This method is the simplest approach falling under the class of boundary-fitted grid methods, where the grid is suitably adapted to the boundary of the domain.

More accurate and well-known methods of this class are represented by the Finite Element Methods (FEM), successfully adopted in several scientific contexts (e.g., [9, 10, 35, 5, 37, 26, 36, 22]). Although they may provide an extremely accurate representation of the domain, the generation of the mesh may become significantly expensive from a computational point of view especially for complex domains with several corners or highly variable curvature. This aspect is even more exasperated for moving domain problems, where a new mesh generation is needed at each time step. Another alternative for working with a Cartesian structure is furnished by the Isogeometric approach, adopted for example in [52, 39], where an evident difficulty is given by the need of using several patches when the domain is complicate and of course this is a concrete possibility when treating the degradation e.g. of a statue from our cultural heritage.

For all these reasons, the mathematical models proposed in this paper, where the domain coincides with the realistic monument with its sharp features, would be more efficiently solved by numerical approaches where the boundary is embedded in a steady Cartesian grid and implicitly described by a level-set function. Another advantage of this approach is that it would be easier to generalize the methods to the case of evolving boundaries or the presence of internal interfaces, advocated by the more modern models in [14, 45, 13]. A pioneering work of this class of methods is the Immersed Boundary Method [47] to model blood flows in the heart, where Peskin presented a first-order accurate method, later extended to higher order by the Immersed Interface Methods proposed by LeVeque and Li in [43].

More recent numerical approaches to discretise partial differential equations on complex domains in a Cartesian grid are the Ghost-Fluid Methods proposed in [28, 33, 34, 44], where the grid points lying outside of $\Omega$ but still close to the boundary are called ghost points and a fictitious (ghost) value of the numerical solution is formally extrapolated onto these grid points in order to maintain a standard discretization for all internal grid points and still achieving the desired accuracy order. The main feature of these methods is that the internal equations, which may contain ghost values for internal grid points that are close to the boundary, are firstly formally solved for the ghost values, resulting in a final linear system with eliminated boundary conditions. This class of methods may fail to obtain a higher accuracy for Neumann boundary condition, and anyway it remains first order accurate in the post-processing reconstruction of the gradient of the solution.

To improve the accuracy order for the gradient of the solution and/or in the presence of Neumann boundary conditions, a new finite-difference method was proposed by Coco and Russo in [15, 20, 17], where the ghost
point values are eventually coupled each other, resulting in a bigger linear system with non-eliminated boundary conditions, solved by a multigrid approach suitably designed for ghost-point methods. The domain is implicitly described by a level-set function, making the method suitable to be embedded in the framework of level-set methods to model moving domains and complex topological changes.

Other numerical methods to solve sharp-edge boundary problems are the finite volume methods [46], the non-symmetric positive definite finite element method [40], the arbitrary Lagrangian Eulerian method (ALE) [29, 25], the penalization methods [42, 8, 11], and the class of Immersed Finite Volume Methods (IFVM) [27]. Although high order accuracy is achieved by these methods, a second order accuracy for the gradient of the solution is either impossible or cumbersome to achieve.

Other recent advancements have been obtained in [38] to achieve higher accuracy in the presence of Neumann boundary conditions, and in [6] to also achieve higher accuracy in the gradient of the solution.

Within this paper we consider a novel numerical technique for the approximation of nonlinear (possibly degenerate) parabolic equations, which relies on the finite difference discretization and efficient solvers of [49, 23, 24] and on the level-set domain description and handling of boundary conditions of [18]. As in [49], the time discretization is the implicit Crank-Nicolson, a large nonlinear system at each time step is solved by a Newton-Raphson procedure, with a tailored multigrid technique for the linear systems. The spatial discretization is achieved by finite differences on a uniform Cartesian grid and, in the bulk of the domain, the numerical scheme is the same as in [23]. However, here, the domain can be of arbitrary shape and is implicitly defined by \( \Omega = \{ \mathbf{x}, \text{s.t.} \, \varphi(\mathbf{x}) < 0 \} \), where the level-set function \( \varphi : \mathbb{R}^N \to \mathbb{R} \) is known at least at the grid nodes. The grid nodes are defined, according to \( \varphi \), as internal (those inside \( \Omega \)), ghosts (first layer of points around the internal ones) and external. The method of [23] is applied only on the internal grid points. In order to close the method, the resulting nonlinear system of equations is augmented, as in [18], by the equations expressing the fulfillment of the boundary conditions on \( \partial \Omega \) in terms of the ghost values and of their first internal neighbour points. The resulting system is then solved by Newton-Raphson and the special smoothing technique of [18] is employed in the multigrid linear solver.

The outline of the paper is the following. In §2 we introduce the mathematical model. The numerical method is discussed in §3, discussing the details of the time discretization in §3.1, the space discretization in §3.2 and §3.3, the Newton-Raphson solver in §3.4 and the multigrid method in §3.5. The numerical tests of §4 include accuracy and efficiency tests, as well as examples of application to nontrivial geometries §4.3. Finally, the main conclusions of the paper and perspectives for future work are discussed in §5.

The numerical results, which are very satisfactory both from the viewpoint of the reconstruction quality and of the computational efficiency, are presented and discussed at the end of the paper.

## 2 Mathematical Model

Here we recall briefly the model of marble sulfation introduced in [4], referring the reader to the original paper for the details and more comprehensive study of the properties of the solutions. In [4], the authors consider the (simplified) chemical reaction

\[
\text{CaCO}_3 + \text{SO}_2 + \frac{1}{2} \text{O}_2 + 2\text{H}_2\text{O} \rightarrow \text{CaSO}_4 \cdot 2\text{H}_2\text{O} + \text{CO}_2.
\]

to account for the transformation of \( \text{CaCO}_3 \) of the marble stone into \( \text{CaSO}_4 \cdot 2\text{H}_2\text{O} \) (gypsum), that is triggered in a moist atmosphere by the availability of \( \text{SO}_2 \) at the marble surface and inside the pores of the stone.

Letting \( \Omega \subset \mathbb{R}^d \) represent the pristine marble piece, the equations governing the process of marble sulfation
are:

\[
\begin{aligned}
\frac{\partial (\phi c) s}{\partial t} &= - \frac{\alpha}{m_c} \phi c s + d \nabla \cdot (\phi c \nabla s) \quad \text{in } \Omega \times [0, T] \\
\frac{\partial c}{\partial t} &= - \frac{\alpha}{m_s} \phi c s \quad \text{in } \Omega \times [0, T] \\
s(x, t) &= s_b \quad \text{for } (x, t) \in \partial \Omega \times [0, T] \\
\frac{\partial c(x, t)}{\partial n} &= 0 \quad \text{for } (x, t) \in \partial \Omega \times [0, T] \\
c(x, 0) &= c_0(x) \quad \text{for } x \in \Omega.
\end{aligned}
\]  

The initial conditions set the value of the marble and gas concentration inside \( \Omega \) at the initial time. As the reaction proceeds, the calcium carbonate concentration is reduced from the initial value \( c_0 \), as CaCO\(_3\) is progressively replaced by gypsum.

The time evolution is described by the diffusion term in the gas equation and by the reaction terms in both differential equations. The porosity \( \phi \) of the material controls the diffusion of the gas in the pores of the marble. Since marble and gypsum have different porosities, \( \phi \) is not a constant but a function of the \( c(x, t) \), making the diffusion equation a nonlinear one. For simplicity, as in [4], we assume that

\[ \phi(c(x, t)) = \alpha c(x, t) + \beta, \quad \text{with } \alpha = 0.01 \text{ and } \beta = 0.1. \]

We point out that more complex relations may be employed, since our method does not rely strongly on the linearity of the above relation.

The third and fourth equations are the boundary conditions (the operator \( \partial / \partial n \) represents the derivative along the outward normal direction), that describe the conditions surrounding the work of art.

Although the model can be cast in higher dimensions, in this paper we focus on the 2D case for simplicity, since it contains already the main difficulties of the full 3D model. Let \( D = [-L, L]^2 \) be the computational domain, \( \Omega \subset D \) the domain representing the marble monument, \( \Gamma = \partial \Omega \) the boundary of the domain. The numerical method will consider a regular grid in \( D \) and a level-set function defined on the grid will be used both to detect the grid points inside \( \Omega \) and to describe the exact location and outward normal for the boundary of \( \Omega \).

### 3 Numerical method

#### 3.1 Time discretization

Equations [1] are discretized in time using the second order accurate Crank-Nicolson scheme

\[
\begin{aligned}
\frac{\phi^{(n+1)} s^{(n+1)} - \phi^{(n)} s^{(n)}}{\Delta t} &= \mathcal{L}^s (s^{(n)}, c^{(n)}) + \mathcal{L}^s (s^{(n+1)}, c^{(n+1)}) \quad \text{in } \Omega \\
\frac{\Delta c^{(n+1)}}{\Delta t} &= \mathcal{L}^c (s^{(n)}, c^{(n)}) + L_c (s^{(n+1)}, c^{(n+1)}) \quad \text{in } \Omega \\
\frac{\partial s^{(0)}}{\partial n} &= 0 \quad \text{on } \partial \Omega \\
c^{(0)} &= c_0
\end{aligned}
\]  

where

\[ \mathcal{L}^s(s, c) = - \frac{\alpha}{m_c} \phi c s + d \nabla \cdot (\phi c \nabla s), \quad \mathcal{L}^c(s, c) = - \frac{\alpha}{m_s} \phi c s \]

are the differential operators representing the right-hand side of [1], while \( s^{(n)} = s^{(n)}(x) \) and \( c^{(n)} = c^{(n)}(x) \) are functions of space only and represent approximations of the solutions \( s(x, t) \) and \( c(x, t) \), respectively, at time \( t_n = n \Delta t \), for \( n = 0, \ldots, N_t \) (with \( \Delta t = T/N_t \)).
3.2 Level-set function

The domain $\Omega$ is implicitly described by a level-set function $\varphi: D \to \mathbb{R}$ such that:

$$\Omega = \{ x \in \mathbb{R}^2 : \varphi(x) < 0 \}, \quad \Gamma = \partial \Omega = \{ x \in \mathbb{R}^2 : \varphi(x) = 0 \}.$$  

From the level-set function it is possible to infer the geometric properties of the boundary $\Gamma$. In fact, the outward unit normal vector $n$ and the curvature $\kappa$ are given by:

$$n = \frac{\nabla \varphi}{|\nabla \varphi|}, \quad \kappa = \nabla \cdot n.$$  

(3)

3.3 Spatial discretization

Discretization in space is performed by a standard finite-difference scheme for the grid points lying inside the domain $\Omega$ and by a ghost-cell technique on the ghost points (to impose high order accurate boundary conditions).

Let $D = [-L, L]^2$ be the computational domain and let $N \geq 1$ be the number of intervals in each direction. We call $h = 2L/N$ the spatial step. Observe that we are assuming for simplicity that $\Delta x = \Delta y = h$, although the method can be easily generalised to the case $\Delta x \neq \Delta y$.

The set of grid points is $D_h = \{(x_i, y_j) \in \mathbb{R}^2 : x_i = -L + ih, y_j = -L + jh, \text{ for } i,j = 0, \ldots, N\}$. Let $\Omega_h = \Omega \cap D_h$ be the discrete counterpart of $\Omega$.

We say that a grid point $(x_i, y_j)$ is a **ghost point** if and only if both of the following conditions are satisfied:

$$(x_i, y_j) \notin \Omega_h, \quad \{(x_i \pm h, y_j), (x_i, y_j \pm h)\} \cap \Omega_h \neq \emptyset.$$  

(4)

In other words, a ghost point is a grid point that is outside the domain $\Omega$ and that has one of its four neighbour grid points inside $\Omega$. We call $\Gamma_h$ the set of ghost points (see Fig. 1).

Let $I_\Omega = \{(i,j) : (x_i, y_j) \in \Omega_h\}$ and $I_\Gamma = \{(i,j) : (x_i, y_j) \in \Gamma_h\}$ be the sets of inside indices and ghost indices, respectively, and $N_i = |I_\Omega|$ and $N_g = |I_\Gamma|$ be their cardinalities.

![Fig. 1: Internal grid points $\Omega_h$ (blu filled circle) and ghost points $\Gamma_h$ (red empty circle), according to the definition of ghost points (4).](image1.png)

![Fig. 2: Five-point stencil adopted for the finite-difference discretization on inside grid points (6).](image2.png)

5
We aim at approximating the solution in $\Omega \cup \Gamma_h$ for any time step. Therefore, the numerical solution at each time step $t_n = n \Delta t$ is expressed by a vector $W^{(n)} = (s^{(n)}, c^{(n)}) \in \mathbb{R}^{2(N_1+N_g)}$, whose components are $s_{ij}^{(n)}$ and $c_{ij}^{(n)}$, with $(i,j)$ varying in $\Omega \cup \Gamma$.

The components of $W$ are ordered by choosing a mapping

$$\mathcal{M}: \{1,2,\ldots,N_1+N_g\} \longrightarrow \Omega \cup \Gamma.$$  

(5)

For the purpose of describing the numerical method, we order all $s$ variables before the $c$ ones, i.e. $W^{(n)} = (s^{(n)}, c^{(n)}) \in \mathbb{R}^{2(N_1+N_g)}$. Of course, the actual layout of the vector in the computer memory will be chosen to achieve optimal efficiency, e.g. like $W = (s_1, c_1, s_2, \ldots, s_{N_1}, c_{N_1}, \ldots, s_{N_1+N_g}, c_{N_1+N_g})$.

In order to compute $W^{(n+1)}$ from $W^{(n)}$, a system of $2(N_1+N_g)$ nonlinear equations in $2(N_1+N_g)$ unknowns must be solved at each time step. The nonlinear system is obtained as follows. The $2N_g$ nonlinear equations related to inside grid points are obtained by discretizing the first two equations of (2) on inside grid points $(x_i, y_j) \in \Omega$ using the standard five-point stencil (Fig. (2)) finite-difference scheme:

$$\phi \left( c_{ij}^{(n+1)} \right) s_{ij}^{(n+1)} - \phi \left( c_{ij}^{(n)} \right) s_{ij}^{(n)} - L_h^s(s^{(n)}, c^{(n)}) + L_h^c(s^{(n+1)}, c^{(n+1)}) = 0,$$

(6)

$$c_{ij}^{(n+1)} - c_{ij}^{(n)} - L_h^c(s^{(n)}, c^{(n)}) + L_h^c(s^{(n+1)}, c^{(n+1)}) = 0,$$

(7)

where

$$L_h^s(s, c) = -\frac{a}{m_c} \phi(c_{ij}) s_{ij} c_{ij} + \frac{d}{2} \sum_{(i',j') \in N_{ij}} \left( \phi(c_{i',j'}) + \phi(c_{ij}) \right) (s_{i',j'} - s_{ij})$$

and $N_{ij}$ is the set of four neighbouring index pairs for $(i,j)$, namely:

$$N_{ij} = \{(i \pm 1, j), (i, j \pm 1)\}.$$

The $2N_g$ linear equations related to ghost points are obtained by enforcing boundary conditions on $\Gamma$ using a ghost point extrapolation technique that was already successfully adopted in the context of elliptic \[18\] \[20\] \[17\] \[16\] \[15\] and hyperbolic \[12\] \[19\] equations. In detail, let $G_{ij} = (x_i, y_j) \in \Gamma_h$ be a ghost point and $n_{ij} = (n_x, n_y)$ be the approximated outward unit normal vector computed by a central finite-difference discretization of Eq. (3):

$$n_x = \frac{\bar{n}_x}{\sqrt{\bar{n}_x^2 + \bar{n}_y^2}}, \quad n_y = \frac{\bar{n}_y}{\sqrt{\bar{n}_x^2 + \bar{n}_y^2}}, \quad \text{with} \quad \bar{n}_x = \frac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2h}, \quad \bar{n}_y = \frac{\varphi_{i,j+1} - \varphi_{i,j-1}}{2h}, \quad \varphi_{ij} = \varphi(x_i, y_j).$$

(8)

Let $ST_{ij}$ be the nine-point stencil in the upwind direction with respect to $n_{ij}$. More precisely,

$$ST_{ij} = \left\{ (x_i, y_j) - h (m_x k_x, m_y k_y), (k_x, k_y) \in \{0,1,2\} \right\},$$

where $m_x = \text{SIGN}(n_x)$ and $m_y = \text{SIGN}(n_y)$, with $\text{SIGN}(x) = 1$ if $x \geq 0$ and $\text{SIGN}(x) = -1$ if $x < 0$ (we observe that we take conventionally $\text{SIGN}(0) = 1$). Since this nine-point stencil is taken in the upwind direction with respect to the normal $n_{ij}$, it can be easily proven that $ST_g \subseteq \Omega_h \cup \Gamma_h$, provided that the grid is sufficiently fine (i.e. $h$ is sufficiently small).

The linear equations are obtained by prescribing the boundary conditions of Eq. (2) (i.e. third and fourth equations) on $B_{ij} \subset \Gamma$, where $B_{ij}$ is the normal projection of $G_{ij}$ onto $\Gamma$, obtained by the following algorithm (see Fig. 3):
• compute \( P_{ij} = G_{ij} - 2h_n \) (we have \( P_{ij} \in \Omega \) provided that the grid is sufficiently fine, i.e. \( h \) is sufficiently small);

• apply the bisection method to solve \( \tilde{\varphi}(P_{ij} + \alpha (G_{ij} - P_{ij})) = 0 \) in the unknown \( \alpha \in [0, 1] \) (with the tolerance criterion: \( \min\{|\tilde{\varphi}(P_{ij} + \alpha_k (G_{ij} - P_{ij}))|, |\alpha_k - \alpha_{k-1}|\} < 10^{-6} \)), where \( \tilde{\varphi} \) is a biquadratic interpolation of \( \varphi \) on the stencil \( ST_{ij} \);

• compute \( B_{ij} = P_{ij} + \alpha (G_{ij} - P_{ij}) \).

Finally, the linear equations are:

\[
\begin{align*}
\text{(9)} & \quad s_b - \tilde{s}^{(n+1)}(B_{ij}) = 0, \\
\text{(10)} & \quad -\left(\nabla \tilde{c}^{(n+1)} \cdot \frac{\nabla \tilde{\varphi}}{|\nabla \tilde{\varphi}|}\right)_{B_{ij}} = 0,
\end{align*}
\]

where \( \tilde{s}^{(n+1)}, \tilde{c}^{(n+1)} \) and \( \tilde{\varphi} \) are the biquadratic interpolations of \( s^{(n+1)}, c^{(n+1)} \) and \( \varphi \), respectively, on the stencil \( ST_{ij} \).

### 3.4 Newton-Raphson method

Eqs. (6), (7), (9) and (10) constitute the system of nonlinear equations that we need to solve in order to advance in time from \( t_n \) to \( t_{n+1} \). Let \( F \) be an operator \( F : \mathbb{R}^{2(N_i + N_g)} \rightarrow \mathbb{R}^{2(N_i + N_g)} \) such that the nonlinear system (6), (7), (9), (10) can be represented by \( F(W^{(n+1)}) = 0 \). This system is solved by the Newton-Raphson method, i.e. by the following iterative scheme:

1. take the initial guess \( W^{(n+1,0)} = W^{(n)} \);
2. for \( k = 1, \ldots \), repeat the steps:
   a. solve the linear system
      \[
      J_F(W^{(n+1,k)}) \cdot \Delta W = F(W^{(n+1,k)}),
      \]
      where \( J_F(W^{(n+1,k)}) \) is the Jacobian matrix of \( F \);
(b) update the current guess $W^{(n+1,k+1)} = W^{(n+1,k)} - \Delta W$.
until a suitable tolerance is reached, e.g.:
\[
\min \left\{ \| F(W^{(n+1,k+1)}) \|_\infty, \frac{\| W^{(n+1,k+1)} - W^{(n+1,k)} \|_\infty}{\| W^{(n+1,k)} \|_\infty} \right\} < 10^{-9}.
\]

The Jacobian matrix can be represented in the compact form
\[
J_F(W^{(n+1,k)}) = \begin{pmatrix}
J^{ss}(W^{(n+1,k)}) & J^{sc}(W^{(n+1,k)}) \\
J^{cs}(W^{(n+1,k)}) & J^{cc}(W^{(n+1,k)})
\end{pmatrix},
\]
where $\{J^{ss}, J^{sc}, J^{cs}, J^{cc}\} (W^{(n+1,k)})$ are the four $(N_i + N_g) \times (N_i + N_g)$ matrices detailed below. Note that (12) describes the logical structure of the Jacobian matrix, and that its actual layout in the computer memory may be different and will match the layout chosen for the vectors $W^{(n)}$.

We represent the rows of a $(N_i + N_g) \times (N_i + N_g)$ matrix by $3 \times 3$ stencils in the following way: we say that the stencil
\[
\begin{bmatrix}
 a_{-1,1} & a_{0,1} & a_{1,1} \\
 a_{-1,0} & a_{0,0} & a_{1,0} \\
 a_{-1,-1} & a_{0,-1} & a_{1,-1}
\end{bmatrix}
\]
represents the $r$-th row of a $(N_i + N_g) \times (N_i + N_g)$ matrix $A$, with $M(r) = (i,j)$, if for any $1 \leq q \leq N_i + N_g$ we have that
\[
A_{rq} = \begin{cases} 
 a_{k_i,k_j} & \text{if } M(q) = (i+k_i,j+k_j), \text{ with } k_i,k_j \in \{-1,0,1\} \\
 0 & \text{otherwise.}
\end{cases}
\]
We have boxed the central element $a_{0,0}$ to emphasize that this element is on the main diagonal of the matrix $A$, i.e. $A_{rr} = a_{0,0}$. Sometimes (in particular for ghost points) it may be easier to have the element of the main diagonal not necessarily at the center of the $3 \times 3$ stencil. Therefore, we use for instance the stencil
\[
\begin{bmatrix}
 a_{0,0} & a_{1,0} & a_{2,0} \\
 a_{0,1} & a_{1,1} & a_{2,1} \\
 a_{0,2} & a_{1,2} & a_{2,2}
\end{bmatrix}
\]
if we want to represent the $r$-th row of $A$ as:
\[
A_{rq} = \begin{cases} 
 a_{k_i,k_j} & \text{if } M(q) = (i+k_i,j+k_j), \text{ with } k_i \in \{0,1,2\}, k_j \in \{-2,-1,0\} \\
 0 & \text{otherwise.}
\end{cases}
\]

**Representation of the matrix $J^{ss}(W^{(n+1,k)})$.** Using this notation, if $M(r) \in I_\Omega$ the $r$-th row of $J^{ss}(W^{(n+1,k)})$ is:
\[
\begin{bmatrix}
 0 & 0 & 0 \\
 0 & \frac{\phi c^{(n+1)}_{ij}}{\Delta t} + \frac{a}{2m_e} \phi c^{(n+1)}_{ij} & 0 \\
 0 & 0 & 0
\end{bmatrix}
\]
Let $\mathcal{M}(r) \in I_\Gamma$ and $n_x \geq 0$, $n_y \geq 0$ (see Fig. 4 and Eq. (8)). Referring to Fig. 5, observe that the three coefficients

$$\left[ \frac{\vartheta(\vartheta - 1)}{2}, \quad (1 - \vartheta)(1 + \vartheta), \quad \frac{\vartheta(1 + \vartheta)}{2} \right]$$

are the 1D quadratic interpolation coefficients on $\Gamma$ for grid points $x_{i-2}$, $x_{i-1}$, $x_i$, respectively, with $\vartheta = (\Gamma - x_{i-1})/h$.

The 2D biquadratic interpolation is obtained as dimension by dimension quadratic interpolations. Therefore, the $r$-th row of $J_{ss}(W(n+1,k))$ is represented by

$$\begin{bmatrix}
\vartheta_x \vartheta_y(1 + \vartheta_y) \\
(1 - \vartheta_y)(1 + \vartheta_x) \\
\vartheta_y(\vartheta_y - 1)
\end{bmatrix} \cdot \begin{bmatrix}
\vartheta_x(\vartheta_x - 1) \\
(1 - \vartheta_x)(1 + \vartheta_x) \\
\vartheta_x(1 + \vartheta_x)
\end{bmatrix} = \begin{bmatrix}
\vartheta_x(1 + \vartheta_x) \\
(1 + \vartheta_x)(1 + \vartheta_x) \\
\vartheta_x(1 + \vartheta_x)(1 + \vartheta_y)
\end{bmatrix} \cdot \begin{bmatrix}
\vartheta_y(1 + \vartheta_y) \\
(1 - \vartheta_y)(1 + \vartheta_y) \\
\vartheta_y(1 + \vartheta_y)
\end{bmatrix},$$

with

$$(\vartheta_x, \vartheta_y) = \frac{|B_{ij} - (x_{i+1}, y_{j-1})|}{h}.$$
Observe that the coefficients of the other three cases are the same as in (15), but in a different order, and that the boxed coefficient (i.e. the coefficient that will populate the main diagonal of the matrix) has the same expression in all cases.

**Representation of the matrix** $J^{sc}(W^{(n+1,k)})$. The $r$-th row of $J^{sc}(W^{(n+1,k)})$ is

\[
\begin{bmatrix}
0 & 0 & 0 & 0 \\
\frac{\phi'(c^{(n+1)}_{ij})}{\Delta t} s^{(n+1)}_{ij} + \frac{a}{2 m_c} \left( \phi'(c^{(n+1)}_{ij}) \right) s^{(n+1)}_{ij} c^{(n+1)}_{ij} + \phi(c^{(n+1)}_{ij}) s^{(n+1)}_{ij} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

\[+ \frac{d}{4} \sum_{(i^*,j^*) \in N_{ij}} \left( s^{(n+1)}_{ij} - s^{(n+1)}_{i^*,j^*} \right) \begin{bmatrix}
0 & \phi'(c^{(n+1)}_{i,j+1}) & 0 \\
\phi'(c^{(n+1)}_{i,j-1}) & 0 & \phi'(c^{(n+1)}_{i,j+1}) \\
0 & \phi'(c^{(n+1)}_{i+1,j}) & 0 \\
\end{bmatrix} \tag{16}
\]

if $M(r) \in \Omega$, or a null row if $M(r) \in \Gamma$.

**Representation of the matrix** $J^{cs}(W^{(n+1,k)})$. The $r$-th row of $J^{cs}(W^{(n+1,k)})$ is

\[
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & \phi'(c^{(n+1)}_{ij}) s^{(n+1)}_{ij} c^{(n+1)}_{ij} & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

\[\tag{17}
\]

if $M(r) \in \Omega$, or a null row if $M(r) \in \Gamma$.

**Representation of the matrix** $J^{cc}(W^{(n+1,k)})$. If $M(r) \in \Omega$, the $r$-th row of $J^{cc}(W^{(n+1,k)})$ is represented by

\[
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & \frac{a}{m_x} \phi'(c^{(n+1)}_{ij}) s^{(n+1)}_{ij} c^{(n+1)}_{ij} & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

\[\tag{18}
\]

If $M(r) \in \Gamma$ and $n_x, n_y \geq 0$, in order to obtain the $r$-th row of $J^{cc}(W^{(n+1,k)})$ we need to use the condition (10):

\[
\tilde{n}_x \frac{\partial \tilde{\phi}}{\partial x} + \tilde{n}_y \frac{\partial \tilde{\phi}}{\partial y} = 0, \quad \text{with} \quad \tilde{n}_x = \frac{\tilde{n}_x^*}{\sqrt{(\tilde{n}_x^*)^2 + (\tilde{n}_y^*)^2}}, \quad \tilde{n}_y = \frac{\tilde{n}_y^*}{\sqrt{(\tilde{n}_x^*)^2 + (\tilde{n}_y^*)^2}}, \quad \tilde{n}_x^* = \frac{\partial \tilde{\phi}}{\partial x}, \quad \tilde{n}_y^* = \frac{\partial \tilde{\phi}}{\partial y}. \tag{19}
\]
Since the coefficients of the 1D quadratic approximation of the first derivative on $\Gamma$ are (see Fig. 5)

\[
\begin{bmatrix}
\frac{1}{h} \left[ \vartheta - \frac{1}{2}, -2\vartheta, \vartheta + \frac{1}{2} \right]
\end{bmatrix}
\]

(20)

for grid points $x_{i-2}, x_{i-1}, x_i$, respectively, then the $r$-th row of $J^{cc}(W^{(n+1,k)})$ is represented by (from [19], see Fig. 4)

\[
\tilde{n}_x = \frac{1}{h} \begin{bmatrix}
\vartheta_y(1 + \vartheta_y) \\
(1 - \vartheta_y)(1 + \vartheta_y) \\
\vartheta_y(\vartheta_y - 1)
\end{bmatrix}
\]

\[
\begin{bmatrix}
\frac{\vartheta_y(1 + \vartheta_y)}{2} \\
\frac{\vartheta_y(\vartheta_y - 1)}{2} \\
\frac{\vartheta_y + 1}{2}
\end{bmatrix}
\cdot \begin{bmatrix}
\frac{1}{h} \left[ \vartheta x - \frac{1}{2}, -2\vartheta x, \vartheta x + \frac{1}{2} \right]
\end{bmatrix}
\]

\[
\tilde{n}_y = \frac{1}{h} \begin{bmatrix}
\vartheta_x(1 + \vartheta_x) \\
(1 - \vartheta_x)(1 + \vartheta_x) \\
\vartheta_x(\vartheta_x - 1)
\end{bmatrix}
\]

\[
\begin{bmatrix}
\frac{\vartheta_x(1 + \vartheta_x)}{2} \\
\frac{\vartheta_x(\vartheta_x - 1)}{2} \\
\frac{\vartheta_x + 1}{2}
\end{bmatrix}
\cdot \begin{bmatrix}
\frac{1}{h} \left[ \vartheta y - \frac{1}{2}, -2\vartheta y, \vartheta y + \frac{1}{2} \right]
\end{bmatrix}
\]

\[
+ \tilde{n}_y \frac{1}{h} \begin{bmatrix}
\vartheta_y(1 + \vartheta_y) \\
(1 - \vartheta_y)(1 + \vartheta_y) \\
\vartheta_y(\vartheta_y - 1)
\end{bmatrix}
\begin{bmatrix}
\frac{1}{h} \left[ \vartheta x - \frac{1}{2}, -2\vartheta x, \vartheta x + \frac{1}{2} \right]
\end{bmatrix}
\]

\[
+ \tilde{n}_y \frac{1}{h} \begin{bmatrix}
\vartheta_x(1 + \vartheta_x) \\
(1 - \vartheta_x)(1 + \vartheta_x) \\
\vartheta_x(\vartheta_x - 1)
\end{bmatrix}
\begin{bmatrix}
\frac{1}{h} \left[ \vartheta y - \frac{1}{2}, -2\vartheta y, \vartheta y + \frac{1}{2} \right]
\end{bmatrix}
\]

\[
+ \tilde{n}_y \frac{1}{h} \begin{bmatrix}
\vartheta_y(1 + \vartheta_y) \\
(1 - \vartheta_y)(1 + \vartheta_y) \\
\vartheta_y(\vartheta_y - 1)
\end{bmatrix}
\begin{bmatrix}
\frac{1}{h} \left[ \vartheta x - \frac{1}{2}, -2\vartheta x, \vartheta x + \frac{1}{2} \right]
\end{bmatrix}
\]

\[
+ \tilde{n}_y \frac{1}{h} \begin{bmatrix}
\vartheta_x(1 + \vartheta_x) \\
(1 - \vartheta_x)(1 + \vartheta_x) \\
\vartheta_x(\vartheta_x - 1)
\end{bmatrix}
\begin{bmatrix}
\frac{1}{h} \left[ \vartheta y - \frac{1}{2}, -2\vartheta y, \vartheta y + \frac{1}{2} \right]
\end{bmatrix}
\]

\[
\tilde{n}_x = \frac{\tilde{n}_x^*}{\sqrt{(\tilde{n}_x^*)^2 + (\tilde{n}_y^*)^2}}, \quad \tilde{n}_y = \frac{\tilde{n}_y^*}{\sqrt{(\tilde{n}_x^*)^2 + (\tilde{n}_y^*)^2}}.
\]

\[
\tilde{n}_x = \frac{1}{h} \begin{bmatrix}
\varphi_{i-j,1} \\
\varphi_{i-1,j} \\
\varphi_{i,j}
\end{bmatrix} \odot \begin{bmatrix}
\frac{\vartheta_y(1 + \vartheta_y)}{2} \\
\frac{\vartheta_x(1 + \vartheta_x)}{2} \\
\vartheta_y(\vartheta_y - 1)
\end{bmatrix}
\begin{bmatrix}
\frac{1}{h} \left[ \vartheta x - \frac{1}{2}, -2\vartheta x, \vartheta x + \frac{1}{2} \right]
\end{bmatrix}
\]

\[
\tilde{n}_y = \frac{1}{h} \begin{bmatrix}
\varphi_{i-2,j-2} \\
\varphi_{i-2,j-1} \\
\varphi_{i-2,j}
\end{bmatrix} \odot \begin{bmatrix}
\frac{\vartheta_y(1 + \vartheta_y)}{2} \\
\frac{\vartheta_x(1 + \vartheta_x)}{2} \\
\vartheta_y(\vartheta_y - 1)
\end{bmatrix}
\begin{bmatrix}
\frac{1}{h} \left[ \vartheta y - \frac{1}{2}, -2\vartheta y, \vartheta y + \frac{1}{2} \right]
\end{bmatrix}
\]
we implement a collective Gauss-Seidel scheme, i.e. a two-dimensional differential equations, where the collective Gauss-Seidel scheme is usually preferred (see Section 8). In this paper we focus on the update $\Delta s$ (with $\omega = 1$) that the residual linear system is well represented on a coarser grid (see Section 2.1). Well known relaxation techniques such as the high frequency components of the defect should be dumped quickly after few relaxations, in such a way that the residual linear system is well represented on a coarser grid.

Let us write the linear system (11) as:

$$
\frac{\partial}{\partial x}(1+\varphi_x)\left(\frac{\partial x}{2}(\varphi_x - 1)\right) + \frac{\partial}{\partial y}(1+\varphi_y)\left(\frac{\partial y}{2}(\varphi_y - 1)\right) = \left[ \begin{array}{ccc}
\varphi_{x-2,j} & \varphi_{x-1,j} & \varphi_{x,j} \\
\varphi_{x-2,j-1} & \varphi_{x-1,j-1} & \varphi_{x,j-1} \\
\varphi_{x-2,j-2} & \varphi_{x-1,j-2} & \varphi_{x,j-2}
\end{array} \right] \circ
\left[ \begin{array}{ccc}
\varphi_{x-2,j} & \varphi_{x-1,j} & \varphi_{x,j} \\
\varphi_{x-2,j-1} & \varphi_{x-1,j-1} & \varphi_{x,j-1} \\
\varphi_{x-2,j-2} & \varphi_{x-1,j-2} & \varphi_{x,j-2}
\end{array} \right]
$$

where the product operator $\circ$ means the product component-wise between the two $3 \times 3$ matrices and the sum over all components (inner products between the two vector representations of the two matrices).

The other three possible cases $\{n_x < 0, n_y \geq 0\}$, $\{n_x \geq 0, n_y < 0\}$ and $\{n_x < 0, n_y < 0\}$ are obtained similarly.

Summarizing, the four sub-matrices of the Jacobian matrix (12) can be represented in matrix form (see Eqs. 13, 15, 16, 17, 18, 21):

$$
J^{ss}(W^{n+1,k}) = D_{ss}(W^{n+1,k}) + \frac{d}{4} M_{ss}(W^{n+1,k}) + R_{ss}, \\
J^{sc}(W^{n+1,k}) = D_{sc}(W^{n+1,k}) + \frac{d}{4} M_{sc}(W^{n+1,k}),
$$

where $D_{ss}(W^{n+1,k})$, $D_{sc}(W^{n+1,k})$, $D_{cc}(W^{n+1,k})$, $D_{cc}(W^{n+1,k})$ are four $(N_x + N_y) \times (N_x + N_y)$ diagonal matrices, $M_{ss}(W^{n+1,k})$, $M_{sc}(W^{n+1,k})$ are two $(N_x + N_y) \times (N_x + N_y)$ penta-diagonal matrices, while $R_{ss}$, $R_{cc}$ are two $(N_x + N_y) \times (N_x + N_y)$ nine-diagonal matrices that do not depend on $W^{n+1,k}$ and then can be precomputed at the beginning of the numerical simulation.

### 3.5 Multigrid method

The linear system (11) is solved by a multigrid approach, as described in this section. In particular, we will introduce the relaxation operator (Section 3.5.1) and the transfer (restriction and interpolation) operators (Sections 3.5.2 and 3.5.3). The multigrid method can then be easily implemented from these operators (we refer the reader to any book on multigrid methods for a comprehensive presentation, such as [51]). In this paper we implement the $W$-cycle multigrid scheme and compare its convergence factor against the one predicted by the Local Fourier Analysis for $W$-cycle multigrid schemes in rectangular domains. Extensions to multigrid schemes more efficient than $W$-cycle such as Full-multigrid [51] Ch. 2.6] can be also easily implemented following the same approach proposed in this paper.

#### 3.5.1 Relaxation operator

Let us write the linear system (11) as:

$$
\begin{bmatrix}
J^{ss}(W^{n+1,k}) & J^{sc}(W^{n+1,k}) \\
J^{cs}(W^{n+1,k}) & J^{cc}(W^{n+1,k})
\end{bmatrix}
\begin{bmatrix}
\Delta s \\
\Delta c
\end{bmatrix} =
\begin{bmatrix}
F^{ss}(W^{n+1,k}) \\
F^{cc}(W^{n+1,k})
\end{bmatrix}
$$

In order to have an efficient multigrid method, the relaxation operator must satisfy the smoothing property, i.e. the high frequency components of the defect should be dumped quickly after few relaxations, in such a way that the residual linear system is well represented on a coarser grid (see [51] Ch. 2.1]). Well known relaxation operators that satisfy the smoothing property for scalar elliptic equations are Gauss-Seidel and weighted-Jacobi (with $\omega = 4/5$). It is known that the classical Gauss-Seidel scheme may underperform for systems of partial differential equations, where the collective Gauss-Seidel scheme is usually preferred (see [51] Ch. 8). In this paper we implement a collective Gauss-Seidel scheme, i.e. a 2 x 2 linear system is solved at each internal grid point to update $\Delta s_{ij}$ and $\Delta c_{ij}$ simultaneously, while an appropriate relaxation is performed on ghost points (18, [20, 17]).
In detail, the relaxation scheme spans all \( r = 1, \ldots, N + N_g \). If \( \mathcal{M}(r) = (i, j) \in \mathcal{I}_r \), then \( \Delta s_{ij} \) and \( \Delta c_{ij} \) are updated as follows:
\[
\begin{pmatrix}
(\Delta s^{(m+1)})_r \\
(\Delta c^{(m+1)})_r
\end{pmatrix}
= 
\begin{pmatrix}
(\Delta s^{(m)})_r \\
(\Delta c^{(m)})_r
\end{pmatrix}
+ P^{-1} \begin{pmatrix}
F^s(W^{(n+1,k)})_r - (J^{ss}(W^{(n+1,k)})_r \cdot \Delta s^*)_r \\
F^c(W^{(n+1,k)})_r - (J^{sc}(W^{(n+1,k)})_r \cdot \Delta c^*)_r
\end{pmatrix},
\]
where \( F^s(W^{(n+1,k)})_r \) and \( F^c(W^{(n+1,k)})_r \) are the \( r \)-th components of the vectors \( F^s(W^{(n+1,k)}) \) and \( F^c(W^{(n+1,k)}) \), respectively, and \( J^{ss,sc,cs,cc}(W^{(n+1,k)})_r \) is the \( r \)-th row of the matrix \( J^{ss,sc,cs,cc}(W^{(n+1,k)}) \) (see Eqs. (13),(16),(17),(18)).

We have denoted by \( \Delta s^* \) and \( \Delta c^* \) the current approximations of the Gauss-Seidel iteration, i.e. they are vectors whose \( q \)-th component is defined by:
\[
(\Delta s^*)_q = \begin{cases} (\Delta s^{(m+1)})_q & \text{if } 1 \leq q < r \\ (\Delta s^{(m)})_q & \text{if } r \leq q \leq N_i + N_g \end{cases}, \quad (\Delta c^*)_q = \begin{cases} (\Delta c^{(m+1)})_q & \text{if } 1 \leq q < r \\ (\Delta c^{(m)})_q & \text{if } r \leq q \leq N_i + N_g \end{cases}
\]
and
\[
P = \begin{pmatrix}
(J^{ss}(W^{(n+1,k)}))_r & 0 \\
(J^{sc}(W^{(n+1,k)}))_r & (J^{cc}(W^{(n+1,k)}))_r
\end{pmatrix}.
\]

We observe that the classical (pointwise) Gauss-Seidel scheme can be obtained by replacing the matrix \( \Delta \) with
\[
P = \begin{pmatrix}
(J^{ss}(W^{(n+1,k)}))_r & 0 \\
0 & (J^{cc}(W^{(n+1,k)}))_r
\end{pmatrix}.
\]

If \( \mathcal{M}(r) = (i, j) \in \mathcal{I}_r \), then \( \Delta s_{ij} \) and \( \Delta c_{ij} \) are updated as follows:
\[
\begin{pmatrix}
(\Delta s^{(m+1)})_r \\
(\Delta c^{(m+1)})_r
\end{pmatrix}
= 
\begin{pmatrix}
(\Delta s^{(m)})_r + \tau^s (F^s(W^{(n+1,k)}))_r - J^{ss}(W^{(n+1,k)})_r \cdot \Delta s^* \\
(\Delta c^{(m)})_r + \tau^c (F^c(W^{(n+1,k)}))_r - J^{cc}(W^{(n+1,k)})_r \cdot \Delta c^*
\end{pmatrix},
\]
where \( J^{ss,cc}(W^{(n+1,k)})_r \) is the \( r \)-th row of the matrix \( J^{ss,cc}(W^{(n+1,k)}) \) (see Eqs. (15),(16)), and \( \Delta s^* \) and \( \Delta c^* \) are defined as in (22). If we choose \( \tau^s \) and \( \tau^c \) as in (24) to have a Gauss-Seidel iteration, i.e.
\[
\tau^s = \left( \frac{\vartheta_x(1 + \vartheta_x)\vartheta_y(1 + \vartheta_y)}{4} \right)^{-1}, \quad \tau^c = \left( \frac{\vartheta_x}{h} + \frac{\vartheta_y}{h} + \frac{\vartheta_x}{h} + \frac{\vartheta_y}{h} \right)^{-1},
\]
the relaxation scheme may not converge (we observe for example that rows (15) and (21) are not diagonally dominant). Following the idea proposed in (13),(16),(17),(18), the parameters \( \tau^s \) and \( \tau^c \) are chosen in such a way that a proper CFL condition is satisfied for the iterations (25). In particular, we want to ensure that the absolute value of the coefficient of \( \Delta(s,c)^{(m)}_r \) in the right-hand side of (25) is smaller than one, i.e.
\[
\left| 1 - \tau^s \vartheta_x(1 + \vartheta_x)\vartheta_y(1 + \vartheta_y) \right| < 1
\]
\[
\left| 1 - \tau^c \left( \vartheta_x + \frac{1}{2} \right) \left( \vartheta_x + \frac{1}{2} \right) + \vartheta_y \left( \vartheta_y + \frac{1}{2} \right) \right| < 1.
\]
Parameters \( \tau^s \) and \( \tau^c \) should not depend on \( \vartheta_x, \vartheta_y \) and must be chosen in such a way that conditions (26) are satisfied for any \( 0 \leq \vartheta_x, \vartheta_y \leq 1 \) and for any \( (\tilde{n}_x,\tilde{n}_y) \): \( (\tilde{n}_x)^2 + (\tilde{n}_y)^2 = 1 \). This is achieved by:
\[
0 < \tau^s < 1, \quad 0 < \tau^c < \frac{2\sqrt{2} h}{3}.
\]
For practical purposes, we will choose
\[
\tau^s = 0.9, \quad \tau^c = 0.9 \frac{2\sqrt{2} h}{3}.
\]

Finally, although in this paper we will use the lexicographic order of the map (5), more efficient (collective) Gauss-Seidel schemes for multigrid methods, such as Red-Black Gauss-Seidel, can be easily implemented.
3.5.2 Restriction operator

After $\nu_1$ pre-relaxation iterations (Sect. 3.5.1), we compute the defects $r_{\nu}^h$ and $r_{\nu}^c$:

$$r_{\nu}^h = F_c^e(W^{(n+1,k)}) - (J_c^e(W^{(n+1,k)}) \cdot \Delta s^{(\nu_1)} + J_c^c(W^{(n+1,k)}) \cdot \Delta c^{(\nu_1)})$$
$$r_{\nu}^c = F_c^c(W^{(n+1,k)}) - (J_c^e(W^{(n+1,k)}) \cdot \Delta s^{(\nu_1)} + J_c^c(W^{(n+1,k)}) \cdot \Delta c^{(\nu_1)})$$

that will be restricted to the coarser grid (with spatial step $2h$) by a suitable restriction operator $\mathcal{I}_{2h}^h$:

$$r_{\nu}^g = \mathcal{I}_{2h}^h r_{\nu}^h, \quad r_{\nu}^c = \mathcal{I}_{2h}^h r_{\nu}^c.$$ 

We observe that the defects are discontinuous across the boundary, because the defects on $\mathcal{I}_h$ are related to the internal equations, while their values on $\mathcal{I}_g$ are referred to the boundary conditions (see [18]). For this reason, the restriction of the internal equations must use values only from $\mathcal{I}_h$. In detail, we perform the usual full-weighting restriction away from the boundary, while we modify the restriction for inner equations close to the boundary. We recall the full-weighting restriction operator (see [51] Ch. 2.3.3)):

$$\mathcal{I}_{2h}^h = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}^{2h}.$$  \hspace{1cm} (27)

In general, by the stencil notation

$$\mathcal{I}_{2h}^h = \left[ \begin{array}{cccc} \vdots & \vdots & \vdots & \vdots \\ \cdots & t_{-1,-1} & t_{-1,0} & t_{-1,1} \\ \cdots & t_{0,-1} & t_{0,0} & t_{0,1} \\ \cdots & t_{1,-1} & t_{1,0} & t_{1,1} \\ \vdots & \vdots & \vdots & \vdots \end{array} \right]^{2h}$$

we will denote the restriction operator $\mathcal{I}_{2h}^h$ defined by:

$$\mathcal{I}_{2h}^h w_h(x,y) = \sum_{(i,j) \in R_k} t_{i,j} w_h(x + jh, y + ih),$$

where only a finite number of coefficients $t_{i,j}$ is different from zero, and $R_k \equiv \{-k, \ldots, k\}^2$ for some positive integer $k$. In practice, $k = 1$ allows second order restriction operator.

Following the same technique of [18] [29], we modify the restriction operator when we are close to the boundary in such a way we only use values on internal grid points (i.e. we do not use the values of the defects on ghost points). Let $(x, y) \in D_{2h}$ a grid point of the coarse grid with spatial step $2h$ and let $\mathcal{N}(x,y) = \{(x + jh, y + ih) : j, i = -1, 0, 1\}$ be the neighborhood of $(x, y)$ in the finer grid. Let $\mathcal{T}$ be the full rectangle with maximum area that does not cross the boundary (see Fig. 6). The stencil used in $(x, y)$ to transfer $w_h$ to a coarser grid depends on the size of $\mathcal{T}$. In particular, if $\mathcal{T} \cap D_h$ is a $3 \times 3$ point stencil (i.e. $\mathcal{N}(x,y) \subseteq \Omega_h$), then we can use the standard full-weighting stencil (27). Now let $\mathcal{T} \cap D_h$ be a $3 \times 2$ point stencil (the case $2 \times 3$ is similar). Without loss of generality, we can suppose that the vertices of $\mathcal{T}$ are $(x + jh, y + ih)$, with $j \in \{-1, 0\}$, $i \in \{-1,1\}$. In this case, the restriction operator is:

$$(\mathcal{I}_{2h}^h w_h)(x,y) = \frac{1}{16} \begin{bmatrix} 2 & 2 & 0 \\ 4 & 4 & 0 \\ 2 & 2 & 0 \end{bmatrix}^{2h} (x,y).$$ \hspace{1cm} (28)
while, if \( \mathcal{T} \) is a \( 2 \times 2 \) point stencil, with vertex \((x + jh, y + ih)\), \( j, i \in \{-1, 0\}\), the restriction operator is:

\[
(I_{2h}^h w_h)(x, y) = \frac{1}{16} \begin{bmatrix} 0 & 0 & 0 \\ 4 & 4 & 0 \\ 4 & 4 & 0 \end{bmatrix}_{2h}^h(x, y), \tag{29}
\]

These three cases are summarized in Fig. 6.

![Fig. 6: Top: nine point stencil \( \mathcal{N}(x, y) \) (circles) and the boundary of the rectangle \( \mathcal{T} \) (bold line). The bold circle is on both the coarser and finer grids, while the smaller circles are only on the finer grid. The arrows represent the action of the restriction operators. Bottom: the respective stencils in matrix form used by the restriction operator (Eqs. (27), (28) and (29)).](image)

The restriction of the boundary conditions is performed using the same idea, namely using only points outside the domain (i.e. either ghost points or inactive grid points), provided that the defects are firstly defined in the neighbouring inactive points by extrapolating the ghost value constantly along the normal direction to the boundary \( \Gamma \). This can be achieved by solving the transport equations

\[
\frac{\partial r_h^{(s,c)}}{\partial \tau} + \nabla r_h^{(s,c)} \cdot n = 0.
\]

for a few steps of a fictitious time \( \tau \), (using for example Euler explicit method), where \( n = \nabla \varphi / |\nabla \varphi| \) is the unit normal vector.

### 3.5.3 Interpolation operator

The defect equations

\[
\begin{bmatrix} J^{ss}(W^{(n+1,k)}) \\ J^{sc}(W^{(n+1,k)}) \\ J^{cs}(W^{(n+1,k)}) \\ J^{cc}(W^{(n+1,k)}) \end{bmatrix} \cdot \begin{bmatrix} e_{2h}^s \\ e_{2h}^c \end{bmatrix} = \begin{bmatrix} r_{2h}^s \\ r_{2h}^c \end{bmatrix}
\]

are solved recursively on the coarser grid (where \( J^{ss,sc,cs,cc}(W^{(n+1,k)}) \) are approximated in the coarser grid using the same technique as in the fine grid), and then the error is interpolated back to the finer grid:

\[
e_h^s = I_{2h}^{2h} e_{2h}^s, \quad e_h^c = I_{2h}^{2h} e_{2h}^c.
\]
Since errors $e_h^s$ and $e_h^c$ are continuous across the boundary, we do not need to modify the stencil for particular cases and we are allowed to use the standard linear interpolation operator even close to the boundary:

$$T_h^{2h} = \frac{1}{4} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} h^2.$$

Finally, $\nu_2$ post-relaxation iterations (Sect. 3.5.1) are performed on the finer grid.

4 Numerical tests

In this section we confirm numerically the second order accuracy of the numerical method and we investigate the efficiency of the multigrid approach. We choose the following parameters in Eq. 1 (see [49]):

$$a = 10^4, \quad d = 0.1, \quad m_s = 64.06, \quad m_c = 100.09, \quad \phi(c) = 0.1 + 0.01c. \quad (30)$$

We choose $\Delta t = \Delta x = \Delta y = h$ and we compute the numerical solution up to the final time $t = 1$. The W-cycle iteration scheme of the multigrid is performed with $\nu_1 = 2$ pre-relaxation, $\nu_2 = 1$ post-relaxation and with an $8 \times 8$ grid as the coarsest grid.

4.1 Accuracy test

In order to test the accuracy, we modify the numerical method to solve a more general problem than (1) by adding source terms $f_1, f_2: \Omega \times [0, T] \to \mathbb{R}$ and boundary values $g_1, g_2: \partial\Omega \times [0, T] \to \mathbb{R}$:

$$\begin{align*}
\frac{\partial (c(t))}{\partial t} &= -\frac{a}{m_e} \phi(c) s c + d \nabla \cdot (\phi(c) \nabla s) + f_1 \quad \text{in } \Omega \times [0, T] \\
\frac{\partial c}{\partial t} &= \phi(c) s c + f_2 \quad \text{in } \Omega \times [0, T] \\
s(x, t) &= g_1(x, t) \quad \text{for } (x, t) \in \partial\Omega \times [0, T] \\
\frac{\partial c}{\partial n}(x, t) &= g_2(x, t) \quad \text{for } (x, t) \in \partial\Omega \times [0, T] \\
s(x, 0) &= s_0(x) \quad \text{for } x \in \Omega \\
c(x, 0) &= c_0(x) \quad \text{for } x \in \Omega
\end{align*}$$

We choose $f_1, f_2, g_1, g_2$ in such a way that the exact solutions are:

$$s^{\text{exa}}(x, t) = 2 + \sin(x) \cos(y) \sin(t + \sqrt{2}), \quad c^{\text{exa}}(x, t) = 3 + \sin(0.5x) \cos(3y) \sin(2t + \sqrt{3})$$

and then we compute the $L^p$ errors at time $t = 1$ on the solutions

$$e_h^s = \frac{\|s_h - s^{\text{exa}}_h\|_p}{\|s^{\text{exa}}_h\|_p}, \quad e_h^c = \frac{\|c_h - c^{\text{exa}}_h\|_p}{\|c^{\text{exa}}_h\|_p}$$

and on the gradients

$$e_h^{\nabla s} = \frac{\|\nabla s_h - |\nabla s^{\text{exa}}_h|\|_p}{\|\nabla s^{\text{exa}}_h\|_p}, \quad e_h^{\nabla c} = \frac{\|\nabla c_h - |\nabla c^{\text{exa}}_h|\|_p}{\|\nabla c^{\text{exa}}_h\|_p},$$

where $s_h$ and $c_h$ are the numerical solutions and $\nabla s_h$ and $\nabla c_h$ are computed by central differences. We perform two tests. In Test 1, the domain is represented by a circle and the level-set function is:

$$\varphi(x, y) = \sqrt{(x - x_0)^2 + (y - y_0)^2} - R, \quad \text{where } x_0 = \sqrt{2}/30, \quad y_0 = \sqrt{3}/40, \quad R = 1.486. \quad (31)$$
In Test 2 the domain is represented by the union of a square and four circles centred on the vertices

\[ \varphi(x, y) = \min \{ \varphi_1(x, y), \varphi_2(x, y) \} \]  

(32)

where

\[ \varphi_1(x, y) = \max \{ |x|, |y| \} - L, \quad \varphi_2(x, y) = \sqrt{(|x| - L)^2 + (|y| - L)^2} - D, \quad L = 0.9567, \quad D = 0.3. \]

The domains for Test 1 and Test 2 are represented in Fig. 7. \( L_1 \) and \( L_\infty \) errors for the solutions and the gradients for Test 1 can be found in Table 1 (for SO\(_2\)) and Table 2 (for CaCO\(_3\)), and for Test 2 in Table 3 (for SO\(_2\)) and Table 4 (for CaCO\(_3\)). Bestfit lines in bilogarithmic plots for the errors versus \( N \) can be found in Figs. 8 (SO\(_2\) in Test 1), 9 (CaCO\(_3\) in Test 1), 10 (SO\(_2\) in Test 2) and 11 (CaCO\(_3\) in Test 2). We note that in all cases second order convergence is achieved.

| No. of points | \( L_1 \) error of \( s \) | order | \( L_\infty \) error of \( s \) | order |
|---------------|------------------|------|------------------|------|
| 16 \times 16  | 1.00 \cdot 10^{-2} | -    | 2.25 \cdot 10^{-1} | -    |
| 32 \times 32  | 1.02 \cdot 10^{-3} | 3.30 | 1.67 \cdot 10^{-2} | 3.75 |
| 64 \times 64  | 1.38 \cdot 10^{-4} | 2.88 | 9.13 \cdot 10^{-3} | 0.87 |
| 128 \times 128| 2.98 \cdot 10^{-5} | 2.21 | 3.16 \cdot 10^{-4} | 4.85 |
| 256 \times 256| 7.49 \cdot 10^{-6} | 1.99 | 8.41 \cdot 10^{-5} | 1.91 |

| No. of points | \( L_1 \) error of \( |\nabla s| \) | order | \( L_\infty \) error of \( |\nabla s| \) | order |
|---------------|------------------|------|------------------|------|
| 16 \times 16  | 3.33 \cdot 10^{-2} | -    | 4.59 \cdot 10^{-1} | -    |
| 32 \times 32  | 5.03 \cdot 10^{-3} | 2.73 | 7.91 \cdot 10^{-2} | 2.54 |
| 64 \times 64  | 6.74 \cdot 10^{-4} | 2.90 | 7.66 \cdot 10^{-2} | 0.05 |
| 128 \times 128| 1.31 \cdot 10^{-4} | 2.37 | 5.66 \cdot 10^{-3} | 3.76 |
| 256 \times 256| 3.24 \cdot 10^{-5} | 2.01 | 2.93 \cdot 10^{-3} | 0.95 |
Table 2: Test 1. Accuracy order in the solution (top) and in the gradient (bottom) for $c$ (CaCO$_3$).

| No. of points | $L^1$ error of $c$ | order | $L^\infty$ error of $c$ | order |
|---------------|-------------------|-------|--------------------------|-------|
| $16 \times 16$ | $3.92 \cdot 10^{-4}$ | -     | $1.39 \cdot 10^{-2}$ | -     |
| $32 \times 32$ | $8.42 \cdot 10^{-4}$ | -     | $3.14 \cdot 10^{-3}$ | 2.15  |
| $64 \times 64$ | $1.90 \cdot 10^{-4}$ | 2.15  | $8.82 \cdot 10^{-4}$ | 1.83  |
| $128 \times 128$ | $4.59 \cdot 10^{-5}$ | 2.05  | $2.24 \cdot 10^{-4}$ | 1.98  |
| $256 \times 256$ | $1.13 \cdot 10^{-5}$ | 2.02  | $5.74 \cdot 10^{-5}$ | 1.96  |

| No. of points | $L^1$ error of $|\nabla c|$ | order | $L^\infty$ error of $|\nabla c|$ | order |
|---------------|-------------------|-------|--------------------------|-------|
| $16 \times 16$ | $2.32 \cdot 10^{-2}$ | -     | $1.02 \cdot 10^{-1}$ | -     |
| $32 \times 32$ | $5.99 \cdot 10^{-3}$ | 1.95  | $3.14 \cdot 10^{-2}$ | 1.70  |
| $64 \times 64$ | $1.60 \cdot 10^{-3}$ | 1.90  | $7.68 \cdot 10^{-3}$ | 2.02  |
| $128 \times 128$ | $4.11 \cdot 10^{-4}$ | 1.97  | $1.95 \cdot 10^{-3}$ | 1.99  |
| $256 \times 256$ | $1.03 \cdot 10^{-4}$ | 1.99  | $4.84 \cdot 10^{-4}$ | 2.01  |

Table 3: Test 2. Accuracy order in the solution (top) and in the gradient (bottom) for $s$ (SO$_2$).

| No. of points | $L^1$ error of $s$ | order | $L^\infty$ error of $s$ | order |
|---------------|-------------------|-------|--------------------------|-------|
| $16 \times 16$ | $1.54 \cdot 10^{-2}$ | -     | $2.56 \cdot 10^{-1}$ | -     |
| $32 \times 32$ | $1.40 \cdot 10^{-3}$ | 3.46  | $2.20 \cdot 10^{-2}$ | 3.54  |
| $64 \times 64$ | $1.27 \cdot 10^{-4}$ | 3.47  | $5.68 \cdot 10^{-4}$ | 5.27  |
| $128 \times 128$ | $3.61 \cdot 10^{-5}$ | 1.81  | $1.25 \cdot 10^{-3}$ | 1.13  |
| $256 \times 256$ | $7.39 \cdot 10^{-6}$ | 2.29  | $5.22 \cdot 10^{-5}$ | 4.58  |

| No. of points | $L^1$ error of $|\nabla s|$ | order | $L^\infty$ error of $|\nabla s|$ | order |
|---------------|-------------------|-------|--------------------------|-------|
| $16 \times 16$ | $4.96 \cdot 10^{-2}$ | -     | $5.55 \cdot 10^{-1}$ | -     |
| $32 \times 32$ | $7.02 \cdot 10^{-3}$ | 2.82  | $1.04 \cdot 10^{-1}$ | 2.42  |
| $64 \times 64$ | $6.49 \cdot 10^{-4}$ | 3.41  | $9.36 \cdot 10^{-3}$ | 3.47  |
| $128 \times 128$ | $2.77 \cdot 10^{-4}$ | 1.23  | $2.05 \cdot 10^{-2}$ | 1.13  |
| $256 \times 256$ | $4.22 \cdot 10^{-5}$ | 2.72  | $1.84 \cdot 10^{-3}$ | 3.48  |

4.2 Multigrid efficiency

In this section we solve Eq. (1) with data (30) and the following initial and boundary conditions:

\[ s_0(x) = 0, \quad c_0(x) = 10, \quad s_B = 1. \]

We perform two tests: Test 3 and Test 4, with the domain represented by the level-sets \((31)\) and \((32)\), respectively (see Fig. 7).

Solutions at time $t = 1$ are plotted in Figs. 12 and 13. For each W-cycle of the multigrid method, we compute the convergence factor as:

\[ \rho_{\text{MG}}^{(q)} = \frac{\|r_h^{(q)}\|_\infty}{\|r_h^{(q-1)}\|_\infty}, \]

where

\[ r_h^{(q)} = \begin{bmatrix} r_h \vline r_h \end{bmatrix} = \begin{bmatrix} F_s(W^{(n+1,k)}) - (J^{ss}(W^{(n+1,k)}) \cdot \Delta s + J^{sc}(W^{(n+1,k)}) \cdot \Delta c) \\ F_c(W^{(n+1,k)}) - (J^{cs}(W^{(n+1,k)}) \cdot \Delta c + J^{cc}(W^{(n+1,k)}) \cdot \Delta c) \end{bmatrix} \]
Table 4: Test 2. Accuracy order in the solution (top) and in the gradient (bottom) for $c$ (CaCO$_3$).

| No. of points | $L^1$ error of $c$ | order | $L^\infty$ error of $c$ | order |
|---------------|--------------------|-------|--------------------------|-------|
| $16 \times 16$ | $5.88 \cdot 10^{-3}$ | -     | $3.38 \cdot 10^{-2}$     | -     |
| $32 \times 32$ | $1.06 \cdot 10^{-3}$ | 2.47  | $3.65 \cdot 10^{-3}$     | 3.21  |
| $64 \times 64$ | $2.30 \cdot 10^{-4}$ | 2.21  | $9.48 \cdot 10^{-4}$     | 1.95  |
| $128 \times 128$ | $5.52 \cdot 10^{-5}$ | 2.06  | $2.47 \cdot 10^{-4}$     | 1.94  |
| $256 \times 256$ | $1.35 \cdot 10^{-5}$ | 2.03  | $6.36 \cdot 10^{-5}$     | 1.96  |

| No. of points | $L^1$ error of $|\nabla c|$ | order | $L^\infty$ error of $|\nabla c|$ | order |
|---------------|-----------------|-------|-------------------|-------|
| $16 \times 16$ | $2.37 \cdot 10^{-2}$ | -     | $6.93 \cdot 10^{-2}$ | -     |
| $32 \times 32$ | $6.46 \cdot 10^{-3}$ | 1.87  | $2.61 \cdot 10^{-2}$ | 1.41  |
| $64 \times 64$ | $1.74 \cdot 10^{-3}$ | 1.90  | $7.27 \cdot 10^{-3}$ | 1.84  |
| $128 \times 128$ | $4.47 \cdot 10^{-4}$ | 1.96  | $1.79 \cdot 10^{-3}$ | 2.02  |
| $256 \times 256$ | $1.13 \cdot 10^{-4}$ | 1.98  | $4.96 \cdot 10^{-4}$ | 1.85  |

is the defect after $q$ W-cycles. Convergence factors are plotted in Fig. 14 versus the W-cycle iterations. The first convergence factor obtained in each linear system (11) (either of the same Newton-Raphson step or a new time step) is circled (in red). The convergence factors of the first few linear systems are slightly higher due to the inconsistency of the initial and boundary conditions for $s$. After a few linear systems, the convergence factors are mainly distributed around $\rho = 0.119$, which is the predicted value by the Local Fourier Analysis for scalar multigrid in rectangular domains [51, Table 4.1, page 117], showing that the multigrid efficiency has not been degraded by the non-rectangular domain and the ghost-point approach.

4.3 Complex geometries

In this section we show how the method performs on more complex geometries. We test three domains: a woman head profile (Test 5), a shark (Test 6) and a necklace (Test 7). The solutions at time $t = 1$ are represented in Figs. 15, 17 and 19 respectively, while zooms on some relevant regions, with the contour plot of CaCO$_3$ at
Fig. 10: Test 2: bestfit lines of the errors in the solution and in the gradient for SO$_2$ in $L^1$ and $L^\infty$ norms (Table 3).

Fig. 11: Test 2: bestfit lines of the errors in the solution and in the gradient for CaCO$_3$ in $L^1$ and $L^\infty$ norms (Table 4).

Fig. 12: Solutions for Test 3 at time $t = 1$ with $N = 256$.

times $t = 0.25$, $t = 0.5$, $t = 0.75$ and $t = 1$, are represented in Figs. 16, 18 and 20 respectively. In general, we can see that the reaction is quicker around corner points, i.e. regions where the boundary has a higher curvature. For example, in Test 5 the eyelash of the woman is entirely gypsum already at time $t = 0.25$ (top-right plot of Fig. 16), while the hair strands progressively draw back at times $t = 0.25$, $t = 0.5$ and $t = 0.75$, and almost disappear at time $t = 1$ (top-left plot of Fig. 16). The gypsum formation is also quicker around reentrant corners, as we can see in the bottom-left and bottom-right plots of Fig. 16 for example. Similar conclusions may be drawn for the shark in Test 6, where the teeth (top-right plot of Fig. 18) and the secondary dorsal fin (top-left plot of Fig. 18) are transformed quickly. The damage on the caudal fin is quicker around the tips (top-left plot of Fig. 18) and around the reentrant corner of the pectoral fin (bottom-left plot of Fig. 18) and of the primary dorsal fin (bottom-right plot of Fig. 18). Tests 5 and 6 confirm the qualitative observation made by practitioners in the field of conservation of cultural heritage that the parts of a manufact that are
most quickly affected by the sulfation of marble are the higher details of the decoration and the sharp edges. It is important to take into account that gypsum is soluble in water and very prone to breaking due to thermal shocks and thus that the areas with high gypsum content, in a real case, would be quickly lost by dissolution into rainwater or by exfoliation by the dilatation due to the cyclic seasonal temperature variations. Test 7 shows qualitatively analogous results. In fact, the lace connecting the beads is the most quickly damaged part, due to its reduced thickness, followed by the beads in order of size. The larger beads and the diamond shaped pendant are less sulfated and suffer damages almost only close to the reentrant corners. Fig. 19 shows that in this case gypsum dissolution in water or its exfoliation would cause a dramatic topological change, disconnecting the necklace into many separated pieces.

5 Conclusion

Having in mind the modeling of marble degradation under chemical pollutants e.g. the sulfation process, we considered the governing nonlinear equations and their numerical approximation. The space domain is implicitly defined using a level-set approach. We employed a Crank-Nicolson in time, while for the space variables the discretization is performed by a standard Finite-Difference scheme for grid points inside the domain and by a ghost-cell technique on the ghost points (by using boundary conditions).

The solution of the large nonlinear system has been obtained by a Newton-Raphson procedure and by a tailored multigrid technique. All the numerical experiments have given very satisfactory results both from the viewpoint of the reconstruction quality and of the computational efficiency.

As future steps we can include, from the numerical analysis point of view, the spectral analysis of the resulting matrices from a GLT viewpoint [30, 31] having in mind a rigorous convergence analysis of the considered multigrid techniques.

From a modelling point of view, it would be interesting to extend the computational techniques introduced in this paper to the models of degradation processes that employ an evolving domain (e.g. [14]) or that include internal moving interfaces among layers of materials with different properties (e.g. [13, 45]). In this respect, we point out that the level-set technique introduced in this paper would be able to track correctly the pristine marble domain even if it was disconnected during the time evolution as in the example of Fig. 19.
Fig. 14: Convergence factors versus the W-cycle iterations for Test 3 (top) and Test 4 (bottom). The first convergence factor obtained in each linear system (either of the same Newton-Raphson step or a new time step) is circled (in red). The convergence factors of the first few linear systems are slightly higher due to the inconsistency of the initial and boundary conditions for $s$. After a few linear systems, the convergence factors are mainly distributed around $\rho = 0.119$, which is the predicted value by the Local Fourier Analysis for scalar multigrid in rectangular domains, showing that the multigrid efficiency has not been degraded by the non-rectangular domain and the ghost-point approach.

References

[1] G. Ali, M. Bisi, G. Spiga, and I. Torcicolo. Kinetic approach to sulphite chemical aggression in porous media. *Int. J. Nonlinear Mech.*, 47(7):769–776, 2012.

[2] G. Ali, V. Furuholm, R. Natalini, and I. Torcicolo. A mathematical model of sulphite chemical aggression of limestones with high permeability. Part I. Modeling and qualitative analysis. *Transp. Porous Media*, 69(1):109–122, 2007.

[3] P. Angot, C.-H. Bruneau, and P. Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numer. Math.*, 81, 1999.
Fig. 15: Solutions for Test 5 at time $t = 1$ with $N = 512$.

[4] D. Aregba Driollet, F. Diele, and R. Natalini. A mathematical model for the $SO_2$ aggression to calcium carbonate stones: numerical approximation and asymptotic analysis. SIAM J. Appl. Math., 64(5):1636–1667, 2004.

[5] I. Babuška. The finite element method for elliptic equations with discontinuous coefficients. Comput., 5:207–213, 1970.

[6] D. Bochkov and F. Gibou. Solving Poisson-type equations with Robin boundary conditions on piecewise smooth interfaces. J. Comput. Phys., 376:1156–1198, 2019.

[7] E. Bonetti, C. Cavaterra, F. Freddi, M. Grasselli, and R. Natalini. A nonlinear model for marble sulphation including surface rugosity: theoretical and numerical results. Comm. Pure Appl. Anal., 18(2):977–998, 2019.

[8] J. Bramble and J. King. A finite element method for interface problems in domains with smooth boundaries and interfaces. Adv. Comput. Math., 6:109–138, 1996.

[9] S. Brenner and R. Scott. The mathematical theory of finite element methods, volume 15. Springer Science & Business Media, 2007.

[10] F. Brezzi and M. Fortin. Mixed and hybrid finite element methods, volume 15. Springer Science & Business Media, 2012.

[11] F. Chantalat, C.-H. Bruneau, C. Galusinski, and A. Iollo. Level-set, penalization and cartesian meshes: A paradigm for inverse problems and optimal design. J. Comput. Phys., 228:6291–6315, 2009.

[12] A. Chertock, A. Coco, A. Kurganov, and G. Russo. A Second-Order Finite-Difference Method for Compressible Fluids in Domains with Moving Boundaries. Commun. Comput. Phys., 23(1):230–263, 2018.

[13] F. Clarelli, B. De Filippo, and R. Natalini. Mathematical model of copper corrosion. Appl. Math. Model., 38(19-20):4804–4816, 2014.

[14] F. Clarelli, A. Fasano, and R. Natalini. Mathematics and monument conservation: Free boundary models of marble sulfation. SIAM J. Appl. Math., 69(1):149–168, 2008.

[15] A. Coco, G. Currenti, J. Gottsmann, G. Russo, and C. Del Negro. A hydro-geophysical simulator for fluid and mechanical processes in volcanic areas. J. Math. Ind., 6(1):6, 2016.
Fig. 16: Contour plot of $c = 5$ (CaCO$_3$) for Test 5 at times $t = 0.25$, $t = 0.5$, $t = 0.75$ and $t = 1$. The boundary of the domain is the most external line, while the four contour lines can be easily recognised as they move away from the boundary as $t$ increases. In the electronic version of the paper, they are colored black ($t = 0.25$), cyan ($t = 0.5$), magenta ($t = 0.75$) and blue ($t = 1$). The boundary of the domain is red.

[16] A. Coco, G. Currenti, C. D. Negro, and G. Russo. A Second Order Finite-Difference Ghost-Point Method for Elasticity Problems on unbounded domains with applications to Volcanology. *Commun. Comput. Phys.*, 16:983–1009, 2014.

[17] A. Coco and G. Russo. Second order multigrid methods for elliptic problems with discontinuous coefficients on an arbitrary interface, I: one dimensional problems. *Numer. Math. Theory, Methods Appl.*, 5(1):19–42, 2012.

[18] A. Coco and G. Russo. Finite-Difference Ghost-Point Multigrid Methods on Cartesian Grids for Elliptic Problems in Arbitrary Domains. *J. Comput. Phys.*, 241:464–501, 2013.

[19] A. Coco and G. Russo. Boundary Treatment in Ghost Point Finite Difference Methods for Compressible Gas Dynamics in Domain with Moving Boundaries. In *Hyperbolic Problems: Theory, Numerics, Applications, Proceedings to HYP2012*, pages 455–465, 2014.

[20] A. Coco and G. Russo. Second order finite-difference ghost-point multigrid methods for elliptic problems with discontinuous coefficients on an arbitrary interface. *J. Comput. Phys.*, 361:299–330, 2018.
Fig. 17: Solutions for Test 6 at time $t = 1$ with $N = 512$.

[21] F. Di Turo, C. Proietti, A. Screpanti, M. Fornasier, I. Cionni, G. Favero, and A. De Marco. Impacts of air pollution on cultural heritage corrosion at European level: What has been achieved and what are the future scenarios. Environ. Pollut., 218:586–594, 2016.

[22] J. Dolbow and I. Harari. An efficient finite element method for embedded interface problems. Int. J. for Num. Meth. Eng., 78(229–252), 2009.

[23] M. Donatelli, M. Semplice, and S. Serra-Capizzano. Analysis of multigrid preconditioning for implicit PDE solvers for degenerate parabolic equations. SIAM J. Matrix Anal., 32(4):1125–1148, 2011.

[24] M. Donatelli, M. Semplice, and S. Serra-Capizzano. AMG preconditioning for nonlinear degenerate parabolic equations on nonuniform grids with application to monument degradation. Appl. Numer. Math., 68:1–18, 2013.

[25] J. Donea. An arbitrary Lagrangian-Eulerian finite element method for transient fluid-structure interactions. Comput. Methods Appl. Mech. Eng., 33:689–723, 1982.

[26] M. Dryja. A Neumann-Neumann algorithm for mortar discretization of elliptic problems with discontinuous coefficients. Num. Math., 99(645–656), 2005.

[27] R. E. Ewing, Z. Li, T. Lin, and Y. Lin. The immersed finite volume element methods for the elliptic interface problems. Math. Comput. Simul., 50(1):63–76, 1999.

[28] R. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). J. Comput. Phys., 152:457–492, 1999.

[29] L. Formaggia and F. Nobile. Stability analysis of second-order time accurate schemes for ALE-FEM. Comput. Methods Appl. Mech. Eng., 193:4097–4116, 2004.

[30] C. Garoni and S. Serra-Capizzano. Generalized locally Toeplitz sequences: theory and applications. Vol. I. Springer, Cham, New York, 2017.

[31] C. Garoni and S. Serra-Capizzano. Generalized locally Toeplitz sequences: theory and applications. Vol. II. Springer, Cham, New York, 2018.
Fig. 18: Contour plot of $c = 5$ (CaCO$_3$) for Test 6 at times $t = 0.25$, $t = 0.5$, $t = 0.75$ and $t = 1$. The boundary of the domain is the most external line, while the four contour lines can be easily recognised as they move away from the boundary as $t$ increases. In the electronic version of the paper, they are colored black ($t = 0.25$), cyan ($t = 0.5$), magenta ($t = 0.75$) and blue ($t = 1$). The boundary of the domain is red.

[32] C. Giavarini, M. Santarelli, R. Natalini, and F. Freddi. A non-linear model of sulphation of porous stones: Numerical simulations and preliminary laboratory assessments. *J. Cult. Herit.*, 9(1):14–22, 2008.

[33] F. Gibou and R. Fedkiw. A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. *J. Comput. Phys.*, 176:205–227, 2002.

[34] F. Gibou and R. Fedkiw. A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem. *J. Comput. Phys.*, 202:577–601, 2005.

[35] V. Girault and P.-A. Raviart. *Finite element methods for Navier-Stokes equations: theory and algorithms*, volume 5. Springer Science & Business Media, 2012.

[36] S. Groi and A. Reusken. An extended pressure finite element space for two-phase incompressible flows with surface tension. *J. Comput. Phys.*, 224(40–58), 2007.

[37] A. Hansbo and P. Hansbo. A finite element method for the simulation of strong and weak discontinuities in solid mechanics. *Comput. Meth. Appl. Mech. Eng.*, 1993(3523–3540), 2004.
Fig. 19: Solutions for Test 7 at time $t = 1$ with $N = 512$.

[38] Á. Helgadóttir, Y. T. Ng, C. Min, and F. Gibou. Imposing mixed Dirichlet–Neumann–Robin boundary conditions in a level-set framework. *Comput. Fluids*, 121:68–80, 2015.

[39] C. Hesch and P. Betsch. Isogeometric analysis and domain decomposition methods. *Comput. Methods Appl. Mech. Eng.*, 213:104–112, 2012.

[40] S. Hou, W. Wang, and L. Wang. Numerical method for solving matrix coefficient elliptic equation with sharp-edged interfaces. *J. Comput. Phys.*, 229:7162–7179, 2010.

[41] J. Huang and J. Zou. A mortar element method for elliptic problems with discontinuous coefficients. *IMA J. Numer. Anal.*, 22(549–576), 2001.

[42] D. Lacanette, S. Vincent, A. Sarthou, P. Malaurent, and J.-P. Caltagirone. An Eulerian/Lagrangian method for the numerical simulation of incompressible convection flows interacting with complex obstacles: Application to the natural convection in the Lascaux cave. *Int. J. Heat Mass Transf.*, 52(11-12):2528–2542, 2009.

[43] R. J. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *J. Numer. Analysis*, 31:1019–1044, 1994.

[44] Y. T. Ng, C. Min, and F. Gibou. An efficient fluid-solid coupling algorithm for single-phase flows. *J. Comput. Phys.*, 228:8807–8829, 2009.

[45] C. Nikolopoulos. Mathematical modelling of a mushy region formation during sulphation of calcium carbonate. *Netw. Heterog. Media*, 9(4):635–654, 2014.

[46] M. Oevermann, C. Scharfenberg, and R. Klein. A sharp interface finite volume method for elliptic equations on Cartesian grids. *J. Comput. Phys.*, 228:5184–5206, 2009.

[47] C. S. Peskin. Numerical analysis of blood flow in the heart. *J. Comput. Phys.*, 25:220–252, 1977.

[48] M. Saba, E. Quiones-Bolaos, and A. Barbosa Lpez. A review of the mathematical models used for simulation of calcareous stone deterioration in historical buildings. *Atmospheric Environ.*, 180:156–166, 2018.

[49] M. Semplice. Preconditioned implicit solvers for nonlinear PDEs in monument conservation. *SIAM J. Sci. Comput.*, 32(5):3071–3091, 2010.
Fig. 20: Contour plot of $c = 5$ (CaCO$_3$) for Test 7 at times $t = 0.25$, $t = 0.5$, $t = 0.75$ and $t = 1$. The boundary of the domain is the most external line, while the four contour lines can be easily recognised as they move away from the boundary as $t$ increases. In the electronic version of the paper, they are colored black ($t = 0.25$), cyan ($t = 0.5$), magenta ($t = 0.75$) and blue ($t = 1$). The boundary of the domain is red.

[50] G. H. Shortley and R. Weller. The numerical solution of Laplace’s equation. *J. Appl. Phys.*, 9:334–348, 1938.

[51] U. Trottemberg, C. Oosterlee, and A. Schuller. *Multigrid*. Academic Press, 2000.

[52] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Parameterization of computational domain in isogeometric analysis: methods and comparison. *Comput. Methods Appl. Mech. Eng.*, 200(23-24):2021–2031, 2011.

[53] S. Yu, Y. Zhou, and G. Wei. Matched Interface and Boundary (MIB) method for elliptic problems with sharp-edged interfaces. *J. Comput. Phys.*, 224:729–756, 2007.